

HubSpot CMS 開発者ドキュメント

ご注意事項：

少しでも早くお客様に役立つ情報をお届けするため、本ドキュメントの日本語版は人間の翻訳者を介さない自動翻訳で提供されております。正確な最新情報については、英語版 (<https://designers.hubspot.com/docs>) をご覧ください。また、カスタムデザインに関するご質問等には HubSpot のカスタマーサポートでは対応しかねますので、何卒ご了承くださいますようお願い申し上げます。

HubSpot CMS 開発者ドキュメントサイトへようこそ。HubSpot CMS は、美しいウェブサイト、ブログ、ランディングページ、Eメールを作成するための強力で柔軟なプラットフォームです。[CMS 開発者コミュニティ](#)のフィードバックに基づいて、コア CMS 開発者ドキュメントが改善されました。

このドキュメントに加えて、以下も必ずご確認ください。

- HubSpot の新機能: [CMS 開発者ディスカバリーキット](#) (無料の CMS 開発サンドボックスアカウントが含まれています)。
- [HubSpot CMS 開発者コミュニティ](#)に参加する。
- [CMS 開発者フォーラム](#)を参照。

HubSpot の CMS 開発者ドキュメントは、次のセクションに編成されています。

- **CMS リファレンス**

[一般的な CMS デザイン機能とコンセプトに関連する参照ドキュメント](#)

- **HubL リファレンス**

[HubSpot マークアップ言語の包括的な参照ガイド](#)

- **CMS ツール**

[CMS 開発者向けの開発ツール](#)

- **チュートリアル**

[概要とコードスニペット](#)

CMS 開発者ドキュメントを検索

CMS リファレンス

CMS レスポンシブグリッド

CMS レスポンシブグリッド

HubSpot のドラッグ & ドロップテンプレートレイアウトは、12 列レスポンシブグリッドシステムを使用して構築されています。すべてのドラッグ & ドロップテンプレートには、さまざまなスクリーンサイズでグリッドをレスポンシブにする最小限の CSS ファイル ([layout.css](#)) が自動的に組み込まれています。ファイルはブートストラップ 2 に基づいていますが、モバイルデバイスのレイアウトスタックの行と列を作成するために必要なコードのみを含めるように削減されました。layout.css の詳細については、[こちら](#)をご覧ください。

この記事では、既定の HubSpot テンプレートレイアウトのクラスマップと説明を記載しています。カスタムクラスは追加していません。

[レスポンシブグリッドの Photoshop ファイルをダウンロード](#)できます。この PSD には、モックアップを作成する際に使用するデザイナー向けグリッドガイドが記載されています。

別のグリッドシステムを使用する場合は、グリッドの CSS で layout.css ファイルをオーバーライドする必要があります。また、layout.css で検出されるクラス名に対して適切なフォールバックを提供することが推奨されます。これにより、ドラッグ & ドロップテンプレートを使用して作成したレイアウトに悪影響が生じることを懸念せずに、より高度なグリッドシステムを使用できます。

グリッドの説明

CMS レスポンシブグリッドは行と列のシステムに依存しています。各列がビューポート全体の幅の比率を決めます。各行には **row-fluid** というクラスが与えられています。行内の各列には **span1-span12** というクラスが与えられます。span12 の列は行全体を埋めます。span1 は行幅の 6.38%に相当します。例として、以下のコードに簡略版の 2 列分割を示します。各列のクラスは"span6"であり、コンテナの約 50%を占めています。

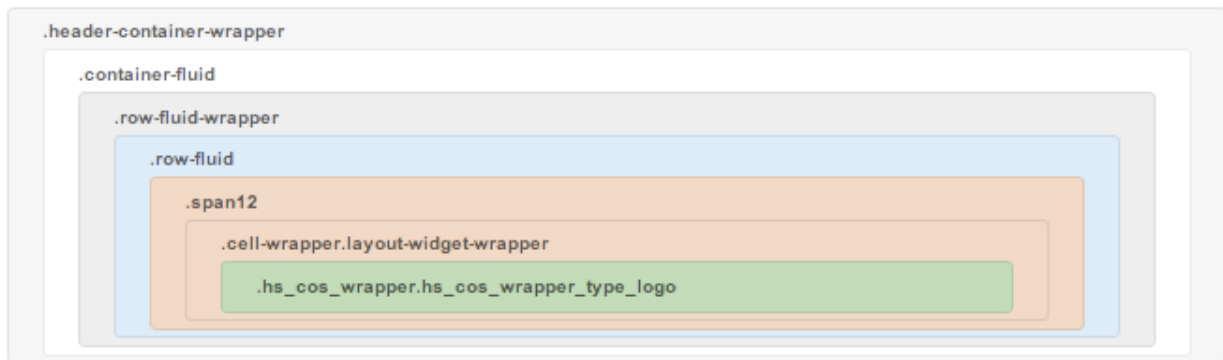
```
<div class="row-fluid ">
  <div class="span6">
    Column 1 content
  </div>
  <div class="span6">
    Column 2 content
  </div>
</div>
```

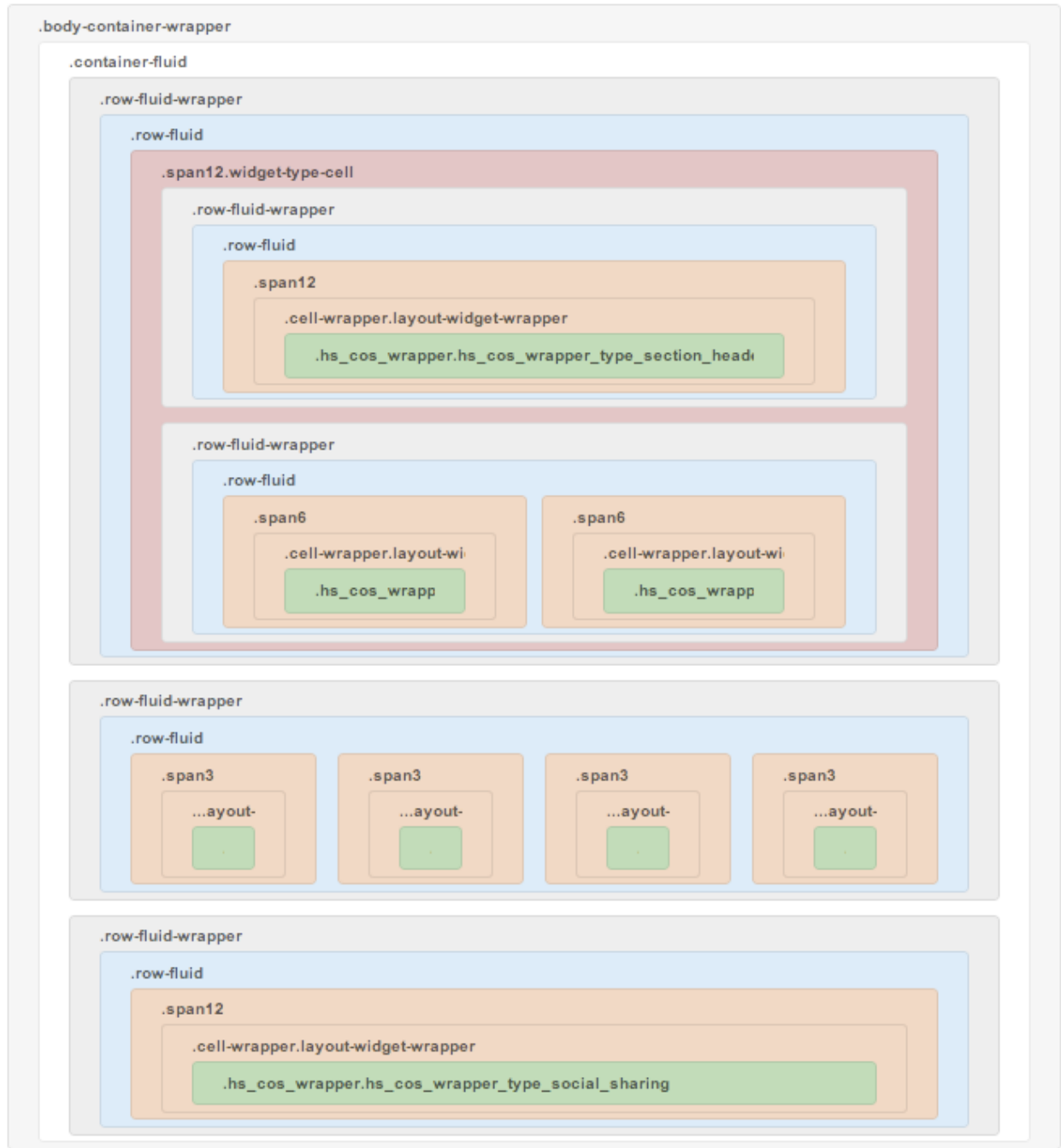
layout.css では各 **span** クラスの幅の割合が事前に定義されています。通常は、span クラスの合計が 12 以下になるようにします。合計がこれよりも大きい場合、行の幅が 100%を超えます span クラスでは既定で float: left も定義されています。デザインによっては、拡張する幅を制限するため[セクションのコンテナに max-width を設定する](#)必要がある場合もあります。

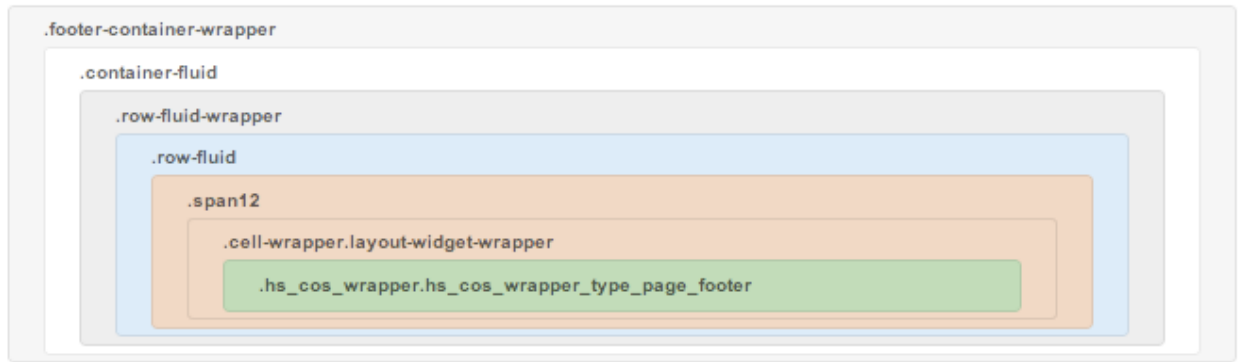
767px のビューポートサイズでは、すべての列モジュールがコンテナの幅の 100%に変わり、float は削除されます。これにより、モバイルデバイスではスタックして縮小するようになります。より詳細なブレイクポイントを使用する場合は、各自の CSS にそのようなブレイクポイントを含める必要があります。

レイアウトの例

クラスマップ







ヘッダー、本文、フッターの各セクション

HubSpot のテンプレートレイアウトはドラッグ&ドロップの区切りにより 3 つのセクションに分割されています。この各セクションは次の各クラスを指定した div で囲まれています。

- header-container-wrapper
- body-container-wrapper
- footer-container-wrapper

これらの各 div 直下に各セクションのプライマリコンテンツのラッパーとして使用できるコンテナ div があります (ただし必須ではない)。これらの div のクラスは container-fluid で、さらに以下のクラスが指定されています。

- header-container
- body-container
- footer-container

行 (row-fluid-wrapper および row-fluid)

これらのプライマリコンテナ内の各行に **row-fluid-wrapper** クラスのラッパーがあります。ラッパー内には **row-fluid** クラスの行の div があります。モジュールをグループ化すると、追加の行が作成されます。

列 (span1-span12)

前述したとおり、テンプレートビルダーUIの幅に基づいて、列のdivには **span1-span12** クラスが動的に割り当てられます。このクラスにより、レンダリングされる幅の割合が決定されます。モジュールをグループ化すると、追加の列が作成されます。テンプレートビルダーUI経由でカスタムクラスを追加すると、そのクラスがこれらのdivに適用されます。

セルとモジュールのラッパー

各列には **cell-wrapper** クラスのdivが含まれます。各モジュールは通常、モジュールタイプ特有のクラスと **hs_cos_wrapper** を指定されたspanタグでラップされます。

出力例

```
<!doctype html>
<!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]-->
<!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8" lang="en"> <![endif]-->
<!--[if IE 8]> <html class="no-js lt-ie9" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
<html class="no-js wf-ffisawebpro-n4-active wf-proximanova-n1-active wf-proximanova-n3-active wf-proximanova-n4-active wf-proximanova-n6-active wf-proximanova-n7-active wf-proximanovacondensed-i4-active wf-proximanovacondensed-i7-active wf-proximanovacondensed-n4-active wf-proximanovacondensed-n7-active wf-active" lang="en"><!--<![endif]--><head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="author" content="HubSpot">
  <meta name="description" content="">
  <meta name="generator" content="HubSpot">
```



```
<title></title>

<script type="text/JavaScript" src="//static.hsstatic.net/jquery-libs/static-1.1/jquery/jquery-1.7.1.js"></script>
  <script type="text/JavaScript">hsjQuery = window['jQuery']</script>
  <link href="//static.hsstatic.net/content_shared_assets/static-1.3949/css/public_common.css" rel="stylesheet">
  <meta property="og:description" content="">
  <meta property="og:title" content="">
  <meta name="twitter:description" content="">
  <meta name="twitter:title" content="">
  <link rel="canonical" href="null">

<!--[if lt IE 9]>
  <script src="https://static.hsstatic.net/content_shared_assets/static-1.3949/js/css3-mediaqueries.js"></script>
  <![endif]-->
  <meta name="viewport" content="width=device-width

<link rel="SHORTCUT ICON" href="//static.hubspot.com/favicon.ico" type="image/x-icon">

  <!-- The style tag has been deprecated.Attached stylesheets are included in the required_head_tags page variable. -->
```

```

</head>
<body class=" hs-content-id-1 hs-landing-page hs-page hs-content-path-c91cf69d-
17bb-48ba-ab8f-dd824950a52c hs-content-name- " style="">
  <div class="header-container-wrapper">
    <div class="header-container container-fluid">
      <div class="row-fluid-wrapper row-depth-1 row-number-1 ">
        <div class="row-fluid ">
          <div class="span12 widget-span widget-type-logo " style="" data-widget-
type="logo" data-x="0" data-w="12">
            <div class="cell-wrapper layout-widget-wrapper">
              <span id="hs_cos_wrapper_logo" class="hs_cos_wrapper
hs_cos_wrapper_widget hs_cos_wrapper_type_logo" style="" data-hs-cos-general-
type="widget" data-hs-cos-type="logo"><a href="//designers.hubspot.com" id="hs-link-
logo" style="border-width:0px;border:0px;"></a></span>
            </div><!--end layout-widget-wrapper -->
          </div><!--end widget-span -->
        </div><!--end row-->
      </div><!--end row-wrapper -->
    </div><!--end header -->
  </div><!--end header wrapper -->
  <div class="body-container-wrapper">
    <div class="body-container container-fluid">
      <div class="row-fluid-wrapper row-depth-1 row-number-1 ">

```

```

<div class="row-fluid ">
  <div class="span12 widget-span widget-type-section_header " style="" data-
widget-type="section_header" data-x="0" data-w="12">
    <div class="cell-wrapper layout-widget-wrapper">
      <span id="hs_cos_wrapper_section_header" class="hs_cos_wrapper
hs_cos_wrapper_widget hs_cos_wrapper_type_section_header" style="" data-hs-cos-
general-type="widget" data-hs-cos-type="section_header"><div class="page-header
section-header"><h1>Inbound Marketing</h1><p class="secondary-header"><span
id="hs_cos_wrapper_subheader" class="section-subheader">Will take you to new
ground</span></p></div></span>
      </div><!--end layout-widget-wrapper -->
    </div><!--end widget-span -->
  </div><!--end row-->
</div><!--end row-wrapper -->
<div class="row-fluid-wrapper row-depth-1 row-number-2 ">
  <div class="row-fluid ">
    <div class="span6 widget-span widget-type-rich_text " style="" data-widget-
type="rich_text" data-x="0" data-w="6">
      <div class="cell-wrapper layout-widget-wrapper">
        <span id="hs_cos_wrapper_left_column" class="hs_cos_wrapper
hs_cos_wrapper_widget hs_cos_wrapper_type_rich_text" style="" data-hs-cos-general-
type="widget" data-hs-cos-type="rich_text"><h2>Something Powerful</h2>
<h3>Tell The Reader More</h3>
<p>The headline and subheader tells us what you're <a href="#">offering</a>, and the
form header closes the deal.Over here you can explain why your offer is so great it's
worth filling out a form for.</p>
<p>Remember:</p>
<ul>
<li>Bullets are great</li>
<li>For spelling out <a href="#">benefits</a> and</li>
<li>Turning visitors into leads.</li>
</ul></span>
      </div><!--end layout-widget-wrapper -->
    </div>
  </div>

```

```

</div><!--end widget-span -->

<div class="span6 widget-span widget-type-form " style="" data-widget-
type="form" data-x="6" data-w="6">
    <div class="cell-wrapper layout-widget-wrapper">
        <span id="hs_cos_wrapper_module_14316630794899224"
class="hs_cos_wrapper hs_cos_wrapper_widget hs_cos_wrapper_type_form" style=""
data-hs-cos-general-type="widget" data-hs-cos-type="form"><h3
id="hs_cos_wrapper_module_14316630794899224_title" class="hs_cos_wrapper form-
title" data-hs-cos-general-type="widget_field" data-hs-cos-type="text"></h3>

<form class="hs-form stacked">
    <div class="field hs-form-field">
        <label>First Name</label>
        <div class="input">
            <input class="hs-input" type="text">
        </div>
    </div>
    <div class="field hs-form-field">
        <label>Last Name</label>
        <div class="input">
            <input class="hs-input" type="text">
        </div>
    </div>
    <div class="field hs-form-field">
        <label>Company</label>
        <div class="input">
            <input class="hs-input" type="text">
        </div>
    </div>
    <div class="field hs-form-field">
        <label>Email</label>

```

```

<div class="input">
  <input class="hs-input" type="text">
</div>
</div>
<a class="hs-button large primary" href="#">Get Free Widget</a>
</form>
</span>

  </div><!--end layout-widget-wrapper -->
</div><!--end widget-span -->
</div><!--end row-->
</div><!--end row-wrapper -->
<div class="row-fluid-wrapper row-depth-1 row-number-3 ">
  <div class="row-fluid ">
    <div class="span3 widget-span widget-type-linked_image " style="" data-widget-
type="linked_image" data-x="0" data-w="3">
      <div class="cell-wrapper layout-widget-wrapper">
        <span id="hs_cos_wrapper_module_14316630473435465"
class="hs_cos_wrapper hs_cos_wrapper_widget hs_cos_wrapper_type_linked_image"
style="" data-hs-cos-general-type="widget" data-hs-cos-type="linked_image"></span>
      </div><!--end layout-widget-wrapper -->
    </div><!--end widget-span -->
    <div class="span3 widget-span widget-type-linked_image " style="" data-widget-
type="linked_image" data-x="3" data-w="3">
      <div class="cell-wrapper layout-widget-wrapper">
        <span id="hs_cos_wrapper_module_14316630550566425"
class="hs_cos_wrapper hs_cos_wrapper_widget hs_cos_wrapper_type_linked_image"
style="" data-hs-cos-general-type="widget" data-hs-cos-type="linked_image"></span>

```

```

        </div><!--end layout-widget-wrapper -->
    </div><!--end widget-span -->
    <div class="span3 widget-span widget-type-linked_image " style="" data-widget-
type="linked_image" data-x="6" data-w="3">
        <div class="cell-wrapper layout-widget-wrapper">
            <span id="hs_cos_wrapper_module_14316630637577372"
class="hs_cos_wrapper hs_cos_wrapper_widget hs_cos_wrapper_type_linked_image"
style="" data-hs-cos-general-type="widget" data-hs-cos-type="linked_image"></span>
        </div><!--end layout-widget-wrapper -->
    </div><!--end widget-span -->
    <div class="span3 widget-span widget-type-linked_image " style="" data-widget-
type="linked_image" data-x="9" data-w="3">
        <div class="cell-wrapper layout-widget-wrapper">
            <span id="hs_cos_wrapper_module_14316630734738319"
class="hs_cos_wrapper hs_cos_wrapper_widget hs_cos_wrapper_type_linked_image"
style="" data-hs-cos-general-type="widget" data-hs-cos-type="linked_image"></span>
        </div><!--end layout-widget-wrapper -->
    </div><!--end widget-span -->
</div><!--end row-->
</div><!--end row-wrapper -->
<div class="row-fluid-wrapper row-depth-1 row-number-4 ">
    <div class="row-fluid ">
        <div class="span12 widget-span widget-type-social_sharing " style="" data-
widget-type="social_sharing" data-x="0" data-w="12">
            <div class="cell-wrapper layout-widget-wrapper">
                <span id="hs_cos_wrapper_social_sharing" class="hs_cos_wrapper
hs_cos_wrapper_widget hs_cos_wrapper_type_social_sharing" style="" data-hs-cos-
general-type="widget" data-hs-cos-type="social_sharing"><a

```

```
href="http://www.facebook.com/share.php?u=http%3A%2F%2Fexample.com%2Fmy-
page%3Futm_medium%3Dsocial%26utm_source%3Dfacebook" target="_blank"
style="width:24px;border-width:0px;border:0px;"></a> <a
href="http://www.linkedin.com/shareArticle?mini=true&url=http%3A%2F%2Fexample.co
m%2Fmy-page%3Futm_medium%3Dsocial%26utm_source%3Dlinkedin"
target="_blank" style="width:24px;border-width:0px;border:0px;"></a> <a
href="https://twitter.com/intent/tweet?original_referer=http%3A%2F%2Fexample.com%
2Fmy-
page%3Futm_medium%3Dsocial%26utm_source%3Dtwitter&url=http%3A%2F%2Fexa
mple.com%2Fmy-
page%3Futm_medium%3Dsocial%26utm_source%3Dtwitter&source=tweetbutton&text
=" target="_blank" style="width:24px;border-width:0px;border:0px;"></a> <a
href="https://plus.google.com/share?url=http%3A%2F%2Fexample.com%2Fmy-
page%3Futm_medium%3Dsocial%26utm_source%3Dgoogle_plus" target="_blank"
style="width:24px;border-width:0px;border:0px;"></a> <a
href="http://pinterest.com/pin/create/button/?url=http%3A%2F%2Fexample.com%2Fmy-
page%3Futm_medium%3Dsocial%26utm_source%3Dpinterest&media="
target="_blank" style="width:24px;border-width:0px;border:0px;"></a> <a href="mailto:?subject=Check out
http%3A%2F%2Fexample.com%2Fmy-
page%3Futm_medium%3Dsocial%26utm_source%3Demail &body=Check out
```

```

http%3A%2F%2Fexample.com%2Fmy-
page%3Futm_medium%3Dsocial%26utm_source%3Demail" target="_blank"
style="width:24px;border-width:0px;border:0px;"></a></span>

    </div><!--end layout-widget-wrapper -->

</div><!--end widget-span -->

</div><!--end row-->

</div><!--end row-wrapper -->

</div><!--end body -->
</div><!--end body wrapper -->

<div class="footer-container-wrapper">
  <div class="footer-container container-fluid">

    <div class="row-fluid-wrapper row-depth-1 row-number-1 ">
      <div class="row-fluid ">
        <div class="span12 widget-span widget-type-page_footer " style="" data-widget-
type="page_footer" data-x="0" data-w="12">
          <div class="cell-wrapper layout-widget-wrapper">
            <span id="hs_cos_wrapper_page_footer" class="hs_cos_wrapper
hs_cos_wrapper_widget hs_cos_wrapper_type_page_footer" style="" data-hs-cos-
general-type="widget" data-hs-cos-type="page_footer">
<footer>
  <span class="hs-footer-company-copyright">© 2015 HubSpot</span>
</footer>
</span>
          </div><!--end layout-widget-wrapper -->
        </div><!--end widget-span -->

```



```
</div><!--end row-->

</div><!--end row-wrapper -->

</div><!--end footer -->
</div><!--end footer wrapper -->

<script type="text/JavaScript" src="//static.hsstatic.net/content_shared_assets/static-1.3949/js/public_common.js"></script>

<script type="text/JavaScript">
var hsVars = {
  ticks:1431665701934,
  page_id:1,
  content_group_id:0,
  portal_id:327485,
  app_hs_base_url: "https://app.hubspot.com"
}
</script>

<!-- Generated by the HubSpot Template Builder - template version 1.03 -->
```

```
<!-- end coded_template: id:2859210188
path:custom/page/web_page_basic/Layout_Demo.html -->
</body></html>
```

含まれる CMS ファイル

含まれる CMS ファイル

ファイルすべての CMS テンプレートに自動的に添付される JavaScript ファイルと CSS ファイルがあります。削除できるファイルと削除できないファイルがあります。この記事では各ファイルの目的を説明します。ここで説明するファイルの他に、各自のカスタム [CSS ファイル](#)と [JavaScript ファイル](#)をテンプレートに添付できます。

CMS コンテンツにスタイルシートを添付する順序については、[この記事をご覧ください](#)。

jQuery

[jQuery](#) は、すべての HubSpot テンプレートの head タグに取り込まれます。HubL 変数 [standard_header_includes](#) の一部としてレンダリングされます。[\[設定\]](#) > [\[ウェブサイト\]](#) > [\[ページ\]](#) で jQuery 1.11.x の最新バージョンまたはバージョン 1.7.1 を選択できます。最新バージョンの jQuery を使用する場合は、古いブラウザとの後方互換性を維持するために jQuery 移行スクリプトを取り込むこともできます。また、フッターからこのライブラリを読み込むこともできます。

public_common.css

public_common.css は、主に HubSpot アプリのナビゲーションのために取り込まれるファイルです。このファイルには、アプリで使用されるさまざまなアニメーションと、HubSpot の一般的な要素 (ブログのサムネイル画像やパンくず形式のナビゲーションなど) のスタイルを設定するコードスニペットが含まれています。このファイルは [こちら](#) で確認できます。

layout.css

layout.css (旧称 required_base.css) は、HubSpot の [レスポンシブグリッド](#) のスタイル設定のためのファイルです。このファイルは常に、ドラッグ&ドロップテンプレートに取り込まれます。既定では、カスタムコードテンプレートには取り込まれません。

このファイルにはレスポンシブグリッド CSS の他に、さまざまなビューポートで要素を表示または非表示にするために使用できるクラスが含まれています。このファイルは [こちら](#) で確認できます。

public_common.js

2019 年 4 月の時点では、このファイルは既定で取り込まれなくなりました。

public_common.js に含まれていたファイルを以下に示します。

- [jquery.flexslider.js](#)
- [vex.js](#)
- [vex.gallery.js](#)

すべての CMS ページでは、テンプレートのフッターに public_common.js が取り込まれます。このファイルは [standard footer includes](#) の一部としてレンダリングされ、削除できません。この JavaScript ファイルはアプリの多くの機能や、HubSpot の slider モジュールなどのウェブサイト機能で使用されます。

HubSpot Tracking Code

HubSpot トラッキングコードは、すべての HubSpot テンプレートに自動的に追加されます。これは HubL 変数 [standard_footer_includes](#) を使用して取り込まれます。トラッキングコードにより、アナリティクス JavaScript ファイルが読み込まれます。このファイルの名前は HubID.js (例 158015.js) です。

E メールテンプレートのマークアップ

E メールテンプレートのマークアップ

以下の記事では、標準的な HubSpot E メールテンプレートのコーディング方法について説明します。E メールクライアントでは、[特定の HTML 機能と CSS 機能](#)のみがサポートされます。このため、クライアント間で一貫して表示される E メールテンプレートをコーディングするには、かなりの経験と忍耐が必要です。HubSpot の E メールテンプレートは、ユーザーフレンドリーな[ドラッグ&ドロップテンプレート](#)または[コードファイル](#)として作成することができます。

テンプレートレイアウトとコードファイル

ドラッグ&ドロップテンプレートのレイアウトには、「レスポンシブ」と「基本」があります。「レスポンシブ」はレイアウトがビューポートのサイズに合わせて調整されることを意味し、「基本」はレイアウトが静的なままであることを意味します。

既定ではレスポンシブですが、インスペクターパネルで[応答 E メールテンプレート]をクリックして[基本 E メールテンプレート]を選択することで、基本テンプレートに変換できます。

レイアウトをカスタマイズする場合、デザイナーはデザインマネージャーで既定の色とフォントをカスタマイズしたり ([設定]>[マーケティング]>[E メール])、ドラッグ & ドロップテンプレートのモジュールにインライン CSS を追加したりできます (モジュールのギアアイコン>[CSS を編集])。

HTML に複製することで、テンプレートレイアウトのマークアップにいつでもアクセスできます ([アクション]>[HTML に複製])。コードテンプレートでは、以下の変数を含めることにより、Eメールのマークアップを完全にカスタマイズできます。

必須の E メールテンプレート変数

HubSpot で E メールを送信するには、テンプレートに特定の情報を含める必要があります。

変数

タイプ

説明

```
{{ site_settings.company_city }}
```

string

会社の市区町村 ([設定]>[マーケティング]>[E メール]>[フッター]で設定) を印刷します。

```
{{ site_settings.company_name }}
```

string

会社名 ([設定]>[マーケティング]>[E メール]>[フッター]で設定) を印刷します。

```
{{ site_settings.company_state }}
```

string

会社の都道府県 ([設定]>[マーケティング]>[E メール]>[フッター]で設定) を印刷します。

```
{{ site_settings.company_street_address_1 }}
```

string

会社の住所 ([設定]>[マーケティング]>[E メール]>[フッター]で設定) を印刷します。

```
{{ unsubscribe_link }}
```

string

受信者がサブスクリプション設定を管理したり、E メール通信の配信を停止したりできるページの URL を印刷します。この変数は<a>の href 属性で使用する必要があります。

コードテンプレートを操作する場合は、ユーザーに編集可能領域を提供するために [HubL モジュール](#) を追加する必要があります。

hs-inline-css と data-hse-inline-css

すべてのクライアントで正しく表示される E メールテンプレートをコーディングする際の問題点の 1 つは、<head>内の<style>で [CSS がサポートされない](#) ことです。たとえば、Gmail のデスクトップウェブアプリは<head>の情報を完全に無視します。

E メールテンプレートのコーディングを簡単にするため、HubSpot のコード化された E メールテンプレートでは特別なスタイルタグがサポートされます。デザイナーがこのタグを使用して記述した CSS は、コンパイルされてからインライン CSS に変換され、ターゲット要素に追加されます。「hs-inline-css」という ID を持つスタイルタグに追加されたコードは、すべてターゲットタグに追加されます。

たとえば、Microsoft Outlook では、<td>タグにインラインでフォントファミリーを指定しない限り、そのテーブル列に含まれるすべてのテキストに既定のフォントファミリーが適用されます。以下の例では、「hs-inline-css」スタイルタグを使用してテンプレート内のすべてのテーブル列にフォントファミリーを追加します。メディアクエリはインラインで指定できないため、別の<style>に含める必要があることに注意してください。

注：hs-inline-css の代わりに、ドラッグ & ドロップテンプレートの[編集]>[Head の編集]で<style>タグに **data-hse-inline-css** 属性を設定することで、同じ目的を達成できます。コードファイルでは、どちらの方法も使用できます (テンプレートに含まれている **style#hs-inline-css** が 1 つだけの場合)。複数の **style[data-hs-inline-css="true"]** 要素を追加できます。

- HubL
- 出力例

```
<!doctype html>
<html>
  <head>
    {# coded email templates ONLY: #}
    <style type="text/css" id="hs-inline-css">
```

```
        td {  
            font-family:Arial, sans-serif;  
        }  
    </style>  
    {# coded OR drag-and-drop email templates: #}  
    <style type="text/css" data-hse-inline-css="true">  
        table {  
            font-family:'Courier New', sans-serif;  
        }  
    </style>  
    </head>  
    <body>  
    <table>  
        <tr>  
            <td>Column 1</td>  
            <td>Column 2</td>  
        </tr>  
    </table>  
    </body>  
</html>
```

レスポンシブ E メールテンプレート

HubSpot の既定の E メールテンプレートレイアウトでは、以下のマークアップが使用されます。レスポンシブレイアウトには、画像やテーブルをレスポンシブにするメディアクエリが含まれています。レスポンシブレイアウトでは、[設定]>[マーケティング]>[E メール]に接続する色とフォントの変数が使用されます。

HubSpot のレスポンシブレイアウトを使用する場合は、テキストの「列」を含むクラスが指定された<td>がレスポンシブになります。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>{% if content.html_title and content.html_title !=
" %}{% content.html_title %}{% else %}{% content.body.subject %}{% endif %}</title>
    <meta property="og:title" content="{% if content.html_title and
content.html_title != " %}{% content.html_title %}{% else %}{% content.body.subject %}{%
endif %}">
    <meta name="twitter:title" content="{% if content.html_title and
content.html_title != " %}{% content.html_title %}{% else %}{% content.body.subject %}{%
endif %}">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    {% if content.meta_description %}<meta name="description"
content="{% content.meta_description %}" />{% endif %}
    <style type="text/css" id="hs-inline-css">
/*<![CDATA[*]
/* everything in this node will be inlined */
/* ===== Page Styles ===== */
body, #backgroundTable {
```

```
background-color: {{ background_color }}; /* Use body to determine  
background color */
```

```
font-family: {{ primary_font }};
```

```
}
```

```
#templateTable {
```

```
width: {{ email_body_width }}px;
```

```
background-color: {{ body_color }};
```

```
-webkit-font-smoothing: antialiased;
```

```
}
```

```
h1, .h1, h2, .h2, h3, .h3, h4, .h4, h5, .h5, h6, .h6 {
```

```
color:{{ primary_font_color }};
```

```
display:block;
```

```
font-family: {{ primary_font }};
```

```
font-weight:bold;
```

```
line-height:100%;
```

```
margin-top:0;
```

```
margin-right:0;
```

```
margin-bottom:10px;
```

```
margin-left:0;
```

```
text-align:left;
```

```
}
```

```
h1, .h1 {  
    font-size:26px;  
}
```

```
h2, .h2 {  
    font-size:20px;  
}
```

```
h3, .h3 {  
    font-size:15px;  
}
```

```
h4, .h4 {  
    font-size:13px;  
}
```

```
h5, .h5 {  
    font-size:11px;  
}
```

```
h6, .h6 {  
    font-size:10px;  
}
```

```
/* ===== Header Styles ===== */

#headerTable {
    background-color: {{ background_color }};
    color:{{ primary_font_color }};
    font-family:{{ primary_font }};
    font-size:10px;
    line-height:120%;
    text-align:right;
    border-collapse: separate !important;
    padding-right: {{email_body_padding}}px;
}

#headerTable a:link, #headerTable a:visited, /* Yahoo!Mail Override */
#headerTable a .yshortcuts /* Yahoo!Mail Override */{
    font-weight:normal;
    text-decoration:underline;
}

/* ===== Template Wrapper Styles ===== */

#contentCell {
    padding: 10px 20px;
```

```

        background-color: {{ background_color }};
    }

    #contentTableOuter {

        border-collapse: separate !important;

        background-color: {{ body_color }};

        {% if email_main_body_box_shadow_css and
email_main_body_box_shadow_css != " %}

        box-shadow: {{ email_main_body_box_shadow_css }};

        {% endif %}

        padding: {{email_body_padding}}px;
    }

    #contentTableInner {

        width: {{email_body_width}}px;
    }

    /* ===== Body Styles ===== */

    .bodyContent {

        color:{{ primary_font_color }};

        font-family:{{ primary_font }};

```

```

    font-size: {{primary_font_size }};

    line-height:150%;

    text-align:left;

}

/* ===== Column Styles ===== */

table.columnContentTable {

    border-collapse:separate !important;

    border-spacing:0;

    background-color: {{ body_color }};

}

td[class~="columnContent"] {

    color:{{ primary_font_color }};

    font-family:{{ primary_font }};

    font-size:{{primary_font_size }};

    line-height:120%;

    padding-top:20px;

    padding-right:20px;

    padding-bottom:20px;

    padding-left:20px;

```

```
}

/* ===== Footer Styles ===== */

#footerTable {
    background-color: {{ background_color }};
}

#footerTable a {
    color: {{ secondary_font_color }};
}

#footerTable {
    color:{{ secondary_font_color }};
    font-family:{{ primary_font }};
    font-size:12px;
    line-height:120%;
    padding-top:20px;
    padding-right:20px;
    padding-bottom:20px;
    padding-left:20px;
    text-align:center;
}
```

```

#footerTable a:link, #footerTable a:visited, /* Yahoo!Mail Override */ #footerTable
a .yshortcuts /* Yahoo!Mail Override */{

    font-weight:normal;

    text-decoration:underline;

}

.hs-image-social-sharing-24 {

    max-width: 24px;

    max-height: 24px;

}

/* ===== Standard Resets ===== */

.ExternalClass{

    width:100%;

} /* Force Hotmail to display emails at full width */

.ExternalClass, .ExternalClass p, .ExternalClass span, .ExternalClass
font, .ExternalClass td, .ExternalClass div {

    line-height: 100%;

} /* Force Hotmail to display normal line spacing */

body, table, td, p, a, li, blockquote{

    -webkit-text-size-adjust:100%;

    -ms-text-size-adjust:100%;

} /* Prevent WebKit and Windows mobile changing default text sizes */

```



```

table, td {
    mso-table-lspace:0pt;
    mso-table-rspace:0pt;
} /* Remove spacing between tables in Outlook 2007 and up */

img {
    vertical-align: bottom;
    -ms-interpolation-mode:bicubic;
} /* Allow smoother rendering of resized image in Internet Explorer */

/* Reset Styles */
body {
    margin:0;
    padding:0;
}
table {
    border-collapse:collapse !important;
}
body, #backgroundTable, #bodyCell{
    height:100% !important;
    margin:0;
    padding:0;
    width:100% !important;
}

```

```

a:link, a:visited {
    border-bottom: none;
}

/* iOS automatically adds a link to addresses */

/* Style the footer with the same color as the footer text */
#footer a {
    color: {{ secondary_font_color }};
    -webkit-text-size-adjust: none;
    text-decoration: underline;
    font-weight: normal
}

/* Hide preview text on rendering */
#preview_text {
    display: none;
}

/*]]>*/
</style>

<style type="text/css">
/**/

/* ===== Mobile Styles ===== */
</pre>
</div>
<div data-bbox="483 918 514 938" data-label="Page-Footer">
<p>34</p>
</div>
```

```
/* Constrain email width for small screens */

@media screen and (max-width: 650px) {

    table[id="backgroundTable"] {

        width: 95% !important;

    }

    table[id="templateTable"] {

        max-width:{{ email_body_width }}px !important;

        width:100% !important;

    }

    table[id="contentTableInner"] {

        max-width:{{ email_body_width }}px !important;

        width:100% !important;

    }

    /* Makes image expand to take 100% of width*/

    img {

        width: 100% !important;

        height: auto !important;

    }

    #contentCell {
```

```

padding: 10px 10px !important;
}

#headerTable {
padding-right: {{email_body_padding_num / 2 |int}}px !important;
}

#contentTableOuter {
padding: {{email_body_padding_num / 2 |int}}px !important;
}
}

@media only screen and (max-width: 480px) {
    /* ===== Client-Specific Mobile Styles ===== */
    body, table, td, p, a, li, blockquote{
    -webkit-text-size-adjust:none !important;
    } /* Prevent Webkit platforms from changing default text sizes */
    body{
width:100% !important;
min-width:100% !important;
    } /* Prevent iOS Mail from adding padding to the body */

    /* ===== Mobile Reset Styles ===== */

```

```
td[id="bodyCell"] {  
padding:10px !important;  
}  
  
/* ===== Mobile Template Styles ===== */  
  
table[id="templateTable"] {  
max-width:{{ email_body_width }}px !important;  
width:100% !important;  
}  
  
table[id="contentTableInner"] {  
max-width:{{ email_body_width }}px !important;  
width:100% !important;  
}  
  
/* ===== Image Alignment Styles ===== */  
  
h1, .h1 {  
font-size:26px !important;  
line-height:125% !important;  
}
```

```
h2, .h2 {  
  font-size:20px !important;  
  line-height:125% !important;  
}
```

```
h3, .h3 {  
  font-size:15px !important;  
  line-height:125% !important;  
}
```

```
h4, .h4 {  
  font-size:13px !important;  
  line-height:125% !important;  
}
```

```
h5, .h5 {  
  font-size:11px !important;  
  line-height:125% !important;  
}
```

```
h6, .h6 {  
  font-size:10px !important;  
  line-height:125% !important;
```

```
}

.hide {
display:none !important;
} /* Hide to save space */

/* ===== Body Styles ===== */

td[class="bodyContent"] {
font-size:16px !important;
line-height:145% !important;
}

/* ===== Footer Styles ===== */

td[id="footerTable"]{
padding-left: 0px !important;
padding-right: 0px !important;
font-size:12px !important;
line-height:145% !important;
}

/* ===== Image Alignment Styles ===== */
```

```

table[class="alignImageTable"] {
width: 100% !important;
}

td[class="imageTableTop"] {
display: none !important;
/*padding-top:10px !important;*/
}

td[class="imageTableRight"] {
display: none !important;
}

td[class="imageTableBottom"] {
padding-bottom: 10px !important;
}

td[class="imageTableLeft"] {
display: none !important;
}

/* ===== Column Styles ===== */

td[class~="column"] {
display: block !important;

```



```
width: 100% !important;
padding-top: 0 !important;
padding-right: 0 !important;
padding-bottom: 0 !important;
padding-left: 0 !important;
}

td[class~=columnContent] {
font-size:14px !important;
line-height:145% !important;

padding-top: 10px !important;
padding-right: 10px !important;
padding-bottom: 10px !important;
padding-left: 10px !important;
}

#contentCell {
padding: 10px 0px !important;
}

#headerTable {
padding-right: {{email_body_padding_num / 2 |int}}px !important;
```

```

    }

    #contentTableOuter {
        padding: {{email_body_padding_num / 2 |int}}px !important;
    }
}

#preview_text {
    display: none;
}

/*]]>*/
</style>

<!--
http://www.emailon@cid.com/blog/details/C13/ensure\_that\_your\_entire\_email\_is\_rendered\_by\_default\_in\_the\_iphone\_ipad -->

<!-- -->
<!-- -->
<!--      _  _      _  _/_/_/      _
-->
<!--      _  _/_/_/_/_/_/_/_/_  _/_/_/_  _/_
_/_/_/_/      -->
<!--      _/_/_/_/_/_  _/_/_/_/_/_  _/_/_/_/_/_  _/_
-->
<!--      _  _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_
-->

```

```

<!--      _/   _/   _/_/_/_/_/_/_   _/_/_/_/_/_/_   _/_/_   _/_/_
-->

<!--                _/                                     -->
<!--                _/                                     -->
<!--                                                     -->
<!--                Extra White Space!
-->

<!--                                                     -->
<!-- -----
----->

</head>

<body leftmargin="0" marginwidth="0" topmargin="0" marginheight="0" offset="0">

  <!-- Preview text (text which appears right after subject) -->

  <div id="preview_text" style="display:none!important">{% text "preview_text"
label="Preview Text <span class=help-text>This will be used as the preview text that
displays in some email clients</span>", value="", no_wrapper=True %}</div>

  <!-- The backgroundTable table manages the color of the background and then
the templateTable maintains the body of

  the email template, including preheader & footer.This is the only table you set the
width of to, everything else is set to

  100% and in the CSS above.Having the width here within the table is just a small
win for Lotus Notes. -->

  <!-- Begin backgroundTable -->

  <table align="center" bgcolor="{{ background_color }}" border="0"
cellpadding="0" cellspacing="0" height="100%" width="100%" id="backgroundTable">

```

```

<tr>

    <td align="center" valign="top" id="bodyCell"> <!-- When nesting tables
within a TD, align center keeps it well, centered. -->

    <!-- Begin Template Container -->

    <!-- This holds everything together in a nice container -->

    <table border="0" cellpadding="0" cellspacing="0" id="templateTable">

    <tr>

        <td align="center" valign="top">

            <!-- Begin Template Preheader -->

            <table border="0" cellpadding="0" cellspacing="0" width="100%"
id="headerTable">

                <div class="header-container-wrapper">

                    <tr>

                        <td align="center" valign="top" class="bodyContent" width="100%"
colspan="12">

                            <table cellpadding="0" cellspacing="0" border="0" width="100%"
class="templateColumnWrapper">

                                <tr>

                                    <td align="center" valign="top" colspan="12" width="100.0%"
class="column" style="text-align: left; font-family: {{ primary_font }}; font-size:
{{ primary_font_size }}; line-height: 1.5em; color: {{ primary_font_color }};">

                                        <div class="widget-span widget-type-email_view_as_web_page "
style="" data-widget-type="email_view_as_web_page">

                                            {% if content.create_page %}

                                                <div style="padding-top: 15px; font-family: Geneva, Verdana, Arial,
Helvetica, sans-serif; text-align: right; font-size: 9px; line-height: 1.34em; color:
{{ secondary_font_color }}">

```

Not rendering correctly? View this email as a web page

</div>

{% endif %}

</div><!--end widget-span -->

</td>

</tr>

</table>

</td>

</tr>

</div><!--end header wrapper -->

</table>

<!-- End Template Preheader -->

</td>

</tr>

<tr>

<td align="center" valign="top" id="contentCell">

<!-- Begin Template Wrapper -->

<!-- This separates the preheader which usually contains the "open in browser, etc" content

from the actual body of the email.Can alternatively contain the footer too, but I choose not

to so that it stays outside of the border. -->

```

        <table border="0" cellpadding="0" cellspacing="0" width="100%"
id="contentTableOuter" style="{{email_body_border_css}}">

        <tr>

                <td align="center" valign="top">

                        <table border="0" cellpadding="0" cellspacing="0"
id="contentTableInner">

                                <div class="body-container-wrapper">

                                        <tr>

                                                <td align="center" valign="top" class="bodyContent" width="100%"
colspan="12">

                                                        <table cellpadding="0" cellspacing="0" border="0" width="100%"
class="templateColumnWrapper">

                                                                <tr>

                                                                        <td align="center" valign="top" colspan="12" width="100.0%"
class="column" style="text-align: left; font-family: {{ primary_font }}; font-size:
{{ primary_font_size }}; line-height: 1.5em; color: {{ primary_font_color }};">

                                                                                <div class="widget-span widget-type-logo " style="" data-widget-
type="logo">

                                                                                        <div class="layout-widget-wrapper">

                                                                                                {% logo "logo_image" suppress_company_name=True,
overrideable=True, label='Logo' %}

                                                                                                        </div><!--end layout-widget-wrapper -->

                                                                                                                </div><!--end widget-span -->

                                                                                                                        </td>

                                                                                                                                </tr>

                                                                                                                                        </table>

                                                                                                                                                </td>

                                                                                                                                </tr>

                                                                                                                                        </table>

                                                                                                                                                </td>

```

```

</tr>

<tr>

  <td align="center" valign="top" class="bodyContent" width="100%"
  colspan="12">

    <table cellpadding="0" cellspacing="0" border="0" width="100%"
  class="templateColumnWrapper">

      <tr>

        <td align="center" valign="top" colspan="12" width="100.0%"
  class="column" style="text-align: left; font-family: {{ primary_font }}; font-size:
  {{ primary_font_size }}; line-height: 1.5em; color: {{ primary_font_color }};">

          <div class="widget-span widget-type-email_body " style="" data-
  widget-type="email_body">

            {% content_attribute "email_body" %}<p>Hi {{ contact.firstname }},</p>

            <p>Describe what you have to offer the customer.Why should they read?What did you
            promise them in the subject line?Tell them something cool.Make them laugh.Make them
            cry.Well, maybe don't do that...</p>

            <p>Use a list to:</p>

            <ul>

              <li>Explain the value of your offer</li>

              <li>Remind the reader what they'll get out of taking action</li>

              <li>Show off your skill with bullet points</li>

              <li>Make your content easy to scan</li>

            </ul>

            <p><a href="http://hubspot.com">LINK TO A LANDING PAGE ON YOUR SITE</a>
            (This is the really important part.)</p>

            <p>Now wrap it all up with a pithy little reminder of how much you love them.</p>

```

```

<p>Aw.You silver-tongued devil, you.</p>

<p>Sincerely,</p>

<p>Your name</p>{% end_content_attribute %}

        </div><!--end widget-span -->

    </td>

</tr>

</table>

</td>

</tr>

<tr>

    <td align="center" valign="top" class="bodyContent" width="100%"
colspan="12">

        <table cellpadding="0" cellspacing="0" border="0" width="100%"
class="templateColumnWrapper">

            <tr>

                <td align="center" valign="top" colspan="12" width="100.0%"
class="column" style="text-align: left; font-family: {{ primary_font }}; font-size:
{{ primary_font_size }}; line-height:1.5em; color: {{ primary_font_color }}; ">

                    <div class="widget-span widget-type-text " style="padding: 10px
0px 0px 0px; text-align: right; " data-widget-type="text">

                        <div class="layout-widget-wrapper">

                            {% text "social_sharing_label" label='Social Sharing Label' %}

                        </div><!--end layout-widget-wrapper -->

```



```

        </div><!--end widget-span -->

    </td>

</tr>

</table>

</td>

</tr>

<tr>

    <td align="center" valign="top" class="bodyContent" width="100%"
colspan="12">

        <table cellpadding="0" cellspacing="0" border="0" width="100%"
class="templateColumnWrapper">

            <tr>

                <td align="center" valign="top" colspan="12" width="100.0%"
class="column" style="text-align: left; font-family: {{ primary_font }}; font-size:
{{ primary_font_size }}; line-height:1.5em; color: {{ primary_font_color }};">

                    <div class="widget-span widget-type-social_sharing "
style="padding: 10px 0px 0px 0px; text-align: right; " data-widget-type="social_sharing">

                        <div class="layout-widget-wrapper">

                            {% widget_block social_sharing "Social Sharing" label='Social
Sharing', use_page_url=False %}

                            {% end_widget_block %}

                        </div><!--end layout-widget-wrapper -->

                    </div><!--end widget-span -->

                </td>

            </tr>

        </table>

```

```

        </td>
    </tr>
</div><!--end body wrapper -->

        </table>
    </td>
</tr>
</table>
<!-- End Template Wrapper -->
</td>
</tr>
<tr>
    <td align="center" valign="top">
        <!-- Begin Template Footer -->
        <table border="0" cellpadding="0" cellspacing="0" width="100%"
id="footerTable">
            <div class="footer-container-wrapper">
                <tr>
                    <td align="center" valign="top" class="bodyContent" width="100%"
colspan="12">
                        <table cellpadding="0" cellspacing="0" border="0" width="100%"
class="templateColumnWrapper">
                            <tr>
                                <td align="center" valign="top" colspan="12" width="100.0%"
class="column" style="text-align: left; font-family: {{ primary_font }}; font-size:
{{ primary_font_size }}; line-height:1.5em; color: {{ primary_font_color }}; ">

```

```

<div class="widget-span widget-type-email_can_spam " style=""
data-widget-type="email_can_spam">

    <p id="footer" style="font-family: Geneva, Verdana, Arial, Helvetica,
sans-serif; text-align: center; font-size: 12px; line-height: 1.34em; color:
{{ secondary_font_color }}; display: block">

{{ site_settings.company_name }}

    {{ site_settings.company_street_address_1 }}

    {{ site_settings.company_street_address_2 }}

    {{ site_settings.company_city }},

    {{ site_settings.company_state }}

    {{ site_settings.company_zip }}

    {{ site_settings.company_country }}

<br/><br/>

You received this email because you are subscribed to {{ subscription_name }} from
{{ site_settings.company_name }}.

<br/><br/>

Update your <a class="hubspot-mergetag" data-unsubscribe="true"
href="{{ unsubscribe_link }}" style="text-decoration: underline; whitespace: nowrap;
color: {{ secondary_font_color }};">email preferences</a> to choose the types of emails
you receive.

<br/><br/>

<a class="hubspot-mergetag" data-unsubscribe="true" href="{{ unsubscribe_link_all }}"
style="text-decoration: underline; whitespace: nowrap; color:
{{ secondary_font_color }};">Unsubscribe from all future emails</a>

</p>

</div><!--end widget-span -->

```

```
        </td>
      </tr>
    </table>
  </td>
</tr>
</div><!--end footer wrapper -->
      <tr>
        <td>
          {% hubspot_footer %}
        </td>
      </tr>
    </table>
    <!-- End Template Footer -->
  </td>
</tr>
</table>
<!-- End Template Container -->
</td>
</tr>
</table>
</body>
</html>
```

基本 E メールテンプレート


```

    {% if email_main_body_box_shadow_css and
email_main_body_box_shadow_css != " " %}

    #main_body {

        box-shadow: {{ email_main_body_box_shadow_css }};

    }

    {% endif %}

    /* Hide preview text on rendering */

    #preview_text {

        display: none;

    }

    /*]]>*/

</style>

<!--
http://www.emailon@cid.com/blog/details/C13/ensure\_that\_your\_entire\_email\_is\_rende
red\_by\_default\_in\_the\_iphone\_ipad -->

<!-- -->

<!-- -->

<!--      _/  _/      _/  _/_/_/      _/
-->

<!--      _/  _/_/_/  _/_/_/  _/      _/_/_/      _/_/
_/_/_/_/      -->

<!--      _/_/_/_/  _/  _/_/_/  _/_/_/  _/      _/_/_/_/  _/
-->

```

```

--> <!--          _/  _/ _/ _/ _/  _/  _/  _/ _/ _/  _/
--> <!--          _/  _/  _/_/_/  _/_/_/  _/_/_/  _/_/_/  _/_/_/  _/_/_/
--> <!--          _/
--> <!--          _/
--> <!--
--> <!--          Extra White Space!
--> <!--
--> <!-------
----->

</head>

<body bgcolor="{{ background_color }}" style="margin: 0; padding: 0; background-
color: {{ background_color }}" marginheight="0" marginwidth="0" topmargin="0">

<!-- Preview text (text which appears right after subject) -->

<div id="preview_text" style="display:none!important">{% text "preview_text"
label="Preview Text <span class=help-text>This will be used as the preview text that
displays in some email clients</span>", value="", no_wrapper=True %}</div>

<!-- start container -->

<table bgcolor="{{ background_color }}" style="background-color:
{{ background_color }}; -webkit-font-smoothing: antialiased; -webkit-text-size-adjust:
none" cellpadding="0" cellspacing="0" border="0" width="100%">

<tr>

<td bgcolor="{{ background_color }}" style="background-color:
{{ background_color }};">

```

```

        <div align="center">

        <table cellpadding="0" width="{{ email_body_width }}" cellspacing="0"
border="0">

                <tr>

                        <td align="center" bgcolor="{{ background_color }}"
style="background-color: {{ background_color }}">

                                <!-- start header section -->

                                <div class="header-container-wrapper">

                                        <table class="wrappertable" cellpadding="0" cellspacing="0" border="0"
width="{{ email_body_width }}">

<tr class="scaffold" style="font-size:0; height: 0px" height="0"><td colspan="1"
height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td><td
colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height:
0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size:
0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height:
0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%"
style="line-height: 0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0"
width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td><td colspan="1"
height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td><td
colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height:
0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size:
0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height:
0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%"
style="line-height: 0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0"
width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td></tr>

                <tr>

                        <td valign="top" colspan=12 width="100.0%" class="" style="width:100.0%; text-
align: left; padding: 0; font-family: {{ primary_font }}; font-size: {{ primary_font_size }};
line-height: 1.5em; color: {{ primary_font_color }}; ">

                                <div class="widget-span widget-type-email_view_as_web_page " style=""
data-widget-type="email_view_as_web_page">

```



```

{% if content.create_page %}

    <div style="padding-top: 15px; font-family: Geneva, Verdana, Arial,
Helvetica, sans-serif; text-align: right; font-size: 9px; line-height: 1.34em; color:
{{ secondary_font_color }}">

    Not rendering correctly?View this email as a web page <a class="hubspot-mergetag"
style="color: {{ secondary_font_color }}; text-decoration: underline; white-space:
nowrap" data-viewaswebpage="true" href="{{ view_as_page_url }}">here</a>.

</div>

{% endif %}

    </div><!--end widget-span -->

</td>

</tr>

</table>

</div><!--end header wrapper -->

    <!-- end header section -->

</td>

</tr>

</table>

</div>

</td>

</tr>

<tr>

    <td bgcolor="{{ background_color }}" style="padding: 10px 20px;
background-color: {{ background_color }}">

```

```

        <div align="center">

        <table cellpadding="0" width="{{ email_body_width }}" cellspacing="0"
border="0">

                <tr>

                        <td id="main_body" width="{{ email_body_width }}"
bgcolor="{{ body_color }}" style="padding: {{ email_body_padding }}px; background-
color: {{ body_color }}; {{ email_body_border_css }}; border-radius: 4px">

                                <div align="center">

                                        <table cellpadding="0" width="{{ email_body_width }}"
cellspacing="0" border="0">

                                                <!-- start content -->

                                                <tr>

                                                        <td style="padding: 0; background-color:
{{ body_color }}">

                                                                <div align="center">

                                                                        <table cellpadding="0" cellspacing="0" border="0"
width="100%" style="font-family: {{ primary_font }}; font-size: {{ primary_font_size }};
line-height: 1.5em; color: {{ primary_font_color }}">

                                                                                <tr>

                                                                                        <td>

                                                                                                <div align="center">

                                                                                                        <!-- start body section -->

                                                                                                        <div class="body-container-wrapper">

                                                                                                                <table class="wrappertable" cellpadding="0" cellspacing="0" border="0"
width="{{ email_body_width }}">

                                                                                                                        <tr class="scaffold" style="font-size:0; height: 0px" height="0"><td colspan="1"
height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td><td
colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height:

```

```

0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size:
0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height:
0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%"
style="line-height: 0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0"
width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td><td colspan="1"
height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td><td
colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height:
0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size:
0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height:
0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%"
style="line-height: 0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0"
width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td></tr>

```

```

<tr>

```

```

    <td valign="top" colspan=12 width="100.0%" class="" style="width:100.0%; text-
align: left; padding: 0; font-family: {{ primary_font }}; font-size: {{ primary_font_size }};
line-height: 1.5em; color: {{ primary_font_color }}; ">

```

```

        <div class="widget-span widget-type-logo " style="" data-widget-
type="logo">

```

```

            <div class="layout-widget-wrapper">

```

```

                {% logo "logo_image" suppress_company_name=True,
overrideable=True, label='Logo' %}

```

```

            </div><!--end layout-widget-wrapper -->

```

```

        </div><!--end widget-span -->

```

```

    </td>

```

```

</tr>

```

```

<tr>

```

```

    <td valign="top" colspan=12 width="100.0%" class="" style="width:100.0%; text-
align: left; padding: 0; font-family: {{ primary_font }}; font-size: {{ primary_font_size }};
line-height: 1.5em; color: {{ primary_font_color }}; ">

```

```
<div class="widget-span widget-type-email_body " style="" data-widget-
type="email_body">

{% content_attribute "email_body" %}<p>Hi {{ contact.firstname }},</p>

<p>Describe what you have to offer the customer.Why should they read?What did you
promise them in the subject line?Tell them something cool.Make them laugh.Make them
cry.Well, maybe don't do that...</p>

<p>Use a list to:</p>

<ul>

<li>Explain the value of your offer</li>

<li>Remind the reader what they'll get out of taking action</li>

<li>Show off your skill with bullet points</li>

<li>Make your content easy to scan</li>

</ul>

<p><a href="http://hubspot.com">LINK TO A LANDING PAGE ON YOUR SITE</a>
(This is the really important part.)</p>

<p>Now wrap it all up with a pithy little reminder of how much you love them.</p>

<p>Aw.You silver-tongued devil, you.</p>

<p>Sincerely,</p>

<p>Your name</p>{% end_content_attribute %}

</div><!--end widget-span -->

</td>

</tr>
```

```

<tr>

    <td valign="top" colspan=12 width="100.0%" class="" style="width:100.0%; text-align: left; padding:0; font-family: {{ primary_font }}; font-size: {{ primary_font_size }}; line-height:1.5em; color: {{ primary_font_color }}; ">

        <div class="widget-span widget-type-text " style="padding: 10px 0px 0px 0px; text-align: right; " data-widget-type="text">

            <div class="layout-widget-wrapper">

                {% text "social_sharing_label" overrideable=True, label='Social Sharing Label' %}

            </div><!--end layout-widget-wrapper -->

        </div><!--end widget-span -->

    </td>

</tr>

<tr>

    <td valign="top" colspan=12 width="100.0%" class="" style="width:100.0%; text-align: left; padding:0; font-family: {{ primary_font }}; font-size: {{ primary_font_size }}; line-height:1.5em; color: {{ primary_font_color }}; ">

        <div class="widget-span widget-type-social_sharing " style="padding: 10px 0px 0px 0px; text-align: right; " data-widget-type="social_sharing">

            <div class="layout-widget-wrapper">

                {% widget_block social_sharing "Social Sharing" label='Social Sharing', overrideable=True, use_page_url=False %}

                {% end_widget_block %}

            </div><!--end layout-widget-wrapper -->

        </div><!--end widget-span -->

    </td>

</tr>

```

```

</table>
</div><!--end body wrapper -->
<!-- end body section -->
</div>
</td>
</tr>
</table>
</div>
</td>
</tr>
<!-- end content -->
</table>
</div>
</td>
</tr>
</table>
</div>
</td>
</tr>
<td bgcolor="{{ background_color }}" style="background-color:
{{ background_color }}; padding: 13px {{ email_body_padding }}px">
<div align="center">

```

```

        <table cellpadding="0" width="{{ email_body_width }}" cellspacing="0"
border="0">

        <tr>

        <td align="center" bgcolor="{{ background_color }}"
style="background-color: {{ background_color }}">

        <!-- start footer section -->

        <div class="footer-container-wrapper">

        <table class="wrappertable" cellpadding="0" cellspacing="0" border="0"
width="{{ email_body_width }}">

<tr class="scaffold" style="font-size:0; height: 0px" height="0"><td colspan="1"
height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td><td
colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height:
0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size:
0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height:
0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%"
style="line-height: 0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0"
width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td><td colspan="1"
height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td><td
colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size: 0px; height:
0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height: 0px; font-size:
0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%" style="line-height:
0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0" width="8.33%"
style="line-height: 0px; font-size: 0px; height: 0px"> </td><td colspan="1" height="0"
width="8.33%" style="line-height: 0px; font-size: 0px; height: 0px"> </td></tr>

        <tr>

        <td valign="top" colspan=12 width="100.0%" class="" style="width:100.0%; text-
align: left; padding:0; font-family: {{ primary_font }}; font-size: {{ primary_font_size }};
line-height:1.5em; color: {{ primary_font_color }}; ">

        <div class="widget-span widget-type-email_can_spam " style="" data-
widget-type="email_can_spam">

```

```
<p id="footer" style="font-family: Geneva, Verdana, Arial, Helvetica, sans-serif; text-align: center; font-size: 12px; line-height: 1.34em; color: {{ secondary_font_color }}; display: block">
```

```
{{ site_settings.company_name }}
```

```
  {{ site_settings.company_street_address_1 }}
```

```
  {{ site_settings.company_street_address_2 }}
```

```
  {{ site_settings.company_city }},
```

```
  {{ site_settings.company_state }}
```

```
  {{ site_settings.company_zip }}
```

```
  {{ site_settings.company_country }}
```

```
<br/><br/>
```

```
You received this email because you are subscribed to {{ subscription_name }} from {{ site_settings.company_name }}.
```

```
<br/><br/>
```

```
Update your <a class="hubspot-mergetag" data-unsubscribe="true" href="{{ unsubscribe_link }}" style="text-decoration: underline; whitespace: nowrap; color: {{ secondary_font_color }};">email preferences</a> to choose the types of emails you receive.
```

```
<br/><br/>
```

```
<a class="hubspot-mergetag" data-unsubscribe="true" href="{{ unsubscribe_link_all }}" style="text-decoration: underline; whitespace: nowrap; color: {{ secondary_font_color }};">Unsubscribe from all future emails</a>
```

```
</p>
```

```
</div><!--end widget-span -->
```

```
</td>
```

```
</tr>
```



```
</table>
</div><!--end footer wrapper -->
    <!-- end footer section -->
    </td>
    </tr>
    <tr>
        {% hubspot_footer %}
    </tr>
</table>
</div>
</td>
</tr>
</table>
<!-- end container -->
</body>
</html>
```

最初から作成する

デザインマネージャーで一から.html Eメールを作成すると、以下のマークアップが自動的に生成されます。

```
<!doctype html>
```

```

<html>
  <head>
    <title>{% if content.html_title and content.html_title !=
" %}{{ content.html_title }}{% else %}{{ content.body.subject }}{% endif %}</title>
    {% if content.meta_description %}<meta name="description"
content="{{ content.meta_description }}" />{% endif %}
    <style type="text/css" id="hs-inline-css">

      /*<![CDATA[* /

      /* everything in this style tag will be inlined onto matching elements */

      .sample-rule {

      }

      /*]]>*/

    </style>

  </head>

  <body>

    <!-- Preview text (text which appears right after subject in certain email clients) -
->

    <div id="preview_text" style="display:none!important">{% text "preview_text"
label="Preview Text <span class=help-text>This will be used as the preview text that
displays in some email clients</span>", value="", no_wrapper=True %}</div>

    <!-- View as webpage link -->

    {% if content.create_page %}

    <div>

      Not rendering correctly?View this email as a web page <a class="hubspot-
mergetag" data-viewaswebpage="true" href="{{ view_as_page_url }}">here</a>.

    </div>

    {% endif %}

```

フォームのマークアップ

HubSpot フォームのマークアップ

HubSpot のフォームは JavaScript を使用して、HubSpot でホストされているページまたは外部でホストされているページに、フォームのマークアップを動的にレンダリングします。この記事では、フォームの埋め込みコードによりレンダリングされるマークアップについて説明します(さまざまなフィールドタイプのスタイル設定に使用できるセレクターを含む)。

フォームのマークアップの説明

First name

Last name

Email*

Single Line Text Property
Help text

Multi-line Text Property

Dropdown Select Property

This is a form Header

Radio Select Property

Option 1
 Option 2
 Option 3

Multiple Checkboxes Property

Option 1
 Option 2
 Option 3

Single Checkbox Property

Number Property

File Upload Property
 No file chosen

Date Picker Property

HubSpot のフォームエディターには、1 列、2 列、または 3 列のフォームマークアップを出力する使いやすい UI があります。HubSpot で作成するフォームに基づいてマークアップが生成されることを覚えておいてください。label タグ、placeholder 属性、ヘル

プテキスト、列、初期フィールド値はすべて、HubSpot のフォームエディター ([マーケティング] > [リードの獲得] > [フォーム]) から制御されます。

埋め込みコードにより **form** タグと **div** ラッパー、および **Forms API** 送信エンドポイントから HubSpot に送信する **action** 属性が生成されます。各入力行は **fieldset** タグで囲まれています。fieldset タグ内には、プログレッシブプロファイリングフィールドと依存フォームフィールドに使用される React JS **div** タグが含まれています。ほとんどのフィールドは、**label** タグと **input** タグ、および **div** 内のヘルプテキストでレンダリングされます。ドロップダウン選択フィールドには **select** タグと **option** タグがあり、その他の複数選択フィールドでは各自のオプションが **ul** タグと **li** タグを使用してレンダリングされます。CSS セレクターの詳細については、[以下](#)を参照してください。

各入力には一意の ID 属性の他に、フィールドのタイプを示す **type** 属性があります。検証エラーメッセージは **ul**、**li**、および **label** タグ内にレンダリングされます。フォームの終わりには **hs_context** という名前の非表示フィールドがあります。このフィールドは、JSON ブロックのコンテキスト情報 (ユーザーのユーザートークン Cookie (hubspot_utm) など) を収集し、送信の一部として渡します。HubSpot はこの情報を自動的に生成します。

各標準 HubSpot コンタクトプロパティを使用して表現されたフォームの出力例を以下に示します。

フォーム出力例

```
<span id="hs_cos_wrapper_my_form" class="hs_cos_wrapper hs_cos_wrapper_widget hs_cos_wrapper_type_form" style="" data-hs-cos-general-type="widget" data-hs-cos-type="form">
  <h3 id="hs_cos_wrapper_my_form_title" class="hs_cos_wrapper form-title" data-hs-cos-general-type="widget_field" data-hs-cos-type="text"></h3>
  <div id="hs_form_target_my_form">
```

```
<form novalidate="" accept-charset="UTF-8"
action="https://forms.hsforms.com/submissions/v3/public/submit/formsnext/multipart/63
33443/f57de912-d759-4f67-ae76-efe001bbf926" enctype="multipart/form-data"
id="hsForm_f57de912-d759-4f67-ae76-efe001bbf926_5033" method="POST"
class="hs-form stacked hs-custom-form hs-form-private hsForm_f57de912-d759-4f67-
ae76-efe001bbf926 hs-form-f57de912-d759-4f67-ae76-efe001bbf926 hs-form-
f57de912-d759-4f67-ae76-efe001bbf926_2b318804-7ef5-417d-b02f-541fef8aa532"
data-form-id="f57de912-d759-4f67-ae76-efe001bbf926" data-portal-id="6333443"
target="target_iframe_f57de912-d759-4f67-ae76-efe001bbf926_5033" data-
reactid=".hspt-forms-0" _lpchecked="1">

  <fieldset class="form-columns-2" data-reactid=".hspt-forms-0.1:$0">

    <div class="hs_firstname hs-firstname hs-fieldtype-text field hs-form-field" data-
reactid=".hspt-forms-0.1:$0.1:$firstname">

      <label id="label-firstname-f57de912-d759-4f67-ae76-efe001bbf926_5033"
class="" placeholder="Enter your First name" for="firstname-f57de912-d759-4f67-ae76-
efe001bbf926_5033" data-reactid=".hspt-forms-0.1:$0.1:$firstname.0"><span data-
reactid=".hspt-forms-0.1:$0.1:$firstname.0.0">First name</span></label>

      <legend class="hs-field-desc" style="display:none;" data-reactid=".hspt-
forms-0.1:$0.1:$firstname.1"></legend>

      <div class="input" data-reactid=".hspt-forms-
0.1:$0.1:$firstname.$firstname"><input id="firstname-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="hs-input" type="text" name="firstname" value=""
placeholder="" autocomplete="given-name" data-reactid=".hspt-forms-
0.1:$0.1:$firstname.$firstname.0" style="background-image:
url(&quot;data:image/png;base64,iVBORw0KGgoAAAANSUgAAABAAAAAQCAYA
AAAF8/9hAAABHKIEQVQ4EaVTO26DQBD1ohQWaS2lg9JybZ+AK7hNwx2oloVf4UPQ0
Lj1FdKktevIpel8AKNUkDcWMxpgSalEaTVv3sx7uztiTdu2s/98DywOw3Dued4Who/M2al
x5IZV1aEsy0+qiwHELyi+Ytl0PQ69SxAxkWIA4RMRTdNsKE59juMcuZd6xIAFeZ6fGCdJ
8kY4y7KAuTRNGd7jyEBXsdOPE3a0QGPsniOnnYMO67LgSQN9T41F2QGrQRRFCwy
zoIF2qyBuKKbcOgPXdVeY9rMWgNsjf9ccYesJhk3f5dYT1HX9gR0LLQR30TnjkuEcx2u
luS4Rnl+aj6sJR0AM8AaumPaM/rRehyWhXqbFAA9kh3/8/NvHxAYGAsZ/il8IalkCLBfNV
AAAAABJRU5ErkJggg==&quot;); background-repeat: no-repeat; background-
attachment: scroll; background-size: 16px 18px; background-position: 98% 50%; cursor:
auto;"></div>

    </div>

    <div class="hs_lastname hs-lastname hs-fieldtype-text field hs-form-field" data-
reactid=".hspt-forms-0.1:$0.1:$lastname">
```

```

    <label id="label-lastname-f57de912-d759-4f67-ae76-efe001bbf926_5033"
class="" placeholder="Enter your Last name" for="lastname-f57de912-d759-4f67-ae76-
efe001bbf926_5033" data-reactid=".hspt-forms-0.1:$0.1:$lastname.0"><span data-
reactid=".hspt-forms-0.1:$0.1:$lastname.0.0">Last name</span></label>

    <legend class="hs-field-desc" style="display:none;" data-reactid=".hspt-
forms-0.1:$0.1:$lastname.1"></legend>

    <div class="input" data-reactid=".hspt-forms-
0.1:$0.1:$lastname.$lastname"><input id="lastname-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="hs-input" type="text" name="lastname" value=""
placeholder="" autocomplete="family-name" data-reactid=".hspt-forms-
0.1:$0.1:$lastname.$lastname.0"></div>

    </div>
</fieldset>

<fieldset class="form-columns-1" data-reactid=".hspt-forms-0.1:$1">

    <div class="hs_email hs-email hs-fieldtype-text field hs-form-field" data-
reactid=".hspt-forms-0.1:$1.1:$email">

        <label id="label-email-f57de912-d759-4f67-ae76-efe001bbf926_5033"
class="" placeholder="Enter your Email" for="email-f57de912-d759-4f67-ae76-
efe001bbf926_5033" data-reactid=".hspt-forms-0.1:$1.1:$email.0"><span data-
reactid=".hspt-forms-0.1:$1.1:$email.0.0">Email</span><span class="hs-form-
required" data-reactid=".hspt-forms-0.1:$1.1:$email.0.1">*</span></label>

        <legend class="hs-field-desc" style="display:none;" data-reactid=".hspt-
forms-0.1:$1.1:$email.1"></legend>

        <div class="input" data-reactid=".hspt-forms-
0.1:$1.1:$email.$email"><input id="email-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="hs-input" type="email" name="email" required=""
placeholder="" value="" autocomplete="email" data-reactid=".hspt-forms-
0.1:$1.1:$email.$email.0"></div>

    </div>
</fieldset>

<fieldset class="form-columns-1" data-reactid=".hspt-forms-0.1:$2">

    <div class="hs_single_line_text_property hs-single_line_text_property hs-
fieldtype-text field hs-form-field" data-reactid=".hspt-forms-
0.1:$2.1:$single_line_text_property">

```

```
<label id="label-single_line_text_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="" placeholder="Enter your Single Line Text Property" for="single_line_text_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-reactid=".hbspt-forms-0.1:$2.1:$single_line_text_property.0"><span data-reactid=".hbspt-forms-0.1:$2.1:$single_line_text_property.0.0">Single Line Text Property</span></label>
```

```
<legend class="hs-field-desc" style="display:block;" data-reactid=".hbspt-forms-0.1:$2.1:$single_line_text_property.1">Help text</legend>
```

```
<div class="input" data-reactid=".hbspt-forms-0.1:$2.1:$single_line_text_property.$single_line_text_property"><input id="single_line_text_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-input" type="text" name="single_line_text_property" value="" placeholder="Placeholder text" data-reactid=".hbspt-forms-0.1:$2.1:$single_line_text_property.$single_line_text_property.0"></div>
```

```
</div>
```

```
</fieldset>
```

```
<fieldset class="form-columns-1" data-reactid=".hbspt-forms-0.1:$3">
```

```
<div class="hs_multi_line_text_property hs-multi_line_text_property hs-fieldtype-textarea field hs-form-field" data-reactid=".hbspt-forms-0.1:$3.1:$multi_line_text_property">
```

```
<label id="label-multi_line_text_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="" placeholder="Enter your Multi-line Text Property" for="multi_line_text_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-reactid=".hbspt-forms-0.1:$3.1:$multi_line_text_property.0"><span data-reactid=".hbspt-forms-0.1:$3.1:$multi_line_text_property.0.0">Multi-line Text Property</span></label>
```

```
<legend class="hs-field-desc" style="display:none;" data-reactid=".hbspt-forms-0.1:$3.1:$multi_line_text_property.1"></legend>
```

```
<div class="input" data-reactid=".hbspt-forms-0.1:$3.1:$multi_line_text_property.$multi_line_text_property"><textarea id="multi_line_text_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-input" name="multi_line_text_property" placeholder="" data-reactid=".hbspt-forms-0.1:$3.1:$multi_line_text_property.$multi_line_text_property.0">Default value</textarea></div>
```

```
</div>
```

```
</fieldset>
```

```
<fieldset class="form-columns-1" data-reactid=".hbspt-forms-0.1:$4">
```



```

<div class="hs_dropdown_select_property hs-dropdown_select_property hs-
fieldtype-select field hs-form-field" data-reactid=".hbspt-forms-
0.1:$4.1:$dropdown_select_property">
    <label id="label-dropdown_select_property-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="" placeholder="Enter your Dropdown Select Property"
for="dropdown_select_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-
reactid=".hbspt-forms-0.1:$4.1:$dropdown_select_property.0"><span data-
reactid=".hbspt-forms-0.1:$4.1:$dropdown_select_property.0.0">Dropdown Select
Property</span></label>
    <legend class="hs-field-desc" style="display:none;" data-reactid=".hbspt-
forms-0.1:$4.1:$dropdown_select_property.1"></legend>
    <div class="input" data-reactid=".hbspt-forms-
0.1:$4.1:$dropdown_select_property.$dropdown_select_property">
        <select id="dropdown_select_property-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="hs-input" name="dropdown_select_property" data-
reactid=".hbspt-forms-
0.1:$4.1:$dropdown_select_property.$dropdown_select_property.0">
            <option value="" disabled="" selected="" data-reactid=".hbspt-forms-
0.1:$4.1:$dropdown_select_property.$dropdown_select_property.0.0">Please
Select</option>
            <option value="Item 1" data-reactid=".hbspt-forms-
0.1:$4.1:$dropdown_select_property.$dropdown_select_property.0.1:$Item 1">Item
1</option>
            <option value="Item 2" data-reactid=".hbspt-forms-
0.1:$4.1:$dropdown_select_property.$dropdown_select_property.0.1:$Item 2">Item
2</option>
        </select>
    </div>
</div>
</fieldset>
<fieldset class="form-columns-0" data-reactid=".hbspt-forms-0.1:$5">
    <div class="hs-richtext hs-main-font-element" data-reactid=".hbspt-forms-
0.1:$5.0">
        <h1>This is a form Header</h1>
    </div>

```

```

</fieldset>
<fieldset class="form-columns-1" data-reactid=".hbspt-forms-0.1:$6">
  <div class="hs_radio_select_property hs-radio_select_property hs-fieldtype-radio
field hs-form-field" data-reactid=".hbspt-forms-0.1:$6.1:$radio_select_property">
    <label id="label-radio_select_property-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="" placeholder="Enter your Radio Select Property"
for="radio_select_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-
reactid=".hbspt-forms-0.1:$6.1:$radio_select_property.0"><span data-reactid=".hbspt-
forms-0.1:$6.1:$radio_select_property.0.0">Radio Select Property</span></label>
    <legend class="hs-field-desc" style="display:none;" data-reactid=".hbspt-
forms-0.1:$6.1:$radio_select_property.1"></legend>
    <div class="input" data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property">
      <ul role="checkbox" class="inputs-list multi-container" data-
reactid=".hbspt-forms-0.1:$6.1:$radio_select_property.$radio_select_property.0">
        <li class="hs-form-radio" role="radio" data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option 1"><label
for="radio_select_property0-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-
form-radio-display" data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option 1.0"><input
id="radio_select_property0-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-
input " type="radio" name="radio_select_property" value="Option 1" aria-
labelledby="label-radio_select_property-f57de912-d759-4f67-ae76-efe001bbf926_5033"
data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option 1.0.0"><span data-
reactid=".hbspt-forms-0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option
1.0.1">Option 1</span></label></li>
        <li class="hs-form-radio" role="radio" data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option 2"><label
for="radio_select_property1-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-
form-radio-display" data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option 2.0"><input
id="radio_select_property1-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-
input " type="radio" name="radio_select_property" value="Option 2" aria-
labelledby="label-radio_select_property-f57de912-d759-4f67-ae76-efe001bbf926_5033"
data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option 2.0.0"><span data-

```

```

reactid=".hbspt-forms-0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option
2.0.1">Option 2</span></label></li>
    <li class="hs-form-radio" role="radio" data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option 3"><label
for="radio_select_property2-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-
form-radio-display" data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option 3.0"><input
id="radio_select_property2-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-
input " type="radio" name="radio_select_property" value="Option 3" aria-
labelledby="label-radio_select_property-f57de912-d759-4f67-ae76-efe001bbf926_5033"
data-reactid=".hbspt-forms-
0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option 3.0.0"><span data-
reactid=".hbspt-forms-0.1:$6.1:$radio_select_property.$radio_select_property.0.$Option
3.0.1">Option 3</span></label></li>
    </ul>
  </div>
</div>
</fieldset>
<fieldset class="form-columns-1" data-reactid=".hbspt-forms-0.1:$7">
  <div class="hs_multiple_checkboxes_property hs-multiple_checkboxes_property
hs-fieldtype-checkbox field hs-form-field" data-reactid=".hbspt-forms-
0.1:$7.1:$multiple_checkboxes_property">
    <label id="label-multiple_checkboxes_property-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="" placeholder="Enter your Multiple Checkboxes Property"
for="multiple_checkboxes_property-f57de912-d759-4f67-ae76-efe001bbf926_5033"
data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.0"><span data-
reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.0.0">Multiple
Checkboxes Property</span></label>
    <legend class="hs-field-desc" style="display:none;" data-reactid=".hbspt-
forms-0.1:$7.1:$multiple_checkboxes_property.1"></legend>
    <div class="input" data-reactid=".hbspt-forms-
0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property">
      <ul role="checkbox" class="inputs-list multi-container" data-
reactid=".hbspt-forms-
0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0">

```

```
<li class="hs-form-checkbox" role="checkbox" data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 1"><label for="multiple_checkboxes_property0-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-form-checkbox-display" data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 1.0"><input id="multiple_checkboxes_property0-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-input " type="checkbox" name="multiple_checkboxes_property" value="Option 1" aria-labelledby="label-multiple_checkboxes_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 1.0.0"><span data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 1.0.1">Option 1</span></label></li>
```

```
<li class="hs-form-checkbox" role="checkbox" data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 2"><label for="multiple_checkboxes_property1-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-form-checkbox-display" data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 2.0"><input id="multiple_checkboxes_property1-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-input " type="checkbox" name="multiple_checkboxes_property" value="Option 2" aria-labelledby="label-multiple_checkboxes_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 2.0.0"><span data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 2.0.1">Option 2</span></label></li>
```

```
<li class="hs-form-checkbox" role="checkbox" data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 3"><label for="multiple_checkboxes_property2-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-form-checkbox-display" data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 3.0"><input id="multiple_checkboxes_property2-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-input " type="checkbox" name="multiple_checkboxes_property" value="Option 3" aria-labelledby="label-multiple_checkboxes_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-reactid=".hbspt-forms-0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option 3.0.0"><span data-reactid=".hbspt-forms-
```

```

0.1:$7.1:$multiple_checkboxes_property.$multiple_checkboxes_property.0.$Option
3.0.1">Option 3</span></label></li>
    </ul>
  </div>
</div>
</fieldset>
<fieldset class="form-columns-1" data-reactid=".hbspt-forms-0.1:$8">
  <div class="hs_single_checkbox_property hs-single_checkbox_property hs-
fieldtype-booleancheckbox field hs-form-field" data-reactid=".hbspt-forms-
0.1:$8.1:$single_checkbox_property">
    <legend class="hs-field-desc" style="display:none;" data-reactid=".hbspt-
forms-0.1:$8.1:$single_checkbox_property.1"></legend>
    <div class="input" data-reactid=".hbspt-forms-
0.1:$8.1:$single_checkbox_property.$single_checkbox_property">
      <ul class="inputs-list" data-reactid=".hbspt-forms-
0.1:$8.1:$single_checkbox_property.$single_checkbox_property.0">
        <li class="hs-form-booleancheckbox" data-reactid=".hbspt-forms-
0.1:$8.1:$single_checkbox_property.$single_checkbox_property.0.0"><label
for="single_checkbox_property-f57de912-d759-4f67-ae76-efe001bbf926_5033"
class="hs-form-booleancheckbox-display" data-reactid=".hbspt-forms-
0.1:$8.1:$single_checkbox_property.$single_checkbox_property.0.0.0"><input
id="single_checkbox_property-f57de912-d759-4f67-ae76-efe001bbf926_5033"
class="hs-input" type="checkbox" name="single_checkbox_property" value="true" data-
reactid=".hbspt-forms-
0.1:$8.1:$single_checkbox_property.$single_checkbox_property.0.0.0.0"><span data-
reactid=".hbspt-forms-
0.1:$8.1:$single_checkbox_property.$single_checkbox_property.0.0.0.1">Single
Checkbox Property</span></label></li>
      </ul>
    </div>
  </div>
</fieldset>
<fieldset class="form-columns-1" data-reactid=".hbspt-forms-0.1:$9">

```

```

    <div class="hs_number_property hs-number_property hs-fieldtype-number field
hs-form-field" data-reactid=".hbspt-forms-0.1:$9.1:$number_property">
        <label id="label-number_property-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="" placeholder="Enter your Number Property"
for="number_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-
reactid=".hbspt-forms-0.1:$9.1:$number_property.0"><span data-reactid=".hbspt-forms-
0.1:$9.1:$number_property.0.0">Number Property</span></label>
        <legend class="hs-field-desc" style="display:none;" data-reactid=".hbspt-
forms-0.1:$9.1:$number_property.1"></legend>
        <div class="input" data-reactid=".hbspt-forms-
0.1:$9.1:$number_property.$number_property"><input id="number_property-f57de912-
d759-4f67-ae76-efe001bbf926_5033" class="hs-input" type="number"
name="number_property" value="" placeholder="" step="any" data-reactid=".hbspt-
forms-0.1:$9.1:$number_property.$number_property.0"></div>
    </div>
</fieldset>
<fieldset class="form-columns-1" data-reactid=".hbspt-forms-0.1:$10">
    <div class="hs_file_upload_property hs-file_upload_property hs-fieldtype-file field
hs-form-field" data-reactid=".hbspt-forms-0.1:$10.1:$file_upload_property">
        <label id="label-file_upload_property-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="" placeholder="Enter your File Upload Property"
for="file_upload_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-
reactid=".hbspt-forms-0.1:$10.1:$file_upload_property.0"><span data-reactid=".hbspt-
forms-0.1:$10.1:$file_upload_property.0.0">File Upload Property</span></label>
        <legend class="hs-field-desc" style="display:none;" data-reactid=".hbspt-
forms-0.1:$10.1:$file_upload_property.1"></legend>
        <div class="input" data-reactid=".hbspt-forms-
0.1:$10.1:$file_upload_property.$file_upload_property"><input
id="file_upload_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" class="hs-
input" type="file" name="file_upload_property" size="30" data-reactid=".hbspt-forms-
0.1:$10.1:$file_upload_property.$file_upload_property.0"></div>
    </div>
</fieldset>
<fieldset class="form-columns-1" data-reactid=".hbspt-forms-0.1:$11">

```

```

<div class="hs_date_picker_property hs-date_picker_property hs-fieldtype-date
field hs-form-field" data-reactid=".hbspt-forms-0.1:$11.1:$date_picker_property">
  <label id="label-date_picker_property-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="" placeholder="Enter your Date Picker Property"
for="date_picker_property-f57de912-d759-4f67-ae76-efe001bbf926_5033" data-
reactid=".hbspt-forms-0.1:$11.1:$date_picker_property.0"><span data-reactid=".hbspt-
forms-0.1:$11.1:$date_picker_property.0.0">Date Picker Property</span></label>
  <legend class="hs-field-desc" style="display:none;" data-reactid=".hbspt-
forms-0.1:$11.1:$date_picker_property.1"></legend>
  <div class="input" data-reactid=".hbspt-forms-
0.1:$11.1:$date_picker_property.$date_picker_property">
    <div class="hs-dateinput" data-reactid=".hbspt-forms-
0.1:$11.1:$date_picker_property.$date_picker_property.0">
      <input id="date_picker_property-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="hs-input" type="text" name="date_picker_property"
value="" readonly="" data-reactid=".hbspt-forms-
0.1:$11.1:$date_picker_property.$date_picker_property.0.0">
      <div class="hs-datepicker" style="position:absolute;z-index:10000;" data-
reactid=".hbspt-forms-0.1:$11.1:$date_picker_property.$date_picker_property.0.1">
        <div class="pika-single fn-date-picker is-hidden is-bound" style="position:
static; left: auto; top: auto;"></div>
      </div>
    </div>
  </div>
</fieldset>
<fieldset class="form-columns-1" data-reactid=".hbspt-forms-0.1:$12">
  <div class="hs_hidden_field hs-hidden_field hs-fieldtype-text field hs-form-field"
style="display:none;" data-reactid=".hbspt-forms-0.1:$12.1:$hidden_field">
    <label id="label-hidden_field-f57de912-d759-4f67-ae76-
efe001bbf926_5033" class="" placeholder="Enter your Hidden Field" for="hidden_field-
f57de912-d759-4f67-ae76-efe001bbf926_5033" data-reactid=".hbspt-forms-
0.1:$12.1:$hidden_field.0"><span data-reactid=".hbspt-forms-
0.1:$12.1:$hidden_field.0.0">Hidden Field</span></label>

```

```

    <legend class="hs-field-desc" style="display:none;" data-reactid=".hspt-
forms-0.1:$12.1:$hidden_field.1"></legend>

    <div class="input" data-reactid=".hspt-forms-
0.1:$12.1:$hidden_field.$hidden_field"><input name="hidden_field" class="hs-input"
type="hidden" value="" data-reactid=".hspt-forms-
0.1:$12.1:$hidden_field.$hidden_field.0"></div>

    </div>

</fieldset>

<noscript data-reactid=".hspt-forms-0.2"></noscript>

<div class="hs_submit hs-submit" data-reactid=".hspt-forms-0.5">

    <div class="hs-field-desc" style="display:none;" data-reactid=".hspt-forms-
0.5.0"></div>

    <div class="actions" data-reactid=".hspt-forms-0.5.1"><input type="submit"
value="Submit" class="hs-button primary large" data-reactid=".hspt-forms-
0.5.1.0"></div>

    </div>

    <div class="hubspot-link__container sproket" data-reactid=".hspt-forms-
0.6"><span data-reactid=".hspt-forms-0.6.1"><span data-
reactid=".hspt-forms-0.6.1.0">Create your own</span><span data-reactid=".hspt-
forms-0.6.1.1"> </span></span><a
href="https://app.hubspot.com/signup/marketing?hubs_medium=virality&hubs_cam
paign=hubspot-forms-virality&hubs_id=forms-branding-
control&hubs_source=preview.hs-
sites.com&intent=marketingFreeForms&opt_sidebar=forms" target="_blank"
class="hubspot-link" data-reactid=".hspt-forms-0.6.2"><span class="hubspot-link-text"
data-reactid=".hspt-forms-0.6.2.0">free form with HubSpot</span></a></div>

    <input name="hs_context" type="hidden"
value="{&quot;rumScriptExecuteTime&quot;:378.519999969285,&quot;rumServiceRes
ponseTime&quot;:556.6899999976158,&quot;rumFormRenderTime&quot;:2.03000003
2391399,&quot;rumTotalRenderTime&quot;:560.3000000119209,&quot;rumTotalReque
stTime&quot;:176.39999999664724,&quot;lang&quot;:&quot;en&quot;,&quot;pageUrl&q
uot;:&quot;https://preview.hs-
sites.com/_hcms/preview/template/multi?domain=undefined&portalId=6333443&a
mp;tc_deviceCategory=undefined&template_file_path=Coded%20files/test.html&a
mp;updated=1569336070447&quot;,&quot;source&quot;:&quot;FormsNext-static-
3.375&quot;,&quot;timestamp&quot;:1569336341672,&quot;userAgent&quot;:&quot;Mo

```



```
zilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90
Safari/537.36&quot;,&quot;referrer&quot;:&quot;https://app.hubspot.com/design-
manager/6333443/code/12514064886&quot;,&quot;originalEmbedContext&quot;:{&quot;
;formsBaseUrl&quot;:&quot;/_hcms/forms/&quot;,&quot;portalId&quot;:&quot;6333443&
quot;,&quot;formId&quot;:&quot;f57de912-d759-4f67-ae76-
efe001bbf926&quot;,&quot;formInstanceId&quot;:&quot;5033&quot;,&quot;pageId&quot;
:&quot;1&quot;,&quot;deactivateSmartForm&quot;:true,&quot;target&quot;:&quot;#hs_f
orm_target_my_form&quot;,&quot;contentType&quot;:&quot;landing-
page&quot;,&quot;formData&quot;:{&quot;cssClass&quot;:&quot;hs-form stacked hs-
custom-
form&quot;,&quot;hutk&quot;:&quot;2700394130afb3130850416114874868&quot;,&qu
ot;pageId&quot;:&quot;1&quot;,&quot;boolCheckBoxFields&quot;:&quot;single_checkb
ox_property&quot;,&quot;dateFields&quot;:&quot;date_picker_property&quot;,&quot;for
mInstanceId&quot;:&quot;5033&quot;,&quot;urlParams&quot;:{&quot;domain&quot;:&qu
ot;undefined&quot;,&quot;portalId&quot;:&quot;6333443&quot;,&quot;tc_deviceCategor
y&quot;:&quot;undefined&quot;,&quot;template_file_path&quot;:&quot;Coded
files/test.html&quot;,&quot;updated&quot;:&quot;1569336070447&quot;,&quot;formVali
dity&quot;:{&quot;multi_line_text_property&quot;:{&quot;valid&quot;:true},&quot;firstnam
e&quot;:{&quot;valid&quot;:true}},&quot;formTarget&quot;:&quot;#hs_form_target_my_f
orm&quot;,&quot;correlationId&quot;:&quot;271fac8d-d068-4df1-822c-
35c3d69088fa&quot;,&quot;contentType&quot;:&quot;landing-
page&quot;,&quot;hutk&quot;:&quot;2700394130afb3130850416114874868&quot;,&qu
ot;isHostedOnHubspot&quot;:true}" data-reactid=".hspt-forms-0.7"><iframe
name="target_iframe_f57de912-d759-4f67-ae76-efe001bbf926_5033"
style="display:none;" data-reactid=".hspt-forms-0.8"></iframe>
</form>
</div>
</span>
```

フォーム ID に基づくフォームのターゲット ング

フォーム ID に基づいて HubSpot フォームのスタイル設定またはターゲティングを行う必要がある場合は、`.hs-form-[id]` クラスをターゲットにすることをおすすめします。たとえば、`hs-form-6db82756-8353-4984-be34-6c093194290f` のようにします。同じように見えるものの、アンダースコアに続いて別の ID (ターゲット ID) が含まれるクラス名は変更される可能性があるため、ターゲットにしないでください。ここに具体的に記載するセレクターは問題なく使用できます。その他のセレクターは今後機能しなくなる可能性があります。 `type` のような標準的な要素属性はターゲットにしても問題ありません。

変更はできるだけ最小限にするよう努めていますが、すべてのユーザーに優れたエクスペリエンスを提供するために、フォームの HTML は変更される可能性があります。HubSpot フォームのマークアップ HTML を完全に制御する必要がある場合は、代わりに [Forms API](#) を使用してください。

```
/*  
When targeting a specific form, using the form id class, and .hs-form together  
is more specific and will override any .hs-form styling you add by default.  
*/
```

```
.hs-form-c1e8c8c9-fda9-4d96-b699-9af79ddbe47c.hs-form{  
  background:#cecece;  
  padding:15px;  
}
```

```
/*  
When targeting a specific field inside of a specific form, it's best to combine  
the more specific form selector above, with the input you'd like to target based  
on the styling forms css example.
```

```

*/

.hs-form-c1e8c8c9-fda9-4d96-b699-9af79ddbe47c.hs-form .hs-button.primary,
.hs-form-c1e8c8c9-fda9-4d96-b699-9af79ddbe47c.hs-form input[type="submit"],
.hs-form-c1e8c8c9-fda9-4d96-b699-9af79ddbe47c.hs-form input[type="button"] {

  background-color:red;

  border-radius:0;

  box-shadow:none;

  text-shadow:none;

}

```

フォームのスタイル設定

HubSpot フォームのスタイルを設定するには、標準属性セレクターを使用してフィールドをターゲットにします。よく使用されるセレクターを以下に示します。HubSpot フォームのみを明示的にスタイル設定するために、セレクターは `class="hs-form"` フォームタグ要素内にネストされます。すべてのフォームをスタイル設定する場合は、`.hs-form` を"body"で置き換えるか、ネストをすべて削除します。その他のセレクターについては、[ボイラープレート CSS ファイルを参照してください](#)。

```

/* All HubSpot Forms

=====

===== */

/* Form Field (selector for form field wrapper) */

```

```
.hs-form .hs-form-field {}

/* Descriptions (targets class applied to Help Text divs) */

.hs-form .hs-field-desc {}

/* Labels (selects field labels and error messages) */

.hs-form label {}

.hs-form .hs-form-field > label {}

/* Inputs (selectors for all inputs) */

.hs-form input[type="text"],
.hs-form input[type="password"],
.hs-form input[type="datetime"],
.hs-form input[type="datetime-local"],
.hs-form input[type="date"],
.hs-form input[type="month"],
.hs-form input[type="time"],
.hs-form input[type="week"],
.hs-form input[type="number"],
.hs-form input[type="email"],
.hs-form input[type="url"],
.hs-form input[type="search"],
.hs-form input[type="tel"],
```

```
.hs-form input[type="color"],
.hs-form input[type="file"],
.hs-form textarea,
.hs-form select {}

/* Inputs in focus (selectors for all inputs when clicked) */
.hs-form input[type="text"]:focus,
.hs-form input[type="password"]:focus,
.hs-form input[type="datetime"]:focus,
.hs-form input[type="datetime-local"]:focus,
.hs-form input[type="date"]:focus,
.hs-form input[type="month"]:focus,
.hs-form input[type="time"]:focus,
.hs-form input[type="week"]:focus,
.hs-form input[type="number"]:focus,
.hs-form input[type="email"]:focus,
.hs-form input[type="url"]:focus,
.hs-form input[type="search"]:focus,
.hs-form input[type="tel"]:focus,
.hs-form input[type="color"]:focus,
.hs-form input[type="file"]:focus,
.hs-form textarea:focus,
.hs-form select:focus {}
```

```
/* Multi-line inputs (selectors to target multi-line fields) */  
  
.hs-form textarea {}  
  
.hs-form textarea:focus {}  
  
/* Dropdowns (selectors for dropdowns) */  
  
.hs-form select {}  
  
.hs-form select:focus {}  
  
/* Multi-select (selectors for multi-select fields) */  
  
.hs-form form.hs-form .hs-form-field ul.inputs-list {}  
  
.hs-form form.hs-form .hs-form-field ul.inputs-list li input {}  
  
.hs-form input[type="radio"] {}  
  
.hs-form input[type="checkbox"] {}  
  
/* Required (selectors for fields, when they do not pass validation) */  
  
.hs-form input:focus:required:invalid,  
  
.hs-form textarea:focus:required:invalid,  
  
.hs-form select:focus:required:invalid {}  
  
.hs-form input:focus:required:invalid:focus,  
  
.hs-form textarea:focus:required:invalid:focus,
```

```

.hs-form select:focus:required:invalid:focus {}

/* Error message (selector for validation messages) */
.hs-form .hs-error-msgs label{}

/* Placeholder Text (styles the placeholder attribute text) */
::-webkit-input-placeholder { /* Webkit Browsers */}
:-moz-placeholder { /* Firefox 18- */}
::-moz-placeholder { /* Firefox 19+ */}
:-ms-input-placeholder { /* IE10 */}

/* Multi Column Form (selectors for fieldsets and field wrappers)

=====
===== */

.hs-form .hs-form fieldset.form-columns-1 {}
.hs-form .hs-form fieldset.form-columns-1 .hs-form-field {}

.hs-form .hs-form fieldset.form-columns-2 {}
.hs-form .hs-form fieldset.form-columns-2 .hs-form-field {}

.hs-form .hs-form fieldset.form-columns-3 {}
.hs-form .hs-form fieldset.form-columns-3 .hs-form-field {}

```

```
/* Submit buttons (selectors for all non-CTA buttons)
```

```
===== */
```

```
body .hs-button.primary,
```

```
body input[type="submit"],
```

```
body input[type="button"] {}
```

```
body .hs-button.primary:hover,
```

```
body input[type="submit"]:hover,
```

```
body input[type="button"]:hover {}
```

```
body .hs-button.primary:focus,
```

```
body input[type="submit"]:focus,
```

```
body input[type="button"]:focus {}
```

既定の埋め込みフォーム CSS

フォームを埋め込んで既定の CSS を無効にしないと、以下のコードは外部ページの head タグ内の style タグにレンダリングされます。

```
.hs-button-reset,
```



```
.hs-form .hs-button {  
  border: none;  
  font-family: Helvetica, Arial, sans-serif;  
    line-height: 1;  
  margin: 0;  
  outline: 0;  
  padding:0  
}
```

```
.base-hs-button-styles,
```

```
.hs-form .hs-button {  
  text-decoration: none;  
  cursor: pointer;  
  display: inline-block;  
  font-size: 12px;  
  font-weight: 700;  
  line-height: 12px;  
  padding: 7px 18px 7px;  
  position: relative;  
  text-align: center  
}
```

```
.base-hs-button-styles:hover,
```

```
.hs-form .hs-button:hover {  
  text-decoration: none  
}
```

```
input.hs-input[type=checkbox],
```

```
input.hs-input[type=radio] {  
  cursor: pointer
```

```
}  
  
input.hs-input,  
textarea.hs-input,  
select.hs-input {  
    display: inline-block;  
    width: 210px;  
    height: 18px;  
    padding: 4px;  
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
    font-size: 13px;  
    font-weight: normal;  
    line-height: 18px;  
    color: #666;  
    border: 1px solid #ccc;  
    -webkit-border-radius: 3px;  
    -moz-border-radius: 3px;  
    -ms-border-radius: 3px;  
    border-radius: 3px;  
}  
  
textarea.hs-input {  
    padding-top: 5px;  
}  
  
html.webkit textarea.hs-input {  
    padding-top: 6px;  
    padding-left: 5px;  
}
```

```
input.hs-input[type=checkbox],
input.hs-input[type=radio] {
  width: auto;
  height: auto;
  padding: 0;
  margin: 3px 0;
  line-height: normal;
  border: none
}
```

```
input.hs-input[type=file] {
  background-color: #fff;
  padding: initial;
  border: initial;
  line-height: initial;
  -webkit-box-shadow: none;
  -moz-box-shadow: none;
  box-shadow: none
}
```

```
input.hs-input[type=button],
input.hs-input[type=reset],
input.hs-input[type=submit] {
  width: auto;
  height: auto
}
```

```
input.hs-input[type=search] {
  -webkit-border-radius: 16px;
  -moz-border-radius: 16px;
```

```

-ms-border-radius: 16px;
border-radius: 16px;
background-image: url("/common_assets/static-2.158/img/form/search.png");
background-repeat: no-repeat;
padding-left: 26px;
padding-right: 6px;
padding-top: 5px;
padding-bottom: 3px;
background-position: 8px 8px
}

html.firefox input.hs-input[type=search] {
  position: relative;
  top: 1px
}

input.hs-input[type=search].transparent-image {
  background-image: url("/common_assets/static-2.158/img/form/search-transparent.png")
}

input.hs-input[type="search"]::-webkit-search-decoration,
input.hs-input[type="search"]::-webkit-search-cancel-button,
input.hs-input[type="search"]::-webkit-search-results-button,
input.hs-input[type="search"]::-webkit-search-results-decoration {
  display: none
}

select.hs-input,
input.hs-input[type=file] {

```

```

    height:27px
}

select[multiple].hs-input {
    height: inherit
}

textarea.hs-input {
    height: auto
}

.hs-input:-moz-placeholder {
    color: #bfbfbf
}

.hs-input::-webkit-input-placeholder {
    color: #bfbfbf
}

input.hs-input,
textarea.hs-input {
    -webkit-transition: border 0.2s linear, box-shadow 0.2s linear;
    -moz-transition: border 0.2s linear, box-shadow 0.2s linear;
    transition: border 0.2s linear, box-shadow 0.2s linear;
    -webkit-box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1);
    -moz-box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1);
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1)
}

input.hs-input:focus,

```

```

textarea.hs-input:focus {
  outline: none;
  border-color: rgba(82, 168, 236, 0.8);
  -webkit-box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1), 0 0 8px rgba(82, 168, 236, 0.6);
  -moz-box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1), 0 0 8px rgba(82, 168, 236, 0.6);
  box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1), 0 0 8px rgba(82, 168, 236, 0.6)
}

input.hs-input.error,
.hs-form div.field.error input,
.hs-form div.field.error textarea,
.hs-form div.field.error .chzn-choices,
textarea.hs-input.error {
  border-color: #c87872;
  -webkit-box-shadow: 0 0 3px rgba(171, 41, 32, 0.25);
  -moz-box-shadow: 0 0 3px rgba(171, 41, 32, 0.25);
  box-shadow: 0 0 3px rgba(171, 41, 32, 0.25)
}

input.hs-input.error:focus,
.hs-form div.field.error input:focus,
.hs-form div.field.error textarea:focus,
.hs-form div.field.error .chzn-choices:focus,
textarea.hs-input.error:focus {
  border-color: #b9554d;
  -webkit-box-shadow: 0 0 6px rgba(171, 41, 32, 0.5);
  -moz-box-shadow: 0 0 6px rgba(171, 41, 32, 0.5);
  box-shadow: 0 0 6px rgba(171, 41, 32, 0.5)
}

```

```
.input-mini.hs-input,  
input.mini.hs-input,  
textarea.mini.hs-input,  
select.mini.hs-input {  
  width:60px  
}
```

```
.input-small.hs-input,  
input.small.hs-input,  
textarea.small.hs-input,  
select.small.hs-input {  
  width:90px  
}
```

```
.input-medium.hs-input,  
input.medium.hs-input,  
textarea.medium.hs-input,  
select.medium.hs-input {  
  width:150px  
}
```

```
.input-large.hs-input,  
input.large.hs-input,  
textarea.large.hs-input,  
select.large.hs-input {  
  width:210px  
}
```

```
.input-xlarge.hs-input,
```

```
input.xlarge.hs-input,  
textarea.xlarge.hs-input,  
select.xlarge.hs-input {  
    width:270px  
}  
  
.input-xxlarge.hs-input,  
input.xxlarge.hs-input,  
textarea.xxlarge.hs-input,  
select.xxlarge.hs-input {  
    width:530px  
}  
  
.input-stretch.hs-input,  
input.stretch.hs-input,  
textarea.stretch.hs-input,  
select.stretch.hs-input {  
    box-sizing: border-box;  
    width:100%  
}  
  
textarea.hs-input.xxlarge {  
    overflow-y: auto  
}  
  
input.hs-input.span1,  
textarea.hs-input.span1,  
select.hs-input.span1 {  
    display: inline-block;  
    float: none;
```



```
width: 30px;
margin-left:0
}

input.hs-input.span2,
textarea.hs-input.span2,
select.hs-input.span2 {
  display: inline-block;
  float: none;
  width: 90px;
  margin-left:0
}

input.hs-input.span3,
textarea.hs-input.span3,
select.hs-input.span3 {
  display: inline-block;
  float: none;
  width: 150px;
  margin-left:0
}

input.hs-input.span4,
textarea.hs-input.span4,
select.hs-input.span4 {
  display: inline-block;
  float: none;
  width: 210px;
  margin-left:0
}
```

```
input.hs-input.span5,  
textarea.hs-input.span5,  
select.hs-input.span5 {  
    display: inline-block;  
    float: none;  
    width: 270px;  
    margin-left:0  
}
```

```
input.hs-input.span6,  
textarea.hs-input.span6,  
select.hs-input.span6 {  
    display: inline-block;  
    float: none;  
    width: 330px;  
    margin-left:0  
}
```

```
input.hs-input.span7,  
textarea.hs-input.span7,  
select.hs-input.span7 {  
    display: inline-block;  
    float: none;  
    width: 390px;  
    margin-left:0  
}
```

```
input.hs-input.span8,  
textarea.hs-input.span8,
```

```
select.hs-input.span8 {  
  display: inline-block;  
  float: none;  
  width: 450px;  
  margin-left:0  
}
```

```
input.hs-input.span9,  
textarea.hs-input.span9,  
select.hs-input.span9 {  
  display: inline-block;  
  float: none;  
  width: 510px;  
  margin-left:0  
}
```

```
input.hs-input.span10,  
textarea.hs-input.span10,  
select.hs-input.span10 {  
  display: inline-block;  
  float: none;  
  width: 570px;  
  margin-left:0  
}
```

```
input.hs-input.span11,  
textarea.hs-input.span11,  
select.hs-input.span11 {  
  display: inline-block;  
  float: none;
```

```
width: 630px;
margin-left:0
}

input.hs-input.span12,
textarea.hs-input.span12,
select.hs-input.span12 {
    display: inline-block;
    float: none;
    width: 690px;
    margin-left:0
}

input.hs-input.span13,
textarea.hs-input.span13,
select.hs-input.span13 {
    display: inline-block;
    float: none;
    width: 750px;
    margin-left:0
}

input.hs-input.span14,
textarea.hs-input.span14,
select.hs-input.span14 {
    display: inline-block;
    float: none;
    width: 810px;
    margin-left:0
}
```

```
input.hs-input.span15,  
textarea.hs-input.span15,  
select.hs-input.span15 {  
    display: inline-block;  
    float: none;  
    width: 870px;  
    margin-left:0  
}
```

```
input.hs-input.span16,  
textarea.hs-input.span16,  
select.hs-input.span16 {  
    display: inline-block;  
    float: none;  
    width: 930px;  
    margin-left:0  
}
```

```
input.hs-input[disabled],  
select.hs-input[disabled],  
textarea.hs-input[disabled] {  
    background-color: #f5f5f5;  
    border-color: #ddd;  
    cursor: not-allowed  
}
```

```
.help-inline,  
.help-block {  
    font-size: 11px;
```

```
    line-height: 18px;
    color: #818181
}

.help-inline {
    padding-left: 5px
}

.help-block {
    display: block;
    max-width: 600px
}

.inline-inputs {
    color: #818181
}

.inline-inputs span,
.inline-inputs input {
    display: inline-block
}

.inline-inputs input.mini {
    width: 60px
}

.inline-inputs input.small {
    width: 90px
}
```

```
.inline-inputs span {
  padding:0 2px 0 1px
}

.input-prepend input,
.input-append input {
  -webkit-border-radius: 0 3px 3px 0;
  -moz-border-radius: 0 3px 3px 0;
  -ms-border-radius: 0 3px 3px 0;
  border-radius:0 3px 3px 0
}

.input-prepend .add-on,
.input-append .add-on {
  position: relative;
  background: #f5f5f5;
  border: 1px solid #ccc;
  z-index: 2;
  float: left;
  display: block;
  width: auto;
  min-width: 16px;
  height: 18px;
  padding: 4px 4px 4px 5px;
  margin-right: -1px;
  font-weight: normal;
  line-height: 18px;
  color: #bfbfbf;
  text-align: center;
  text-shadow: 0 1px 0 #fff;
```

```
-webkit-border-radius: 3px 0 0 3px;  
-moz-border-radius: 3px 0 0 3px;  
-ms-border-radius: 3px 0 0 3px;  
border-radius:3px 0 0 3px  
}
```

```
.input-prepend .active,  
.input-append .active {  
  background: #bfe0bf;  
  border-color: #59ad59  
}
```

```
.input-append input {  
  float: left;  
  -webkit-border-radius: 3px 0 0 3px;  
  -moz-border-radius: 3px 0 0 3px;  
  -ms-border-radius: 3px 0 0 3px;  
  border-radius:3px 0 0 3px  
}
```

```
.input-append .add-on {  
  -webkit-border-radius: 0 3px 3px 0;  
  -moz-border-radius: 0 3px 3px 0;  
  -ms-border-radius: 0 3px 3px 0;  
  border-radius: 0 3px 3px 0;  
  margin-right: 0;  
  margin-left: -1px  
}
```

```
.hs-form {
```



```
margin-bottom:18px
}

.hs-form fieldset {
margin-bottom: 18px;
padding-top:18px
}

.hs-form fieldset legend {
display: block;
margin-left: 150px;
font-size: 19.5px;
line-height: 1;
color: #414141
}

.hs-form .clearfix,
.hs-form .field,
.hs-form .hs-form-field {
margin-bottom:18px
}

.hs-form .clearfix:after,
.hs-form .field:after,
.hs-form .hs-form-field:after {
content: "";
display: table;
clear: both
}
```

```
.hs-form label {
  padding-top: 5px;
  font-size: 13px;
  line-height: 18px;
  font-weight: bold;
  float: left;
  width: 130px;
  text-align: right;
  color: #414141
}

.hs-form .input {
  margin-left: 150px
}

.hs-form div.field.error {
  background: #fae5e3;
  padding: 6px 0;
  margin-bottom: 18px;
  -webkit-border-radius: 4px;
  -moz-border-radius: 4px;
  -ms-border-radius: 4px;
  border-radius: 4px
}

.hs-form div.field.error>label,
.hs-form div.field.error span.help-inline,
.hs-form div.field.error span.help-block {
  color: #9d261d
}
```

```
.hs-form div.field.error .input-prepend span.add-on,  
.hs-form div.field.error .input-append span.add-on {  
  background: #f4c8c5;  
  border-color: #c87872;  
  color: #b9554d  
}  
  
.hs-form .hsErrMsgContainer ul {  
  margin-top: 6px;  
  margin-bottom: 0px;  
  margin-left: 7px;  
  list-style: none  
}  
  
.hs-form .hsErrMsgContainer ul li {  
  font-size: 12px;  
  color: #a44e47  
}  
  
.hs-form .hsErrMsgContainer .hs-block-message {  
  min-width: 200px;  
  width:200px  
}  
  
.hs-form .hsErrMsgContainer .hs-block-message ul {  
  margin-top:0px  
}  
  
.hs-form .actions {
```

```

background: #f5f5f5;
margin-top: 18px;
margin-bottom: 18px;
padding: 17px 20px 18px 150px;
border-top: 1px solid #ddd;
-webkit-border-radius: 0 0 3px 3px;
-moz-border-radius: 0 0 3px 3px;
-ms-border-radius: 0 0 3px 3px;
border-radius:0 0 3px 3px
}

.hs-form .actions .secondary-action {
  float: right
}

.hs-form .actions .secondary-action a {
  line-height:30px
}

.hs-form .actions .secondary-action a:hover {
  text-decoration: underline
}

.hs-form .inputs-list {
  margin: 0 0 5px;
  width:100%
}

.hs-form .inputs-list>li {
  display: block;

```

```
padding: 0;
width:100%
}

.hs-form .inputs-list label {
  display: block;
  float: none;
  width: auto;
  padding: 0;
  line-height: 18px;
  text-align: left;
  white-space: normal;
  font-weight: normal
}

.hs-form .inputs-list label strong {
  color: #818181
}

.hs-form .inputs-list label small {
  font-size: 11px;
  font-weight: normal
}

.hs-form .inputs-list .inputs-list {
  margin-left: 25px;
  margin-bottom: 10px;
  padding-top:0
}
```

```
.hs-form .inputs-list:first-child {
  padding-top:6px
}

.hs-form .inputs-list>li+li {
  padding-top:2px
}

.hs-form .inputs-list input[type=radio],
.hs-form .inputs-list input[type=checkbox] {
  margin-bottom:0
}

.hs-form.stacked {
  padding-left:20px
}

.hs-form.stacked fieldset {
  padding-top:9px
}

.hs-form.stacked legend {
  margin-left:0
}

.hs-form.stacked label {
  display: block;
  float: none;
  width: auto;
  font-weight: bold;
```

```
text-align: left;
line-height: 20px;
padding-top: 0;
margin-bottom:4px
}

.hs-form.stacked .field {
margin-bottom:18px
}

.hs-form.stacked .field div.input {
margin-left:0
}

.hs-form.stacked .field div.input>input+a,
.hs-form.stacked .field div.input select+a,
.hs-form.stacked .field div.input textarea+a {
margin-top: 4px;
display: block
}

.hs-form.stacked .inputs-list {
margin-bottom:0
}

.hs-form.stacked .inputs-list>li {
padding-top:0
}

.hs-form.stacked .inputs-list>li label {
```

```
font-weight: normal;
padding-top:0
}

.hs-form.stacked div.error {
padding: 3px 10px 6px;
margin-top: 0;
margin-left: -10px;
margin-bottom:9px
}

.hs-form.stacked .actions {
margin-left: -20px;
padding-left:20px
}

.hs-form {
padding-left:0px
}

.hs-form.stacked {
padding-left:0px
}

.hs-form.stacked fieldset {
padding:0
}

.hs-form .hs-button {
color: #fff;
```



```

text-shadow: 0 -1px 0 #3574e3;
background-color: #3574e3;
background-image: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(0%, #7ec8f4), color-stop(100%, #3574e3));
background-image: -webkit-linear-gradient(top, #7ec8f4, #3574e3);
background-image: -moz-linear-gradient(top, #7ec8f4, #3574e3);
background-image: linear-gradient(top, #7ec8f4, #3574e3);
box-shadow: inset 0px 1px 0px rgba(255, 255, 255, 0.5);
border-top: 1px solid #64baf0;
border-bottom: 1px solid #1c4ed5;
border-right: 1px solid #468ee6;
border-left: 1px solid #468ee6;
border-radius: 4px;
-webkit-transition: opacity 0.15s linear;
-moz-transition: opacity 0.15s linear;
transition: opacity 0.15s linear;
-moz-user-select: none;
-webkit-user-select: none;
user-select: none
}

.hs-form .hs-button:visited,
.hs-form .hs-button:hover {
  color: #fff
}

.hs-form .hs-button:hover:not(.inactive),
.hs-form .hs-button:focus:not(.inactive),
.hs-form .hs-button.hovered:not(.inactive) {
  box-shadow: inset 0px 1px 0px rgba(255, 255, 255, 0.5), 0 0 5px #3574e3;
}

```

```

border-top: 1px solid #4db0ee;
border-bottom: 1px solid #1946be;
border-right: 1px solid #3080e3;
border-left: 1px solid #3080e3
}

.hs-form .hs-button:active:not(.inactive):not(.link),
.hs-form .hs-button.depressed:not(.inactive):not(.link),
.dropdown-open>.hs-form .hs-button:not(.inactive):not(.link) {
  background-color: #599eeb;
  background-image: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(0%,
#3574e3), color-stop(100%, #599eeb));
  background-image: -webkit-linear-gradient(top, #3574e3, #599eeb);
  background-image: -moz-linear-gradient(top, #3574e3, #599eeb);
  background-image: linear-gradient(top, #3574e3, #599eeb);
  box-shadow: inset 0px 1px 2px rgba(0, 0, 0, 0.3), inset 0px -1px 0px rgba(255, 255,
255, 0.3);
  border-top: 1px solid #1c4ed5;
  border-bottom: 1px solid #4084e2;
  border-right: 1px solid #468ee6;
  border-left: 1px solid #468ee6
}

.dropdown-open>.hs-form .hs-button:not(.inactive):not(.link) {
  box-shadow: inset 0px 1px 2px rgba(0, 0, 0, 0.3)
}

.hs-form .hs-button.disabled.disabled.disabled.disabled,
.hs-form .hs-button[disabled][disabled][disabled] {
  background-color: #6387c5;

```

```

background-image: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(0%,
#88b1ce), color-stop(100%, #6387c5));
background-image: -webkit-linear-gradient(top, #88b1ce, #6387c5);
background-image: -moz-linear-gradient(top, #88b1ce, #6387c5);
background-image: linear-gradient(top, #88b1ce, #6387c5);
box-shadow: inset 0px 1px 0px rgba(255, 255, 255, 0);
border-top: 1px solid #7baacc;
border-bottom: 1px solid #5774be;
border-right: 1px solid #6c94c7;
border-left: 1px solid #6c94c7;
cursor: default;
color: #ededed;
text-shadow: none;
opacity:0.6
}

```

```

.hs-form .hs-button.disabled.disabled.disabled.disabled.next:before,
.hs-form .hs-button[disabled][disabled][disabled].next:before {
opacity:0.5
}

```

```

.hs-form .hs-button.disabled.disabled.disabled.disabled.previous:before,
.hs-form .hs-button[disabled][disabled][disabled].previous:before {
opacity:0.5
}

```

```

.hs-form .hs-button.next:before,
.hs-form .hs-button.previous:before {
background: url(/style_guide/static-8.319/img/hs-button-arrows.png) center 0 no-
repeat;

```

```
display: block;
width: 16px;
height: 16px;
position: absolute;
top: 5px;
content: " ";
z-index:99
}

.hs-form .hs-button.next {
padding-right:32px
}

.hs-form .hs-button.next:before {
right:9px
}

.hs-form .hs-button.previous {
padding-left:32px
}

.hs-form .hs-button.previous:before {
background-position: center -16px;
left:9px
}

.hs-form .hs-button.inactive {
cursor: default
}
```

```
.hs-form ul {
  list-style: none
}

.hs-form label {
  display: block;
  font-family:"Helvetica Neue", Helvetica, Arial, sans-serif
}

.hs-form label.hidden {
  display: none
}

.hs-form .hs-field-desc {
  color: #aaa;
  margin: 0px 0px 5px 150px;
  font-size: 11px;
  font-family:"Helvetica Neue", Helvetica, Arial, sans-serif
}

.hs-form .hs-form-required {
  color: red
}

.hs-form .field {
  margin-bottom:9px
}

.hs-form .hs-richtext {
  margin-bottom: 3px;
```

```
font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
line-height: 18px;
font-size: 14px;
color: #414141
}

.hs-form .hs-richtext hr {
  text-align: left;
  margin-left: 0;
  width:91%
}

.hs-form .email-correction,
.hs-form .email-validation {
  padding-top: 3px;
  font-size: 12px;
  font-family:"Helvetica Neue", Helvetica, Arial, sans-serif
}

.hs-form .email-correction a,
.hs-form .email-validation a {
  cursor: pointer
}

.hs-form .inputs-list {
  padding-left: 5px;
  list-style: none
}

.hs-form .inputs-list li input {
```

```

margin:3px 5px 3px 0px
}

.hs-form input[type=checkbox],
.hs-form input[type=radio] {
margin-right:5px
}

.hs-form input:not([type="image"]),
.hs-form textarea {
box-sizing: content-box
}

.hs-form.stacked .hs-field-desc {
margin:0px 0px 2px 0px
}

.hs-form .hs-input,
.hs-form textarea.hs-input {
box-sizing: border-box;
max-width: 500px;
width:90%
}

.hs-フォーム.hs-入力:[type =change],
.hs-form .hs-input:not[type=radio],
.hs-form textarea.hs-input:not[type=checkbox],
.hs-form textarea.hs-input:not[type=radio] {
min-height: 18px;

```

```
min-width:100px
}

.hs-form .actions {
  background: transparent;
  margin-top: 18px;
  margin-bottom: 18px;
  padding: 17px 20px 18px 0px;
  border-top: none;
  -webkit-border-radius: 0 0 3px 3px;
  -moz-border-radius: 0 0 3px 3px;
  -ms-border-radius: 0 0 3px 3px;
  border-radius:0 0 3px 3px
}

.hs-form .actions .secondary-action {
  float: right
}

.hs-form .actions .secondary-action a {
  line-height:30px
}

.hs-form .actions .secondary-action a:hover {
  text-decoration: underline
}

.pika-single {
  z-index: 9999;
  display: block;
```



```
position: relative;
color: #333;
background: #fff;
border: 1px solid #ccc;
border-bottom-color: #bbb;
font-family:"Helvetica Neue", Helvetica, Arial, sans-serif
}

.pika-single.is-hidden {
  display: none
}

.pika-single.is-bound {
  position: absolute;
  box-shadow:0 5px 15px -5px rgba(0, 0, 0, 0.5)
}

.pika-single {
  *zoom:1
}

.pika-single:before,
.pika-single:after {
  content: " ";
  display: table
}

.pika-single:after {
  clear: both
}
```

```
.pika-lendar {
  float: left;
  width: 240px;
  margin: 8px;
}

.pika-title {
  position: relative;
  text-align: center;
}

.pika-title select {
  cursor: pointer;
  position: absolute;
  z-index: 9998;
  margin: 0;
  left: 0;
  top: 5px;
  filter: alpha(opacity=0);
  opacity: 0;
}

.pika-label {
  display: inline-block;
  *display: inline;
  position: relative;
  z-index: 9999;
  overflow: hidden;
  margin: 0;
```

```
padding: 5px 3px;
font-size: 14px;
line-height: 20px;
font-weight: bold;
background-color: #fff
}

.pika-prev,
.pika-next {
display: block;
cursor: pointer;
position: relative;
outline: none;
border: 0;
padding: 0;
width: 20px;
height: 30px;
text-indent: 20px;
white-space: nowrap;
overflow: hidden;
background-color: transparent;
background-position: center center;
background-repeat: no-repeat;
background-size: 75% 75%;
opacity: 0.5;
*position: absolute;
*top:0
}

.pika-prev:hover,
```

```

.pika-next:hover {
  opacity:1
}

.pika-prev.is-disabled,
.pika-next.is-disabled {
  cursor: default;
  opacity:0.2
}

.pika-prev,
.is-rtl .pika-next {
  float: left;
  background-image:
url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABQAAAAeCAYAAAAs
Ej5rAAAAUkIEQVR42u3VMQoAIBADQf8Pgj+OD9hG2CtONJB2ymQkKe0HbwAP0xuc
DiQWARITIDEBEnMgMQ8S8+AqBII6kKgHiXqQqAeJepBo/z38J/U0uAHlaBkBI9I4GwA
AAABJRU5ErkJggg==");
  *left:0
}

.pika-next,
.is-rtl .pika-prev {
  float: right;
  background-image:
url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABQAAAAeCAYAAAAs
Ej5rAAAAU0IEQVR42u3VOwoAMAgE0dwfAnNjU26bYkBCFGwfiL9VVWoO+BJ4Gf3gts
EKKoFBNTCoCAYVwaAiGNQGMUHMkjGbgjk2mIONuXo0nC8XnCf1JXgArVIZAqh5T
KYAAAAASUVORK5CYII=");
  *right:0
}

```

```
.pika-select {
  display: inline-block;
  *display: inline
}

.pika-table {
  width: 100%;
  border-collapse: collapse;
  border-spacing: 0;
  border:0
}

.pika-table th,
.pika-table td {
  width: 14.28571%;
  padding:0
}

.pika-table th {
  color: #999;
  font-size: 12px;
  line-height: 25px;
  font-weight: bold;
  text-align: center
}

.pika-table abbr {
  border-bottom: none;
  cursor: help
}
```

```
.pika-button {
  cursor: pointer;
  display: block;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
  outline: none;
  border: 0;
  margin: 0;
  width: 100%;
  padding: 5px;
  color: #666;
  font-size: 12px;
  line-height: 15px;
  text-align: right;
  background: #f5f5f5
}

.is-today .pika-button {
  color: #3af;
  font-weight: bold
}

.is-selected .pika-button {
  color: #fff;
  font-weight: bold;
  background: #3af;
  box-shadow: inset 0 1px 3px #178fe5;
  border-radius:3px
}
```

```
.is-disabled .pika-button {
  pointer-events: none;
  cursor: default;
  color: #999;
  opacity:0.3
}

.pika-button:hover {
  color: #fff !important;
  background: #ff8000 !important;
  box-shadow: none !important;
  border-radius: 3px !important
}

.pika-week {
  font-size: 11px;
  color: #999
}

.hs-form fieldset {
  border: 0;
  padding: 0;
  margin: 0;
  max-width:500px
}

.hs-form fieldset.form-columns-1 .hs-input {
  width:95%
}
```

```
.hs-form fieldset.form-columns-1 .input {
  margin-right:8px
}

.hs-form fieldset.form-columns-1 input[type="checkbox"],
.hs-form fieldset.form-columns-1 input[type="radio"] {
  width: auto
}

.hs-form fieldset.form-columns-2 .hs-form-field {
  width: 50%;
  float: left
}

.hs-form fieldset.form-columns-2 .input {
  margin-right:8px
}

.hs-form fieldset.form-columns-3 .hs-form-field {
  width: 32.7%;
  float: left
}

.hs-form fieldset.form-columns-3 .input {
  margin-right:8px
}

.hs-form fieldset .hs-form-field.hidden-label {
  margin-top:24px
}
```



```
}  
  
.hs-form .hs-social-login {  
  border-radius: 3px;  
  box-sizing: content-box;  
  width: 95%;  
  max-width: 480px;  
  height: 90px;  
}  
  
.hsformerror {  
  margin: 0 0 2px;  
  padding: 2px 6px;  
  height: auto;  
  background-color: #fdd2d0;  
  font-size: 11px;  
  border: 1px solid #fcb3af;  
  padding: 4px 16px 4px 10px;  
  color: #000;  
  display: none;  
  background-image: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(0%, #fefefe), color-stop(100%, #fdd2d0));  
  background-image: -webkit-linear-gradient(#fefefe, #fdd2d0);  
  background-image: -moz-linear-gradient(#fefefe, #fdd2d0);  
  background-image: -o-linear-gradient(#fefefe, #fdd2d0);  
  background-image: linear-gradient(#fefefe, #fdd2d0);  
  -webkit-border-radius: 4px;  
  -moz-border-radius: 4px;  
  -ms-border-radius: 4px;  
  -o-border-radius: 4px;
```

```
border-radius: 4px;
-webkit-box-shadow: 0 0 6px #ddd;
-moz-box-shadow: 0 0 6px #ddd;
box-shadow: 0 0 6px #ddd;
z-index:99999
}

.hsformerror em {
border: 10px solid;
border-color: #fdd2d0 transparent transparent;
bottom: -17px;
display: block;
height: 0;
left: 60px;
position: absolute;
width:0
}

.hsformerror p {
font-family: Lucida Grande, Lucida Sans Unicode, bitstream vera sans, trebuchet ms,
verdana, sans-serif;
margin: 0;
float: left;
margin-right:8px
}

.hsformerror:hover {
cursor: default
}
```

```

.hsformerror .close-form-error {
  float: right;
  display: inline;
  top: 3px;
  position: absolute;
  font-family: Verdana !important;
  color: #b17c79 !important;
  cursor: pointer !important;
  font-size: 11px !important;
  font-weight: normal !important
}

.hsformerror .close-form-error:hover {
  color: #cc8884
}

@media (max-width: 400px),
(min-device-width: 320px) and (max-device-width: 480px) {
  form.hs-form .form-columns-2 .hs-form-field,
  form.hs-form .form-columns-3 .hs-form-field {
    float: none;
    width:100%
  }
  form.hs-form .form-columns-2 .hs-form-field .hs-input,
  form.hs-form .form-columns-3 .hs-form-field .hs-input {
    width:95%
  }
}

@media (max-width: 422px) {

```

```
.all-social-types {
  height:100px
}

}

@media (max-width: 342px) {
  .hs-social-login {
    height:72px
  }
}
```

フォームの埋め込みコードの変更

フォームのスタイル設定の他に、埋め込みコードを変更してマークアップと動作を変更することもできます。埋め込みコード JavaScript の変更については、[こちらの記事](#)をご覧ください。

Forms API

マークアップを完全に制御するには、独自のマークアップを作成し、HubSpot の Forms API を使用して、[コンタクトデータを HubSpot に送信](#)できます。

フォームでの GDPR テキストのターゲティング

GDPR に関連するフォームでテキストをスタイル設定する場合の出発点として、以下の CSS を使用できます。チームが選択した同意オプションによっては、変更が必要となることがあります。

```
.legal-consent-container .hs-richtext
```

フォームのテーマと iframe

フォームエディターでは、CSS のスキルがなくても基本的なスタイルを設定できます。このために、フォームエディターの [スタイルとプレビュー] タブにテーマとスタイルオプションが用意されています。iframe を使用して HubSpot フォームがレンダリングされる場合には、この機能が使用されている可能性があります。

```
<!--[if lte IE 8]>
<script charset="utf-8" type="text/JavaScript" src="//js.hsforms.net/forms/v2-legacy.js"></script>
<![endif-->
<script charset="utf-8" type="text/JavaScript"
src="//js.hsforms.net/forms/v2.js"></script>
<script data-hubspot-rendered="true">
  hbspt.forms.create({
    portalId: "123456",
    formId: "fbb27ef9-9999-4c49-8663-bd5c05f684b0"
  });
</script><div class="hbspt-form" id="hbspt-form-1567055267882-127981741"><iframe
id="hs-form-iframe-0" class="hs-form-iframe" scrolling="no" height="155" width="873"
style="position: static; border: none; display: block; overflow: hidden; width: 873px;
height: 155px;"></iframe></div>
```

CSS を使用してスタイル設定を制御する場合は、フォームエディターに移動し ([マーケティング] > [リードの獲得] > [フォーム])、編集するフォームを見つけ、[スタイルとプレビュー] タブに移動し、[未加工の HTML フォームとして設定] を切り替えます。

これによりフォームが変更され、現在のページ内の HTML フォーム入力要素を使用してレンダリングされるようになります。

メニューのマークアップ

HubSpot メニューのマークアップ

メニューモジュールは、さまざまな設定とこれらの設定に関連付けられているメニューツリーに基づいて、テンプレートとページにナビゲーションメニューをレンダリングします。この記事では、メニューモジュールにより出力されるマークアップと、スタイル設定に使用できる CSS セレクターの一部について説明します。

メニューモジュールには、メニューモジュール (以前の詳細メニュー) とシンプルメニューモジュールの 2 種類があります。

主な違いは、メニュー情報のソースです。標準のメニューモジュールは、[\[コンテンツ設定\]](#) > [\[ナビゲーション\]](#) からメニュー構造を取得します。シンプルメニューモジュールは、テンプレートの既定メニュー構造、またはページレベルで設定されているメニュー構造 (設定されている場合) を表示します。

メニューのマークアップを完全に制御するには、メニューモジュールの代わりに [menu\(\)関数](#) を使用する必要があります。

メニューのマークアップの説明

HubSpot の他のモジュールと同様に、メニューモジュールはモジュールラッパーマークアップで囲まれています。div タグおよび span タグにより、コンテンツエディター UI でモジュールが編集できるようになり、またモジュールタイプのクラスと、ドラッグ&ドロップテンプレートの [レスポンシブグリッド](#) が組み込まれます。メニューモジュールとシンプルメニューモジュールのメニューマークアップは、ラッパーとメニュー div に適用される一部のクラスを除いて同じです。

実際のメニューは `class="hs-menu-wrapper"` div で囲まれている `ul` としてレンダリングされます。hs-menu-wrapper div には、[メニューモジュールの設定](#) に基づいて追加の

クラスがあります。ul 内の各メニュー項目は、li で囲まれた a タグです。li タグには、メニューツリー内の項目の深さを示すクラスがあります (例: `hs-menu-depth-1`)。

li 項目にネストした子メニューが含まれている場合、li 項目には追加のクラス `hs-item-has-children` が含まれています。子メニューは、クラス `hs-menu-children-wrapper` を使用してネストした ul としてレンダリングされます。メニューツリーに含まれているページを訪問すると、クラス `active-branch` が親 li 項目に追加され、クラス `active` がそのページの特定の li 項目に追加されます。

ここに示す高度なメニューツリーのマークアップを以下に示します。

メニューのマークアップの例

```
<div id="hs_cos_wrapper_my_menu" class="hs_cos_wrapper
hs_cos_wrapper_widget hs_cos_wrapper_type_module widget-type-menu" style=""
data-hs-cos-general-type="widget" data-hs-cos-type="module">
  <span id="hs_cos_wrapper_my_menu_" class="hs_cos_wrapper
hs_cos_wrapper_widget hs_cos_wrapper_type_menu" style="" data-hs-cos-general-
type="widget" data-hs-cos-type="menu">
    <div id="hs_menu_wrapper_my_menu_" class="hs-menu-wrapper active-branch
flyouts hs-menu-flow-horizontal" role="navigation" data-sitemap-name="default" data-
menu-id="12514065329" aria-label="Navigation Menu">
      <ul>
        <li class="hs-menu-item hs-menu-depth-1"><a href="https://www.hubspot.com"
role="menuitem">Top Level Item 1</a></li>
```



```
<li class="hs-menu-item hs-menu-depth-1"><a href="https://www.hubspot.com"
role="menuitem">Top Level Item 2</a></li>

<li class="hs-menu-item hs-menu-depth-1 hs-item-has-children" role="menu"><a
href="https://www.hubspot.com" aria-haspopup="true" aria-expanded="false"
role="menuitem">Top Level Item 3</a>

  <ul class="hs-menu-children-wrapper">

    <li class="hs-menu-item hs-menu-depth-2"><a href="https://www.hubspot.com"
role="menuitem">Second Level Item 1</a></li>

    <li class="hs-menu-item hs-menu-depth-2"><a href="https://www.hubspot.com"
role="menuitem">Second Level Item 2</a></li>

    <li class="hs-menu-item hs-menu-depth-2 hs-item-has-children"><a
href="https://www.hubspot.com" role="menuitem">Second Level Item 3</a></li>

  </ul>

</li>

</ul>

</div>

</span>

</div>
```

設定によって追加されるクラス

詳細メニューモジュールの設定は、HubSpot でレンダリングされるマークアップを制御し、特定のメニュー機能のスタイル設定に使用するクラスを追加します。[詳細メニュータイプ] 設定と [最大レベル数] 設定により、メニューマップでメニューツリーのどの部分がレンダリングされるかが制御されます。また、[メニュー] コントロールにより、使用する詳細メニューが選択されます。

[方向] 設定により、メニューラッパーdiv に **hs-menu-flow-horizontal** または **hs-menu-flow-vertical** が追加されます。このクラスを CSS セレクターで使用して、縦メニューと横メニューを異なる方法でスタイル設定できます。

[フライアウトを有効にしますか?] 設定により、クラス **flyouts** または **no-flyouts** がメニューラッパーdiv に追加されます。このクラスを CSS セレクターで使用して、ドロップダウンがあるメニューまたはドロップダウンがないメニューをターゲットにし、この設定に基づいて非表示にできます。

メニューのスタイル設定

HubSpot メニューのスタイル設定に使用できる CSS セレクターの例を以下に示します。HubSpot の [ボイラープレート CSS ファイル](#) で、CSS 宣言とともに使用されるいくつかのセレクターを確認できます。

```
/* Menu */
.hs-menu-wrapper ul {
  /* Targets all unordered lists within HubSpot menu */
}

/* Horizontal Menu
```

```
===== */
===== */

.hs-menu-wrapper.hs-menu-flow-horizontal ul {
/* Targets all unordered lists within horizontal menus */
}

.hs-menu-wrapper.hs-menu-flow-horizontal ul li{
/* Targets all list items within horizontal menus */
}

.hs-menu-wrapper.hs-menu-flow-horizontal ul li a{
/* Targets all links within horizontal menus */
}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul {
/* Targets the top-level unordered list within horizontal menus */
}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-menu-depth-1 {
/* Targets top-level list items within the top-level unordered list */
}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul li a {
/* Targets top-level list links within the top-level unordered list */
}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-item-has-children {
/* Targets list items with children with the top-level unordered list */
}
```

```

}

.hs-menu-wrapper.hs-menu-flow-horizontal.flyouts > ul li.hs-item-has-children ul.hs-
menu-children-wrapper {

/* Targets second-level unordered lists when flyouts are enabled (for styling dropdowns)
*/

}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-item-has-children ul.hs-menu-
children-wrapper li a {

/* Targets links within second-level unordered lists */

}

.hs-menu-wrapper.hs-menu-flow-horizontal.flyouts > ul li.hs-item-has-children ul.hs-
menu-children-wrapper li.hs-item-has-children ul.hs-menu-children-wrapper {

/* Targets third-level unordered lists (for styling dropdowns)*/

}

.hs-menu-wrapper.hs-menu-flow-horizontal.flyouts > ul li.hs-item-has-children:hover >
ul.hs-menu-children-wrapper {

/* Targets second-level unordered list when top-level menu item is hovered (use to
reveal dropdowns) */

}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-item-has-children.active-branch{

/* Targets top-level active branch unordered list */

}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-item-has-children.active-branch >
ul.hs-menu-children-wrapper {

/* Targets second-level unordered list within active branch */

}

```

```

.hs-menu-wrapper.hs-menu-flow-horizontal li.active a{
/* Targets the link within the active list item */
}

/* Vertical Menu
=====
===== */

.hs-menu-wrapper.hs-menu-flow-vertical ul {
/* Targets all unordered lists within vertical menus */
}

.hs-menu-wrapper.hs-menu-flow-vertical ul li a {
/* Targets all list items within vertical menus */
}

.hs-menu-wrapper.hs-menu-flow-vertical ul li a {
/* Targets all links within vertical menus */
}

.hs-menu-wrapper.hs-menu-flow-vertical > ul {
/* Targets the top-level unordered list within vertical menus */
}

.hs-menu-wrapper.hs-menu-flow-vertical > ul li.hs-menu-depth-1 > a {
/* Targets top-level links in vertical menus */
}

```

```

.hs-menu-wrapper.hs-menu-flow-vertical > ul li.hs-item-has-children {
/* Targets top-level list items with children */
}

/* No flyouts
=====
===== */

.hs-menu-wrapper.hs-menu-flow-vertical.no-flyouts .hs-menu-children-wrapper {
/* Targets child menus when flyouts are disabled */
}

.hs-menu-wrapper.hs-menu-flow-horizontal.no-flyouts > ul li.hs-item-has-children ul.hs-
menu-children-wrapper {
/* Targets second-level child menus when flyouts are disabled */
}

```

モバイルメニュー

CSS だけを使ってメニューをモバイル対応にできますが、マルチレベルのナビゲーションをサポートする動的モバイルメニューを作成するには、JavaScript または高度な CSS テクニックを使用する必要があります。JavaScript を使用したこのようなモバイルメニューの実装については、[このチュートリアル](#)で説明します。

デザインマネージャーユーザーガイド

デザインマネージャーを使ってみる

最終更新日:2019年7月30日

HubSpot のデザインマネージャーでは、デザイナーが美しくレスポンシブなウェブサイトを作成するために必要なすべてのツールを揃えているだけでなく、マーケターも直観的なインターフェイスで柔軟にコンテンツを作成、編集できるようになっています。

このガイドでは、デザインマネージャーの使用を開始するためのリソースのリストをご紹介します。デザインのインスピレーションについては、カスタマー送信の [デザイナーコレクション](#) をご確認ください。

ウェブ開発者の方は、[デザイナードキュメント](#) を参考にできます。また、[CMS Discovery Kit](#) に従って独自の開発を開始することもできます。独自に設計する際にサポートが必要になった場合は、[HubSpot パートナー](#) に協力を求めるか、[デザインコミュニティ](#) に投稿してください。

- [デザインマネージャーのクイックツアー](#)
- [レイアウトエディターでページ、Eメール、ブログのテンプレートを作成する方法](#)
- [テンプレートレイアウトの構成とカスタマイズの方法](#)
- [サイトのナビゲーションメニューのセットアップ方法](#)
- [ファイルマネージャーにサイトアセットをアップロードする方法](#)
- [テンプレートで既定のモジュールを使用する方法](#)
- [テンプレートに既定のコンテンツを追加する方法](#)
- [インスペクターを使用してテンプレートをスタイル設定する方法](#)
- [CSS ファイルを作成、編集、添付して、サイトのスタイルを設定する方法](#)

- [複数のテンプレートにわたってグローバルコンテンツを使用する方法](#)
- [システムテンプレートを使用して、エラーページ、サブスクリプションページ、パスワードプロンプトページをカスタマイズする方法](#)
- [モジュールを作成して編集する方法](#)
- [HubSpot でカスタムコードテンプレートを作成する方法](#)
- [HubSpot に JavaScript ファイルを追加する方法](#)
- [デザインマネージャーに関するよくある質問](#)
- [デザインマネージャー設定をカスタマイズする](#)

コンテンツをアクセシビリティ対応にする

コンテンツをアクセシビリティ対応にする

企業にとってウェブサイトの重要性が増し続ける中、これまでになく重要になっているのが、あらゆる訪問者がウェブサイトのコンテンツにアクセスできるようにすることです。米国では、これは [508 条](#)コンプライアンスと呼ばれることがよくあります。これは連邦政府関係機関が障がい者に電子情報や情報テクノロジーへのアクセスを可能にすることを要求する 1998 年の改訂リハビリテーション法の条文を指します。508 条コンプライアンスは適切なベースラインとは言え、ウェブ上でアクセシビリティ対応コンテンツを作成するための標準としてはしだいに [Web Content Accessibility Guidelines \(WCAG\)](#) が使用されるようになっていきます。

アクセシビリティは機能ではありません

法的側面が、訪問者に優れたユーザーエクスペリエンスを提供する動機付けであってはなりません。[CDCによると](#)、アメリカ人の4人に1人が何らかの障がいを持っています。優れたユーザーエクスペリエンスを提供する上で、訪問者の25%を無視するということは、店舗が4人の買い物客のうち1人に対して入店を拒否するようなものです。入店を拒否された買い物客は気分を害するだけでなく、再び店に足を運ぶことも、その店のサービスを他の人にすすめることもないでしょう。

開発者たちの間では「アクセシビリティは機能ではない」というフレーズがよく口に出されます。つまり、アクセシビリティを付加価値としてみなさないこと、あるいはプロジェクトの最後で対処することとして捉えないことを意味します。

アクセシビリティを確保するには、思慮、共感、理解が必要です。プロジェクトの早い段階でアクセシビリティに対処すれば、後で設計し直す必要に迫られずに、ソリューションを設計して開発できます。

多くの場合、アクセシビリティに対処すると、開発者としての他の優先事項にも対処できます。ユーザーエクスペリエンス、パフォーマンス、SEOなどの懸念事項はアクセシビリティと直接関係するからです。1つのことを改善すると、他の改善にもつながることはよくあります。

共感はアクセシビリティを確保する上で重要な部分であるものの、見落とされがちです。開発者はすべてのものを自動化しようと目指すものです。何かを処理するのが難しい、あるいは処理するには時間がかかる場合、それを自動的に処理するツールや、正しい処理方法を指示してくれるツールを求めます。けれども残念ながら、こうしたツールは表面をかじるだけに過ぎません。ツールは人間ではなく、真の意味で人間と同じ体験を共有することはできません。ほとんどの自動アクセシビリティテストに見事に合格するウェブページを作成するのは、技術的には可能です。けですが、技術だけを頼りに作成したウェブページは、視覚や聴覚に障がいを持つ人、色盲の人、運動機能が限られているか運動機能に障がいを持つ人も含め、万人にとって使い物になりませ

ん。要件を技術的に満たすことはできても、使いにくく操作するのが苛立たしいウェブページでは訪問者が遠ざかってしまいます。このことから言える重要な点は、ツールに依存せずに成果物をテストし、ユーザーに共感し、フィードバックを集めることです。

最終的なガイドではありませんが、ここではコンテンツをアクセスしやすいものにするための措置と、詳細を確認できるリソースを紹介します。

テキスト以外のコンテンツに代替テキストを提供する

すべての画像、アイコン、動画、ボタンに代替テキストが必要です。これは、スクリーンリーダーを使用してコンテンツを利用している訪問者に適していることに加え、検索エンジン対応にも有効です。代替テキストでは、画像の内容について説明する必要があります。サイトのほとんどの画像はページエディターで編集することが可能です。[ページエディターでは簡単に代替テキストを設定できます](#)。ただし、カスタムモジュールとテンプレートの内部で、ページエディターで指定した代替テキストが優先されることを確認する必要があります。

つまり、ユーザーが入力する代替テキストで `img` 要素が実際に使用されることを確認する必要があります。

Good:

```

```

Bad:

```

```

何らかの理由でページエディターで画像を編集できないようにしている場合は、代替テキストをハードコーディングしてください。

代替テキストに関するこのルールには例外が 1 つあります。画像に追加のコンテキストを用意せず、その画像をプレゼンテーション目的でのみ使用する場合は、alt 属性を省略するよりも、`"null" alt text` 値を使用することをおすすめします。

Good if only presentational:

```

```

動画の場合、キャプションをサポートするサービスを利用し、字幕の組み込みを検討します。字幕のアップロードをサポートするサービスには [Vidyard](#)、[Youtube](#)、[Vimeo](#)、[Wistia](#) があります。

すべてのリンクとフォーム入力に説明文があることを確認する

アイコンを使用している場合、スクリーンリーダー用テキストを提供します。たとえば、一部の標準 HubSpot テンプレートではアイコンショートコードが使用されています。

```
<a href="https://www.facebook.com"><i class="fab fa-facebook-f"></i><span class="screen-reader-text">Facebook</span></a>
```

フォームラベルやその他の説明文を非表示にするための `'display: none'` を使用しないでください。これを使用すると、視覚障がいを持つユーザーにテキストが読み上げられません。また、テキストは見えても、プレースホルダーのテキストとのコントラストを認識できないためにテキストを見分けにくいユーザーの気分を害することになります。ユーザーがフォームに入力していない場合やフォームが自動で入力される場合は言うまで

もなく、フォームのフィールドラベルが見にくいと、ユーザーはフォームが正しく入力されているかどうか見当が付きません。

どうしてもテキストを非表示にしなければならない場合、またはテキストを非表示にすると、視覚障がいのないユーザーには不要なコンテキストが追加される場合は、CSSを使用してスクリーンリーダーからテキストを隠さずに非表示することや、'.screen-reader-text'クラスを追加することができます。

```
.screen-reader-text {  
  clip: rect(1px, 1px, 1px, 1px);  
  height: 1px;  
  overflow: hidden;  
  position: absolute !important;  
  width: 1px;  
}
```

コンテンツリンクへのスキップを追加する

訪問者がスクリーンリーダーを使用している場合、ページの繰り返し部分（ヘッダーなど）をスキップする方法があると非常に役立ちます。これはページの主要なコンテンツをポイントするリンクを追加することで実現できます。

ヘッダーに、以下の内容のHTMLを追加します。

```
<a href="#content" class="screen-reader-text">Five ways to make your site  
more accessible</a>
```

次に、すべてのテンプレートにコンテンツのID属性を持つ要素が含まれていることを確認します。この例では、テキストとして移動先の記事のタイトルを使用しています。こうすると、検索エンジンがリンク先の内容を把握できます。これは、意味を理解してコンテンツに移動する方法となります。

```
<main id="content">
<!-- your page or post's actual content -->
</main>
```

セマンティックマークアップを使用する

セマンティックマークアップとは、要素の意味を伝える HTML のことです。セマンティック要素の例としては、`<header>` `<main>` `<footer>` `<nav>` `<time>` `<code>` `<aside>` `<article>`が挙げられます

セマンティックでない要素の例としては、`<div>` ``が挙げられます。

セマンティックでない要素にもそれなりに意味がありますが、これらの要素は主にプレゼンテーション目的で使用し、意味を伝えるためには使用しないでください。

セマンティック要素は、検索エンジンでその意味を理解できるので、SEO にポジティブな影響を与えるのにも、ユーザーエクスペリエンスを向上させるのにも役立つテクノロジーです。

aria 属性を使用する際、優れたユーザーエクスペリエンスを実現できると確信できる場合を除き、`<div>`要素をインタラクティブ要素 (ボタン、タブ、リンクなど) として使用しないでください。

リンク (`<a />`) と `<button />` を適切に使用してください。

リンクは、同じページ内のセクションにリンクする場合や他のページに移動する場合に使用します。

ボタンは、ページを離れたり id に移動したりすることにならないインタラクションに使用します。

テーブルを処理する場合、テーブルにその内容を説明するタイトルがあれば、そのタイトルを<table>要素の直後にある<caption>要素に含めます。

テーブルの列と行の見出しには、テーブルの適切な scope 属性を設定した<th>要素を使用し、各セクションを示すには<thead>、<tbody>、<tfoot>を使用します。

<th>属性に scope 属性を設定して、行または列のどちらの見出しであるかを示します。

アコーディオンが必要な場合は、<details>要素と<summary>要素を使用すると、アコーディオンの機能のほとんどを無償で使用できるだけでなく、使いやすいアコーディオンにすることができます。

キーボードナビゲーション

一部のユーザーは、キーボードを使用して任意のウェブページやフォームにナビゲーションするのにキーボードを使用します。ウェブサイトのナビゲーションを行う際に、訪問者によっては、キーボードを使用する必要がある場合や、キーボードをエミュレートするある種のサポートデバイスを使用する必要がある場合があります。

1. [':focus'](#)スタイルが設定されていることを確認してください。これにより、ユーザーがキーボードを使用してナビゲーションする際にどこにフォーカスしているかを確認できます。CSS の `outline` プロパティを無効にするのは、許容できる別のビジュアルインジケータを提供する場合に限ってください。[:focus-within](#) がユーザーの注意を向けるのに役立つ場合は、これを使用します。

2. `:hover`または JavaScript を使用してページの一部の表示/非表示を切り替える場合、ユーザーが JavaScript を使用してそのコントロールをナビゲーションできること、また、情報への代替パスがあることを確認してください。

必要に応じ、セマンティック要素を使用して役割にフォールバックする

テンプレートやコードファイルを作成する際は、ウェブブラウザとスクリーンリーダーに対し、要素にどのような内容が含まれているかを伝えるために、適切なセマンティック要素 (`<header>`、`<main>`、`<footer>`、`<nav>`など) を使用してください。

特定のシナリオで、セマンティック要素では意味が伝わる見込みが薄い場合は、メインのコンテンツブロックへ役割に関する属性を追加してください。それには、デザインマネージャーで (ドラッグ & ドロップテンプレートの) HTML をラッパーとして使用します。

ランドマークを追加すると、スクリーンリーダーを使用しているユーザーがウェブページの主要なセクションの間を移動しやすくなります。HubSpot は、メニュー、シンプルメニュー、Google 検索モジュールに「役割」属性を組み込みます。

デザインに関するその他のヒント

1. ユーザーがブラウザのズーム率を 200%にしても、コンテンツが表示されたままの状態を読みやすいことを確認してください。
2. フォントサイズは 16px 以上にすることをおすすめします。これより小さいと、訪問者がコンテンツを読みにくくなります。

3. 色だけで情報を伝えようとしないでください。色盲は世界人口に対して驚くほど高い割合を占めています。
4. 視力の弱いユーザーがコンテンツを読めるよう、テキストの色と背景色には十分なコントラストをとるようにしてください。
5. すばやく（1秒あたり3回以上）点滅するアニメーションは使用しないようにしてください。訪問者によっては発作を起こす原因となるおそれがあります。
6. サイトでアニメーションを使用する場合は、[prefers-reduced-motion](#)をサポートすることを検討してください。
7. 現在、レスポンシブなウェブサイトの構築は必須とみなされています。インタラクティブな要素のすべてがモバイル上で期待どおりに機能すること、タップ領域の間に十分なスペースが設けられていることを確認してください。

コンテンツの作成

テキストコンテンツでもアクセシビリティを考慮する必要があることは忘れがちです。

1. 「左側のサイドバーを参照」といった指示は使用しないようにしてください。ユーザーがモバイルデバイスを使用している場合、コンテンツは一般に画面内で積み上げられるため、このような指示は意味をなしません。
2. <h1> <h2>のような見出しを使用して、見出しを連続的にネストしてください。視覚効果のために見出しをスキップしないでください。
3. リンクを追加する場合は、「ここをクリック」というリンクテキストを使ってもスクリーンリーダーに意味は伝わりません。また、タッチスクリーンやその他のタイプのデバイスでは意味がありません。代わりに、テキストでそのリンク先の内容を説明してください。このようにすると、検索エンジンがリンク先の内容を把握できるため、SEOにも効果があります。

4. テキストに特殊なスタイルを設定して意味を伝える場合は、<mark> <sup>などの適切なセマンティックタグを使用するようにしてください。そうでないと、一部の訪問者には意味が伝わりません。
5. 世界中の訪問者をターゲットにする場合は、地域に固有の冗談を使用しないようにしてください。サイトが翻訳される場合、語呂合わせの使用は避ける必要があります。気の利いた冗談がどんなに好まれているとしても、うまく翻訳されることはほとんどありません。
6. 文法やスペルの誤りを避けるためのツールを使用してください。これによりテキストを理解しやすくなるだけでなく、翻訳もしやすくなります。
7. サイトのオーディエンスに合わせてコンテンツを作成してください。学識のある物理学者を相手にするのと同じように小学5年生の子供に天体物理学について説明することはしないはずで、響きが良いというだけの理由で難しい言葉を使うのではなく、[わかりやすさを優先](#)してください。

アクセシビリティに関する情報

アクセシビリティを念頭に置いたウェブサイトを構築するための実に優れたリソースがあります。これらのリソースをぜひご確認ください。

- [A11Yプロジェクト](#)
- [MDNのアクセシビリティに関するドキュメント](#)
- [Webアクセシビリティ Initiative](#)
- [WebAIM](#)
- [簡易版WCAG 2.1チェックリスト](#)
- [508条チェックリスト](#)
- [WCAG 2.1の推奨事項詳細](#)
- [WAVE](#) (ウェブページのアクセシビリティをテストするためのツール)

HubL リファレンス

HubL 構文

HubL 構文

HubSpot の CMS で使用される基礎的な構文は、HubSpot Markup Language または HubL (「Hubble」と発音) と呼ばれます。このテンプレート作成言語はテンプレート作成エンジン [Jinja](#) に基づいていますが、HubL では、[Jinja](#) の一部の機能をサポートしていない、HubSpot 独自のマークアップがかなり使用されます。

この記事では、HubL の機能と構文の基本について説明します。これは HubL を紹介する内容のドキュメントであり、HubL のさまざまな機能について詳しくは、これ以降のリファレンス記事で説明します。

区切り文字のタイプ

よく利用される他のテンプレート作成言語 (PHP など) と同様に、コードテンプレートファイルまたは HubL テンプレートモジュールで HubL を HTML に組み込むことができます。HubL の開始 i 位置と終了位置を示す区切り文字として機能するいくつかの主な記号を把握しておく必要があります。

`{% %}` - 文の区切り文字

`{{ }}` - 式の区切り文字

文

HubL 文は、編集可能なモジュールの作成、条件付きテンプレートロジックの定義、for ループのセットアップ、変数の定義などに使用します。文は"{%"で区切ります。

式

式は、テンプレートのコンテキストに格納されている値を出力します。式は“{{”で区切ります。たとえば、変数は文として定義する必要がありますが、変数を出力するには HubL 式が使用されます。

式の文

'do'タグは、何も出力しない点を除き、標準の変数式“{{ ... }}"と同様に機能します。これはリストや辞書を変更するときに使用できます。

注:配列に追加する場合には `.append()` を使用し、オブジェクトに追加する場合には `.update()` を使用することがあります。

```
# Arrays
{% set navigation = ['Home', 'About'] %}
{% do navigation.append('Contact Us') %}

# Objects
{% set book = {'name': 'Rocking HubL', 'author': 'John Smith'} %}
{% do demographics.update({'ebook': 'yes'}) %}
```

コメント

HubL を使用した開発で見かけ、使用することがある区切り文字の最後のタイプとして、HubL コメントがあります。コメントは"{#"により定義されます。

モジュール

モジュールは、エンドユーザーがコンテンツエディターを使用してカスタマイズできるテンプレートの動的領域です。たとえば、テンプレートファイルを新規にコーディングする場合には、HubL タグを使用してモジュールを作成し、コンテンツ作成者がページの特定の領域を編集できるようにします。

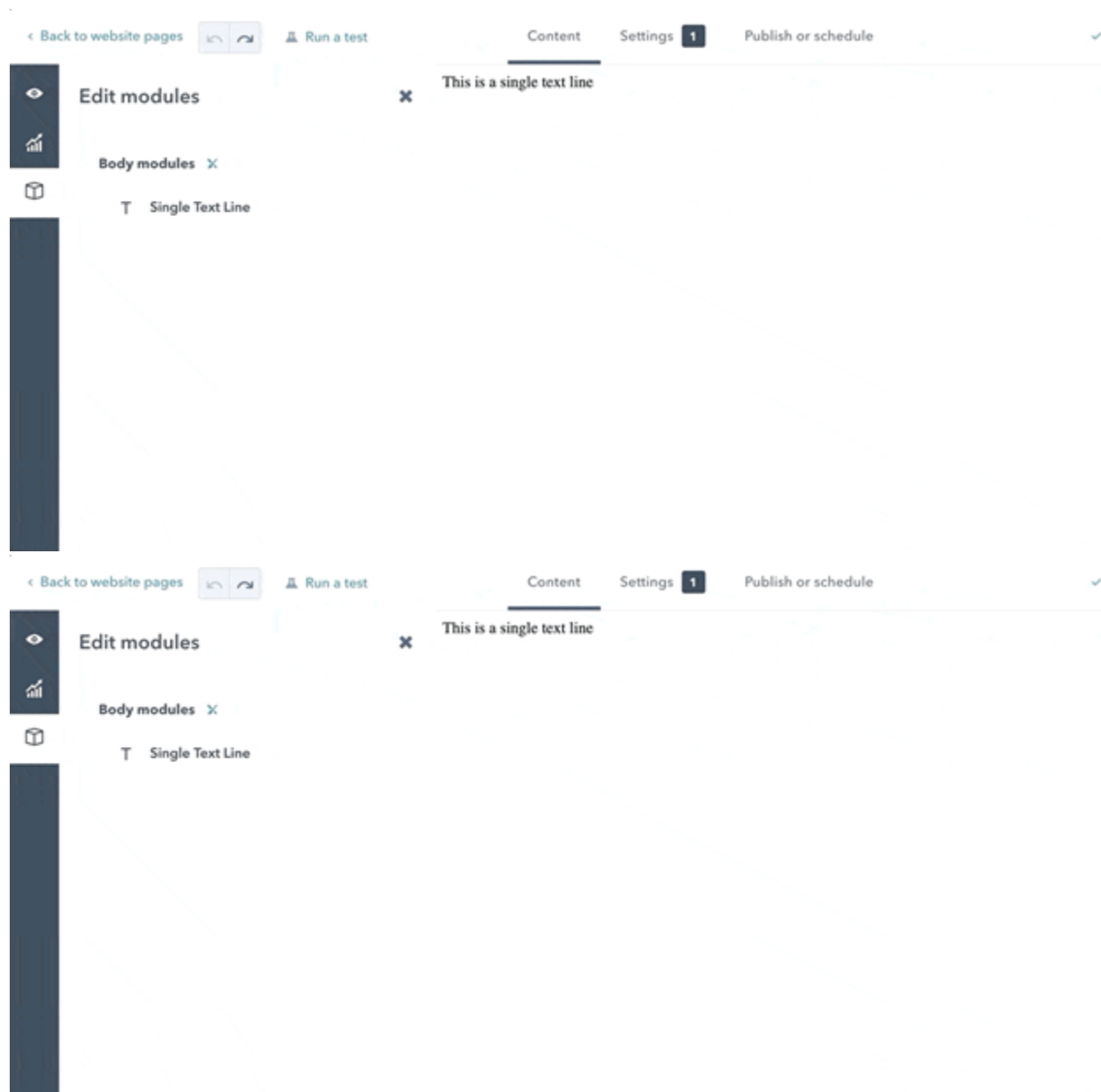
モジュールタグは以下のコンポーネントからなります。

- **モジュールのタイプ** - レンダリングするモジュールを指定します。使用可能な各種モジュールタイプのリストについては、[HubL でサポートされているタグページ](#)を参照してください。
- **モジュールの一意の名前** - テンプレートのコンテキストにおける一意の ID をモジュールに指定します。
- **パス (タグに応じて異なります)** - デザインマネージャーでのモジュールの位置を定義します。HubSpot の既定モジュールのパスは常に@hubspot/で始まり、その後モジュールのタイプが続きます。詳細については、以下の例と [HubL でサポートされているタグ](#)のページをご覧ください。
- **パラメーター (任意)** - 追加のモジュール情報を指定します。

- [HubL](#)
- [出力例](#)

```
{% module "unique_module_name" path="@hubspot/text" label="Single Text Line" value="This is a single text line" %}
```

上記の例は、label パラメーターと value パラメーターが指定されているテキストモジュールです。label はエディターのヘルプテキストを定義し、value はそのモジュールの既定のテキストを設定します。以下のサンプル gif で、エディター内部でこれがどのように表示されるかを確認してください。



モジュール構文の詳細については、[こちら](#)をご覧ください。すべてのモジュールタイプとそのパラメーターのリストを確認するには、[こちら](#)をご覧ください。

変数とマクロ

HubL には、アプリから役立つ値を出力する事前定義 HubSpot 変数が多数あります。

また、テンプレートで独自の変数を定義することもできます。以下の例では、

`primaryColor` という変数が文で定義されており、HubL 式により出力されます。この例では HubL 変数と CSS が混在しています。

- HubL
- 出力例

```
{% set primaryColor = '#F7761F' %} {# defines variable and assigns HEX color #}

a {
color: {{ primaryColor }}; {# prints variable HEX value #}
}
```

- HubL マクロでは、動的な値を使用して複数の文を出力できます。この手法は、同じ基本的なコードを繰り返し記述するが、変更する必要があるのが特定の値だけである場合に役立ちます。以下の例では、マクロを使用して、すべてのベンダープレフィックスが含まれている CSS3 transition プロパティを出力しています。マクロの詳細については、[こちらをご覧ください](#)。

- HubL
- 出力例

```
{% macro trans(value) %}
-webkit-transition: {{value}};
-moz-transition: {{value}};
-o-transition: {{value}};
-ms-transition: {{value}};
```

```
transition: {{value}};
{% endmacro %}

a {
  {{ trans('all .2s ease-in-out') }}
}
```

フィルターと関数

テンプレート変数の値を変換または変更するには、フィルターを HubL に追加できます。日付変数を形式設定するシンプルな例を以下に示します。フィルターでは | (パイプライン記号) が使用されており、これらのフィルターはスペースなしで変数に適用されます。

以下の例では、ブログ記事が 4 月 29 日に公開されていることを前提としています。記事の公開日の形式が、`datetimeformat` フィルターにより再設定されています。[すべてのフィルターの一覧はこちらをご覧ください。](#)

- [HubL](#)
- [出力例](#)

```
{{ content.publish_date_local_time|datetimeformat('%B %e, %Y') }}
```

フィルターは変数のレンダリング方法に影響しますが、関数は値とアカウント情報を処理してその情報をレンダリングします。たとえば、特定のブログの記事の合計数を取得する場合や、色変数を特定の量だけ明るくするかまたは暗くする場合には、関数を使用できます。

この例では、designers.hubspot.com/blog のブログ記事の合計数が出力されます。ブログ ID (ブログダッシュボードの URL で使用可能) パラメーターを使用して、ターゲットにするブログを指定します。すべての関数の一覧はこちらをご覧ください。

```
{{ blog_total_post_count(359485112) }}
```

If 文

If 文では、条件文を使用して、テンプレートでどのように "if"、"elif"、"else"、および "endif" の HubL 文で条件文をレンダリングするかを指示できます。if 文は "if" で始まり "endif" で終わります。この記事では、if 文についての詳細な情報を参照できます。

エンドユーザーがドロップダウンから部門を選択できるようにする choice モジュールを定義する例を以下に示します。ユーザーがエディターで選択した内容に基づいて、テンプレートにより採用情報ページの動的見出しが生成されます。この例では export_to_template_context パラメーターを使用する必要があります。このパラメーターの詳細については、こちらをご覧ください。

```
{% choice "department" label='Choose department', value='Marketing',  
choices='Marketing, Sales, Dev, Services' export_to_template_context=True %}
```

```
{% if widget_data.department.value == 'Marketing' %}
```

```
<h3>Want to join our amazing Marketing team?!</h3>
```

```
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}  
team.</h4>
```

```
{% elif widget_data.department.value == 'Sales' %}
```



```
<h3>Are you a Sales superstar?</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% elif widget_data.department.value == 'Dev' %}

<h3>Do you love to ship code?</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% else %}

<h3>Want to work with our awesome customers?</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% endif %}
```

テンプレートに記述された条件文は、コンタクトデータベースの情報ではなく、テンプレートレベルで使用可能な情報のみをベースにできる点に注意してください。特定のコンタクトのページのレンダリング方法を変更するには、[スマートコンテンツ](#)を使用します。

for ループ

for ループを使用すると、格納されている項目を順次繰り返し処理できます。ブログコンテンツをリスト形式でレンダリングする場合に最もよく使用されます。for ループは、"for"文で始まり"endfor"文で終わります。

以下の例では、熊の種類が変数"bears"として格納されます。for ループを使用してさまざまな熊の種類が繰り返し処理され、種類ごとにリスト項目が出力されます。for ループの詳細については[こちら](#)をご覧ください。

- HubL
- 出力例

```
{% set bears = ['panda bear', 'polar bear', 'black bear', 'grizzly bear', 'koala bear'] %}

<h1>Types of bears</h1>
<ul>
  {% for bear in bears %}
  <li>{{ bear }}</li>
  {% endfor %}
</ul>
```

その他の HubL 機能

HubL を使用した開発の際に役立つ可能性のあるその他のテンプレート機能を以下で説明します。

HubL 区切り文字のエスケープ

他の多くの言語では HubL と同じ区切り文字が使用されており、これが原因で CMS でコードファイル进行处理する際に問題が生じることがあります。別の言語で HubL の区切り文字を使用する場合は、そのコードを囲む必要があります。

```
{% raw %} {{Code you want to escape}} {% endraw %}
```

ファイルに別のファイルをインクルードする

include タグを使用して、複数の.html ファイルを 1 つの HubL テンプレートにインクルードできます。標準テンプレート変数を必要としない HTML ファイルを作成するには、[このテンプレートを新しいコンテンツで利用できるようにする] オプションをオフにする必要があります。グローバルグループをインクルードするには、デザインマネージャーURL からグループの ID をコピーし、ディレクトリ `generated_global_groups` からインクルードします。以下に構文を示します。

```
{% include "custom/page/web_page_basic/my_footer.html" %}
{% include 'generated_global_groups/2781996615.html' %}
{% include "hubspot/styles/patches/recommended.css" %}
```

また、同じ include タグを使用して、複数の CSS ファイルを 1 つの CSS ファイルにコンパイルできます。親ファイルを公開すると、親のコードを使用して子ファイルが 1 つの縮小 CSS ファイルにコンパイルされます。複数の CSS ファイルを 1 つのファイルにコンパイルする場合には、変更を反映するため、子ファイルの変更後に親ファイルを再公開する必要があります。

ブロックと拡張

複雑なテンプレートの作成時に使用できるより高度な手法として、マスターテンプレートを拡張するコンパートメント化されたブロックを作成する方法があります。必要な `standard_header_includes` 変数と `standard_footer_includes` 変数を含むマスターテンプレートを作成できます。そのテンプレート内で、以下の構文を使用して一意のブロックを定義する必要があります (`my_sidebar` は一意の名前です)。

次に、そのブロックにデータを提供する子 HTML ファイルを作成します。まず、親のパスを参照する `extends` 文を宣言する必要があります。このコードブロックはマスターテンプレートでレンダリングされますが、別の小さな管理しやすいファイルで保持さ

れます。この手法はすべてのユーザー向けではありませんが、複雑な E メールテンプレートやページテンプレートをコーディングする際に、整理された状態を維持する上で役立ちます。この手法を使用する場合、コンテンツの作成時に子テンプレートを選択する必要があります。

```
{% extends "custom/page/web_page_basic/my_template.html" %}
{% block my_sidebar %}
<h3>Sidebar title</h3>
<ul>
  <li>Bullet 1</li>
  <li>Bullet 2</li>
  <li>Bullet 3</li>
</ul>
{% endblock %}
```

HubL モジュールの構文とパラメータ

HubL モジュールの構文とパラメータ

モジュールとは、HubSpot ページまたは Eメールの編集可能な領域です。ドラッグ & ドロップテンプレートビルダーを使用してテンプレートレイアウトにモジュールを追加することや、HubL を使用してモジュールを作成することができます。HubL モジュー

ルは、コードテンプレートファイル、HubL テンプレートモジュール、またはブログコンテンツモジュールで定義できます。すべてのモジュールとそのパラメーターのリストを以下に示します。

基本的なモジュール構文

モジュールタグは以下のコンポーネントからなります。

- **モジュールのタイプ** - レンダリングするモジュールを指定します。使用可能な各種モジュールタイプのリストについては、[HubL でサポートされているタグ](#) ページを参照してください。
- **モジュールの一意の名前** - テンプレートのコンテキストにおける一意の ID をモジュールに指定します。
- **パス (タグに応じて異なります)** - デザインマネージャーでのモジュールの位置を定義します。HubSpot の既定モジュールのパスは常に @hubspot/ で始まり、その後モジュールのタイプが続きます。詳細については、以下の例と [HubL でサポートされているタグ](#) のページをご覧ください。
- **パラメーター (任意)** - 追加のモジュール情報を指定します。

```
{% module "unique_module_name" path="@hubspot/module_type"
parameterString='String parameter value', parameterBoolean=True %}

{# Sample of a default HubSpot text module #}

{% module "job_title" path="@hubspot/text", label="Job Title", value="Chief Morale
Officer" %}

{# Sample of a custom module #}

{% module "faqs"
path="/Marketplace/HubSpotSiteSetup/Vast_Site_Setup/Custom_Modules/Vast FAQ
Module", label="Vast FAQ Module" %}
```

HubL モジュールタグは"`{%}`"で区切られており、モジュールタイプと一意の名前を指定する必要があります。モジュールのタイプの後に、一意の名前を引用符で囲んで指定する必要があります。モジュール名には、スペースまたはダッシュではなく下線を使用する必要があります。

この一意の名前は、ページ/Eメールエディターで入力されたコンテンツと、対応する HubL モジュールタグをマッチングするときに使用されます。たとえば、テンプレートの 2 か所のエリアで同名の HubL モジュールタグをコーディングする場合、ユーザーがエディターで編集するモジュールは 1 つだけですが、そのモジュールに対する変更は両方のエリアに適用されます。

モジュールタグの 2 つの必須コンポーネントの他に、HubL モジュールでは、モジュールの動作やモジュールのレンダリング方法を指定する各種パラメーターがサポートされています。パラメーターはキーと値のペアであり、カンマで区切られます。パラメーターには、文字列、ブーリアン、整数、列挙、JSON リストなどのさまざまなタイプがあります。文字列パラメーターでは値を引用符で囲みますが*、ブーリアンパラメーターではその True 値と False 値を引用符で囲む必要はありません。label パラメーターと value パラメーターが指定されている基本的なテキストモジュールの例を以下に示します。

*文字列パラメーター値は一重引用符または二重引用符で囲むことができます。このドキュメントでは、一意のモジュール名は常に二重引用符で囲まれ、パラメーター値は一重引用符で囲まれています。この構文は、パラメーター値に複数の属性を持つ HTML マークアップが含まれている場合に役立ちます。ブーリアンパラメーター値は、読みやすくするため最初の文字が大文字で示されています。

- HubL
- 出力例

```
{% module "simple_section_title" path="@hubspot/text" label='Enter text for this section title', value='This is a section title' %}
```

ブロック構文

ほとんどのモジュールには既定のコンテンツを制御するパラメーターがありますが、モジュールの既定のコンテンツに大きなコードブロックを追加する必要がある場合があります。たとえば、リッチテキストまたは HTML モジュールの既定のコンテンツとして大きな HTML ブロックを含めることがあります。value パラメーターにこのコードを記述する代わりに、HubL ブロック構文を使用できます。

```
{% widget_block type_of_module "unique_module_name" parameterString='String parameter value', parameterBoolean=True %}  
  {% widget_attribute "parameter_of_default_content" %}  
    Block of default content  
  {% end_widget_attribute %}  
{% end_widget_block %}
```

`type_of_module` は、`rich_text` や `raw_html` などのモジュールタイプに置き換える必要があります。追加パラメーターは HubL の先頭行に追加できます。2 行目で、ブロックのコンテンツが適用されるパラメーターを定義します。たとえば、`rich_text` モジュールではこれは `html` パラメーターです。raw_html モジュールでは、これは `value` パラメーターです (この 2 つの例については以下を参照してください)。

- HubL
- 出力例

```
{% widget_block rich_text "my_rich_text_module" overrideable=True, label='My rich-text module' %}
```

```

{% widget_attribute "html" %}
  <h2>New Module</h2>
  <p>Add content here.</p>
{% end_widget_attribute %}
{% end_widget_block %}

{% widget_block raw_html "my_custom_html_module" overrideable=True, label='My
custom HTML module' %}
  {% widget_attribute "value" %}
  <div>My HTML block</div>
  {% end_widget_attribute %}
{% end_widget_block %}

```

content_attribute

標準構文とブロック構文の他に、事前定義のコンテンツ変数の既定コンテンツとして大きなブロックを指定することがあります。この最も一般的な例として、

content.email_body 変数があります。この変数は、コンテンツエディターで変更できる標準の E メール本文を出力します。これは標準 HubL モジュールではないので、**content_attribute** タグを使用して既定コンテンツのブロックを指定します。

email_body 変数に既定のコンテンツコードブロックが取り込まれる例を以下に示します。

```

{% content_attribute "email_body" %}
  <p>Hi {{ contact.firstname }},</p>
  <p>Describe what you have to offer the customer.Why should they read?What did you promise them in the subject line?Tell them something cool.Make them laugh.Make them cry.Well, maybe don't do that...</p>
  <p>Use a list to:</p>

```



```

<ul>
  <li>Explain the value of your offer</li>
  <li>Remind the reader what they'll get out of taking action</li>
  <li>Show off your skill with bullet points</li>
  <li>Make your content easy to scan</li>
</ul>
<p><a href="http://hubspot.com">LINK TO A LANDING PAGE ON YOUR
SITE</a> (This is the really important part.)</p>
<p>Now wrap it all up with a pithy little reminder of how much you love them.</p>
<p>Aw.You silver-tongued devil, you.</p>
<p>Sincerely,</p>
<p>Your name</p>
{% end_content_attribute %}

```

ブロック構文での content_attribute の使用

大きなモジュールでは、content_attribute とブロック構文の両方を使用して、既定のコンテンツを読みやすくできます。widget_block の中で content_attribute を使用できます。その後 content_attribute を "is_json=True" に設定します。これにより、パラメーターが JSON 形式で設定されます。

```

{% widget_block social_sharing "module_12345" %}
  {% content_attribute "email" is_json=True %}
  {
    "default": true,
    "enabled": false,
    "img_src": "https://..."
  }
  {% end_content_attribute %}
{% end_widget_block %}

```

すべてのモジュールで使用可能なパラメーター

一部の特殊な変数は使えるモジュールが限定されていますが、すべてのモジュールでサポートされる変数もあります。すべてのモジュールでサポートされる変数は以下のとおりです。

パラメーター名	タイプ	説明	既定値
label	string	コンテンツエディターでモジュールの内部名を指定します。このパラメーターは、ユーザーに追加の指示を与える場合にも使用できます。	なし
overrideable	boolean	モジュールをコンテンツエディターで編集できるかどうかを制御します。このパラメーターは、テンプレートビルダーのUIでモジュールをロックする機能と同様に使用できます。	True

no_wrapper

boolean

no_wrapper の値が True に設定されると、モジュールのコンテンツのラッパーであるマークアップが削除されます。ページに対するモジュールの出力は、特殊なクラスで常にラップされており、このラッパーであるマークアップが含まれることで、プレビューペインでモジュールをクリックすると、エディターがそのモジュールまでスクロールします。テキストモジュールを使ってアンカータグの href 属性のリンク先を指定する場合など、ラッパーを削除する必要がある場合があります。

False

extra_classes	string	<p>extra_classes パラメーターを追加すると、これらのクラスがモジュールコンテンツのラッピングに追加されます。複数のクラスを一括で追加するには、クラスをスペースで区切って指定します (例: extra_classes='full-width panel')。</p>	なし
export_to_template_context	boolean	<p>True の場合は HTML をレンダリングする代わりに、このウィジェットのパラメーターがテンプレートコンテキストで使用可能になります。HubL でパラメーターを取得するには、widget_data タグを使用します。 このパラメーターとウィジェットデータタグの使い方については、この記事をご覧ください。</p>	False

<code>unique_in_loop</code>	boolean	このパラメーターは、モジュールがループ内で定義されている場合に、 <code>loop.index</code> に一意のモジュール名が付加するときに使用できます。True に設定されている場合は、ループの繰り返しごとにモジュールの異なるバージョンを出力できます。 このパラメーターの用途についてはこちらをご覧ください。	False
-----------------------------	---------	---	-------

すべてのモジュールタイプとそのパラメーターのリストを確認するには、[ここをクリックしてください。](#)

変数とマクロ構文

HubL 変数とマクロ構文

HubL では、値の保存やテンプレートへの出力に変数を使用します。変数はテンプレートのロジックで使用することも、for ループで繰り返し使用することもできます。テンプレートで繰り返し使用されるコードの動的な部分出力するには、変数のほかに、マクロを使用することもできます。

変数は"}}"で囲まれた式です。変数の基本的な構文は以下のとおりです。

```
{{ variable }}
{{ dict.attribute }}
```

変数

変数は、式の中の単一の語、または辞書の属性のいずれかです。HubL では、辞書 (**dict**) と呼ばれる Python ベースのデータ構造を使用して、さまざまな変数のセットを保存します。たとえば、HubSpot では辞書「content」を使用して、そのテンプレートで作成されたコンテンツに関する多くの属性を格納しています。

content.absolute_url と指定した場合、特定のコンテンツの URL が出力されます。

HubSpot には、ページやブログ、Eメールのテンプレートに使用できる多数の定義済み変数が用意されています。サポートされる変数すべてのリファレンス一覧については、[こちら](#)を参照してください。ポータルページを参照している場合には、[Developer Info](#) も表示されます。

変数の値や辞書の属性をテンプレートに出力するほかに、ユーザー独自の変数を定義することもできます。1つの変数内に文字列、Boolean、整数、シーケンスを格納したり、辞書を作成したりできます。変数は、文の区切り文字の内側で、「set」というワードを使って定義します。格納された変数を出力するには、変数名を式として記述します。以下では、さまざまなタイプの変数に格納し、出力しています。

変数は1つの単語にするか、スペースの代わりに下線を使用する (`my_variable` など) 必要があります。HubL では、ハイフンで連結した変数名はサポートされません。

- HubL

```
{% set string_var = "This is a string value stored in a variable" %}
{{ string_var }}
```

```
{% set bool_var = True %}
{{ bool_var }}

{% set int_var = 53 %}
{{ int_var }}

{% set seq_var = ['Item 1', 'Item 2', 'Item 3'] %}
{{ seq_var }}

{% set var_one = "String 1" %}
{% set var_two = "String 2" %}
{% set sequence = [var_one, var_two] %}
{{ sequence }}

{% set dict_var = {'name': 'Item Name', 'price': '$20', 'size': 'XL'} %}
{{ dict_var.price }}
```

- 出力例

This is a string value stored in a variable

True

53

[Item1, Item2, Item3]

[String 1, String 2]

上記の例はそれぞれに異なるタイプの変数を格納しています。最後の例では2つの別々の変数を1つのシーケンスとして格納しています。

変数は、値を出力するほかに、**if文**で使用したり、**フィルター**パラメーターや**関数**パラメーターとして使用したりすることもできます。また、**forループ**で繰り返し使用することもできます (シーケンス変数のみ)。

一般的な使用例の1つに、スタイルシートでよく使われるCSSの値を変数を使用して定義するという方法があります。たとえば、ある色をCSSファイル全体で繰り返し使用しているとします。この色を変数を使用して定義しておけば、変数の値を変更するだけで、色を変更できます。次にファイルを公開したときに、その変数に対する参照がすべて更新されます。

- HubL

```
{% set primaryColor = '#F7761F' %} {# defines variable and assigns HEX color #}

a {
  color: {{ primaryColor }}; {# prints variable HEX value #}
}
```

- 出力例

```
a {
  color:#F7761F;
}
```


マクロ

HubL マクロでは、動的な値を使用して複数の文を出力できます。たとえば、同じコードブロックを何度も記述するような場合は、マクロを使用すると便利です。マクロに渡す引数を切り替えることで、そのコードブロックを出力できます。

マクロは、HubL 文の中で定義し、名前をつけ、引数を指定します。このマクロを別の文で呼び出して動的な値を渡すと、動的な引数を持つ最終的なコードブロックが出力されます。マクロの基本的な構文は以下のとおりです。

```
{% macro name_of_macro(argument_name, argument_name2) %}
  {{ argument_name }}
  {{ argument_name2 }}
{% endmacro %}
{{ name_of_macro('value to pass to argument 1', 'value to pass to argument 2') }}
```

マクロから空白文字が改行として返される場合は、テンプレート内の空白文字を手動で取り除くことができます。ブロック、コメント、変数の式の先頭または末尾にマイナス記号 (-) を追加すると、そのブロックの前または後の空白文字が削除されます。

```
{% macro name_of_macro(argument_name, argument_name2) -%}
  {{ argument_name }}
  {{ argument_name2 }}
{%- endmacro %}
```

以下の実用例では、さまざまなベンダープレフィックスを含む CSS3 プロパティを動的な値と合わせて出力するために、マクロを使用しています。この例では、1つの macro タグで 5 行のコードを出力できます。

- HubL

```
{% macro trans(value) %}
```

```

-webkit-transition: {{value}};
-moz-transition: {{value}};
-o-transition: {{value}};
-ms-transition: {{value}};
transition: {{value}};
{% endmacro %}

a {
  {{ trans('all .2s ease-in-out') }}
}

```

- 出力例

```

a {
-webkit-transition: all .2s ease-in-out;
-moz-transition: all .2s ease-in-out;
-o-transition: all .2s ease-in-out;
-ms-transition: all .2s ease-in-out;
transition: all .2s ease-in-out;
}

```

コール

場合によっては、動的な追加情報をマクロブロックに渡す必要があるかもしれません。たとえば、引数のほかに、長いコードをマクロに挿入するような場合です。これを行うには、call ブロックと caller() を使用します。call ブロックは基本的にマクロと同じよう

に機能しますが、call ブロックに名前は指定しません。caller()式で、call ブロックの内容をどこに表示するかを指定します。

以下の例では、2つの引数に加えて、1つの<p>がマクロに追加されています。

• HubL

```
{% macro render_dialog(title, class) %}
<div class="{{ class }}">
  <h2>{{ title }}</h2>
  <div class="contents">
    {{ caller() }}
  </div>
</div>
{% endmacro %}

{% call render_dialog('Hello World', 'greeting') %}
  <p>This is a paragraph tag that I want to render in my.</p>
{% endcall %}
```

• 出力例

```
<div class="greeting">
  <h2>Hello World</h2>
  <div class="contents">

  <p>This is a simple dialog rendered by using a macro and
  a call block.</p>
</div>
</div>
```

インポート

マクロには、あるテンプレートファイルから別のテンプレートファイルにインポートして使用できるという便利な機能もあります。これには、**import** タグを使用する必要があります。import タグでは、マクロが含まれているテンプレートへの Design Manager (デザインマネージャー) ファイルパスを指定し、インポート先テンプレート内でのマクロ名を指定することができます。その後、それらのマクロに値を渡すことができます。マクロを再定義する必要はありません。

たとえば、次の 2 つのマクロが含まれている.html テンプレートファイルがあるとします。1 つのマクロは header タグを設定するために定義し、もう 1 つは footer タグを生成するために定義しています。このファイルはデザインマネージャーに my_macros.html という名前で保存されています。

```
{% macro header(tag, title_text) %}
  <header> <{{ tag }}>{{ title_text }} </{{tag}}> </header>
{% endmacro %}

{% macro footer(tag, footer_text) %}
  <footer> <{{ tag }}>{{ footer_text }} </{{tag}}> </footer>
{% endmacro %}
```

これらのマクロを使用するテンプレートでは、my_macros.html ファイルへのファイルパスを指定した import タグを使用します。また、マクロのグループには名前を付けます (この例では header_footer)。そして、そのインポートしたテンプレートに付けた名前にマクロ名を追加することでマクロを実行できます。以下の例を参照してください。

- HubL

```
{% import 'custom/page/web_page_basic/my_macros.html' as header_footer %}
```

```
{{ header_footer.header('h1', 'My page title') }}  
<p>Some content</p>  
{{ header_footer.footer('h3', 'Company footer info') }}
```

- 出力例

```
<header><h1>My page title</h1></header>  
<p>Some content</p>  
<footer><h3>Company footer info</h3></footer>
```

from

別の.html ファイルに含まれているすべてのマクロではなく、特定のマクロのみをインポートしたい場合は、**from** タグを使用します。from タグでは、インポートするマクロのみを指定します。通常、**import** を使用する方が柔軟性は向上しますが、この代替手段もサポートされます。

次の例では、この記事の前のセクションと同じ my_macros.html ファイルを利用します。ただし、今回は、すべてのマクロをインポートするのではなく、footer マクロのみを利用します。

- HubL

```
{% from 'custom/page/web_page_basic/my_macros.html' import footer %}  
{{ footer('h2', 'My footer info') }}
```

- 出力例

```
<footer><h2>My footer info</h2></footer>
```

ループ内の変数

ループ内で定義した変数は、そのループの範囲内に制限され、ループの外側から呼び出すことはできません。

ループ外で定義した変数をループ内から呼び出すことはできますが、その逆はできません。

また、dict に値を設定する場合やリスト操作を実行する場合にオブジェクトを変化させるために、関数を使用できます。以下の例では `.update` [リスト操作](#) を使用しています。

```
{% set obj = {val : 0} %}
{% for i in range(0, 10) %}
  {% do obj.update('val', obj.val + i) %}
{% endfor %}
{{ obj.val }}
```

If 文

If 文

以下の記事では、HubL の `if` 文と `unless` 文について説明します。if 文には、HubL でサポートされている演算子が含まれていることがよくあります。if 文は、式のテストを実行するときに使用できます。

HubSpot での E メールでは、if 文の中でのパーソナライズ変数 (contact 変数と company 変数) の使用は現在サポートされていないことに注意してください。これらの変数は、HubSpot CMS ページとブログテンプレートのロジックに使用できます。

また、トランザクションメール API から渡された情報は、if 文の中では機能しません。これは、情報が入力される前にテンプレートがコンパイルされるためです。

if 文の基本構文

HubL では、if 文によりテンプレートのロジックの定義が支援されます。HubL の if 文の構文は、Python の条件文に類似しています。if 文は [文の区切り文字](#) で囲まれており、if 文で始まり `endif` で終わります。

if 文の基本構文の例を以下に示します。この例の "condition" は、True または False を評価するブーリアンルールに置き換えられます。

```
{% if condition %}
    If the condition is true print this to template.
{% endif %}
```

基本構文を紹介したので、次に基本的な if 文の実際の例を示します。以下の例に示す if 文は、"my_module" という名前の HubL モジュールと "my_variable" という名前の変数がテンプレートに含まれているかどうかを確認します。演算子がない状態では、この if 文ではテンプレートのコンテキストでモジュールが定義されているかどうか評価されることに注意してください。

```
{% module "my_module" path="@hubspot/rich_text", label='My rich text module',
html='Default module text' export_to_template_context=true %}

{% if widget_data.my_module %}
    A module named "my_module" is defined in this template.
```

```
{% endif %}

{% set my_variable = "A string value for my variable" %}
{% if my_variable %}
    The variable named my_variable is defined in this template.
{% endif %}
```

HubL モジュールの評価時には if 文でモジュール名が引用符で囲まれたままであり、変数のテスト時には変数名が引用符で囲まれないことに注意してください。上記のいずれの例でも、モジュールと変数がテンプレートに存在するため、これらの文が評価され、マークアップが出力されます。これらの例では、モジュールと変数が定義されているかどうかだけが確認され、モジュールと変数に値が含まれているかどうかは確認されないことに注意してください。

次に、モジュールがテンプレートに存在しているかどうかではなく、モジュールに値が含まれているかどうかを評価する if 文について説明します。このためには、**export_to_template_context** パラメーターを使用する必要があります。このパラメーターの詳細については、[こちら](#)をご覧ください。以下の例では、コンテンツエディターでテキストモジュールに値が含まれている場合に、マークアップが出力されます。モジュールのテキストフィールドがクリアされている場合、マークアップはレンダリングされません。カスタムモジュール内で作業している場合は、簡略化された widget.widget_name 構文があります。この構文については[この例](#)で説明しています。

- HubL

```
{% module "product_names" path="@hubspot/text", label='Enter the product names that you would like to render the coupon ad for', value='all of our products', export_to_template_context=True %}
```



```
{% if widget_data.product_names.value %}
<div class="coupon-ad">
<h3>For a limited time, get 50% off {{ widget_data.product_names.value}}! </h3>
</div>
{% endif %}
```

- Example Output

```
<div class="coupon-ad">
<h3>For a limited time get 50% off all of our products! </h3>
</div>
```

elif と else の使用

追加の条件文や、1 つ以上の条件が false の場合に実行するルールを使用して、高度な if 文を作成できます。Elif 文を使用すると、直前の条件の後に評価される追加の条件をロジックに追加できます。Else 文では、他のすべての条件が false の場合に実行されるルールを定義します。1 つの if 文の中で使用できる elif 文の数に制限はありませんが、else 文は 1 つのみ使用できます。

[<=演算子](#)を使用して変数の値をチェックする if 文の基本構文の例を以下に示します。

この例では、テンプレートにより"Variable named number is less than or equal to 6."が出力されます。

```
{% set number = 5 %}

{% if number <= 2 %}
```

```
        Variable named number is less than or equal to 2.
{% elif number <= 4 %}
        Variable named number is less than or equal to 4.
{% elif number <= 6 %}
        Variable named number is less than or equal to 6.
{% else %}
        Variable named number is greater than 6.
{% endif %}
```

choice モジュールを使用してユーザーが選択する部門に基づいて、経歴ページの異なる見出しをレンダリングする例を以下に示します。この例では、choice モジュールの特定の事前定義値をチェックするために[==演算子](#)が使用されています。

```
{% choice "department" label='Choose department', value='Marketing',
choices='Marketing, Sales, Dev, Services' export_to_template_context=True %}

{% if widget_data.department.value == 'Marketing' %}

<h3>Want to join our amazing Marketing team?!</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% elif widget_data.department.value == 'Sales' %}

<h3>Are you a Sales superstar?</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% elif widget_data.department.value == 'Dev' %}
```

```

<h3>Do you love to ship code?</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% else %}

<h3>Want to work with our awesome customers?</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% endif %}

```

unless 文

unless 文は、if 文に似た条件文ですが、if 文とは逆のロジックで機能します。1つのブーリアン条件が True に評価されない場合に、初めのタグと終わりのタグで囲まれたコードをレンダリングおよびコンパイルします。unless 文は **unless** で始まり **endunless** で終わります。elif と else はサポートされていません。

以下の例では、リッチテキストフィールドに値が含まれていない場合に"Under construction"という見出しが出力されます。リッチテキストフィールドに値が含まれている場合は、その値が表示されます。

```

{% module "my_page_content" path="@hubspot/rich_text", label='Enter your page
content', html=" export_to_template_context=true %" }
{{ widget_data.my_page_content.html }}

{% unless widget_data.my_page_content.html %}
<h1>This page is under construction.</h1>

```

```
<h3>Come back soon!</h3>
{% endunless %}
```

ifchanged

HubL では if 文と unless 文の他に、**ifchanged** 文がサポートされています。この文を使用して、前回このタグを呼び出した時点以降に変数に変更された場合にだけマークアップをレンダリングできます。

インライン if 文

HubL ではインライン if 文がサポートされています。インライン if 文を使用すると、条件文を簡潔に作成できます。

```
{% set color = "Blue" if is_blue is truthy else "Red" %} // color == "blue"

{{ "Blue" if is_blue is truthy else "Red" }} // "Blue"

{% set dl = true %}
<a href="http://example.com/some.pdf" {"download" if dl} >Download PDF</a>
```

三項演算子

三項演算子を使用して条件文を簡単に記述することもできます。

```
// If the variable is_blue is true, output "blue", otherwise output "red"
```

```
{{ is_blue is truthy ? "blue" : "red" }}
```

```
// Set the variable is_red to false if is_blue is true, otherwise set to true
```

```
{% set is_red = is_blue is truthy ? false : true %}
```

For ループ

For ループ

オブジェクトのシーケンスを繰り返すには、HubL で for ループを使用できます。最もよく使用されるのは[ブログコンテンツ](#)をリスト形式でレンダリングするときですが、他のシーケンス変数を並べ替える場合にも使用できます。

for ループは **for** 文で始まり、**endfor** 文で終わります。for 文には 1 つのシーケンス項目、**"in"**、シーケンス名がこの順序で指定されています。文の始まりから終わりまでの間のコードは、繰り返しごとに出力されます。このコードには通常、個々のシーケンス項目の出力変数が含まれています。for ループの基本的な構文は以下のとおりです。

```
{% for item in items %}
```

```
    {{ item }}
```

```
{% endfor %}
```

変数値のシーケンスをリストに出力する方法を以下の例に示します。

- HubL
- 出力例

```
{% set languages = ['HTML', 'CSS', 'JavaScript', 'Python', 'Ruby', 'PHP', 'Java'] %}
```

```
<h1>Languages</h1>;
<ul>
{% for language in languages %}
<li>{{ language }}</li>
{% endfor %}
</ul>
```

ループプロパティ

ループを繰り返すときに、[条件文](#)を使用してループの動作を定義できます。変数プロパティ `loop.index` は、ループの現在の繰り返し回数のカウントを維持します。繰り返し回数を異なる方法でカウントするその他の `loop` 変数プロパティがいくつかあります。これらのプロパティについては以下をご覧ください。

変数	説明
<code>loop.index</code>	ループの現在の繰り返し。この変数はカウントを 1 から開始します。
<code>loop.index0</code>	ループの現在の繰り返し。この変数はカウントを 0 から開始します。

<code>loop.revindex</code>	ループの終わりまでの繰り返し回数。1までカウントダウンします。
<code>loop.revindex0</code>	ループの終わりまでの繰り返し回数。1までカウントダウンします。
<code>loop.first</code>	この変数は、ループの最初の繰り返しである場合に <code>true</code> として評価されます。
<code>loop.last</code>	この変数は、ループの最後の繰り返しである場合に <code>true</code> として評価されます。
<code>loop.length</code>	シーケンスの項目の数。
<code>loop.cycle</code>	シーケンスリスト間を循環するヘルパー関数。以下の説明を参照してください。
<code>loop.depth</code>	再帰ループでのレンダリングの現在の深さを示します。レベル1から開始します。

`loop.depth0` 再帰ループでのレンダリングの現在の深さを示します。レベル 0 から開始します。

さまざまなループ変数の使用例を以下にいくつか示します。以下の基本的な例では、`loop.index` を使用して、繰り返しごとに出力されるカウントを保持します。

- [HubL](#)
- [出力例](#)

```
{% set loopy = ['Content', 'Social', 'Contacts', 'Reports'] %}
{% for app in loopy %}
  {{ loop.index }}. {{app}} <br>
{% endfor %}
```

次の例では、条件文を使用して、ループの長さが特定の数字で割り切れるかどうかを確認します。その後、`post-item div` の幅をレンダリングします。この例では標準のブログ記事ループを使用し、ループに 6 件の記事があることを前提としています。

- [HubL](#)
- [出力例](#)

```
{% for content in contents %}
{% if loop.length is divisibleby 4 %}
<div style="width:25%">Post content</div>

{% elif loop.length is divisibleby 3 %}
<div style="width:33.33332%">Post content</div>

{% else %}
```



```
<div style="width:50%">Post content</div>
```

```
{% endif %}
```

```
{% endfor %}
```

ネストしたループ

ループを他のループでネストすることもできます。子 for ループは、親 for ループが繰り返されるたびに実行されます。以下の例では、親項目のの中でネストしているに子項目のリストが出力されます。

- . HubL
- . 出力例

```
{% set parents = ['Parent item 1', 'Parent item 2', 'Parent item 3'] %}
```

```
{% set children = ['Child item 1', 'Child item 2', 'Child item 3'] %}
```

```
<ul>
```

```
{% for parent in parents %}
```

```
<li>{{parent}}<ul>
```

```
  {% for child in children %}
```

```
    <li>{{child}}</li>
```

```
  {% endfor %}
```

```
</ul>
```

```
</li>
```

```
{% endfor %}
```

```
</ul>
```

cycle

繰り返しごとに一連の文字列値を循環して出力するには、for ループで cycle タグを使用します。この手法の最も実地的な利用例の 1 つに、リスト内のブログ記事に交互に切り替わるクラスを適用することがあります。このタグは 3 つ以上の値に使用でき、ループの繰り返し回数が cycle の値よりも多い場合に cycle を繰り返します。以下の例では、"odd"と"even"というクラスがリスト内の記事に適用されます (この例ではループに 5 件の記事が含まれていることを前提としています)。

コンマ区切りの cycle 文字列値の間にはスペースがないことに注意してください。

- HubL
- 出力例

```
{% for content in contents %}  
  <div class="post-item {% cycle 'odd','even' %}">Blog post content</div>  
{% endfor %}
```

ループ内の変数

ループ内で定義した変数は、そのループの範囲内に制限され、ループの外側から呼び出すことはできません。

ループ外で定義した変数をループ内から呼び出すことはできますが、その逆はできません。

ループでのキーと値のペア

ループ対象の情報の dict にキーと値が含まれている場合は、単純な for ループでは値にだけアクセスできます。ループでキーと値の両方にアクセスするには、HubL の形式は以下ようになります。

- . HubL
- . 出力例

```
{% set dict_var = {'name': 'Cool Product', 'price': '$20', 'size': 'XL'} %}  
{% for key, val in dict_var.items() %}  
  {{ key }}: {{ val }}<br>  
{% endfor %}
```

設定した回数の繰り返し

設定した回数だけを繰り返すことがあります。これは、HTML や CSS を生成するとき
に便利です。このためには [range 関数](#) を使用します。

- . HubL
- . 出力例

```
{% for x in range(0,5) %}  
  {{loop.index}}  
{% endfor %}
```

演算子と式のテスト

演算子と式のテスト

テンプレートのロジックと機能を拡張できるよう、HubL ではいくつかの主要な演算子
と式のテストをサポートしています。[演算子](#) を使用すると、数学関数の実行、比較、テ

ンプレートロジックの複雑化、マークアップのレンダリングの変更を行うことができます。この記事では、HubL で使用できる[式のテスト](#)も包括的にリストします。

演算子

演算子は、HubL コンパイラに対して各種の演算を行うよう指示する記号です。これらの演算の結果、マークアップが出力されます。以下のセクションに、サポートされているすべての HubL 演算子のリストを記載します。

数学

標準の数学演算子を使用して、テンプレートのコンテキストで値を計算できます。

記号	説明
+	2つのオブジェクトを加算します。これは通常、数値を加算するために使用します。リストの文字列を連結する場合は、代わりに「~」を使用します。
-	ある数値を別の数値から減算します。
/	数値を除算します。

`%` 値を除算した余りを返します。

`//` 2つの数値を除算した余りを整数に切り捨てて返します。たとえば、
`{{ 20 // 7 }}`とすると、2が返されます。

`*` 数値を乗算します。

`**` 左辺のオペランドを右辺のオペランドで累乗します。

- HubL
- 出力例

```
{% set my_num = 11 %}  
{% set my_number = 2 %}  
  
{{ my_num + my_number }}  
{{ my_num - my_number }}  
{{ my_num / my_number }}  
{{ my_num % my_number }}  
{{ my_num // my_number }}  
{{ my_num * my_number }}  
{{ my_num ** my_number }}
```

比較

比較演算子を使用すると、テンプレートロジックに対して値を評価できます。[このリンク先の if ステートメント](#)で、比較演算子の使用例を確認できます。

記号	説明
<code>==</code>	2つのオブジェクトが等しい場合、 <code>true</code> に評価します。
<code>!=</code>	2つのオブジェクトが等しくない場合、 <code>true</code> に評価します。
<code>></code>	左辺の値が右辺の値よりも大きい場合、 <code>true</code> に評価します。
<code>>=</code>	左辺の値が右辺の値と同じかそれよりも大きい場合、 <code>true</code> に評価します。
<code><</code>	左辺の値が右辺の値よりも小さい場合、 <code>true</code> に評価します。

<= 左辺の値が右辺の値と同じかそれよりも小さい場合、true に評価します。

- HubL
- 出力例

```
{% set my_num = 11 %}  
{% set my_number = 2 %}  
{{ my_num == my_number }}  
{{ my_num != my_number }}  
{{ my_num > my_number }}  
{{ my_num >= my_number }}  
{{ my_num < my_number }}  
{{ my_num <= my_number }}
```

論理

論理演算子を使用すると、複数の式を1つのステートメントに結合できます。

記号	説明
and	左辺のオペランドと右辺のオペランドの両方が true に評価される場合、true を返します。

or 左辺のオペランドまたは右辺のオペランドのいずれかが true に評価される場合、true を返します。

not ステートメントを否定します。is と併せて使用します。以下の例を参照してください。

(expr) 演算順として式をグループ化します。例: (10 - 2) * 変数。

?: [3項演算子](#)は3つの引数を受け入れます。式を評価して、true または false の値を返します。

その他の HubL 演算子

以下に、さまざまなタスクを実行するために使用できる、その他の重要な HubL 演算子をリストします。

記号	説明
----	----

in 値がシーケンスに含まれているかどうかをチェックします。

is [式のテスト](#)を実行します。

| フィルターを適用します。

~ 値を連結します。

式のテスト

式のテストは、論理演算子を使用して評価できる各種のブール条件です。

containing

containing 式のテストでは、変数のリストに値が含まれているかどうかをチェックします。

- HubL
- 出力例

```
{% set numbers = [1, 2, 3] %}
```

```
{% if numbers is containing 2 %}
    Set contains 2!
{% endif %}
```

containingall

containingall 式のテストでは、変数のリストに別のリストのすべての値が含まれているかどうかをチェックします。

- HubL
- 出力例

```
{% set numbers = [1, 2, 3] %}

{% if numbers is containingall [2, 3] %}
    Set contains 2 and 3!
{% endif %}

{% if numbers is containingall [2, 4] %}
    Set contains 2 and 4!
{% endif %}
```

defined

defined 式のテストでは、変数がテンプレートのコンテキスト内で定義済みであるかどうかをチェックします。演算子を使わずに if ステートメントを記述してこの式のテストを使用することもできますが、その場合は既定で、変数が定義されているかどうかだけがチェックされます。

以下の例では、カラーモジュールの color パラメータがテストされます。color パラメータに値がない場合、テンプレートは既定の背景色として黒をレンダリングします。値が定義されている場合、テンプレートはユーザーが設定した背景色をレンダリングします。

- HubL
- 出力例

```
{% color "my_color" color='#930101', export_to_template_context=True %}

<style>
{% if widget_data.my_color.color is defined %}
body{
background: {{ widget_data.my_color.color }};
}
{% else %}
body{
background: #000;
}
{% endif %}
</style>
```

divisibleby

divisibleby 式のテストを使用すると、オブジェクトを別の数値で割り切れるかどうかをテストできます。

たとえば、以下の for ループは、動物の種類のリストを反復処理するために作成されています。動物の各種類が div 内に出力され、5 つ目ごとの div に異なるインラインスタイルが適用されます (幅: 100%)。このコンセプトをブログに適用すると、特定のパ

ターンの投稿に別のマークアップをレンダリングできます。ループと `loop.index` の詳細については、[こちらの記事をご覧ください](#)。

- . HubL
- . 出力例

```
{% set animals = ['lions', 'tigers', 'bears', 'dogs', 'sharks'] %}
{% for animal in animals %}

{% if loop.index is divisibleby 5 %}
<div style="width:100%">{{animal}}</div>
{% else %}
<div style="width:25%">{{animal}}</div>
{% endif %}

{% endfor %}
```

equalto

equalto 式のテストでは、変数の値が定数または別の変数の値と等しいかどうかをチェックします。これと同じテストは、`==`演算子を使用して行うこともできます。

以下の例では、ループ内の投稿の合計数に応じてブログ記事の幅を調整します。出力例は、ブログに4件の投稿があることを前提としています。

- . HubL
- . 出力例

```
{% for content in contents %}
  {% if loop.length is equalto 2 %}
    <div style="width:50%;">Post content</div>
  {% elif loop.length is equalto 3 %}
```

```
<div style="width:33.333332%;">Post content</div>
{% elif loop.length is equalto 4 %}
    <div style="width:25%;">Post content</div>
{% else %}
    <div style="width:100%;">Post content</div>
{% endif %}
{% endfor %}
```

even

even 式のテストでは、数値型変数の値が偶数であるかどうかをチェックします。

以下の例は、単純化したブローグー覧のループを示しています。この例では、ループの現在の反復処理での値が偶数の場合、その投稿アイテムの div に `even-post` クラスを割り当てます。それ以外の場合は、`odd-post` クラスを割り当てます。

- [HubL](#)
- [出力例](#)

```
{% for content in contents %}
    {% if loop.index is even %}
        <div class="post-item even-post">Post content</div>
    {% else %}
        <div class="post-item odd-post">Post content</div>
    {% endif %}
{% endfor %}
```

iterable

iterable 式のテストでは、変数が反復可能であり、ループ処理を適用できるかどうかをチェックします。

以下の例では、「jobs」という変数が反復可能であるかどうかをチェックします。この変数にジョブのリストが格納されている場合、ifステートメントが true に評価されて、ループが実行されます。変数に1つの値しか格納されていない場合は、[ifステートメント](#)により、別のマークアップを使用してその値が出力されます。[ループの詳細について](#)は、[こちらをご覧ください](#)。

- HubL
- 出力例

```
{% set jobs = ['Accountant', 'Developer', 'Manager', 'Marketing', 'Support'] %}
{% if jobs is iterable %}
<h3>Available positions</h3>
<ul>
{% for job in jobs %}
  <li>{{ job }}</li>
{% endfor %}
</ul>
{% else %}
<h3>Available position</h3>
  <div class="single-position">{{ jobs }}</div>
{% endif %}
```

lower

lower 式のテストは、文字列が小文字であれば、true に評価します。

以下の例では、[unless ステートメント](#)と小文字フィルターを使用して、テキストモジュールに入力されるテキストの文字列が常に小文字になるようにしています。

- . HubL
- . 出力例

```
{% module "my_text" path="@hubspot/text" label='Enter text', value='Some TEXT that should be Lowercase', export_to_template_context=True %}
```

```
{% unless widget_data.my_text.value is lower %}
```

```
{{ widget_data.my_text.value|lower }}
```

```
{% endunless %}
```

mapping

mapping 式のテストでは、オブジェクトが `dict` (辞書) であるかどうかをチェックします。

以下の例では、`contact` オブジェクトが辞書であるかどうかをチェックしています。この例の場合、オブジェクトは辞書に該当します。

- . HubL
- . 出力例

```
{% if contact is mapping %}
```

```
This object is a dictionary.
```

```
{% else %}
```

```
This object is not a dictionary.
```

```
{% endif %}
```

none

`none` 式のテストでは、変数に `null` 値が格納されているかどうかをチェックします。

- HubL
- 出力例

```
{% module "user_email" path="@hubspot/text" label='Enter user email',
value='example@hubspot.com', export_to_template_context=True %}
{% unless widget_data.user_email.value is none %}
{{ widget_data.user_email.value }}
{% endunless %}
```

number

`number` 式のテストでは、変数の値が数値であるかどうかをチェックします。

以下の例では、変数をチェックして、それが変数であるかどうかを調べます。変数である場合は百万に変換します。

- HubL
- 出力例

```
{% set my_var = 40 %}
{% if my_var is number %}
{{ my_var * 1000000 }}
{% else %}
my_var is not a number.
{% endif %}
```

odd

`odd` 式のテストでは、数値型変数の値が奇数であるかどうかをチェックします。

以下の例は、前に記載した even 式のテストの例の逆バージョンです。

- . HubL
- . 出力例

```
{% for content in contents %}
  {% if loop.index is odd %}
    <div class="post-item odd-post">Post content</div>
  {% else %}
    <div class="post-item even-post">Post content</div>
  {% endif %}
{% endfor %}
```

sameas

sameas 式のテストでは、2 つの変数の値が同じであるかどうかをチェックします。

以下の例では、2 つの変数を設定してから、これらの変数の値が同じであるかどうかをチェックします。

- . HubL
- . 出力例

```
{% set var_one = True %}
{% set var_two = True %}
{% if var_one is sameas var_two %}
  The variables values are the same.
{% else %}
  The variables values are different.
{% endif %}
```

sequence

sequence 式のテストは、変数がシーケンスであるかどうかをチェックするという点で **iterable** 式のテストと似ています。

以下の例では、変数がシーケンスであるかどうかをチェックしてから、音楽ジャンルのシーケンスを反復処理します。

- HubL
- 出力例

```
{% set genres = ['Pop', 'Rock', 'Disco', 'Funk', 'Folk', 'Metal', 'Jazz', 'Country', 'Hip-Hop', 'Classical', 'Soul', 'Electronica'] %}
{% if genres is sequence %}
<h3>Favorite genres</h3>
<ul>
{% for genre in genres %}
  <li>{{ genre }}</li>
{% endfor %}
</ul>
{% else %}
<h3>Favorite genre:</h3>
  <div class='single-genre'>{{ genres }}</div>
{% endif %}
```

string

string 式のテストでは、変数の値が文字列であるかどうかをチェックします。

以下の例では、変数をチェックして、それが文字列であるかどうかを調べます、文字列である場合はタイトルフィルターを適用して大文字に変更します。

- . HubL
- . 出力例

```
{% set my_var = "title of section" %}
{% if my_var is string %}
{{ my_var|title }}
{% else %}
my_var is not a string
{% endif %}
```

string_containing

このテストでは、文字列が別の文字列に含まれているかどうかをチェックします。この式のテストは「is」演算子と組み合わせて使用します。

- . HubL
- . 出力例

```
{% if content.domain is string_containing ".es" %}
Markup that will only render on content hosted on .es domains
{% elif content.domain is string_containing ".jp" %}
Markup that will only render on content hosted on .jp domains
{% else %}
Markup that will render on all other domains
{% endif %}
```

string_startingwith

この式のテストでは、文字列が特定の文字列で始まっているかどうかをチェックします。このテストは「is」演算子と組み合わせて使用します。

- . HubL
- . 出力例

```
{% if content.slug is string_startingwith "es/" %}
```

Markup that will only render on content hosted in a /es/ subdirectory

```
{% elif content.slug is string_startingwith "jp/" %}
```

Markup that will only render on content hosted in a /jp/ subdirectory

```
{% else %}
```

Markup that will render on all subdirectories

```
{% endif %}
```

truthy

truthy 式のテストでは、ブール型変数の値が true であるかどうかをチェックします。

以下の例では、ブール値チェックボックスモジュールを使用してアラートメッセージを表示します。

- . HubL
- . 出力例

```
{% boolean "check_box" label='Show alert', value=True,  
export_to_template_context=True %}
```

```
{% if widget_data.check_box.value is truthy %}
```

```
<div class='alert'>Danger!</div>
```

```
{% endif %}
```

undefined

undefined 式のテストでは、テンプレートのコンテキスト内で変数が未定義であるかどうかをチェックします。このテストは **none** 式のテストとは異なります。undefined 式のテストで **true** が返されるのは、変数が存在しても、値が格納されていない場合ですが、**none** 式のテストでは変数に **null** 値が格納されている場合に **true** が返されます。

以下の例では、テンプレートで変数「**my_var**」の有無をチェックします。

- . HubL
- . 出力例

```
{% if my_var is undefined %}
A variable named "my_var" does not exist on this template.
{% else %}
{{ my_var }}
{% endif %}
```

upper

upper 式のテストは、文字列が大文字であれば、**true** に評価します。以下の例は、前に記載した **lower** 式のテストの例の逆バージョンです。

- . HubL
- . 出力例

```
{% module "my_text" path="@hubspot/text" label='Enter text', value='Some TEXT that
should be Uppercase', export_to_template_context=True %}

{% unless widget_data.my_text.value is upper %}
{{ widget_data.my_text.value|upper }}
{% endunless %}
```

within

`within` 式のテストでは、変数がリスト内にあるかどうかをチェックします。

- [HubL](#)
- [出力例](#)

```
{% set numbers = [1, 2, 3] %}  
  
{% if 2 is within numbers %}  
    2 is in the list!  
{% endif %}  
  
{% if 4 is within numbers %}  
    4 is in the list!  
{% endif %}
```

ブログコンテンツのマークアップ

ブログコンテンツのマークアップ

[他のモジュール](#)とは異なり、ブログコンテンツモジュールは1つのタグではなく、[条件文](#)と[forループ](#)の組み合わせで構成されています。if文は、1つの記事とリストのどちらのレイアウトを探しているかに応じてマークアップをカスタマイズするために使用されます。forループは、コンテンツを繰り返したり、記事のさまざまな選択結果をリストしたりするために使用されます。

以下の記事では、HubSpot ブログコンテンツモジュールの標準的なマークアップと、いくつかのシンプルなカスタマイズについて説明します。標準的な HubSpot ブログコンテンツモジュールの完全なマークアップについては、[このページの下部](#)をご覧ください。

if is_listing_view 文

標準的な HubSpot ブログコンテンツ領域は、ユーザーがリストと個別の記事のどちらを見るかを評価するマスターif文を使用して作成されます。ドラッグ&ドロップテンプレートビルダーのテンプレートを使用する場合、このif文はブログコンテンツモジュールボタンのUIに組み込まれます。

HTML ブログテンプレートをコーディングし、記事とリストレイアウトの両方で1つのテンプレートを使用する場合は、このif文を定義する必要があります。このif文内に、記事とリストの両方のコードを書き込みます。

```
{% if is_listing_view %}
Markup for blog listing template
{% else %}
Markup for blog post template
{% endif %}
```

if blog_author 文

標準的な HubSpot ブログリストのマークアップの内部には、**if blog_author** 文があります。この文は、執筆者リストページを見ている場合に true と評価されます。執筆者リストページは、1人の執筆者による記事のページです。標準の HubSpot テンプレ

トには、執筆者の名前、プロフィール、およびソーシャルメディアアカウントが含まれています。

```
{% if blog_author %}
<div class="hs-author-profile">
    <h2 class="hs-author-name">{{ blog_author.display_name }}</h2>
    {% if blog_author.avatar %} <div class="hs-author-avatar">  </div> {%
endif %}

    <div class="hs-author-bio">{{ blog_author.bio }}</div>
    {% if blog_author.has_social_profiles %}
    <div class="hs-author-social-section">
    <span class="hs-author-social-label">Find me on:</span>
    <div class="hs-author-social-links">
        {% if blog_author.facebook %}
        <a href="{{ blog_author.facebook }}" target="_blank"
class="hs-author-social-link hs-social-facebook"> </a>
        {% endif %}
        {% if blog_author.linkedin %}
        <a href="{{ blog_author.linkedin }}" target="_blank"
class="hs-author-social-link hs-social-linkedin"> </a>
        {% endif %}
        {% if blog_author.twitter %}
        <a href="{{ blog_author.twitter }}" target="_blank" class="hs-
author-social-link hs-social-twitter"> </a>
        {% endif %}
        {% if blog_author.google_plus %}
        <a href="{{ blog_author.google_plus }}"?rel=author"
target="_blank" class="hs-author-social-link hs-social-google-plus"> </a>
        {% endif %}
    </div>
</div>
```



```
</div>
{% endif %}
</div>
<h3 class="hs-author-listing-header">Recent Posts</h3>
{% endif %}
```

ブログリストの for ループ

記事のリストは、ユーザーの要求に応じてブログ記事を繰り返して記事をリストする [for ループ](#) によって生成されます。for ループは、「contents 内の content」を繰り返します。contents は、そのブログに含まれるすべての記事を含むコンテンツの事前定義されたシーケンスです。

```
{% for content in contents %}
  <div class="post-item">
    Post item markup that renders with each iteration.
  </div>
{% endfor %}
```

if not simple_list_page 文

for ループ内には、シンプルリストと正規リストで表示する内容を決定する if 文があります。シンプルリストは、すべての記事のリストです。シンプルリストは、コンテンツ設定の記事数制限の影響を受けず、通常、すべてのブログ記事へのリンクのみを含んでいます。正規リストは、コンテンツ設定で指定された記事の数だけ繰り返し、それに従ってページ番号を付けます。

以下のコードは、この if 文のシンプルなバージョンであり、シンプルリストページで繰り返す内容を定義します。この if 文では逆のロジックが使用されることに注意してください。このため、else でシンプルリストビューが定義されます。

```
{% if not simple_list_page %}
  Iterated post markup for regular listing
{% else %}
  <h2 class="post-listing-simple"><a
href="{{content.absolute_url}}">{{ content.name }}</a></h2>
{% endif %}
```

記事リストのマークアップ

正規リストの if 文のロジック内では、繰り返される記事のマークアップが定義されます。このコードブロックは、リストビューで各記事をどのように表示するかを指示します。標準的な HubSpot ブログリストのマークアップには、以下の項目が含まれます。

記事タイトル

投稿者

ソーシャルシェアリング

サムネイル画像

記事サマリー

続きを読む

コメント数

関連する HubL を表示して詳細を確認するには、マークアップコンポーネントまたは をクリックしてください

```
<div class="post-header">
    <h2><a href="{{content.absolute_url}}">{{ content.name }}</a></h2>
    <div id="hubspot-author_data" class="hubspot-editable" data-hubspot-form-id="author_data" data-hubspot-name="Blog Author">
        Posted by <a class="author-link" href="{{ group.absolute_url }}/author/{{ content.blog_post_author.slug }}">{{ content.blog_post_author.display_name }}</a> on {{ content.publish_date_localized }}
    </div>
    {% blog_social_sharing "blog_social_sharing" overrideable=False, label='Blog Social Sharing' %}</div>
<div class="post-body clearfix">
    <!--post summary-->
    {% if content.post_list_summary_featured_image %}
    <div class="hs-featured-image-wrapper">
        <a href="{{content.absolute_url}}" title="" class="hs-featured-image-link">
            
        </a>
    </div>
    {% endif %}
    {{ content.post_list_content|safe }}</div>
{% if content_group.show_summary_in_listing %}
    <a class="more-link" href="{{ content.absolute_url }}">Read More</a>
{% endif %}<div class="custom_listing_comments">
    {% set comments_number = content.comment_list|length %}
    {% set comments_label = "Comment" if comments_number == 1 else "Comments" %}
```

```

    {{ comments_number }} {{ comments_label }} <a
href="{{content.absolute_url}}#comments-listing">Click here to read/write
comments</a>
</div>{% if content.topic_list %}
    <p id="hubspot-topic_data"> Topics:
        {% for topic in content.topic_list %}
            <a class="topic-link" href="{{ blog_tag_url(group.id,
topic.slug) }}">{{ topic.name }}</a>{% if not loop.last %},{% endif %}
        {% endfor %}
    </p>
{% endif %}

```

リストのページネーション

HubSpot のブログリストのマークアップには、次の記事、前の記事、およびすべての記事のページ番号が含まれます。このページ番号は、シンプルリスト以外のビューでのみ表示されます。

```

{% if not simple_list_page %}
    <div class="blog-pagination">
        {% if last_page_num %}
            <a class="previous-posts-link"
href="{{ blog_page_link(last_page_num) }}">Previous</a>
        {% endif %}
        <a class="all-posts-link" href="{{ group.absolute_url }}/all">All posts</a>
        {% if next_page_num %}
            <a class="next-posts-link"
href="{{ blog_page_link(next_page_num) }}">Next</a>
        {% endif %}
    </div>
{% endif %}

```

ブログ記事のマークアップ

個別のブログ記事のマークアップは、最初の `if is_listing_view` 文の `else` 文の後に定義するか、ドラッグ & ドロップブログコンテンツモジュールの `[Edit post layout (記事レイアウトの編集)]` セクション内で直接定義します。標準的な個別の記事のマークアップには、以下の項目が含まれます。

記事タイトル

執筆者情報

ソーシャルシェアリング

記事コンテンツ

記事のトピック

記事コメント

関連する HubL を表示して詳細を確認するには、マークアップコンポーネントまたは [こちら](#) をクリックしてください

```
<div class="blog-section">
  <div class="blog-post-wrapper cell-wrapper">
    <div class="section post-header">
      <h1>{{ content.name }}</h1>
      <div id="hubspot-author_data" class="hubspot-editable" data-hubspot-form-id="author_data" data-hubspot-name="Blog Author" >
        <span class="hs-author-label">Posted by</span>
        <a class="author-link">
```

```

href="{{ group.absolute_url }}/author/{{ content.blog_post_author.slug }}">{{ content.blog
_post_author.display_name }}</a> on {{ content.publish_date_localized }}

    {% if content.blog_post_author.avatar %} <div class="hs-author-avatar">
 </div> {% endif %}

    {% if content.blog_post_author.has_social_profiles %}
<div class="hs-author-social-section">
<span class="hs-author-social-label">Find me on:</span>
<div class="hs-author-social-links">
    {% if content.blog_post_author.facebook %}
    <a href="{{ content.blog_post_author.facebook }}" target="_blank"
class="hs-author-social-link hs-social-facebook">Facebook </a>
    {% endif %}
    {% if content.blog_post_author.linkedin %}
    <a href="{{ content.blog_post_author.linkedin }}" target="_blank"
class="hs-author-social-link hs-social-linkedin">LinkedIn</a>
    {% endif %}
    {% if content.blog_post_author.twitter %}
    <a href="{{ content.blog_post_author.twitter }}" target="_blank"
class="hs-author-social-link hs-social-twitter">Twitter </a>
    {% endif %}
    {% if content.blog_post_author.google_plus %}
    <a href="{{ content.blog_post_author.google_plus }}"?rel=author"
target="_blank" class="hs-author-social-link hs-social-google-plus">Google+</a>
    {% endif %}
</div>
</div>

    {% endif %} </div>
    {% blog_social_sharing "blog_social_sharing" overrideable=False,
label='Blog Social Sharing' %} <div class="section post-body">
    {{ content.post_body }}
</div> {% if content.topic_list %}

```

```

<p id="hubspot-topic_data" > Topics:
    {% for topic in content.topic_list %}
        <a class="topic-link" href="{{ blog_tag_url(group.id,
topic.slug) }}">{{ topic.name }}</a>{% if not loop.last %},{% endif %}
    {% endfor %}
</p>
{% endif %}      <div class="section blog-comments">
    {% blog_comments "blog_comments" overrideable=False, label='Blog
Comments' %}
</div> </div>
</div>

```

任意の執筆者ボックス

個々の記事レイアウトの下部には、既定でコメントアウトされる執筆者ボックスがあります。このボックスには、執筆者のプロフィール情報、ソーシャル情報、およびアバターが含まれています。このセクションを表示するには、コードブロックの先頭と末尾から`{#}`区切り文字を削除します。

```

<!-- Optional:Blog Author Bio Box -->
{#{<div class="about-author-sec row-fluid">
    <div class="span2">
        
    </div>
    <div class="span10">
        <h3>Written by <a class="author-link" href="{{ blog_author_url(group.id,
content.blog_post_author.slug) }}">{{ content.blog_post_author.display_name }}</a></h
3>
        <p>{{ content.blog_post_author.bio }}</p>
        {% if content.blog_post_author.has_social_profiles %}

```

```

<div class="hs-author-social-section">
  <div class="hs-author-social-links">
    {% if content.blog_post_author.facebook %}
      <a href="{{ content.blog_post_author.facebook }}" target="_blank"
class="hs-author-social-link hs-social-facebook">Facebook</a>
    {% endif %}
    {% if content.blog_post_author.linkedin %}
      <a href="{{ content.blog_post_author.linkedin }}" target="_blank"
class="hs-author-social-link hs-social-linkedin">LinkedIn</a>
    {% endif %}
    {% if content.blog_post_author.twitter %}
      <a href="{{ content.blog_post_author.twitter }}" target="_blank"
class="hs-author-social-link hs-social-twitter">Twitter</a>
    {% endif %}
    {% if content.blog_post_author.google_plus %}
      <a href="{{ content.blog_post_author.google_plus }}"?rel=author"
target="_blank" class="hs-author-social-link hs-social-google-plus">Google+</a>
    {% endif %}
  </div>
</div>
{% endif %}
</div>
</div>#}

```

if topic 文

執筆者リストページでのみ表示されるマークアップを定義する if **blog_author** 文に加えて、ブログトピックリストのコードのみを表示するために使用できる **topic** 変数があります。この if 文は既定のブログコードに含まれていません。以下の例は、ページタイ

トル変数を使用してトピックリストページの上部にトピック名を自動的に印刷するスニペットです。このスニペットをブログリストのコードに追加することができます。

```
{% if topic %}<h3>Posts about {{ page_meta.html_title|replace(' | ', ' ') }}</h3>{% endif %}
```

ブログのマークアップの例

以下のマークアップは、HubSpot で使用される標準的なブログコンテンツモジュールです。ブログテンプレートを最初からコーディングするときに参照用として使用することができ、テンプレートビルダーではブログコンテンツモジュールに自動的に挿入されます。

```
{% if is_listing_view %}<div class="blog-section">
  <div class="blog-listing-wrapper cell-wrapper">
    <div class="blog-section">
      <div class="blog-listing-wrapper cell-wrapper">
        {# simple_list_page indicates the "blog/all" page, which is a list of links to every
        blog post #}
        <div class="post-listing{% if simple_list_page %}-simple{% endif %}">
          {% if blog_author %}
          <div class="hs-author-profile">
            <h2 class="hs-author-name">{{ blog_author.display_name }}</h2>
            {% if blog_author.avatar %} <div class="hs-author-avatar">  </div> {% endif %}
            <div class="hs-author-bio">{{ blog_author.bio }}</div>
            {% if blog_author.has_social_profiles %}
            <div class="hs-author-social-section">
              <span class="hs-author-social-label">Find me on:</span>
              <div class="hs-author-social-links">
```

```

        {% if blog_author.facebook %}
        <a href="{{ blog_author.facebook }}" target="_blank"
class="hs-author-social-link hs-social-facebook"> </a>
        {% endif %}
        {% if blog_author.linkedin %}
        <a href="{{ blog_author.linkedin }}" target="_blank"
class="hs-author-social-link hs-social-linkedin"> </a>
        {% endif %}
        {% if blog_author.twitter %}
        <a href="{{ blog_author.twitter }}" target="_blank" class="hs-
author-social-link hs-social-twitter"> </a>
        {% endif %}
        {% if blog_author.google_plus %}
        <a href="{{ blog_author.google_plus }}"?rel=author"
target="_blank" class="hs-author-social-link hs-social-google-plus"> </a>
        {% endif %}
    </div>
</div>
{% endif %}
</div>
<h3 class="hs-author-listing-header">Recent Posts</h3>
{% endif %}
{% for content in contents %}
<div class="post-item">
    {% if not simple_list_page %}
    <div class="post-header">
        <h2><a
href="{{content.absolute_url}}">{{ content.name }}</a></h2>
        <div id="hubspot-author_data" class="hubspot-editable" data-
hubspot-form-id="author_data" data-hubspot-name="Blog Author">
            Posted by

```

```

        {%if content.blog_post_author %}
            <a class="author-link" href="{{ blog_author_url(group.id,
content.blog_post_author.slug) }}">{{ content.blog_post_author.display_name }}</a> on
{{ content.publish_date_localized }}
        {% endif %}
    </div>
    {% blog_social_sharing "blog_social_sharing" overrideable=False,
label='Blog Social Sharing' %}
</div>
<div class="post-body clearfix">
<!--post summary-->
{% if content.post_list_summary_featured_image %}
    <div class="hs-featured-image-wrapper">
        <a href="{{content.absolute_url}}" title="" class="hs-
featured-image-link">
            
        </a>
    </div>
{% endif %}
    {{ content.post_list_content|safe }}
</div>
{% if content_group.show_summary_in_listing %}
    <a class="more-link" href="{{ content.absolute_url }}">Read
More</a>
{% endif %}
<div class="custom_listing_comments">
    {% set comments_number = content.comment_list|length %}
    {% set comments_label = "Comment" if comments_number == 1
else "Comments" %}

```

```

        {{ comments_number }} {{ comments_label }} <a
href="{{content.absolute_url}}#comments-listing">Click here to read/write
comments</a>

    </div>

    {% if content.topic_list %}
        <p id="hubspot-topic_data"> Topics:
        {% for topic in content.topic_list %}
            <a class="topic-link" href="{{ blog_tag_url(group.id,
topic.slug) }}">{{ topic.name }}</a>{% if not loop.last %},{% endif %}
        {% endfor %}
        </p>
    {% endif %}
    {% else %}
        <h2 class="post-listing-simple"><a
href="{{content.absolute_url}}">{{ content.name }}</a></h2>
    {% endif %}
</div>
{% endfor %}
</div>

{% if not simple_list_page %}
<div class="blog-pagination">
    {% if last_page_num %}
        <a class="previous-posts-link"
href="{{ blog_page_link(last_page_num) }}">Previous</a>
    {% endif %}
        <a class="all-posts-link" href="{{ group.absolute_url }}/all">All posts</a>
    {% if next_page_num %}
        <a class="next-posts-link"
href="{{ blog_page_link(next_page_num) }}">Next</a>
    {% endif %}

```

```

        </div>
        {% endif %}
    </div>
</div>

</div>
</div>{% else %}<div class="blog-section">
    <div class="blog-post-wrapper cell-wrapper">
        <div class="blog-section">
            <div class="blog-post-wrapper cell-wrapper">
                <div class="section post-header">
                    <h1>{{ content.name }}</h1>
                    <div id="hubspot-author_data" class="hubspot-editable" data-hubspot-
form-id="author_data" data-hubspot-name="Blog Author" >
                        <span class="hs-author-label">Posted by</span>
                        {% if content.blog_post_author %}
                            <a class="author-link" href="{{ blog_author_url(group.id,
content.blog_post_author.slug) }}">{{ content.blog_post_author.display_name }}</a> on
{{ content.publish_date_localized }}
                            {% if content.blog_post_author.avatar %} <div class="hs-author-
avatar">  </div> {% endif %}
                            {% if content.blog_post_author.has_social_profiles %}
                                <div class="hs-author-social-section">
                                    <span class="hs-author-social-label">Find me on:</span>
                                    <div class="hs-author-social-links">
                                        {% if content.blog_post_author.facebook %}
                                            <a href="{{ content.blog_post_author.facebook }}"
target="_blank" class="hs-author-social-link hs-social-facebook">Facebook </a>
                                        {% endif %}
                                        {% if content.blog_post_author.linkedin %}

```

```

        <a href="{{ content.blog_post_author.linkedin }}"
target="_blank" class="hs-author-social-link hs-social-linkedin">LinkedIn</a>
        {% endif %}
        {% if content.blog_post_author.twitter %}
        <a href="{{ content.blog_post_author.twitter }}"
target="_blank" class="hs-author-social-link hs-social-twitter">Twitter </a>
        {% endif %}
        {% if content.blog_post_author.google_plus %}
        <a
href="{{ content.blog_post_author.google_plus }}"?rel=author" target="_blank" class="hs-
author-social-link hs-social-google-plus">Google+</a>
        {% endif %}
    </div>
</div>
{% endif %}
</div>
</div>
    {% blog_social_sharing "blog_social_sharing" overrideable=False,
label='Blog Social Sharing' %}
    <div class="section post-body">
    {{ content.post_body }}
</div>
    {% if content.topic_list %}
    <p id="hubspot-topic_data" > Topics:
    {% for topic in content.topic_list %}
        <a class="topic-link" href="{{ blog_tag_url(group.id,
topic.slug) }}">{{ topic.name }}</a>{% if not loop.last %},{% endif %}
    {% endfor %}
    </p>
    {% endif %}

```

```

        <div class="section blog-comments">
            {% blog_comments "blog_comments" overrideable=False, label='Blog
Comments' %}
        </div>
    </div>
</div>

<!-- Optional:Blog Author Bio Box -->
{#<div class="about-author-sec row-fluid">
    <div class="span2">
        
    </div>
    <div class="span10">
        <h3>Written by <a class="author-link" href="{{ blog_author_url(group.id,
content.blog_post_author.slug) }}">{{ content.blog_post_author.display_name }}</a></h
3>
        <p>{{ content.blog_post_author.bio }}</p>
        {% if content.blog_post_author.has_social_profiles %}
        <div class="hs-author-social-section">
            <div class="hs-author-social-links">
                {% if content.blog_post_author.facebook %}
                <a href="{{ content.blog_post_author.facebook }}" target="_blank"
class="hs-author-social-link hs-social-facebook">Facebook</a>
                {% endif %}
                {% if content.blog_post_author.linkedin %}
                <a href="{{ content.blog_post_author.linkedin }}" target="_blank"
class="hs-author-social-link hs-social-linkedin">LinkedIn</a>
                {% endif %}
                {% if content.blog_post_author.twitter %}

```

```
        <a href="{{ content.blog_post_author.twitter }}" target="_blank"
class="hs-author-social-link hs-social-twitter">Twitter</a>
        {% endif %}
        {% if content.blog_post_author.google_plus %}
        <a href="{{ content.blog_post_author.google_plus }}"?rel=author"
target="_blank" class="hs-author-social-link hs-social-google-plus">Google+</a>
        {% endif %}
    </div>
</div>
{% endif %}
</div>#}
</div>
</div>{% endif %}
```


Export_to_template_context

export_to_template_context

`export_to_template_context` は、HubL タグを実際に表示せずに HubL タグのパラメーターをテンプレート環境で使用できるようにするパラメーターです。このパラメーターはすべての [HubL タグ](#) で使用できます。これらのパラメーターを取得し、変数に保存し、テンプレートのロジックに含めるには、`widget_data` dict を使用します。

HubL タグを実際に表示せずに HubL タグのパラメーターをテンプレートのコンテキストで使用できるようになると、ユーザーはコンテンツエディターでテンプレートの表示方法に影響する意思決定を行うことができます。たとえば、ユーザーがフィールドに値を入力したときに、特定のコードブロックのみを表示する必要があるとします。これは、このパラメーターを使用することで実現できます。

まず、HubL タグに `export_to_template_context=True` を追加する必要があります。次に、`widget_data.module.parameter_you_want_to_retrieve` を使用する必要があります。

```
{% module "job_title" path="@hubspot/text", label="Enter a Job Title", value="Chief  
Morale Officer", export_to_template_context=True %}  
{# Makes the parameters available to be stored and used in template logic #}  
  
{{ widget_data.job_title.body.value }}  
{# Prints the value of the HubL tag but can also be used in template logic #}
```

以下では、この考え方の適用例をいくつか紹介します。

カスタムモジュール内での利用

新しいモジュールフレームワークで該当するすべての HubL フィールドがサポートされるため、カスタムモジュールでは `export_to_template_context=True` はサポートされなくなりました。

ユーザーが選択できる背景画像

この例では、画像 HubL タグを作成しますが、それを表示せずにテンプレートのコンテンツにエクスポートします。widget_data タグで src パラメーターを取得し、style タグの背景画像のソースとして表示します。

```
{% module "background_image" path="@hubspot/image" label='Select a background image',
src='http://cdn2.hubspot.net/hub/428357/file-2117441560-jpg/img/dev-bg-compressed.jpg',
export_to_template_context=True %}

<!--Sample markup -->
<div class="bg-image-section"
style="background:url('{{ widget_data.background_image.src }}'); background-size:cover; min-height: 500px;">
  <h1>Supercharge your app with HubSpot</h1>
  <h3>Build powerful integrations for a global community of users</h3>
</div>
```

これはコードテンプレートでも実現できますが、通常はページエディターのユーザーに最適な編集環境を提供するため、カスタムモジュールを作成することをお勧めします。このような HubL タグは個別のフィールドとして表示されますが、関連するフィールドが複数存在する場合があります。カスタムモジュールを使用すると、ページエディターではモジュールのすべてのフィールドがグループ化された状態で表示されます。

定義済みのマークアップを表示するための 選択フィールド

以下の例では、`export_to_template_context` パラメーターと[選択モジュール](#)を組み合わせ、求人ページのバナーメッセージを変更しています。ユーザーが UI を使用して部門を選択すると、ユーザーが実際にコンテンツを編集しなくても見出しが変更されます。

```
{% module "department" path="@hubspot/choice", label="Choose department",
value="Marketing", choices="Marketing, Sales, Dev, Services"
export_to_template_context=True %}

{% if widget_data.department.value == "Marketing" %}

<h3>Want to join our amazing Marketing team?!</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% elif widget_data.department.value == "Sales" %}

<h3>Are you a Sales superstar?</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% elif widget_data.department.value == "Dev" %}

<h3>Do you love to ship code?</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>
```

```
{% else %}

<h3>Want to work with our awesome customers?</h3>
<h4>We have exciting career opportunities on the {{ widget_data.department.value }}
team.</h4>

{% endif %}
```

これと同じ機能を、カスタムモジュール内の選択フィールドを使用して再現できます。カスタムモジュールの UI では、値とラベルの両方を含む選択オプションを非常に簡単に作成できます。

テンプレートにすでに表示されているモジュールからパラメーターを取得する

ページにすでに表示されているモジュールまたはタグからパラメーターを取得する場合は、**widgets** という名前の dict 内でそのモジュールにアクセスできます。`export_to_template_context` パラメーターは必須ではありません。構文は次のとおりです。

```
{{ content.widgets.name_of_module.body.parameter }}

{{ content.widgets.my_text.body.value }}
```

フィールドによってデータの保存形式が異なるため、表示するデータへのアクセス方法を確認するには、多くの場合、[開発者情報](#)を使用すると便利です。

HubL モジュール情報をブログリストに印刷する

ブログテンプレートは、一般的にはブログに使用されますが、他の種類のリストを作成するために[再利用する](#)こともできます。前述の手法を使用してこの目的を達成できます。

たとえば、会社が受信した報道記事のリストレイアウトを作成するときに、記事にリンクせずに、リストを別のページにリンクしたい場合があります。

academy.hubspot.com/jp/projects でこの考え方の実例を確認できます。プロジェクトリストページはブログリストテンプレートですが、個々の記事は通常の HubSpot ページにリンクしています。コンテンツ作成者はエディターでリンク先を指定します。

個々のブログ記事のコードの先頭で、テキストフィールドを定義します。記事にこのテキストを表示しない場合は、`export_to_template_context` を使用します。

```
{% module "custom_blog_link" path="@hubspot/text", label="Link to external press item", export_to_template_context=True %}
```

このテキストフィールドはブログ記事ごとに編集できます。次に、リスト内のリンクを定義する必要があります。しかし、`widget_data` は記事のコンテキストにのみ存在するため、別の構文を使用してウィジェットデータを取得し、リンクに設定する必要があります。この場合は、`content.widgets.custom_blog_link.body.value` を使用します。`widget_data` はブログリストでは使用できませんが、そのフィールドの値は個々のコンテンツのウィジェットのコンテキスト内に保存されます。

各記事を含むこのカスタムリンクを表示する基本的なブログリストのループを以下に示します。この手法を使用する場合は、ブログ記事ごとに自動的に作成されたサブディレ

クトリを robots.txt ファイルに追加して、空の記事が Google などのクローラーによってクロールされないようにする必要があります。

```
{% for content in contents %}  
<a href="{{ content.widgets.custom_blog_link.body.value }}">  
  Click here to see this press feature!  
</a>  
{% endfor %}
```

開発者情報の使い方

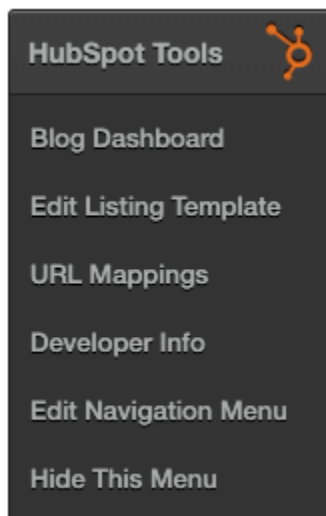
CMS のページの Developer Info の 使い方

Developer Info により、ページ、ランディングページ、ブログのリストページまたは記事のコンテキストで使用可能な情報をすべて確認できます。[CMS で利用可能なよく使われる変数のほとんどはこちらにリストされています。](#) Developer Info では、ページまたは記事で使用可能なすべての情報からなる包括的な JSON ツリーが表示されます。

Developer Info を使用する前に、ブラウザーで JSON をフォーマットするための拡張機能またはプラグインをインストールする必要があります。Chrome を使用している場合は、[JSONView](#) という名前の拡張機能をインストールできます。

Developer Info の表示

[JSON フォーマット用ブラウザ拡張機能](#)のインストールが完了したら、公開中のページ/記事または共有プレビューのスプロケットメニューから Developer Info にアクセスできます。スプロケットをクリックして **[Developer Info]** を選択します。



ページのコンテキストに含まれている情報のほとんどは内部使用を目的としていますが、この JSON を使用すると、特定のページまたは記事にレンダリングできるすべての情報の包括的なビューが表示されます。たとえば、Designers.HubSpot.com の Developer Info の一部の例を以下に示します。

```
page_meta: {
  html_title:"HubSpot Designers Site",
  meta_keywords: "",
  meta_description:"Find all the help, documentation, inspiration you need to bring your ideas to life with HubSpot's Content Management System. ",
  name:"Designers Home V2",
  canonical_url: "http://designers.hubspot.com"
}
```

この JSON ブロックは、page_meta dict に含まれているプロパティを表示します。ページの HTML タイトルを出力する場合は、以下の HubL 式を記述します。

```
{{ page_meta.html_title }}
```

JSON ツリーの各ネストレベルでは、ピリオドを使ってプロパティを連結する必要があります。たとえばブログ記事の執筆者の表示名は、JSON ツリーで 3 レベルをネストしており、以下の式を使用して出力されます。

```
{{ content.blog_post_author.display_name }}
```

HubL でサポートされている変数

HubL でサポートされている変数

HubSpot テンプレートでは、さまざまな事前定義変数を使用できます。これらの事前定義変数を使用すると、便利なウェブサイトや E メール要素を表示できます。このページには、あらかじめ定義されている変数の参照リストが記載されています。HubL テンプレートまたはモジュールで独自の変数を作成する方法については、[こちら](#)をご覧ください。

このページに記載されている変数のほとんどはオプションですが、いくつかの変数は、テンプレートから E メールまたはページを作成する上で必須です。

E メールテンプレートの必須変数

[CAN-SPAM](#) に準拠するため、HubSpot 経由で送信されるすべての E メールには、特定の会社情報とオプトアウト情報が必要です。したがって、HubSpot E メールテンプレ

ートには特定の変数が必要です。その他のオプションの E メール変数は [このページの下の部分](#)に記載されています

変数	タイプ	説明
<code>{{ site_settings.company_city }}</code>	string	会社の市区町村 ([設定]>[マーケティング]>[E メール]>[設定]>[フッター]で設定) を出力します。
<code>{{ site_settings.company_name }}</code>	string	会社名 ([設定]>[マーケティング]>[E メール]>[設定]>[フッター]で設定) を出力します。
<code>{{ site_settings.company_state }}</code>	string	会社の都道府県 ([設定]>[マーケティング]>[E メール]>[設定]>[フッター]で設定) を出力します。
<code>{{ site_settings.company_street_address_1 }}</code>	string	会社の住所 ([設定]>[マーケティング]>[E メール]>[設定]>[フッター]で設定) を出力します。

<code>{{ unsubscribe_link }}</code>	string	受信者がサブスクリプション設定を管理したり、Eメール通信の配信を停止したりできるページの URL を出力します。この変数は<a>の href 属性で使用する必要があります。
-------------------------------------	--------	--

ページテンプレートの必須変数

コードファイルを編集可能なページまたはブログテンプレートとして公開するには、以下の変数を含める必要があります。これらの変数を使用せずに HTML ファイルを公開し、別のテンプレート内で使用するには、[このテンプレートを新しいコンテンツで利用できるようにする]オプションをオフにします。

<code>{{ standard_footer_includes }}</code>	string	HubSpot トラッキングコード、 public_common.js 、 コンテンツ設定 のフッターHTML に含まれるその他のコード、または特定ページのオプションを表示します。このタグは終了本文タグの直前に挿入する必要があります。
---	--------	---

`{{ standard_header_includes }}` string [jQuery、public_common.css、layout.css](#)、添付されているスタイルシート、meta viewport タグ、Google アナリティクス のトラッキングコード、その他のメタ情報、およびドメイン/テンプレート/ページレベルで head タグに追加されたコードを追加します。この変数を HTML テンプレートの `<head>` に追加する必要があります。

すべてのテンプレートで利用可能な変数

E メールテンプレート、ページテンプレート、またはブログテンプレートで使用できる事前定義の HubSpot 変数が多数あります。このような変数を以下に示します。

(注:これらの変数で出力できる内容の詳細を確認するには、[pprint パラメータを使用してください](#)。)

一般的な変数

さまざまなコンテンツを表示する変数を以下に示します。

変数	タイプ	説明
----	-----	----

{{ account }}	dict	<p>この変数は、既知のコンタクトの会社のパーソナライズプロパティを格納するディクショナリです。このディクショナリのプロパティにアクセスするには、ピリオドとプロパティ名を追加します。たとえば account.name と指定すると、コンタクトの会社名が出力されます。</p> <p><i>この変数を使用すると、ページキャッシュが無効になります。</i></p>
{{ company_domain }}	string	<p>[ウェブサイト]>[ページ]>[ブランディング]>[ロゴのリンク]から会社のドメインを出力します。</p>
{{ contact }}	dict	<p>この変数は、既知のコンタクトのコンタクトのパーソナライズプロパティを格納するディクショナリです。このディクショナリのプロパティにアクセスするには、ピリオドとプロパティ名を追加します。たとえば contact.firstname と指定すると、コンタクトの名前が出力されます。</p> <p><i>この変数を使用すると、ページキャッシュが無効になります。</i></p>

<code>{{ content }}</code>	dict	この変数は、特定のコンテンツ (E メール、ページ、記事など) に関連するさまざまなプロパティを格納するディクショナリです。
<code>{{ content.absolute_url }}</code>	string	ページ、記事、投稿、Eメールのウェブページバージョンの完全な URL を出力します。
<code>{{ content.archived }}</code>	boolean	この変数は、ページまたは E メールがユーザーによってアーカイブ済みとしてマークされている場合に、True と評価されます。
<code>{{ content.author_email }}</code>	string	コンテンツ作成者の E メールアドレス。
<code>{{ content.author_name }}</code>	string	コンテンツ作成者の名と姓
<code>{{ content.author_username }}</code>	string	コンテンツ作成者の HubSpot ユーザー名
<code>{{ content.campaign }}</code>	string	このページまたは E メールが関連付けられているマーケティングキャンペーンの GUID。この固有 ID は、キャンペーンツールで特定キャンペーンの URL に含まれています。

{{ content.campaign_name }}	string	このページまたは E メールが関連付けられているマーケティングキャンペーンの名前。
{{ content.created }}	datetime	コンテンツが作成された時刻の日時オブジェクト (UTC 時刻)。この変数は datetime フィルター を使用して書式設定できます。
{{ content.meta_description }}	string	ページのメタディスクリプションを取得する場合は、変数 page_meta.meta_description を使用することをお勧めします。
{{ content.name }}	string	記事、E メール、またはページの名前。ページと Eメールの場合は内部コンテンツ名が出力され、記事の場合は記事のタイトルが出力されます。ブログ記事の場合、表示される記事のタイトルです。その他のコンテンツタイプの場合、一般に内部名です。この変数にはラッパーが含まれているので、ブログ記事に組み込まれている場合には UI で編集できます。ラッパーなしでコンテンツ名を出力する場合は <code>page_meta.name</code> を使用します。

<code>{{ content.publish_date }}</code>	datetime	コンテンツが公開された時刻を表す日時オブジェクト (UTC 時刻)。この変数は datetime フィルター を使用して書式設定できます。
<code>{{ content.publish_date_localized }}</code>	string	コンテンツの公開日時を表す文字列。この日時は HubSpot レポート設定で定義されているローカル時刻です。この変数は、 [設定]>[ウェブサイト]>[ブログ]>[日付形式] の言語と日付の設定の影響も受けます。
<code>{{ content.template_path }}</code>	string	デザインマネージャーでのテンプレートのファイルパス (例: custom/page/web_page_basic/my_template.html)
<code>{{ content.updated }}</code>	datetime	ユーザーが最後にコンテンツを更新した日時を表す日時オブジェクト (UTC 時刻)。この変数は datetime フィルターを使用して書式設定できます。
<code>{{ content_id }}</code>	string	ページ、記事、または Eメールの固有 ID を出力します。この ID はエディターの URL に含まれています。この変数は content.id のエイリアスとして使用できます。

{{ favicon_link }}	string	ファビコンのソース URL を出力します。この画像は、[設定] > [ウェブサイト] > [ページ] > [ブランディング] で設定します。
{{ hub_id }}	string	HubSpot アカウントのポータル ID。
{{ hubspot_analytics_tracking_code }}	string	アナリティクストラッキングコードを含めます。standard_footer_includes によりトラッキングコードがすでに表示されるため、このタグは必須ではありません。
{{ local_dt }}	datetime	レポート設定で定義されているタイムゾーンでの現行時刻を表す日時オブジェクト
{{ local_time_zone }}	string	HubSpot レポート設定で設定されているタイムゾーン
{{ page_meta.canonical_url }}	string	このページにアクセスするための公式 URL。通常はクエリ文字列パラメータを含みません。これは rel='canonical' タグに使用します。HubSpot により URL が自動的に正規化されます。

<code>{{ page_meta.html_title }}</code>	string	ページのタイトル。この変数は HTML テンプレートの<title>タグで使用する必要があります。
<code>{{ page_meta.meta_description }}</code>	string	ページのメタディスクリプション。この変数は HTML テンプレートの"description" <meta> タグで使用する必要があります。
<code>{{ page_meta.name }}</code>	string	content.name のエイリアス
<code>{{ portal_id }}</code>	string	hubs_id のエイリアス
<code>{{ request_contact }}</code>	dict	リクエストされたコンタクトに関するデータが含まれているディクショナリ。 <i>この変数を使用すると、ページキャッシュが無効になります。</i>
<code>{{ site_settings }}</code>	dict	site_settings ディクショナリには、カラーやフォントなどのさまざまな設定が含まれています (以下を参照)。
<code>{{ year }}</code>	string	現在の年を出力します。

カラーとフォントの設定

[設定]>[マーケティング]>[設定]>[カラー]には、テンプレートやファイルに出力できる基本的なカラーとフォントの制御設定があります。CSS ファイルでこれらの変数を使用する場合は、いずれかの設定を変更したときには新しいカラーを適用するために CSS ファイルを再公開/再コンパイルする必要があることに注意してください。

変数	タイプ	説明
<pre> {{ site_settings.background_color }} </pre>	string	<p>[設定]>[マーケティング]>[Eメール]>[設定]>[カラー]での背景色の設定。16進値を出力します。</p>
<pre> {{ site_settings.body_border_color }} </pre>	string	<p>[設定]>[マーケティング]>[Eメール]>[設定]>[カラー]での本文の境界線色の設定。[境界線の色オプション]ドロップダウンで[Eメール本文の境界線色を手動で設定]を選択すると、このオプションが使用可能になります。16進値を出力します。</p>

<pre>{{ site_settings.body_border_color_choice }}</pre>	enumeration	<p>この変数は HubSpot のデフォルトの E メールテンプレートで使用され、境界線を追加するかどうかを決定します。この設定は[コンテンツ設定]>[色とフォント]で制御されます。次の値を出力します: BORDER_AUTOMATIC, BORDER_MANUAL, BORDER_NONE</p>
<pre>{{ site_settings.body_color }}</pre>	string	<p>[設定]>[マーケティング]>[Eメール]>[設定]>[カラー]での本文の色の設定。16 進値を出力します。</p>
<pre>{{ site_settings.color_picker_favorite_1 }}</pre>	string	<p>[設定]>[マーケティング]>[Eメール]>[設定]>[カラー]でのお気に入りの色 1 の設定。16 進値を出力します。タグを変更して他のお気に入りの色設定を設定するには、1 を 2~6 に置き換えます。</p>

<code>{{ site_settings.primary_accent_color }}</code>	string	[設定]>[マーケティング]>[Eメール]>[設定]>[カラー]での基本アクセント色の設定。16進値を出力します。
<code>{{ site_settings.primary_font }}</code>	enumeration	[設定]>[マーケティング]>[Eメール]>[設定]>[フォント]での基本フォントの設定。ドロップダウンの値を出力します。
<code>{{ site_settings.primary_font_color }}</code>	string	[設定]>[マーケティング]>[Eメール]>[設定]>[フォント]での基本フォント色の設定。16進値を出力します。
<code>{{ site_settings.primary_font_size }}</code>	string	[設定]>[マーケティング]>[Eメール]>[設定]>[フォント]での基本フォントサイズの設定。"px"を含みます。
<code>{{ site_settings.secondary_accent_color }}</code>	string	[設定]>[マーケティング]>[Eメール]>[設定]>[カラー]での第2基本フォント色の設定。16進数値を出力します。

<code>{{ site_settings.secondary_font }}</code>	enumeration	[設定]>[マーケティング]>[Eメール]>[設定]>[フォント]での第2基本フォントの設定。ドロップダウンの値を出力します。
<code>{{ site_settings.secondary_font_color }}</code>	string	[設定]>[マーケティング]>[Eメール]>[設定]>[フォント]での第2基本フォント色の設定。16進数値を出力します。
<code>{{ site_settings.secondary_font_size }}</code>	string	[設定]>[マーケティング]>[Eメール]>[設定]>[フォント]での基本フォントサイズの設定。"px"を含みます。

Eメールの変数

HTML Eメールテンプレート、またはEメールレイアウトのHubLテンプレートモジュールのための変数を以下に示します。

変数	タイプ	説明
----	-----	----

<code>{{ background_color }}</code>	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
<code>{{ body_border_color }}</code>	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
<code>{{ body_border_color_choice }}</code>	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
<code>{{ body_color }}</code>	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
<code>{{ content.create_page }}</code>	boolean	Eメールのウェブページバージョンがある場合、この変数は True です。
<code>{{ content.email_body }}</code>		Eメールの本文。この変数はリッチテキストモジュールを表示します。

<code>{{ content.emailbody_plaintext }}</code>	string	プレーンテキストの E メール本文をオプションで上書きします。
<code>{{ content.from_name }}</code>	string	Eメールの送信者名。
<code>{{ content.reply_to }}</code>	string	Eメールの返信用アドレス。
<code>{{ content.subject }}</code>	string	Eメールの件名
<code>{{ email_body_border_css }}</code>	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
<code>{{ email_body_padding }}</code>	string	Eメール本文のパディング設定 この設定は[設定]>[マーケティング]>[Eメール]>[設定]>[サイズ]にあります。
<code>{{ email_body_width }}</code>	string	Eメール本文の幅の設定。この設定は[設定]>[マーケティング]>[Eメール]>[設定]>[サイズ]にあります。

{{ primary_accent_color }}	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
{{ primary_font }}	enumeration	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
{{ primary_font_color }}	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
{{ primary_font_size }}	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
{{ primary_font_size_num }}	string	[設定]>[マーケティング]>[E メール]>[設定]>[フォント]のフォントサイズ番号を出力します。 "px"は含まれません。

{{ secondary_accent_color }}	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
{{ secondary_font }}	enumeration	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
{{ secondary_font_color }}	string	前述のカラーとフォントの設定 に対応する E メールテンプレート専用のエイリアス。
{{ primary_font_size_num }}	string	[設定]>[マーケティング]>[E メール]>[設定]>[フォント]のフォントサイズ番号を出力します。 "px"は含まれません。
{{ site_settings.company_street_address_2 }}	string	[設定]>[マーケティング]>[E メール]>[設定]>[フッター]の住所 2 を出力します。

<code>{{ site_settings.office_location_name }}</code>	string	[設定]>[マーケティング]>[Eメール]>[設定]>[フッター]のオフィス所在地名を出力します。
<code>{{ subscription_confirmation_url }}</code>	string	サブスクリプション設定確認ページの URL を出力します。この URL は送信時に動的に生成されます。
<code>{{ subscription_name }}</code>	string	Eメールに対して指定された Eメールタイプの名前を出力します。
<code>{{ unsubscribe_anchor }}</code>	string	"unsubscribe (サブスクライブ解除)"という語がサブスクライブ解除ページにリンクされているアンカータグを生成します。

<code>{{ unsubscribe_link_all }}</code>	string	サブスクリプション設定を管理するためのリンクではなく、すべての E メールコミュニケーションをサブスクライブ解除するリンクを表示します。
<code>{{ unsubscribe_section }}</code>	string	サブスクライブ解除リンクとヘルプテキストが含まれているサブスクライブ解除セクションを表示します。
<code>{{ view_as_page_section }}</code>	string	Eメールのウェブページバージョンを表示するリンクとヘルプテキストを生成します。
<code>{{ view_as_page_url }}</code>	string	Eメールのウェブページバージョンを表示するリンクを生成します。

ウェブサイトページの変数

サイトページ、ランディングページ、システムページ、ブログに使用できる変数を以下に示します。

変数	タイプ	説明
<code>{{ builtin_body_classes }}</code>	string	この変数は、テンプレートを使用して作成されたコンテンツのマークアップ（コンテンツのタイプ、コンテンツ名など）を区別するために役立つクラスを動的に出力します。これにより、さまざまなタイプのコンテンツや特定のページのスタイルを容易に設定できます。この変数は、コード化テンプレートの本文タグの class 属性で使用する必要があります。
<code>{{ request_contact.is_logged_in }}</code>	string	この変数は、リクエストしているコンタクトがウェブサイトのゲートコンテンツにログインしているかどうかを定義します（詳細については オーディエンスアクセスの制御に関するドキュメント をご覧ください）。この変数は、リクエストしているコンタクトがログインしている場合は True を返し、ログアウトしている場合は False を返します。URL https://www.yourdomain.com/_hcms/mem/logout にコンタクトを移動させることで、コンタクトをログアウトできます。

この変数を使用すると、ページキャッシュが無効になります。

`{{ request_contact.list_memberships }}` string

この変数は、コンタクトがメンバーであるリストを表す ID のディクショナリを返します。

この変数を使用すると、ページキャッシュが無効になります。

`{{ content.language }}` dict

この変数は、ページの言語設定に関する情報のディクショナリを返します。

`{{ content.language.languageTag }}` はページの言語 ID ("en"、"es" など) を返します。

`{{ content.language.textDirection }}` はページの言語のテキストの方向 ("rtl" や "ltr" など) を返します。

HTTP リクエストの変数

HTTP ページリクエストに関する情報を出力する変数を以下に示します。

変数	タイプ	説明
<code>{{ request.cookies }}</code>	dict	Cookie の値にマップされた Cookie 名のディクショナリ。 <i>この変数を使用すると、ページキャッシュが無効になります。</i>
<code>{{ request.domain }}</code>	string	このページにアクセスするために使用するドメイン
<code>{{ request.full_url }}</code>	string	このページにアクセスするために使用する URL とクエリ文字列。
<code>{{ request.path }}</code>	string	URL のパスコンポーネント
<code>{{ request.path_and_query }}</code>	string	URL のパスとクエリコンポーネント
<code>{{ request.query }}</code>	string	URL のクエリ文字列コンポーネント

<code>{{ request.query_dict }}</code>	dict	名前->値ディクショナリに変換されたクエリ文字列
<code>{{ request.referrer }}</code>	string	HTTP リファラー (現在のページにリンクされているページの URL)。 <i>この変数を使用すると、ページキャッシュが無効になります。</i>
<code>{{ request.remote_ip }}</code>	string	訪問者の IP アドレス。 <i>この変数を使用すると、ページキャッシュが無効になります。</i>
<code>{{ request.scheme }}</code>	string	リクエストのプロトコル (http または https)
<code>{{ request.search_engine }}</code>	string	このページの検索に使用された検索エンジン (該当する場合)。google、aol、live、yahoo、images.google など。
<code>{{ request.search_keyword }}</code>	string	このページの検索に使用されたキーワードフレーズ (該当する場合)。
<code>{{ request.headers }}</code>	string	利用可能なリクエストヘッダーのディクショナリ

ブログの変数

ブログテンプレートで使用可能な変数を以下に示します。記事リストでのみ使用できる変数と、ブログ記事でのみ使用できる変数があります。

変数	タイプ	説明
<pre> {{ blog_author }} </pre>	string	<p>この変数には、ブログ執筆者リストページのブログ執筆者情報が含まれています。ブログ執筆者リストのマークアップを表示する条件付きロジックを作成するときに使用できます。また、以下のプロパティも含まれています。</p> <p>blog_author.avatar, blog_author.bio, blog_author.display_name, blog_author.email, blog_author.facebook, blog_author.google_plus, blog_author.has_social_profiles, blog_author.linkedin, blog_author.twitter, and blog_author.website</p>

<code>{{ content.blog_post_author }}</code>	string	<p>この変数には、特定の記事の各ブログ記事執筆者の情報が含まれています。以下のプロパティが含まれています。</p> <p>content.blog_post_author.avatar, content.blog_post_author.bio, content.blog_post_author.display_name, content.blog_post_author.email, content.blog_post_author.facebook, content.blog_post_author.google_plus, content.blog_post_author.has_social_profiles, content.blog_post_author.linkedin, content.blog_post_author.slug, content.blog_post_author.twitter, and content.blog_post_author.website</p>
<code>{{ content.blog }}</code>	string	グループのエイリアス。
<code>{{ content.comment_count }}</code>	integer	現在のブログ記事のコメント数。
<code>{{ content.comment_list }}</code>	string	現在のブログ記事のコメントのリスト
<code>{{ current_page_num }}</code>	integer	ビューの現在のブログ記事ページのインデックス (整数) 。
<code>{{ content.featured_image }}</code>	string	ブログの公開時に選択されたサムネイル画像のソース URL。

<code>{{ content.featured_image_alt_text }}</code>	string	サムネイル画像の代替テキスト
<code>{{ last_page_num }}</code>	integer	ビューの最終ブログ記事ページのインデックス (整数) 。
<code>{{ content.comment_count }}</code>	integer	現在のブログ記事のコメント数。
<code>{{ next_page_num }}</code>	integer	ビューの次のブログ記事ページのインデックス (整数)
<code>{{ content.next_post_featured_image }}</code>	string	次のブログ記事のサムネイル画像の URL (存在する場合) 。
<code>{{ content.next_post_name }}</code>	string	次のブログ記事の名前 (存在する場合) 。
<code>{{ content.next_post_slug }}</code>	string	次のブログ記事の URL スラッグ (存在する場合) 。
<code>{{ content.post_body }}</code>	string	ブログ記事の本文

{{ content.post_list_content }}	string	リストページに表示するために変更されたブログ記事の本文コンテンツ最終的な結果は、[設定]>[ウェブサイト]>[ブログ]の概要設定の影響を受けます。サムネイル画像が設定で有効になっている場合、この変数を指定すると、「続きを読む」セパレーターよりも上の画像がすべて自動的に削除されます。
{{ content.post_list_summary_featured_image }}	string	リストレイアウトのサムネイル画像。この変数は、[設定]>[ウェブサイト]>[ブログ]の設定の影響を受けます。
{{ content.post_summary }}	string	ブログ記事の概要。このコンテンツは、ブログエディターの「続きを読む」セパレーターにより決定します。
{{ content.previous_post_featured_image }}	string	前のブログ記事のサムネイル画像の URL (存在する場合)。
{{ content.previous_post_name }}	string	前のブログ記事の名前 (存在する場合)。

<code>{{ content.previous_post_slug }}</code>	string	前のブログ記事の URL スラッグ (存在する場合)。
<code>{{ content.publish_date_localized }}</code>	string	ブログ記事が公開された日時を表す文字列。ブログの言語と日付書式設定に基づいて書式設定されます。
<code>{{ content.simple_list_page }}</code>	boolean	リクエストされたページが、すべてのブログ記事へのリンクを含む「すべての記事」ページであるかどうかを示すブール値。
<code>{{ content.topic_list }}</code>	string	ブログ記事に関連付けられているトピックオブジェクトのリスト。各トピックオブジェクトから name および slug プロパティにアクセスできます。
<code>{{ contents }}</code>	string	コンテンツは、 ブログリストページ (is_listing_view) で使用可能な for ループで繰り返されるブログ記事のシーケンスです。
<code>{{ contents.total_count }}</code>	integer	リスト (定期、トピック、執筆者など) の記事の合計数

<code>{{ contents.total_page_count }}</code>	integer	ページあたりの記事の数に基づく記事ページの合計数。
<code>{{ contents_topics }}</code>	string	コンテンツ (記事のシーケンス) のすべてのブログトピックのリストを取得します。
<code>{{ group }}</code>	dict	ブログ全体に関連する変数を含むディクショナリ。
<code>{{ group.absolute_url }}</code>	string	ブログのベース URL
<code>{{ group.allow_comments }}</code>	boolean	コメントが許可されている場合は True に評価されます。
<code>{{ group.description }}</code>	string	[設定]>[ウェブサイト]>[ブログ] のブログのメタディスクリプション。特定のリストページのメタディスクリプションに使用されます。
<code>{{ group.header }}</code>	string	ブログのヘッダー
<code>{{ group.html_title }}</code>	string	<title>タグに表示されるこのブログのタイトル

<code>{{ group.id }}</code>	string	ブログの固有の ID。この ID は、特定のブログのブログダッシュボードの URL に含まれています。
<code>{{ group.public_title }}</code>	string	表示されるページの最上部に示されるこのブログのタイトル。
<code>{{ group.slug }}</code>	string	このブログのパス
<code>{{ topic }}</code>	dict	topic 変数を使用して、トピックリストのマークアップを表示できます。またこの変数には、プロパティ <code>topic.name</code> と <code>topic.slug</code> が含まれています。

HubDB の変数

HubDB で動的ページを作成するときに使用される変数を以下に示します。これらの変数は動的テンプレートでのみ使用できます。

<code>{{ dynamic_page_hubdb_table_id }}</code>	long	ページエディターの[詳細設定]タブで選択されているテーブルの ID。
--	------	------------------------------------

<code>{{ dynamic_page_hubdb_row }}</code>	dict	ページリクエストパスに一致する動的ページの HubDB 行。リクエストがリストページを対象としている場合、この値は <code>null</code> になります。
<code>{{ row.hs_id }}</code>	long	HubDB 行の内部 ID。
<code>{{ row.hs_name }}</code>	string	HubDB 行の名前。
<code>{{ row.hs_path }}</code>	string	HubDB 行のパス。 <code>dynamic_page_hubdb_table_id</code> で指定されたテーブルの 1 行に対するリクエストを解決するために使用されます。
<code>{{ row.hs_child_table_id }}</code>	long	HubDB 行の子テーブル ID。ネストしたテンプレートを作成するために使用できます。
<code>{{ row.hs_parent_row }}</code>	dict	HubDB 行の親行。ネストしたテンプレートの子テーブルを使用する場合にのみ使用できます。

`{{ dynamic_page_route_level }}` integer 複数レベルの動的テンプレート内のページの現在の深さ。値は 0 から始まり、テーブルの層が増えるたびに増加します。

メニューノードの変数

[HubL メニュー関数](#)により返されるオブジェクトで使用できる変数を以下に示します。

変数	タイプ	説明
<code>{{ node.label }}</code>	string	ページのメニューラベル。
<code>{{ node.url }}</code>	string	ページの URL。
<code>{{ node.pageId }}</code>	number	ページの ID (HubSpot 内の場合) 。
<code>{{ node.contentGroupId }}</code>	number	ページのブログ ID (HubSpot ブログ記事の場合) 。
<code>{{ node.parentNode }}</code>	object	現在のノードの親ノード。現在のノードは、親ノードの <code>children</code> プロパティに含まれます。

<code>{{ node.children }}</code>	list	現在のノードの子ノードのリスト。
<code>{{ node.activeBranch }}</code>	boolean	ノードが現在のページの最上位分岐内にある場合は True。
<code>{{ node.activeNode }}</code>	boolean	ノードが現在のページである場合は True。
<code>{{ node.level }}</code>	number	上位レベルのノードからの現在のノードのレベルの数 (深さ) 。
<code>{{ node.pageTitle }}</code>	string	コンテンツページの名前 (HubSpot 内の場合) 。
<code>{{ node.slug }}</code>	string	ページのパススラッグ。
<code>{{ node.linkTarget }}</code>	string	ページのターゲットリンク。

HubL でサポートされているタグ

HubL でサポートされているフィルター

HubL でサポートされているフィルター

フィルターは HubL の最終出力に影響を与えます。さまざまな HubL ステートメントと式にフィルターを適用することで、サーバーから出力されるテンプレートマークアップを変更できます。

フィルターの基本構文は `|filtername` です。フィルターは区切り記号で囲まれ、ステートメントまたは式のすぐ後に追加されます。フィルターによっては、追加的なパラメータをカッコ内で指定できる場合があります。文字列パラメータ、数値パラメータ、ブールパラメータを含むフィルターの基本構文は、`|filtername('stringParameter', 10, true)` となります。文字列パラメータを引用符に囲む必要があることに注意してください。また、HubL フィルターにはエイリアスが存在し、それをプライマリフィルターと同じ目的に使用できます。

以下の記事では、サポートされている HubL フィルターをすべて示します。

現在のところ、HubSpot での E メールでは、パーソナライズ変数 (たとえば、contacts 変数、company 変数、deal 変数など) に基づくフィルターは一切サポートされていないことにご注意ください。HubSpot CMS ページおよびブログテンプレートでは、これらのプロパティにフィルターを適用できます。

abs

数値の絶対値を取得します。このフィルターを使用すると、常に正の数値が得られます。

- HubL
- 出力例

```
{% set my_number = -53 %}  
{{ my_number|abs }}
```

add

1つの数値を別の数値に追加します。このフィルターの機能は+演算子と同じです。括弧内のパラメーターは、最初の数値に加えられる加数です。

- HubL
- 出力例

```
{% set my_num = 40 %}  
{{ my_num|add(13) }}
```

attr

ディクショナリの属性をレンダリングします。このフィルターの機能は、content.absolute_url など、ディクショナリに含まれる変数を入力する機能と同等です。パラメーターは、どの属性を入力するかを指定します。

パラメーター: attr('attribute_name')

- HubL
- 出力例

```
{{ content|attr('absolute_url') }}
```

batch

batch フィルターは、1つのシーケンス内の複数項目をグループ化します。2つのパラメーターを受け入れます。最初のパラメーターはバッチに含める項目数を指定し、2番目のパラメーターは、欠落している項目を埋めるために何を含めるかを指定します。

以下の例では、果物の種類からなるシーケンスが変数に入っています。シーケンスを反復処理するループに batch フィルターが適用されます。ネストしたループが3回実行され、行ごとに3種類のフルーツを出力した後、外側ループが再び実行されます。最終的な出力では、フルーツが5種類しかないので、最後の項目が (2番目のパラメーター)に置き換えられることに注意してください。

パラメーター: `batch(linecount, fill_with=None)`

- HubL
- 出力例

```
{% set rows = ['apples', 'oranges', 'pears', 'grapes', 'blueberries'] %}

<table>
{% for row in rows|batch(3, '&nbsp;') %}
  <tr>
    {% for column in row %}
      <td>{{ column }}</td>
    {% endfor %}
  </tr>
{% endfor %}
</table>
```

between_times

指定された時間単位を使用して、2つの日付時刻オブジェクト間の差異を計算します。

有効な時間単位は `nanos`、`micros`、`millis`、`seconds`、`minutes`、`hours`、`half_days`、`days`、`weeks`、`months`、`years`、`decades`、`centuries`、`millennia`、`eras` です。

パラメーター: `between_times(end, timeunit)`

- HubL
- 出力例

```
{% set begin = "2018-07-14T14:31:30+0530"|strptime("yyyy-MM-dd'T'HH:mm:ssZ") %}
{% set end = "2018-07-20T14:31:30+0530"|strptime("yyyy-MM-dd'T'HH:mm:ssZ") %}
{{ begin|between_times(end, 'days') }}
```

bool

テキスト文字列値をブール値に変換します

- HubL
- 出力例

```
{% if "true"|bool == true %}hello world{% endif %}
```

capitalize

変数値の最初の文字を大文字にします。最初の文字が大文字になり、その他の文字は元の大文字または小文字のままです。

- . HubL
- . 出力例

```
{% set sentence = "the first letter of a sentence should always be capitalized." %}  
{{ sentence|capitalize }}
```

center

center フィルターは空白を使用して、指定されたフィールド長の中でテキストを中央に配置します。HubSpot の HTML コンパイラは自動的に空白を除去するので、このフィルターは推奨されず、特に役立つわけでもありませんが、包括的な説明のためにここで説明しています。パラメーターは、テキストを中心に配置するための空白の長さを指定します。以下の例では、pre タグ内の変数に center フィルターが適用され、空白が除去されないようになっています。

パラメーター: center(width=40)

```
<pre>  
{% set var = "string to center" %}  
{{ var|center(80) }}  
</pre>
```

convert_rgb

HEX 値を RGB 文字列に変換します。これは、色変数を RGB に変換して RGBA CSS 宣言で使用する必要がある場合に便利です。以下の例では、色モジュールによって設定された値が RGB 値に変換され、RGBA CSS 宣言で使用されます。

- . HubL
- . 出力例

```
{% set my_color = "#FFFFFF" %}
```

```
{{ my_color|convert_rgb }}
```

```
{% color "my_color" color="#000000", export_to_template_context=True %}
```

```
<div style="background: rgba({{ widget_data.my_color.color|convert_rgb }}, .5)"></div>
```

cut

値から文字列を除去します。このフィルターを使用すると、文字列の特定の部分を照合して切り取ることができます。パラメーターは、削除対象となる文字列部分を指定します。以下の例では、スペースと単語「world」が元の変数値から除去されます。

パラメーター: `cut('characters to cut')`

- [HubL](#)
- [出力例](#)

```
{% set my_string = "Hello world." %}
```

```
{{ my_string|cut(' world') }}
```

datetimeformat

日付オブジェクトの書式を設定します。既定では、HubSpot の日付変数はタイムスタンプとして出力されます。datetimeformat フィルターを使用すると、そのタイムスタンプを読みやすい日時に変換できます。以下の表に示すフィルターのパラメーターは、時間変数が最終的にどのようにレンダリングされるかを決定します。

フィルターの日付入力が null である場合は、現在の日時を想定します。

2 番目のパラメーター (任意指定) を使用すると、タイムゾーンを指定できます。タイムゾーンは、[サポートされる Java 8 形式](#) でなければなりません。

エイリアス: date

- HubL
- 出力例

```
{{ content.updated|datetimeformat('%B %e, %Y') }}  
{{ content.publish_date|datetimeformat('%B %e, %Y %l %p') }}  
{{ content.publish_date|datetimeformat('%B %e, %Y %l %p', 'America/Los_Angeles') }}
```

ディレクト イブ	意味	例
%a	曜日（ロケールで設定された省略名）。	Sun、Mon、... Sat (en_US の場合) So、Mo、... Sa (de_DE の場合)
%A	曜日（ロケールで設定されたフルネーム）。	Sunday、 Monday、... Saturday (en_US の 場合) Sonntag、 Montag、... Samstag (de_DE の 場合)

%w	<p>10 進数で表した曜日。1 は日曜日、7 は土曜日です。</p> <p>注：これは、0 で始まる Python の曜日とは異なります。</p>	1、2、... 7
%d	月の日付 (ゼロでパディングされた 10 進数)。	01, 02, ..., 31
%e	月の日付 (パディングのない 10 進数)。	1、2、... 31
%b	月 (ロケールで設定された省略名)。	<p>Jan、Feb、... Dec (en_US の場合)</p> <p>Jan、Feb、... Dez (de_DE の場合)</p>
%B	月 (ロケールで設定されたフルネーム)。	<p>January、 February、... December (en_US の場合)</p> <p>Januar、 Februar、... Dezember (de_DE の場合)</p>

%m	月 (ゼロでパディングされた 10 進数)。	01、02、... 12
%y	年 (世紀なし、ゼロでパディングされた 10 進数)。	00、01、... 99
%Y	年 (世紀あり、10 進数)。	1970, 1988, 2001, 2013
%H	時間 (24 時間表記、ゼロでパディングされた 10 進数)。	00, 01, ..., 23
%I	時間 (12 時間表記、ゼロでパディングされた 10 進数)。	01, 02, ..., 12
%k	時間 (24 時間表記、0 ~ 23 の範囲の 10 進数、1 桁の場合はその前に空白)。	0, 1, ..., 24
%l	時間 (12 時間表記、0 ~ 12 の範囲の 10 進数、1 桁の場合はその前に空白)。注: これは小文字の L です。	1, 2, ..., 12
%p	AM または PM に相当する、ロケールで設定された表記。	AM、PM (en_US の場合) am、pm (De_DE の場合)

%M	分 (ゼロでパディングされた 10 進数)。	00, 01, ..., 59
%S	秒 (ゼロでパディングされた 10 進数)。	00, 01, ..., 59
%f	ミリ秒 (左側がゼロでパディングされた 10 進数)。	000000, 000001, ..., 999999
%z	+HHMM 形式または -HHMM 形式の UTC オフセット (オブジェクトが naive である場合は空の文字列)。	(空)、+0000、-0400、+1030
%Z	タイムゾーン名 (オブジェクトが naive である場合は空の文字列)。	(空)、UTC、EST、CST
%j	年全体での日付 (ゼロでパディングされた 10 進数)。	001, 002, ..., 366
%U	年全体での週番号 (日曜日を週の第 1 日とする)。ゼロでパディングされた 10 進数。最初の日曜日より前にくる新年のすべての日は、第 0 週と見なされます。	00, 01, ..., 53
%W	年全体での週番号 (月曜日を週の第 1 日とする)。10 進数。最初の月曜日よ	00, 01, ..., 53

り前にくる新年のすべての日は、第 0 週と見なされます。

%c	ロケールで設定された適切な日付/時間表記。	Tue Aug 16 21:30:00 1988 (en_US の場合) Di 16 Aug 21:30:00 1988 (de_DE の場合)
%x	ロケールで設定された適切な日付表記。	08/16/88 (なし) 08/16/1988 (en_US の場合) 16.08.1988 (de_DE の場合)
%X	ロケールで設定された適切な時刻表記。	21:30:00 (en_US の場合) 21:30:00 (de_DE の場合)
%%	%のリテラル文字。	%

default

値が未定義の場合に、最初のパラメーターを返します。それ以外の場合、変数の値を出力します。false と評価される変数で default を使用するには、2 番目のパラメーターを true に設定する必要があります。以下の最初の例では、変数が未定義の場合にメッセージを出力します。2 番目の例では、未定義ではない空の文字列にフィルターを適用しますが、2 番目のパラメーターが原因でメッセージが出力されます。

default フィルターのエイリアス: d

パラメーター: default(default_value='variable is undefined', boolean)

- . HubL
- . 出力例

```
{{ my_variable|default('my_variable is not defined') }}  
{{ ""|default('the string was empty', true) }}
```

dictsort

ディクショナリを並べ替えて「キー、値」のペアを生成します。ディクショナリは既定では並べ替えられません、キーまたは値を基準に並べ替えてディクショナリを出力できます。最初のパラメーターは、並び替えで大文字と小文字を区別するかどうかを決定するブール値です。2 番目のパラメーターは、キーと値のどちらを基準にディクショナリを並べ替えるかを決定します。以下の例では、コンタクトに関する既知の詳細情報をすべて含む、並び替えられたコンタクトディクショナリを出力します。

パラメーター: dictsort(case_sensitive=False, by='value')

- . HubL
- . 出力例

```
{% for item in contact|dictsort(false, 'value') %}  
  {{item}}
```



```
{% endfor %}
```

difference

このフィルターは、2つのセットまたはリストの差異を返します。フィルターから返されるリストには、最初のリストに含まれ、2番目には含まれない一意の要素がすべて格納されます。

パラメーター: `difference(list)`

- HubL
- 出力例

```
{{ [1, 2, 3]|difference([2, 3, 4, 5]) }}
```

divide

現在の値を除数で割り算します。渡されるパラメーターは除数です。このフィルターは、[演算子](#)の代わりにになります。

パラメーター: `divide(divisor=2)`

- HubL
- 出力例

```
{% set numerator = 106 %}  
{{ numerator|divide(2) }}
```

divisible

divisible フィルターは divisibleby 式テストの代わりになります。特定の数で値を割り切れる場合に true と評価されます。

パラメーター: `divisible(divisor=2)`

- HubL
- 出力例

```
{% set num = 10 %}
{% if num|divisible(2) %}
The number is divisible by 2
{% endif %}
```

escape

文字列内の `&` `<` `>` `'` 文字を、HTML セーフなシーケンスに変換します。そのような文字を含む可能性のあるテキストを HTML で表示するには、これを使用できます。戻り値はマークアップ文字列としてマークされます。

エイリアス: `e`

- HubL
- 出力例

```
{% set escape_string = "<div>This markup is printed as text</div>" %}
{{ escape_string|escape }}
```

escape_jinja

文字列内の {} 文字を、Jingava セーフなシーケンスに変換します。そのような文字を含む可能性のあるテキストを Jinjava で表示するには、このフィルターを使用できます。戻り値はマークアップ文字列としてマークされます。

- . HubL
- . 出力例

```
{% set escape_string = "{{This markup is printed as text}}" %}
{{ escape_string|escape_jinja }}
```

escapejs

文字列をエスケープして、JavaScript 変数宣言に文字列を安全に挿入できるようにします

- . HubL
- . 出力例

```
{% set escape_string = "\tThey said 'This string can safely be inserted into
JavaScript.'" %}
{{ escape_string|escapejs }}
```

escapejson

文字列をエスケープして、その文字列を JSON 値として使用できるようにします

- . HubL
- . 出力例

```
{% set your_string = "\tTesting a \"quote for the week\""
```

```
{{ your_string|escapejson }}
```

filesizeformat

値の書式を設定して、「人間が読みやすい」ファイルサイズにします (たとえば 13 kB、4.1 MB、102 Bytes など)。既定では 10 進プレフィックス (Mega、Giga など) が使用され、パラメーターが True に設定されている場合はバイナリプレフィックス (Mebi、Gibi) が使用されます、

パラメーター: `filesizeformat(boolean)`

- . HubL
- . 出力例

```
{% set bytes = 100000 %}  
{{ bytes|filesizeformat }}
```

first

シーケンス内の最初の項目を返します。

- . HubL
- . 出力例

```
{% set my_sequence = ['Item 1', 'Item 2', 'Item 3'] %}  
{{ my_sequence|first }}
```

float

値を浮動小数点値に変換します。正常に変換できない場合は 0.0 を返します。最初のパラメーターを使用すると、この既定の動作をオーバーライドできます。

パラメータ: `float(0.1)`

- . HubL
- . 出力例

```
{% text "my_text" value='25', export_to_template_context=True %}  
{{ widget_data.my_text.value|float + 28 }}
```

forceescape

HTML エスケープを厳密に実施します。実際に HubSpot の環境ではダブルエスケープを使用するケースがないので、これは通常、escape フィルターと同じ動作をします。

- . HubL
- . 出力例

```
{% set escape_string = "<div>This markup is printed as text</div>" %}  
{{ escape_string|forceescape }}
```

format

Python 文字列の書式設定をオブジェクトに適用します。%s を別の変数に置換できます。

- . HubL
- . 出力例

```
{{ "Hi %s %s"|format(contact.firstname, contact.lastname) }}
```

format_currency

パラメーターとして渡されるポータルのもとの既定の通貨とロケールに基づき、特定の数値を通貨として書式設定します。

このフィルターは 1 つの引数 *locale* を受け入れます。これは、あらゆる Java ロケール言語タグを受け入れます。既定はページのロケールです。形

式:ISO639LanguageCodeInLowercase-ISO3166CountryCodeInUppercase

- . HubL
- . 出力例

```
{% set price = 100 %}  
{{ price|format_currency("en-US") }}  
{{ price|format_currency("fr-FR") }}
```

fromjson

JSON 文字列をオブジェクトに変換します。

- . HubL
- . 出力例

```
{% set obj = '{ "name": "Brian", "role": "Owner" }' %}  
{{ obj|fromjson }}
```

geo_distance

地球上の 2 地点間の楕円 2D 距離を計算します。

- . HubL
- . 出力例

```
<!-- in the example below the HubDB Location = 42.3667, -71.1060 (Cambridge, MA) |
Chicago, IL = 37.3435, -122.0344 -->
{{ row.location | geo_distance(37.3435, -122.0344, "mi") }} MI
```

groupby

groupBy フィルターは、共通の属性を基準にしてオブジェクトのシーケンスをグループ化します。パラメーターは、グループ化の基準となる共通属性を設定します。

パラメータ: groupby(attribute='name_of_attribute')

- HubL
- 出力例

```
<ul>
{% for group in contents|groupBy('blog_post_author') %}
  <li>{{ group.grouper }}
    <ul>
      {% for content in group.list %}
        <li>{{ content.name }}</li>
      {% endfor %}
    </ul>
  </li>
{% endfor %}
</ul>
```

indent

indent フィルターは空白を使用して、特定のフィールド長の範囲内でテキストのインデントを設定します。HubSpot の HTML コンパイラは自動的に空白を除去するので、このフィルターは推奨されず、特に役立つわけでもありませんが、包括的な説明のためにここで説明しています。以下の例では、pre タグ内の変数に center フィルターが適用され、空白が除去されないようになっています。最初のパラメーターは空白の量を制御し、2 番目のブール値は第 1 行をインデントするかどうかを切り替えます。

パラメーター: `indent(width=4, indentfirst=False)`

```
<pre>
{% set var = "string to indent" %}
{{ var|indent(2, true) }}
</pre>
```

int

値を整数に変換します。正常に変換できない場合は 0 を返します。最初のパラメーターを使用すると、この既定の動作をオーバーライドできます。

パラメーター: `int(0)`

- HubL
- 出力例

```
{% text "my_text" value='25', export_to_template_context=True %}
{{ widget_data.my_text.value|int + 28 }}
```

intersect

このフィルターは、2 つのセットまたはリストの共通部分を返します。フィルターから返されるリストには、両方のリストに含まれる一意の要素がすべて格納されます。

パラメーター: `intersect(list)`

- . HubL
- . 出力例

```
{{ [1, 2, 3]|intersect([2, 3, 4, 5]) }}
```

ipaddr

値が有効な IPv4 または IPv6 アドレスである場合には、`true` と評価されます。

- . HubL
- . 出力例

```
{% set ip = '1.0.0.1' %}  
{% if ip|ipaddr %}  
  The string is a valid IP address  
{% endif %}
```

join

シーケンス内の文字列を連結した 1 つの文字列を返します。要素間は既定では空文字列で区切られますが、任意指定のパラメーターを使って区切り文字を定義できます。2 番目のパラメーターを使用すると、結合する属性を指定できます。

`join(d=", attribute='name')`

- . HubL
- . 出力例

```
{% set my_list = [1, 2, 3] %}
{% set sep = "---" %}
{{ my_list|join }}
{{ my_list|join('|') }}
{{ my_list|join(sep) }}
```

last

シーケンスの最後の項目を返します。

- . HubL
- . 出力例

```
{% set my_sequence = ['Item 1', 'Item 2', 'Item 3'] %}
{% my_sequence|last %}
```

length

シーケンスまたはマッピング内の項目数を返します。

エイリアス: count

- . HubL
- . 出力例

```
{% set services = ['Web design', 'SEO', 'Inbound Marketing', 'PPC'] %}
{{ services|length }}
```

list

数値を変換して1つのリストにします。文字列である場合は、複数文字からなるリストが返されます。シーケンスに文字列を追加するには、それをシーケンス区切り[]と共に文字列変数に追加します。

- [HubL](#)
- [出力例](#)

```
{% set one = 1 %}
{% set two = 2 %}
{% set three = 3 %}
{% set four = ["four"] %}
{% set list_num = one|list + two|list + three|list + four|list %}
{{ list_num }}
```

log

数値の自然対数を計算します。2番目のパラメーター（任意指定）を含めると、nを低とする対数を計算できます。

パラメーター: `log(base=e)`

- [HubL](#)
- [出力例](#)

```
{{ 10|log }}
{{ 65536|log(2) }}
```

lower

値をすべて小文字に変換します。

パラメーター: int(0)

- . HubL
- . 出力例

```
{{ text "text" value='Text to MAKE Lowercase', export_to_template_context=True }}  
{{ widget_data.text.value|lower }}
```

map

オブジェクトのシーケンスにフィルターを適用するか、属性を検索します。オブジェクトリストを扱うとき、その中の特定の値だけに対象を絞り込むには、これが便利です。

基本的な使用法は、属性に対するマッピングです。たとえば、ディクショナリの特定の属性において特定の値が存在するかどうかを検査する条件付きロジックを使用する場合があります。または、フィルター名と引数を渡してフィルターを起動させることもできます。

パラメーター: map('attribute')または map('filter_to_apply')

- . HubL
- . 出力例

```
{% set seq = ['item1', 'item2', 'item3'] %}  
{{ seq|map('upper') }}  
{{ content|map(attribute='currentState')|join(', ') }}
```

md5

特定のオブジェクトの **md5** ハッシュを計算します。

- . HubL

- 出力例

```
{{ content.absolute_url|md5 }}
```

minus_time

日時オブジェクトで一定量の時間を差し引きます。有効な時間単位は `nanos`、`micros`、`millis`、`seconds`、`minutes`、`hours`、`half_days`、`days`、`weeks`、`months`、`years`、`decades`、`centuries`、`millennia`、`eras` です。

パラメーター: `minus_time(diff, timeunit)`

- HubL
- 出力例

```
{% set date = "2018-07-14T14:31:30+0530"|strtotime("yyyy-MM-dd'T'HH:mm:ssZ") %}  
{{ date }}  
{{ date|minus_time(2, 'months') }}
```

multiply

1 つの値を数値で乗算します。[*演算子](#)と同じ機能です。

- HubL
- 出力例

```
{% set n = 20 %}  
{{ n|multiply(3) }}
```

plus_time

日時オブジェクトに一定量の時間を加えます。有効な時間単位は `nanos`、`micros`、`millis`、`seconds`、`minutes`、`hours`、`half_days`、`days`、`weeks`、`months`、`years`、`decades`、`centuries`、`millennia`、`eras` です。

パラメーター: `plus_time(diff, timeunit)`

- [HubL](#)
- [出力例](#)

```
{% set date = "2018-07-14T14:31:30+0530"|strptime("yyyy-MM-dd'T'HH:mm:ssZ") %}
{{ date }}
{{ date|plus_time(5, 'days') }}
```

pprint

変数をプリティプリントします。これは、変数のタイプや、デバイスに役立つその他の情報を出力します。

- [HubL](#)
- [出力例](#)

```
{% set this_var = "Variable that I want to debug" %}
{{ this_var|pprint }}
```

random

シーケンスから 1 つの項目をランダムに返します。

- [HubL](#)
- [出力例](#)

```
{% for content in contents|random %}
<div class="post-item">Post item markup</div>
```

```
{% endfor %}
```

regex_replace

正規表現パターンを検索して、それを文字シーケンスに置換します。最初の引数は RE2 形式の正規表現パターン、2 番目は置換文字列です。

RE2 正規表現の構文については、[こちら](#)をご覧ください。

- HubL
- 出力例

```
{{ "contact-us-2"|regex_replace("[^a-zA-Z]", "") }}
```

reject

オブジェクトに式テストを適用して適合するものを拒否することにより、オブジェクトのシーケンスをフィルタリングします。

パラメーター: `reject(exp_test='truthy')`

- HubL
- 出力例

```
{% set some_numbers = [10, 12, 13, 3, 5, 17, 22] %}  
{{ some_numbers|reject('even') }}
```

rejectattr

オブジェクトの属性にテストを適用して適合するものを拒否することにより、オブジェクトのシーケンスをフィルタリングします。最初のパラメーターは、選択する属性を指定します。ドット表記を使用すると、ネストされた属性にアクセスできます。2 番目

は、テストを指定します。3番目のパラメーター（任意指定）は、テストで照合の対象となる値を受け入れます。

パラメーター: `rejectattr(attr='attribute_name', exp_test='truthy', val='<string>')`

- . HubL
- . 出力例

```
{% for content in contents|rejectattr('post_list_summary_featured_image') %}
<div class="post-item">
{% if content.post_list_summary_featured_image %}
  <div class="hs-featured-image-wrapper">
    <a href="{{content.absolute_url}}" title="" class="hs-featured-image-link">
      
    </a>
  </div>
{% endif %}
  {{ content.post_list_content|safe }}
</div>
{% endfor %}
```

replace

サブ文字列のすべてのインスタンスを、新しいものに置換します。最初の引数は、置換されるサブ文字列です。2番目は、置換後の文字列です。3番目の引数 `count`（任意指定）が指定されている場合、最初の `count` 回の出現が置換されます。

パラメーター: `replace(old, new, count=None)`

- . HubL

- . 出力例

```
{% if topic %}<h3>Posts about {{ page_meta.html_title|replace('Blog |', '') }}</h3>{% endif %}
```

reverse

オブジェクトを反転します。または、オブジェクトを逆順に反復処理するイテレーターを返します。

- . HubL
- . 出力例

```
{% set nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] %}  
{% for num in nums|reverse %}  
  {{ num }}  
{% endfor %}
```

root

値の平方根を計算します。任意指定の数パラメーターを含めると、数値の n 乗根を計算できます。

パラメーター: `root(nth_root=2)`

- . HubL
- . 出力例

```
{{ 16|root }}  
{{ 625|root(4) }}
```

round

数値を特定の精度に丸めます。最初のパラメーターは精度を指定します (既定は 0)。

2 番目は丸め手法を次のように指定します。

- 'common'は四捨五入
- 'ceil'は常に切り上げ
- 'floor'は常に切り捨て

手法を指定しない場合、'common'が使用されます。

パラメーター: `round(precision, 'rounding_method')`

- [HubL](#)
- [出力例](#)

```
{{ 52.5|round }}  
{{ 52.5|round(0, 'floor') }}
```

safe

値に「セーフ」のマークを付けます。つまり、自動エスケープが有効になっている環境で、この変数はエスケープされないという意味です。

- [HubL](#)
- [出力例](#)

```
{{ content.post_list_content|safe }}
```

select

オブジェクトにテストを適用して適合するものだけを選ぶことにより、オブジェクトのシーケンスをフィルタリングします。

パラメーター: `select(exp_test='truthy')`

- HubL
- 出力例

```
{% set some_numbers = [10, 12, 13, 3, 5, 17, 22] %}
{{ some_numbers|select('even') }}
```

selectattr

オブジェクトの属性にテストを適用して適合するものだけを選ぶことにより、オブジェクトのシーケンスをフィルタリングします。最初のパラメーターは、選択する属性を指定します。ドット表記を使用すると、ネストされた属性にアクセスできます。2番目は、テストを指定します。3番目のパラメーター（任意指定）は、テストで照合対象となる値を受け入れます。

パラメーター: `selectattr(attr='attribute_name', exp_test='truthy', val='<string>')`

以下の例ではブログ記事ループがフィルタリングされて、サムネイル画像を含む記事だけが返されます。

- HubL
- 出力例

```
{% for content in contents|selectattr('post_list_summary_featured_image') %}
<div class="post-item">
  {% if content.post_list_summary_featured_image %}
    <div class="hs-featured-image-wrapper">
```

```

    <a href="{{content.absolute_url}}" title="" class="hs-featured-image-link">
      
    </a>
  </div>
{% endif %}
{{ content.post_list_content|safe }}
</div>
{% endfor %}

```

この例の 3 番目のパラメーターは、`equalto` を使って特定の値をテストする目的で使用されています。

```
{% set people_with_dogs = people|selectattr('animals','equalto','dog') %}
```

shuffle

シーケンス内の反復の順序をランダム化します。以下の例は、標準的なブロググループをシャッフルします。

- HubL
- 出力例

```

{% for content in contents|shuffle %}
<div class="post-item">Markup of each post</div>
{% endfor %}

```

slice

イテレーターをスライスして、これらの項目を含むリストの一覧を返します。最初のパラメーターはスライスされる項目の数を指定し、2番目のパラメーターは空のスライスに埋める文字を指定してください。

パラメーター: `slice(slices, 'filler')`

- HubL
- 出力例

```
{% set items = ['laptops', 'tablets', 'smartphones', 'smart watches', 'TVs'] %}  
<div class="columnwrapper">  
  {% for column in items|slice(3, ' ') %}  
    <ul class="column-{{ loop.index }}">  
      {% for item in column %}  
        <li>{{ item }}</li>  
      {% endfor %}  
    </ul>  
  {% endfor %}  
</div>
```

sort

反復可能データを並べ替えます。このフィルターを使って HubSpot で属性を基準に並べ替える際には、すべてのパラメーターが必須です。最初のパラメーターは、並べ替え順序を逆にするブール値です。2番目のパラメーターは、並び替えで大文字と小文字を区別するかどうかを決定します。最後のパラメーターは、並べ替え基準となる属性を指定します。以下の例では1つのブログの記事がレンダリングされ、名前のアルファベット順で並べ替えられます。

パラメーター: `reverse=False, case_sensitive=False, attribute='name')`

- . HubL
- . 出力例

```
{% set my_posts = blog_recent_posts('default', limit=5) %}  
  
{% for item in my_posts|sort(False, False, 'name') %}  
{{ item.name }}<br>  
  
{% endfor %}
```

split

入力文字列を特定の区切り文字で分割して、リストにします。最初のパラメーターは、変数を分割する基準となる区切り文字を指定します。2番目のパラメーターは、変数の分割回数を決定します。残りの項目が存在する場合、それらはグループ化されたままになります。以下の例では、複数の名前からなる1つの文字列が「;」で分割されます。最初の4回だけ分割されます。

パラメーター: `split('character_to_split_by', number_of_splits)`

- . HubL
- . 出力例

```
{% set string_to_split = "Stephen; David; Cait; Nancy; Mike; Joe; Niall; Tim; Amanda" %}  
{% set names = string_to_split|split(';', 4) %}  
<ul>  
{% for name in names %}  
<li>{{ name }}</li>
```

```
{% endfor %}  
</ul>
```

string

別の種類の変数を文字列に変換します。以下の例では、整数が文字列に変換されます (変数の種類の変化を確認する目的で pprint が使用されています)。

- [HubL](#)
- [出力例](#)

```
{% set number_to_string = 45 %}  
{{ number_to_string|string|pprint }}
```

striptags

SGML/XML タグを除去して、隣接する空白を 1 つのスペースで置換します。このフィルターを使用して、変数内の HTML タグを除去できます。

- [HubL](#)
- [出力例](#)

```
{% set some_html = "<div><strong>Some text</strong></div>" %}  
{{ some_html|striptags }}
```

strtotime

日時文字列と日時書式を変換して、日時オブジェクトにします。

パラメーター: strtotime(datetimeFormat='<string>')

- [HubL](#)
- [出力例](#)

```
{{ "2018-07-14T14:31:30+0530"|strtotime("yyyy-MM-dd'T'HH:mm:ssZ")|unixtimestamp }}
```

sum

シーケンス内の数値をすべて加算します。最初のパラメーターでは任意指定の属性を指定でき、2番目のパラメーターは、変数内に加算するものがない場合に返す値を設定します。

パラメーター: `sum('attribute', nothing_to_sum)`

- HubL
- 出力例

```
{% set sum_this = [1, 2, 3, 4, 5] %}  
{{ sum_this|sum }}  
Total: {{ items|sum(attribute='price') }}
```

symmetric_difference

このフィルターは、2つのセットまたはリストの対称差を返します。フィルターから返されるリストには、最初のリストに含まれ2番目には含まれない一意の要素、または2番目のリストに含まれ最初には含まれない一意の要素がすべて格納されます。

パラメーター: `symmetric_difference(list)`

- HubL
- 出力例

```
{{ [1, 2, 3]|symmetric_difference([2, 3, 4, 5]) }}
```


title

タイトルケース化された値を返します。つまり、単語の先頭が大文字に、残りの文字はすべて小文字になります。

- [HubL](#)
- [出力例](#)

```
{% text "my_title" label='Enter a title', value='My title should be titlecase',  
export_to_template_context=True %}  
{{ widget_data.my_title.value|title }}
```

tojson

オブジェクトを JSON 文字列として書き出します。

- [HubL](#)
- [出力例](#)

```
{% for content in contents %}  
  {{ content.blog_post_author|tojson }}  
{% endfor %}
```

trim

先頭および末尾の空白を除去します。HubSpot 自体がマークアップから空白をトリムしますが、包括的な説明にするためにこのフィルターを含めています。

- [HubL](#)

- 出力例

```
{{ " remove whitespace " }}  
{{ " remove whitespace "|trim }}
```

truncate

特定の文字数の後ろにあるテキストを切り捨てます。既定は 255 です。このカウントには HTML 文字が含まれることに注意してください。長さは最初のパラメーターで指定され、既定は 255 です。2 番目のパラメーターが true である場合、このフィルターはこの長さでテキストを切り捨てます。そうでない場合、最後の単語が破棄されます。テキストが実際に切り捨てられた場合、省略記号 (...) が付加されます。...とは別の省略記号を使用したい場合は、3 番目のパラメーターを使ってそれを指定できます。

パラメーター: `truncate(number_of_character, breakword=False, end='...')`

- HubL
- 出力例

```
{{ "I only want to show the first sentence.Not the second."|truncate(40) }}  
{{ "I only want to show the first sentence.Not the second."|truncate(35, True, '.....') }}
```

truncatehtml

html マークアップを保持しながら、特定の文字列を切り捨てます (つまり、ネストしているすべてのタグが正しく閉じます)。これにより、切り捨て後にタグが開いたままになることはありません。HTML 文字は、合計文字数にカウントされません。このフィルターには長さパラメーターと切り捨て記号パラメーターがあります。3 番目のブール値パラメーター (オプション) は、その長さで単語を切り離すかどうかを指定します。単語の長さを保持する目的で、このパラメーターは既定で false になっています。

パラメーター: `truncatehtml(200, '...', false)`

- . HubL
- . 出力例

```
{% set html_text = "<p>I want to truncate this text without breaking my HTML<p>" %}  
{{ html_text|truncatehtml(28, '..', false) }}
```

union

このフィルターは、2つのセットまたはリストの和集合を返します。フィルターから返されるリストには、いずれかのリストに含まれる一意の要素がすべて格納されます。

パラメーター: `union(list)`

- . HubL
- . 出力例

```
{{ [1, 2, 3]|union([2, 3, 4, 5]) }}
```

unique

このフィルターは、オブジェクトのシーケンスまたはディクショナリから一意のセットを抽出します。(関数から戻される記事のリストなどの)ディクショナリをフィルタリングする場合、どの属性を使ってディクショナリ内の項目の重複を排除するかを指定できます。

パラメーター: `|unique(attr='name_of_attribute')`

- . HubL
- . 出力例

```
{% set my_sequence = ['one', 'one', 'two', 'three'] %}  
{{ my_sequence|unique }}
```

unixtimestamp

このフィルターは日時オブジェクトを Unix タイムスタンプに変換します。

- [HubL](#)
- [出力例](#)

```
{{ local_dt }}  
{{ local_dt|unixtimestamp }}
```

upper

値をすべて大文字に変換します。

- [HubL](#)
- [出力例](#)

```
{% text "text" value='text to make uppercase', export_to_template_context=True %}  
{{ widget_data.text.value|upper }}
```

urlencode

文字列をエスケープして UTF-8 形式で URL エンコードします。

- [HubL](#)
- [出力例](#)

```
{% text "encode" value="Escape & URL encode this string", label="Enter slug",  
export_to_template_context=True %}  
{{ widget_data.encode.value|urlencode }}
```

urlize

プレーンテキスト内の URL を、クリック可能なリンクに変換します。追加的な整数をこのフィルターに渡した場合、URL がその数値に短縮されます。2 番目のパラメーターは、リンクを `rel="no follow"` にするかどうかを指示するブール値です。最後のパラメーターでは、リンクを新しいタブで開くかどうかを指定できます。

パラメーター: `urlize((40,no_follow=true, target='_blank')`

- . HubL
- . 出力例

```
{{ "http://hubspot.com/"|urlize }}
{{ "http://hubspot.com/"|urlize(10,true) }}
{{ "http://hubspot.com/"|urlize("",true) }}
{{ "http://hubspot.com/"|urlize("",false,target='_blank') }}
```

wordcount

文字列内の単語数をカウントします。

文字列に HTML が含まれている場合、正確なカウントを得るには `striptags` フィルターを使用します。

- . HubL
- . 出力例

```
{% set count_words = "Count the number of words in this variable" %}
{{ count_words|wordcount }}
```

wordwrap

指定された文字数でワードラップを行います。HubSpot では既定で空白を除去するので、この機能は<pre>で役立ちます。

パラメーター: `wordwrap(character_count)`

- HubL
- 出力例

```
{% set wrap_text = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam efficitur, ipsum non sagittis euismod, ex risus rhoncus lectus, vel maximus leo enim sit amet dui. Ut laoreet ultricies quam at fermentum." %}
{{ wrap_text|wordwrap(10) }}
```

xmlattr

ディクショナリ内の項目に基づいて HTML/XML 属性文字列を作成します。「値なし」および未定義値を除き、

すべての値が自動的にエスケープされます。最初のパラメーターが `false` でない場合、フィルターから何かが返されると、項目の前に自動的にスペースが追加されます。

パラメーター: `xmlattr(autospace=True)`

- HubL
- 出力例

```
{% set html_attributes = {'class': 'bold', 'id': 'sidebar'} %}
<div {{ html_attributes|xmlattr }}></div>
```

HubL でサポートされている関数

HubL でサポートされている関数

関数は、パラメータを受け入れて値を生成する点でフィルターと類似していますが、すべての関数が初期テンプレート値に適用される必要があるわけではありません。代わりに、関数は HubSpot 環境の他の領域とやり取りします。

append

リストの末尾に 1 つの項目を追加します。

- [HubL](#)
- [出力例](#)

```
{% set numbers_under_5 = [1,2,3] %}  
{% do numbers_under_5.append(4) %}  
{{numbers_under_5}}
```

blog_all_posts_url

blog_all_posts_url 関数は、指定されたブログ内のすべてのブログ記事のリストページを指し示す完全な URL を返します。この関数は、ただ 1 つのパラメータとしてブログ ID または単純なキーワード「default」を受け入れます。

以下の例は、この関数をアンカーの href として使用方法を示しています。

- [HubL](#)
- [出力例](#)

```
<a href="{{ blog_all_posts_url('default') }}">All Marketing blog posts</a>
```

blog_author_url

blog_author_url 関数は、指定されたブログ執筆者のリストページを指し示す完全な URL を返します。

この関数は 2 つのパラメータを受け入れます。最初のパラメータは、その執筆者のリストページがどのブログ内に存在するかを指定します。2 番目のパラメータは、どの執筆者にリンクするかを指定します。このパラメータで **content.blog_post_author.slug** を使用すると、現在の記事の執筆者が指定されます。または、「brian-halligan」などのハイフン付きの小文字の名前を受け入れることもできます。

以下の例は、この関数をアンカーの href として使用方法を示しています。これを **blog_authors** 関数と組み合わせる例については、**blog_authors** 関数の説明を参照してください。

- HubL
- 出力例

```
<a href="{{ blog_author_url('default', 'brian-halligan') }}">Brian Halligan</a>
```

blog_authors

blog_authors は、指定されたブログのブログ執筆者オブジェクトからなるシーケンスを、スラッグ昇順に並べ替えて返します。このシーケンスを変数の中に保存して反復処理すると、カスタム執筆者記事フィルターを作成できます。

各執筆者による公開中の記事の数にアクセスするには、**author.live_posts** を使用できます。

この関数は 2 つのパラメータを受け入れます。最初のパラメータは、どのブログから執筆者を取得するかを指定します。2 番目のパラメータは、取得される執筆者数の上限を設定します。

以下の例の第 1 行は、この関数によって執筆者オブジェクトのシーケンスがどのように返されるかを示しています。例の残りの部分では、シーケンスを変数の中に保存し、執筆者オブジェクトを反復処理して執筆者オブジェクトのリンクをいくつか出力する例を示しています。この例では、ブログに 4 人の執筆者がいることを想定しています。

この関数での執筆者数の上限は 250 です。

- HubL
- 出力例

```
{{ blog_authors('default', 250) }}

{% set my_authors = blog_authors('default', 250) %}
<ul>
{% for author in my_authors %}
<li><a href="{{ blog_author_url(group.id, author.slug) }}">{{ author }}</a></li>
{% endfor %}
</ul>
```

blog_page_link

blog_page_link 関数は、ページ番号付きのブログリスト表示の URL を生成します。この関数は 1 つの数値パラメータを受け入れ、これによって現在のページ、次のページ、前のページ、または特定のページのリンクを生成できます。この関数は通常、ページ番号付きアンカータグの href 属性で使用され、ブローグー覧テンプレートでこの関数を使用する必要があります。

以下に示す例では、アンカーhrefとしてこの関数を使用しています。最初の例では、現在のページを出力します。2番目の例では、第7ページを指定する数値7をパラメータとして受け入れています。3番目の例では、現在のページからの相対ページ番号のリンクを生成する目的で next_page_num 変数を使用しています（前のページを指定するには last_page_num 変数を使用できます）。最後の例では、現在の page_num 変数と+演算子を使用して、現在のページより4つ大きい番号へのリンクを作成しています。

- HubL
- 出力例

```
<a href="{{ blog_page_link(current_page_num) }}">Current page</a>
<a href="{{ blog_page_link(7) }}">Page 7</a>
<a href="{{ blog_page_link(next_page_num) }}">Next</a>
<a href="{{ blog_page_link(current_page_num + 4) }}">Page Plus 4</a>
```

blog_popular_posts

blog_popular_posts 関数は、特定数の人気のある記事をシーケンスとしてレンダリングします。この記事シーケンスを変数の中に保存することができ、for ループでそれを反復処理すると、特に人気が高い記事を示すカスタム記事リストを作成できます。

この関数は最大3つのパラメーターを受け入れます。最初のパラメーターは、どのブログから記事を収集するかを指定します。値は default、または（ブログダッシュボードの URL で取得できる）特定のブログの ID です。2番目のパラメーター（任意指定）は、取得する記事数を指定します。指定しない場合は既定で 10 になります。3番目のパラメーター（任意指定）は、結果をフィルタリングする基準となるトピックを指定します。これを指定すると、このトピックに関連する人気のある記事のみが返されます。

以下の例の第 1 行は、関数がどのようにシーケンスを返すかを示しています。シーケンスは 1 つの変数に保存され、ループ処理されます。[ブログ記事変数](#)では、「content」ではなく個々のループ項目の名前を使用する必要があります。この例では「pop_post.name」が使用されています。この手法は、ブログテンプレートだけでなく通常のページに対しても使用できます。

この関数には 200 件の記事数という制限があります。

- [HubL](#)
- [出力例](#)

```
{% set pop_posts = blog_popular_posts('default', 5, 'marketing-tips') %}
{% for pop_post in pop_posts %}
  <div class="post-title">{{ pop_post.name }}</div>
{% endfor %}
```

blog_post_archive_url

`blog_post_archive_url` 関数は、指定されたブログ上の特定の日付値に関するアーカイブリストページの完全な URL を返します。この関数には 2 つの必須パラメーターと 2 つの任意指定パラメーターがあります。最初のパラメーターはブログ ID または単純なキーワード default です。2 番目は、どの年のアーカイブされた記事を表示するかを指定します。

任意指定パラメーターには、それぞれどの月および日のアーカイブされた記事を表示するかが格納されます。

以下の例は、この関数をアンカーの href として使用方法を示しています。

- [HubL](#)
- [出力例](#)

```
<a href="{{ blog_post_archive_url('default', 2017, 7, 5) }}">Posts from July 5th, 2017</a>
```

blog_post_by_id

この関数は廃止される予定です。 [content_by_id\(\)](#) を使用してください

`blog_post_by_id` 関数は、提供されたブログ記事 ID に基づいて 1 つのブログ記事を返します。この関数が受け入れるパラメーターは数値のブログ ID のみです。

以下の例は、この関数を使ってハイパーリンクリスト項目を生成する方法を示しています。

- [HubL](#)
- [出力例](#)

```
{% set my_post = blog_post_by_id(4715624297) %}  
<ul>  
  <li>  
    <a href="{{ my_post.absolute_url }}">{{my_post.title}}</a>  
  </li>  
</ul>
```

blog_recent_author_posts

`blog_recent_author_posts` 関数は、指定された執筆者のブログ記事オブジェクトからなるシーケンスを、新しい順に並び替えて返します。この記事シーケンスを変数の中に保存することができ、for ループでそれを反復処理すると、特定の執筆者のカスタム記事リストを作成できます。

この関数は 3 つのパラメーターを受け入れます。最初のパラメーターはどのブログから執筆者の記事を収集するかを指定します。値は default、または (ブログダッシュボードの URL で取得できる) 特定のブログの ID です。2 番目のパラメータは、どの執筆者を使用するかを指定します。このパラメータで `content.blog_post_author.slug` を使用すると、現在の記事の執筆者が指定されます。または、「brian-halligan」などのハイフン付きの小文字の名前を受け入れることもできます。3 番目のパラメーター (オプション) は、取得する記事数を指定します。

以下の例の第 1 行は、関数がどのように執筆者の記事シーケンスを返すかを示しています。この例では、執筆者名を正確に指定する代わりに現在の記事執筆者が使用されています。シーケンスは 1 つの変数に保存され、ループ処理されます。[ブログ記事変数](#)では、「content」ではなく個々のループ項目の名前を使用する必要があります。この例では「author_post.name」が使用されています。この手法は、ブログテンプレートだけでなく通常のページに対しても使用できます。

この関数には 200 件の記事数という制限があります。

- [HubL](#)
- [出力例](#)

```
{{ blog_recent_author_posts('default', content.blog_post_author.slug, 5 ) }}
{% set author_posts = blog_recent_author_posts('default',
content.blog_post_author.slug, 5) %}
{% for author_post in author_posts %}
  <div class="post-title">{{ author_post.name }}</div>
{% endfor %}
```

blog_recent_posts

blog_recent_posts 関数は、指定されたブログでのブログ記事オブジェクトからなるシーケンスを、新しい順に並び替えて返します。この記事シーケンスを変数の中に保存することができ、for ループでそれを反復処理すると、特に人気が高い記事を示すカスタム記事リストを作成できます。

この関数は 2 つのパラメーターを受け入れます。最初のパラメーターは、どのブログから人気の高い記事を収集するかを指定します。値は default、または (ブログダッシュボードの URL で取得できる) 特定のブログの ID です。2 番目のパラメーター (オプション) は、取得する記事数を指定します。

以下の例の第 1 行は、関数がどのようにシーケンスを返すかを示しています。シーケンスは 1 つの変数に保存され、ループ処理されます。ブログ記事変数では、「content」ではなく個々のループ項目の名前を使用する必要があります。この例では「rec_post.name」が使用されています。この手法は、ブログテンプレートだけでなく通常のページに対しても使用できます。

この関数には 200 件の記事数という制限があります。

- HubL
- 出力例

```
{{ blog_recent_posts('default', 5) }}
{% set rec_posts = blog_recent_posts('default', 5) %}
{% for rec_post in rec_posts %}
  <div class="post-title">{{ rec_post.name }}</div>
{% endfor %}
```

blog_recent_tag_posts

`blog_recent_tag_posts` 関数は、指定されたタグ（複数可）のブログ記事オブジェクトからなるシーケンスを、新しい順に並び替えて返します。この記事シーケンスを変数の中に保存することができ、for ループでそれを反復処理すると、特定のタグのカスタム記事リストを作成できます。

この関数は 3 つのパラメーターを受け入れます。最初のパラメーターは、どのブログから記事を収集するかを指定します。値は `default`、または（ブログダッシュボードの URL で取得できる）特定のブログの ID です。

2 番目のパラメーターはどのタグを使用するかを指定します。1 つのタグ、またはカンマで区切った最大 10 個のタグの配列を指定できます。このパラメーターでは、`content.tag_list` からの特定のタグを示す `tag.slug` を使用できます。または「marketing-tips」などのハイフン付きの小文字の名前を受け入れることもできます。

3 番目のパラメーターは取得する記事数を指定します。

以下の例の第 1 行は、関数がどのようにタグ別の記事シーケンスを返すかを示しています。第 2 行で使われている関数は、後でループ処理する目的でシーケンスを変数に保存します。[ブログ記事変数](#)では、「content」ではなく個々のループ項目の名前を使用する必要があります。この例では「tag_post.name」が使用されています。この手法は、ブログテンプレートだけでなく通常のページに対しても使用できます。

個々のブログ記事レイアウトの関連記事セクションの基礎として、この関数を使用できます。関連する記事セクションの作成については、[こちらのチュートリアルをご覧ください](#)。

最近の記事数には 100 件の制限があります。

- [HubL](#)
- [出力例](#)

```
{{ blog_recent_tag_posts('default', 'marketing-tips', 5 ) }}
{% set tag_posts = blog_recent_tag_posts('default', ['marketing', 'fun', 'inbound'], 3) %}
{% for tag_post in tag_posts %}
  <div class="post-title">{{ tag_post.name }}</div>
{% endfor %}
```

blog_tag_url

blog_tag_url 関数は、指定されたブログタグのリストページを指し示す完全な URL を返します。

この関数は 2 つのパラメータを受け入れます。最初のパラメータは、そのタグのリストページがどのブログ内に存在するかを指定します。2 番目のパラメータは、どのタグにリンクするかを指定します。このパラメーターでは、**content.topic_list** からの特定のタグを示す **topic.slug** を使用できます。または「marketing-tips」などのハイフン付きの小文字の名前を受け入れることもできます。

以下の例は、この関数をアンカーの href として使用方法を示しています。これを **blog_topics** 関数と組み合わせる例については、**blog_topics** 関数の説明を参照してください。

- [HubL](#)
- [出力例](#)

```
<a href="{{ blog_tag_url('default', 'inbound-marketing') }}">Inbound Marketing</a>
```

blog_topics

(This function has been renamed to '[blog_tags](#)'.)

blog_tags

`blog_tags` 関数は、(関連するブログ記事数に基づき) 指定したブログに関連する特によく使用される 250 個のタグのシーケンスを、ブログ記事カウントで並べ替えて返します。

このシーケンスを変数の中に保存して反復処理すると、カスタムタグ記事フィルターを作成できます。各タグの記事数にアクセスするには、`tag.live_posts` を使用できます。

この関数は 2 つのパラメータを受け入れます。最初のパラメータは、どのブログからタグを取得するかを指定します。2 番目のパラメータは、取得されるタグ数の上限を設定します。

以下の例の第 1 行は、関数がどのようにタグオブジェクトのシーケンスを返すかを示しています。例の残りの部分では、シーケンスを変数の中に保存し、タグオブジェクトを反復処理して執筆者タグのリンクをいくつか出力する例を示しています。この例では、ブログに 4 つのタグがあることを想定しています。

この関数には 250 個のタグという制限があります。

- [HubL](#)
- [出力例](#)

```
{{ blog_tags('default', 250) }}

{% set my_tags = blog_tags('default', 250) %}
<ul>
{% for item in my_tags %}
<li><a href="{{ blog_tag_url(group.id, item.slug) }}">{{ item }}</a></li>
{% endfor %}
</ul>
```

blog_total_post_count

この関数は、指定されたブログでの公開済み記事の総数を返します。パラメーターを指定しない場合、既定のブログ記事をカウントします。あるいは、「default」を指定したり、カウントする別のブログの ID を指定することもできます。ブログ ID は、特定のブログのブログダッシュボードの URL で得られます。

- [HubL](#)
- [出力例](#)

```
{{ blog_total_post_count }}
{{ blog_total_post_count(359485112) }}
```

clear

リストのすべての項目を除去します。pop() とは異なり、返されるデータはありません。

- [HubL](#)
- [出力例](#)

```
{% set words = ["phone","home"] %}
{% do words.clear() %}
{{words}}
```

color_variant

この関数は、HEX 値または色変数を特定の程度だけ明るく/暗くします。最初のパラメーターは HEX 色 (たとえば #FFDDFF)、または HEX 値を格納する変数です。2 番目のパラメーターでは、調整の程度を 0~255 の範囲で指定します。この関数を CSS フ

ファイルで使用すると、色のバリエーションを作成できます。別の使用ケースとして、色モジュールの色パラメーターでこれを使用すると、色バリエーションを自動生成する基本色をユーザーが指定できます。

以下の例では、`base_color` という変数に HEX 色 `#3A539B` が格納されます。その後、色が `-80` 調整され、より濃い青 (`#00034b`) になります。

- HubL
- 出力例

```
{% set base_color = "#3A539B" %}
{{ color_variant(base_color, -80) }}
```

content_by_id

`content_by_id` 関数は、ID を基準にしてランディングページ、ウェブサイトページまたはブログ記事を返します。この関数が受け入れるパラメーターは、数値のコンテンツ ID のみです。

この関数にはページあたり 10 件の呼び出しという制限があります。

以下のコード例は、この関数を使ってハイパーリンクリスト項目を生成する方法を示しています。

- HubL
- 出力例

```
{% set my_content = content_by_id(4715624297) %}
<ul>
  <li>
    <a href="{{ my_content.absolute_url }}">{{my_content.title}}</a>
  </li>
</ul>
```

content_by_ids

特定の ID リストを基準にして、これらの ID にマッチするランディングページ、ウェブサイトページまたはブログ記事からなるディクショナリを返します。

この関数は 1 つのパラメーター、つまり検索するページまたはブログ記事の ID からなるリスト (配列) を受け入れます。最大で 100 個のコンテンツオブジェクトを渡すことができます。この関数にはページあたり 10 件の呼び出しという制限があります。

以下のコード例は、この関数を使ってハイパーリンクリスト項目を生成する方法を示しています。

- [HubL](#)
- [出力例](#)

```
{% set contents = content_by_ids([4715624297, 4712623297, 5215624284]) %}
<ul>
{% for content in contents %}
  <li>
    <a href="{{ content.absolute_url }}">{{content.title}}</a>
  </li>
{% endfor %}
</ul>
```

copy

リストのシャドーコピーを返します。a[:] と同等の機能です。

シャドーコピーでは新しい複合オブジェクトが構築され、元のコピー内のオブジェクトへの参照が (可能な限り) それに挿入されます。

- . HubL
- . 出力例

```
{% set a = ["ham"] %}
{% set b = a.copy() %}
a: {{a}}

b: {{b}}

After Append
{% do a.append("swiss") %}
a: {{a}}

b: {{b}}
```

count

ある変数がリスト内に出現する回数を返します。

- . HubL
- . 出力例

```
{% set attendees = ["Jack","Jon","Jerry","Jessica"] %}
{% set jon_count = attendees.count("Jon") %}
There are {{jon_count}} Jon's in the list.
<p>After append:</p>
{% do attendees.append("Jon") %}
{% set jon_count_after_append = attendees.count("Jon") %}
```

There are now `{{jon_count_after_append}}` Jon's in the list.

crm_associations

HubSpot CRM の 1 つのオブジェクトに関連付けられている CRM オブジェクトのリストを取得します。この関数は、オブジェクトおよび属性 `has_more`、`total`、`offset`、`results` を返します。

`results` は、関数のパラメーターに一致する特定の関連オブジェクトからなる配列を返します。

`has_more` は、このバッチに含まれるよりも多い結果が存在することを示しています (`total > offset`) 。

`total` は、得られる結果の総数です。

`offset` は、この次の結果バッチに使用されるオフセットです。

セキュリティ上の目的から、誰でもアクセスできるページでは Product オブジェクトだけを取得できます。それ以外のオブジェクトタイプは、[パスワードで保護](#)されたページ、または [CMS メンバーシップログイン](#)を必要とするページでホストされる必要があります。`results` 配列内のオブジェクトは、プロパティーと値からなるディクショナリとして返されます。

最初のパラメーター (必須) は、関連付けの元となるオブジェクトを識別する ID です。

2 番目のパラメーター (必須) は、使用する関連付けタイプの ID です。ID は、<https://developers.hubspot.com/docs/methods/crm-associations/crm-associations-overview> で定義されています。

3 番目のパラメーターは、'&'で区切った文字列 (任意指定) です。すべての式は AND で互いに結合されます。クエリで少なくとも 1 つの順序パラメーター (orderBy=) を使用することで、結果を並べ替えることができます。サポートされている演算子は eq (既定)、neq、lt、lte、gt、gte、is_null、not_null です。

4 番目のパラメーター (任意指定の文字列) は、返すプロパティからなるコンマ区切りリストです。既定では、一般的なプロパティからなる小さなセットが返されます。id プロパティは常に返されます。

5 番目のパラメーターはブール値 (任意指定) です。true を渡すと、日付や通貨などの値がポータル設定に従って書式設定されます。文字列を未加工のままにするには、これを省略するか false を渡します。

注：この関数は、ページあたり最大 10 回まで呼び出すことができます。それぞれの crm_objects 呼び出しで、最大 100 個のオブジェクトを返すことができます。既定の制限は 10 個のオブジェクトです。

- HubL
- 出力例

```
{% set associated_contacts = crm_associations(847943847, 2,
'limit=3&years_at_company__gt=2&orderBy=email', 'firstname,email', false) %}
{{ associated_contacts }}
```

crm_object

HubSpot CRM から 1 つのオブジェクトを取得します。

セキュリティ上の目的から、誰でもアクセスできるページでは Product オブジェクトだけを取得できます。それ以外のオブジェクトタイプは、パスワードで保護されたページ、または CMS メンバーシップログインを必要とするページでホストされる必要があります。オブジェクトは、プロパティーと値からなるディクショナリとして返されます。

最初のパラメーター (必須) は、取得する CRM オブジェクトタイプを指定する文字列です (contact、product、company、deal、ticket、または quote)。

2 番目のパラメーター (必須) は、どの CRM オブジェクトを取得するかを指定します。このパラメーターでは数値と文字列のどちらも可能です。ID を基準にして特定の CRM オブジェクトを取得するには、オブジェクトの ID を示す数値を渡します。CRM オブジェクトをクエリすることで特定の CRM オブジェクトを取得するには、クエリ文字列を&で区切って渡します。すべての式は AND で互いに結合されます。サポートされている演算子は eq (既定)、neq、lt、lte、gt、gte、is_null、not_null です。

3 番目のパラメーターは、返すプロパティーからなるコンマ区切りリストです。既定では、一般的なプロパティーからなる小さなセットが返されます。

最後のパラメーターは、日付や通貨などの値をポータル設定に従って書式設定するかどうかを指定します。値を未加工のままにするには、false を渡します。

注:この関数は、1 ページあたり最大で 10 回だけ呼び出すことができます。

- HubL
- 出力例

```
<!-- by query -->
{% set contact = crm_object("contact", "email=contact@company.com",
"firstname,lastname", false) %}

<!-- by id -->
{% set contact = crm_object("contact", 123) %}
{{ contact.firstname }}
{{ contact.lastname }}
```

crm_objects

HubSpot CRM からオブジェクトのリストを取得します。

この関数は、オブジェクトおよび属性 `has_more`、`total`、`offset`、`results` を返します。

`results` 関数のパラメーターに一致する特定のオブジェクトからなる配列を返します。

`has_more` は、このバッチに含まれるよりも多い結果が存在することを示しています
(`total > offset`)。

`total` は、得られる結果の総数です。

`offset` は、この次の結果バッチに使用されるオフセットです。

クエリで少なくとも1つの順序パラメーターを使用することで、結果を並べ替えることができます。たとえば `crm_objects("contact", "firstname=Bob&order=lastname&order=createdate")` の場合、Bob という名 (ファーストネーム) のコンタクトがまず姓の順序、次に作成日順で並べ替えられます。逆順で並べ替えるには、たとえば `order=-createdate` のようにプロパティ名の前に「-」を付加します。

セキュリティ上の目的から、誰でもアクセスできるページでは Product オブジェクトだけを取得できます。それ以外のオブジェクトタイプは、パスワードで保護されたページ、または CMS メンバーシップログインを必要とするページでホストされる必要があります。`results` 配列内のオブジェクトは、プロパティと値からなるディクショナリとして返されます。

最初のパラメーター (必須) は、取得する CRM オブジェクトタイプを指定する文字列です (`contact`、`product`、`company`、`deal`、`ticket`、または `quote`)。

2 番目のパラメーター (必須) は、どの CRM オブジェクトを取得するかを指定します。クエリ文字列を&で区切って渡してください。すべての式は AND で互いに結合されます。サポートされている演算子は `eq` (既定)、`neq`、`lt`、`lte`、`gt`、`gte`、`is_null`、`not_null` です。

3 番目のパラメーターは、`results` 属性値で各オブジェクトに関して返すプロパティからなるコンマ区切りリストです。既定では、一般的なプロパティからなる小さなセットが返されます。

最後のパラメーターは、`results` 属性値で各オブジェクトに関して返される日付や通貨などの値を、ポータル設定に従って書式設定するかどうかを指定します。値を未加工のままにするには、`false` を渡します。

注：この関数は、ページあたり最大 10 回まで呼び出すことができます。それぞれの `crm_objects` 呼び出しで、最大 100 個のオブジェクトを返すことができます。既定の制限は 10 個のオブジェクトです。

- HubL
- 出力例

```
{% set objects = crm_objects("contact", "firstname__not_null=&limit=3",  
"firstname,lastname") %}  
{{ objects }}
```

cta

CTA モジュールには、コードバリエーションを含むパラメーターが多数あるので、`cta` 関数を使用するとテンプレート、ページ、Eメールの中に特定の CTA を簡単に生成できます。リッチテキストエディターを使って CTA を追加するとき、エディターはこの関数を使用します。

この関数は、2つのパラメータを受け入れます。最初のパラメーターは CTA の GUID です。この固有 ID は、特定の CTA の詳細画面の URL に含まれています。2番目のパラメーターは、次の列挙値を受け入れます。

- **justifyleft** - `text-align: left` を CTA ラッパーに追加します。
- **justifycenter** - `text-align: center` を CTA ラッパーに追加します。
- **justifyright** - `text-align: right` を CTA ラッパーに追加します。
- **justifyfull** - CTA ラッパーに行端揃えを追加しません。

- HubL
- 出力例

```
{{ cta('ccd39b7c-ae18-4c4e-98ee-547069bfbc5b') }}  
{{ cta('ccd39b7c-ae18-4c4e-98ee-547069bfbc5b', 'justifycenter') }}
```

datetimeformat

datetimeformat 関数の動作は [datetimeformat フィルター](#) とまったく同じですが、フィルター構文ではなく関数構文を使用します。同じパラメーターを使用して、日付を書式設定します。書式設定パラメーターの全リストについては、[こちら](#)を参照してください。

- [HubL](#)
- [出力例](#)

```
{{ datetimeformat(content.publish_date_local_time, '%B %e, %Y') }}
```

extend

反復可能オブジェクトのすべての項目を付加することで、リストを拡張します。言い換えると、あるリストのすべての項目を別のリストに挿入します。

- [HubL](#)
- [出力例](#)

```
{% set vehicles = ["boat","car","bicycle"] %}  
{% set more_vehicles = ["airplane","motorcycle"] %}  
{% do vehicles.extend(more_vehicles) %}  
{{vehicles}}
```

file_by_id

この関数は、ID を基準にファイルのメタデータを返します。唯一のパラメーターとして、検索するファイルを示す数値 ID を受け入れます。

この関数には、ページあたり 10 件の呼び出しという制限があります。

- . HubL
- . 出力例

```
{% set file = file_by_id(123) %}  
{{ file.friendlyUrl }}
```

geo_distance

注:この関数は、[HubDB](#) でフィルタークエリとして使用されます。

この関数には 4 つのパラメーターがあり、地球上の 2 地点間の楕円 2D 距離を計算します。

最初のパラメーターは、HubDB テーブルからの第 1 地点のロケーション列です (例: row.location) 。

2 番目のパラメーターは、第 2 地点の緯度を示す数値です。

3 番目のパラメーターは、第 2 地点の経度を示す数値です。

4 番目のパラメーターは、返される値の単位を示す文字列値です。オプションとして FT (フィート) 、 MI (マイル) 、 M (メートル) 、 KM (キロメートル) があります。

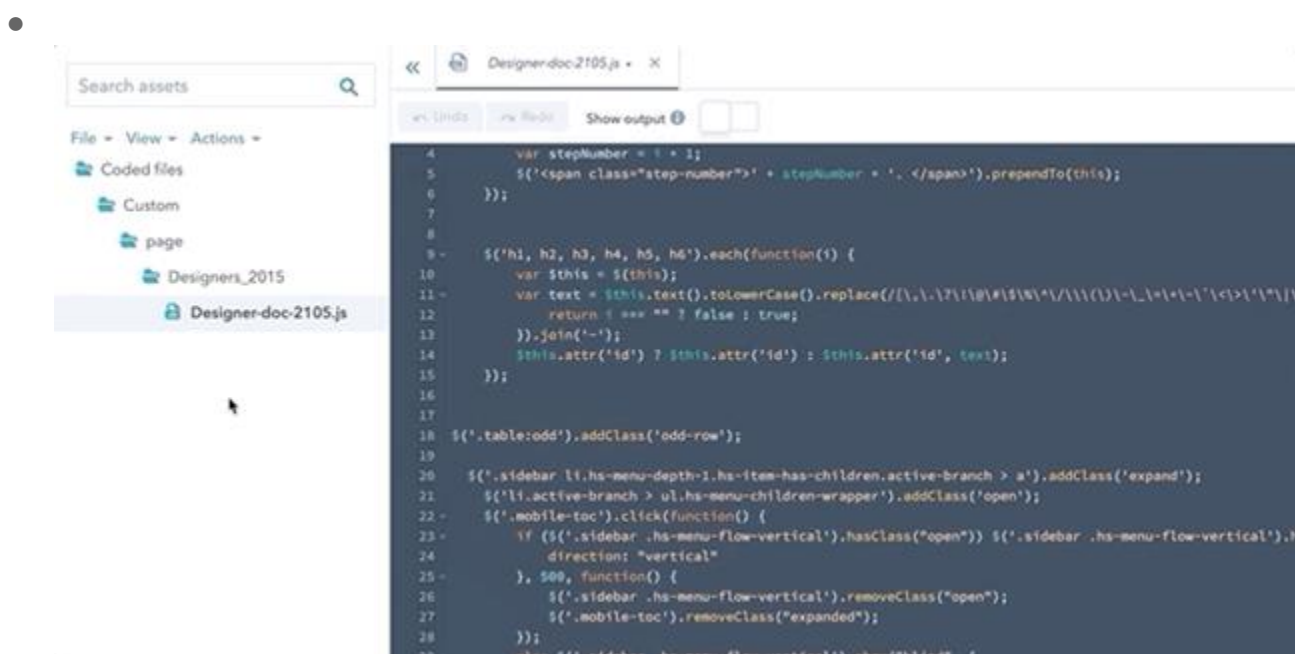
- . HubL
- . 出力例

```
{% for row in hubdb_table_rows(1234567, "geo_distance(loc,1.233,-  
5.678,mi)__gt=500" ) %}  
  <!-- Values matching your filter query will be displayed here -->  
  {{row.name}} <br/>  
{% endfor %}
```

get_asset_url

この関数は、指定したテンプレートまたはコードファイルの公開 URL を返します。この関数のパラメーターは、デザインマネージャーにおけるパスです。コードファイル URL は、公開されるたびに更新されます。したがって、この関数を使用すると常にファイルの最新バージョンを使用していることを確認できます。

アプリ内でこの関数を自動的に生成できます。そうするには、ファイル名を右クリックして[公開 URL をコピー]を選択するか、[アクション]>[公開 URL をコピー]を選択します。以下の例では、デザインマネージャーで作成された JavaScript ファイルの URL を取得します。これを<script>の src として含めることができます。



- HubL
- 出力例

```
{{ get_asset_url("/custom/styles/style.css") }}
```

get_public_template_url

この関数は廃止される予定です。 [get_asset_url\(\)](#) を使用してください。

- . HubL
- . 出力例

```
{{ get_public_template_url("custom/page/Designers_2015/designer-doc-2105.js") }}
```

get_public_template_url_by_id

この関数の動作は `get_public_template_url` と同様で、指定したテンプレートまたはコードファイルの公開 URL を返します。唯一の違いは、この関数のパラメーターがデザインマネージャーのパスではなく、テンプレート ID であることです (この ID はテンプレートまたはコードファイルの URL の中で判別できます)。

- . HubL
- . 出力例

```
{{ get_public_template_url_by_id('2778457004') }}
```

hubdb_table

HubDB の包括的な概要については、CMS Developer ツールセクションの [HubDB ページ](#) をご覧ください。

テーブルの名前、列、最終更新日などの情報を取得するには、`hubdb_table()`関数を使用できます。この関数はただ 1 つのパラメーターとして、取得元の HubDB テーブルの ID を受け入れます。

適切な属性を呼び出すことで、次のような情報を取得できます。

- *id* - テーブルの ID
- *name* - テーブルの名前
- *columns* - 列情報のリスト
- *created_at* - テーブルが初めて作成された時点のタイムスタンプ
- *published_at* - このテーブルが公開された時点のタイムスタンプ。
- *updated_at* - このテーブルの最終更新時点のタイムスタンプ
- *row_count* - テーブルの行数

- . HubL
- . 出力例

```
{% set table_info = hubdb_table(1548215) %}  
{{ table_info.row_count }}
```

hubdb_table_column

HubDB の包括的な概要については、CMS Developer ツールセクションの [HubDB ページ](#) をご覧ください。

hubdb_table_column()関数を使用すると、テーブルの列に関する情報 (ラベル、タイプ、オプションなど) を取得できます。この関数は 2 つのパラメーターを受け入れます。

最初のパラメーターは、取得元の HubDB テーブルの ID です。2 番目のパラメーターは、列の ID または列の名前 (文字列) です。

適切な属性を呼び出すことで、列に関する次のような情報を取得できます。

- *id* - 列の ID
- *name* - 列の名前
- *label* - 列に使用されるラベル
- *type* - 列のタイプ
- *options* - select 列タイプの場合、これは optionId から option 情報へのマップです
- *foreignIds* - foreignId 列タイプの場合、これは foreignId のリストです (id プロパティと name プロパティを含む)

上記の属性に加えて、`getOptionByName("<option name>")` というメソッドを呼び出すこともできます。select タイプの列の場合、これによりオプションの名前を基準に option 情報が取得されます。

- HubL
- 出力例

```
{% set column_info = hubdb_table_column(123456, 6) %}  
{{ column_info.label }}
```

hubdb_table_row

HubDB の包括的な概要については、CMS Developer ツールセクションの [HubDB ページ](#) をご覧ください。

hubdb_table_row 関数を使用すると、HubDB テーブルから 1 つの行を取得できます。この関数は 2 つのパラメーターを受け入れます。

最初のパラメーターは、取得元のテーブルの ID です。2 番目のパラメーターは、取得する行の ID です。この行に関して、該当する属性を呼び出すことで各テーブルセルから次のような情報を取得できます。

(組み込み列とカスタム列の名前では大文字と小文字が区別されません。HS_ID と hs_id は同じです)

- *hs_id* - この行のグローバルに一意な ID
- *hs_created_at* - この行が作成された時点を示すタイムスタンプ
- *hs_path* - 動的ページで使用される場合、この文字列はページの URL パスの最後のセグメントです
- *hs_name* - 動的ページで使用される場合、これはページのタイトルです
- *<column name>* または *["<column name>"]* - 列の名前 (**name**) を基準に、この行の列値を取得します

- [HubL](#)
- [出力例](#)

```
{% set row = hubdb_table_row(1548264, 6726439331) %}
{{ row.role }}
```

hubdb_table_rows

HubDB の包括的な概要については、CMS Developer ツールセクションの [HubDB ページ](#) をご覧ください。

hubdb_table_rows 関数を使用すると、反復処理用に HubDB テーブルの行のリストを取得できます。この関数は 2 つのパラメーターを受け入れます。

最初のパラメーターは、行の取得元となる HubDB テーブルの ID を指定します。2 番目のパラメーターは、互いに AND で結合されたフィルターからなる文字列を受け入れます。これを使用して、どの行を取得するかをフィルタリングします。任意指定のフィルターのリスト、およびこれらのフィルターを使用する構文については、[こちらをご覧ください](#)。

hubdb_table_rows() の 1 回の呼び出しとして定義される CMS ページあたりのテーブルスキャン数には、10 回までの制限があります。

- . HubL
- . 出力例

```
{% for row in hubdb_table_rows(1546258,
"years_at_company__gt=3&orderBy=count") %}
  the value for row {{ row.hs_id }} is {{ row.name }}
{% endfor %}
```

include_css

この関数は廃止される予定です。[require_css\(\)](#)を使用してください

この関数は、CSS ファイルのデザインマネージャーパスを使用して、そのファイルを参照するリンクタグを生成します。パラメーターは、スタイルシートのパスです。

- . HubL
- . 出力例

```
{{ include_css('custom/page/Designers_2015/designers-doc-2015.css') }}
```

include_default_custom_css

この関数は、[プライマリ CSS ファイル](#) (default_custom_style.min.css) を参照するリンクタグを生成します。このファイルはグローバル CSS ファイルとして用意され、すべてのテンプレートにこれを追加できます。この関数によってレンダリングするには、ブール値パラメーターの値 True が必要です。

- [HubL](#)
- [出力例](#)

```
{{ include_default_custom_css(True) }}
```

include_JavaScript

この関数は廃止される予定です。[require_js\(\)](#) を使用してください。

この関数は、Javacript ファイルのデザインマネージャーパスを使用して、そのファイルを参照するスクリプトタグを生成します。パラメーターは、JavaScript ファイルのパスです。

- [HubL](#)
- [出力例](#)

```
{{ include_JavaScript('custom/page/Designers_2015/designer-doc-2105.js') }}
```

index

0 ベースの配列の中で最初にマッチする項目の場所を返します。

この関数は 3 つのパラメーターを受け入れ、最初のパラメーターは必須です。最初のパラメーターは、配列内で検索される項目です。2 番目と 3 番目のパラメーター（それぞれ開始と終了）を使用すると、配列のスライス内でその項目を検出できます。

- . HubL
- . 出力例

```
{% set shapes = ["triangle","square","trapezoid","triangle"] %}
```

```
triangle index: {{shapes.index("triangle")}} <br>
```

```
trapezoid index: {{shapes.index("trapezoid")}}
```

```
<hr/>
```

```
開始と終了を調整 <br>
```

```
triangle index: {{shapes.index("triangle",1,5)}}
```

insert

リスト内の指定されたインデックス場所に要素を挿入します。

この関数は次の 2 つのパラメータを受け入れます。

- Index: 要素を挿入する位置
- Element: 挿入される項目

- . HubL
- . 出力例

```
{% set even_numbers = [2,4,8,10] %}
```

```
{% do even_numbers.insert(2,6) %}
```

```
{{even_numbers}}
```

locale_name

言語コードを判読可能な文字列表記として返し、オプションでターゲット言語に翻訳します。

- . HubL
- . 出力例

```
{{ locale_name('es') }}  
{{ locale_name('es', 'en') }}
```

menu

詳細メニューの、ネストしたリンク構造を返します。メニューノードのプロパティについては、[こちらで確認できます](#)。2番目の引数 (`root_type`) は、どんなメニューを返すかを次のように決定します。

- `site_root` - 最上位メニューノードである子を持つノードを返します
- `top_parent` - そのページが属する最上位メニューノードを返します
- `parent` - その子の中にページが入っているメニューノードを返します
- `page_id` - ページ ID を使ってページノードを返します
- `page_name` - ページ名を使ってページノードを返します
- `breadcrumb` - 現在のページのページノードを返します

必須パラメーター：

`menu_id` - 詳細メニューの ID

任意指定のパラメーター：

`root_type` - 可能な値は `site_root`、`top_parent`、`parent`、`page_id`、`page_name`、`breadcrumb` のいずれかです

root_key -page_id または page_name を使用する場合にメニューノードを特定するための ID/文字列キーです

menu(menu_id, root_type=site_root, root_key)

- . HubL
- . 出力例

```
{% set node = menu(987) %}
{% for child in node.children %}
    {{ child.label }}
{% endfor %}
```

module_asset_url

[リンク済みファイル] > [その他のファイル]によってカスタムモジュールに付加されるアセットの URL を取得します。

- . HubL
- . 出力例

```
{{ module_asset_url("smile.jpg") }}
```

oembed

Eメールでのみ機能します。特定のリクエストに関する OEmbed データディクショナリを返します。

- . HubL
- . 出力例

```
{{ oembed({ url: "https://www.youtube.com/watch?v=KqpFNtbEOh8" }) }}
```

page_by_id

この関数は廃止される予定です。 [content_by_id\(\)](#) を使用してください。

`page_by_id` 関数は、指定された ID に基づいて 1 つのページを返します。この関数が受け入れるパラメーターは数値のページ ID のみです。

以下の例は、この関数を使ってハイパーリンクリスト項目を生成する方法を示しています。

- [HubL](#)
- [出力例](#)

```
{% set my_page = page_by_id(4715624297) %}
<ul>
  <li>
    <a href="{{ my_page.absolute_url }}">{{ my_page.title }}</a>
  </li>
</ul>
```

personalization_token

この関数は 2 つのパラメーターを受け入れます。最初のパラメーターは返されるコンタクト、またはコンタクトに関連するプロパティ値です。2 番目のパラメーター (任意指定) は、式で返す値が存在しない場合に使用する既定値です。

- [HubL](#)

- 出力例

```
Hi {{ personalization_token("contact.firstname", "there") }}!
```

pop

インデックス位置にある項目をリストから除去します。さらに、除去された項目を返します (出力される場合)。

- HubL
- 出力例

```
{% set even_numbers = [2,3,4,6,8,9,10] %}  
{% do even_numbers.pop(1) %}  
{{even_numbers.pop(4)}}  
{{even_numbers}}
```

postal_location

`postal_location` 関数は、特定の郵便番号と国コードに対応するロケーション緯度/経度ペアを返します (ただし米国、カナダ、英国のみ)。最初の必須パラメーターは郵便番号 (文字列) です。2 番目のパラメーター (任意指定) は、郵便番号の国コードです。これを指定しない場合、郵便番号から国が推定されます。

- HubL
- 出力例

```
{{ postal_location("02139") }}  
{% set location = postal_location("02139", "US") %}
```

```
{{ location.latitude }}  
{{ location.longitude }}
```

range

整数からなる算術数列が入ったリストを返します。range で 1 つのパラメーターを指定すると、0 からその値未満までのリストが返されます。range で 2 つのパラメーターを指定すると、最初の値から始まり、1 つずつ増えて、2 番目の値未満までが返されます。3 番目のパラメーターは段階的な増分を指定します。すべての値に負の数を指定できます。無効な範囲を指定すると、空のリストが返されます。range で生成できる値の数は最大 1000 個です。

range を [for loop](#) の中で使用すると、実行する反復処理の回数を指定できます。

- [HubL](#)
- [出力例](#)

```
{{ range(11) }}  
{{ range(5, 11) }}  
{{ range(0, 11, 2) }}  
  
{% for number in range(11) %}  
{{ number }}  
{% endfor %}
```

require_css

この関数は、レンダリングされる CSS ファイルを `<head />` に追加します。すべての CSS `<link />` タグは一緒のグループになり、JavaScript タグの前にレンダリングされま

す。HubL は空の行に置換され、その後で<link />が{{ standard_header_includes }}に追加されます。ここでは絶対 URL が必要とされます。get_public_template_url() 関数を使用することで、既知の相対 URL 付きの COS コンテンツを要求できます。

< style />要素を介して、レンダリングされるインラインスタイルを head に含めるには、代わりに{% require_css %} <style>/*...*/</style>{% end_require_css %} タグを使用してください。

- HubL
- 出力例

```
{{ standard_header_includes }}
<!-- more html -->
{{ require_css("http://example.com/path/to/file.css") }}
{{ require_css(get_public_template_url("/relative/path/to/file.css")) }}
```

require_js

この関数は、レンダリングされるスクリプトを<head/>またはフッターに追加します。HubL が空の行に置換され、既定では{{ standard_footer_includes }}に、または呼び出しに head パラメーターが存在する場合は{{ standard_header_includes }}にそれぞれ含められます。

<script/>要素を介して、レンダリングされるインラインスクリプトを<head/>に含めるには、代わりに{% require_js %}<script> // ... </script>{% end_require_js %} タグを使用してください。

- HubL
- 出力例

```
{{ standard_header_includes }}
```

```
<!-- more html -->
{{ require_js("http://example.com/path/to/footer-file.js", "footer") }}
{{ require_js("http://example.com/path/to/head-file.js", "head") }}

{{ standard_footer_includes }}
```

resize_image_url

ファイルマネージャーに保存されたイメージの URL を書き換えて、リクエスト時にイメージのサイズを変更する URL にします。この関数は 1 つの必須パラメーターと 5 つの任意指定パラメーターを受け入れます。少なくとも 1 つの任意指定パラメーターを渡す必要があります。

必須

- URL - HubSpot でホストされているイメージの URL (文字列)

任意指定

- width - 新しいイメージ幅 (ピクセル単位の数値)
- height - 新しいイメージの高さ (ピクセル単位の数値)
- length - 新しいイメージの長さ (ピクセル単位の数値)
- upscale - 元のイメージの寸法が拡大される場合でもサイズ変更後のイメージを使用するかどうかを示すブール値 (イメージがぼやけて表示される可能性があります)
- upsize - 元のバイト数より増える場合でもサイズ変更後のイメージを返すかどうかを示すブール値

- [HubL](#)
- [出力例](#)

```
{{ resize_image_url("http://your.hubspot.site/hubfs/img.jpg", 0, 0, 300) }}
```

reverse

リスト内の項目の順序を逆にします。受け入れるパラメーターはありません。

- HubL
- 出力例

```
{% set numbers = [1,2,3,4] %}  
{% do numbers.reverse() %}  
{{numbers}}
```

set_response_code

応答コードを、指定されたコードに設定します。現在、指定できるコードは 404 だけです。これを使用すると、ページから 404 エラーが返されます。

- HubL
- 出力例

```
{{ set_response_code(404) }}
```

super

extends タグを使用して親および子テンプレートを作成済みの場合、super は親テンプレートの内容を子テンプレートの中に出力します。たとえば下記のコードでは、「sidebar」という名前の HubL ブロックを使って基本 HTML テンプレートが作成され、parent.html として保存されています。その親ファイルを拡張する 2 番目のテンプレートファイルが作成されます。通常、<h3>は親 HTML の sidebar ブロック内に出力

されます。しかし `super` を使用すると、親テンプレート `sidebar` ブロックの内容が子テンプレートの内容と結合されます。

- [HubL](#)
- [出力例](#)

```
{% extends "custom/page/web_page_basic/parent.html" %}

{% block sidebar %}
  <h3>Table Of Contents</h3>
  {{ super() }}
{% endblock %}
```

today

今日の始まり (午前 0:00) を返します。オプションで、タイムゾーンを既定の UTC から変更するパラメーターを追加できます。

- [HubL](#)
- [出力例](#)

```
{{ today() }}
{{ today("America/New_York") }}
{{ unixtimestamp(today("America/New_York").plusDays(1)) }}
```

to_local_time

HubSpot レポート設定に基づいて、UNIX タイムスタンプをローカル時刻に変換します。その後、`datetimeformat` フィルターを適用して日付を書式設定できます。

- HubL
- 出力例

```
{{ to_local_time(eastern_dt) }}
```

topic_cluster_by_content_id

(渡されたコンテンツ ID で判別される) コンテンツ部分に関連しているトピッククラスターを表す HubL ディクショナリを返します。これには、関連付けられているピラーページ、コアトピック、サブトピックについてのメタデータが含まれます。これを使用すると、関連ピラーページが存在する場合、それをコンテンツ部分に「自動リンク」させることができます。

取得可能なメタデータがある場所は、`attachableContent` (現在のコンテンツのメタデータ)、`topic` (現在のコンテンツの関連トピックメタデータ)、`coreTopic` (関連クラスターのコアトピックメタデータ)、および `pillarPage` (関連ピラーページのメタデータ) です。

可能なプロパティと属性をすべて表示するには、`{{ topicCluster|pprint }}` を使用してください。

- HubL
- 出力例

```
{{ topic_cluster_by_content_id(content.id) }}
{%- if content.id -%}
  {%- set topicCluster = topic_cluster_by_content_id(content.id) -%}
```

```
{%- if topicCluster.pillarPage.url.value and topicCluster.pillarPage.publishState ==
'PUBLISHED' -%}
  <div>Topic: <a
href="{{ topicCluster.pillarPage.url.value }}">{{ topicCluster.coreTopic.phrase }}</a></div
>
  {%- endif -%}
{%- endif -%}
```

truncate

truncate 関数の動作は [truncate フィルター](#) とまったく同じですが、フィルター構文ではなく関数構文を使用します。最初のパラメーターは、文字列を指定します。2番目のパラメーターは、どこで切り捨てるかを長さで指定します。最後のパラメーターは、切り捨てが発生した場合に追加する文字を指定します。

- HubL
- 出力例

```
{{ truncate('string to truncate at a certain length', 19, end='...') }}
```

type

この関数は1つの引数を受け入れ、オブジェクトのタイプを返します。返される値は、"bool"、"datetime"、"dict"、"float"、"int"、"list"、"long"、"null"、"str"、"tuple"のいずれかです。

- HubL
- 出力例

```
{{ type("Blog") }}
```



```
{% set my_type = type("Blog") %}  
<p>{{my_type}}</p>
```

unixtimestamp

この関数は Unix タイムスタンプを返します。既定では現在のタイムスタンプが返されますが、オプションで日時オブジェクトを渡すと、それが Unix タイムスタンプに変換されます。

- . HubL
- . 出力例

```
{{ unixtimestamp() }}  
{{ unixtimestamp(d) }}
```

update

別のディクショナリオブジェクトから、または反復可能なキー/値ペアからの要素を使ってディクショナリを更新します。

- . HubL
- . 出力例

```
{% set dict_var = {'authorName': 'Douglas Judy', 'authorTitle': 'Mastermind' } %}  
{% set test = dict_var.update({'authorFriend': 'Jake'}) %}  
{% set test = dict_var.update({'authorLocation': 'unknown'}) %}  
{{ dict_var }}
```

CMS デベロッパーツール

CMS デベロッパーサンドボックスアカウント

オープンソースプロジェクト

ボイラープレート CSS

HubSpot CMS - ボイラープレート CSS

HubSpot の CMS でウェブサイトのスタイルを設定するときは、さまざまな方法でサイトの外観をコーディングできます。ユーザーの利用開始をサポートし、一般的に使用されている CSS セレクターの例を示すため、HubSpot の技術サービスチームは以下の「ボイラープレート」スタイルシートを作成しました。これらのセレクターを利用すると、作業を簡単に始めることができ、CMS テンプレートの一般的な要素をより明確に理解できます。CSS を記述する場所や情報の整理に関するベストプラクティスについて詳しくは、[デザインマネージャーユーザーガイド](#)を参照してください。

以下のコードスニペットには、空の CSS セレクター、参考になるコード修正、コードを理解するのに役立つコメントが含まれています。これら 2 つのコードブロックを、作成して COS コンテンツに添付する [Primary.css ファイル](#) (すべてのページに適用する場合) またはその他の [カスタム CSS ファイル](#) に貼り付けます。別個のファイルで追加することも、同じスタイルシートに貼り付けることもできます。

この記事のボイラープレート CSS は、HubSpot CMS を初めて使用するユーザーの出発点になります。これは本格的なフレームワークではないため、個別のニーズを満たすためには、このリソースを独自のベーススタイルで拡張する必要があります。

HubSpot CMS のベストプラクティスと高度な機能を示す次世代のボイラープレートを現在開発中です。このボイラープレートは現在初期ベータ版であり、[GitHub 経由で参照し、開発に貢献](#)することができます。

CompanyStyle.css

CompanyStyle.css は、一般的に使用される HubSpot CMS 要素のスタイル設定に使用できる空の CSS セレクターを満載したスタイルシートです。コードはコメントによって複数のセクションに分割され、さまざまなセレクターと HubL トークンの説明が挿入されています。このコードを CSS ファイルに貼り付けると、セレクターは空であるためページの外観は変更されませんが、独自の CSS プロパティを追加するためのテンプレートが提供されます。

```
/* @import url('http://example.com/example_style.css'); */

/**
 * CSS @imports must be at the top of the file.
 * Add them above this section.
 */
```

```
/*
=====
=====
HS COS Boilerplate v1.5.0
=====
===== */

{% include "custom/page/Company_Theme/CompanyModules.css" %}

/* These includes are optional, but helpful. */
{% include "hubspot/styles/patches/recommended.css" %}

/*
=====
=====

Index:
1.BASE STYLES
- Variables
- Macros
```

- Base
- Typography

2.COS STRUCTURE

- Structure
- Header
- Content
- Footer

3.MAIN NAVIGATION

- Custom Menu Primary
- Mobile Menu

4.TEMPLATE MODULE CLASSES

- Common Template Classes
- Site Pages
- Landing Pages
- System Pages

5.COS COMPONENTS

- Blog
- Forms
- Buttons

6.MISCELLANEOUS

7.MOBILE MEDIA QUERIES

```
=====
===== */
```

```
/*
```

```
=====
```

```
=====
```

1.BASE STYLES

```
=====
```

```
===== */
```

```
/*
```

```
=====
```

```
=====
```

Variables

```
=====
```

```
===== */
```

```
/**
```

* Find the most consistently used font styles, colors, and container widths

```

* for your site and assign them to the jinja variables below for easy use
* and reference.
*
* Apply the variable as shown below.Ignore the jinja comment tags.
*
* {# property: {{ variableName }}; #}
*/

/* Colors */

{% set baseColor    = "#333" %} /* {# Dark Gray | Used on 'body' in 'Base' | Ex.
"color: {{ baseColor }};" #} */

{% set linkColor    = "gray" %} /* {# Gray | Used on global anchors in 'Base' | Ex.
"color: {{ linkColor }};" #} */

{% set linkHover    = "#000" %} /* {# Black | Used on global anchors in 'Base' | Ex.
"color: {{ linkHover }};" #} */

{% set accentColor1 = "#666" %} /* {# Medium Gray | Ex. "color:
{{ accentColor1 }};" #} */

{% set accentColor2 = "#ccc" %}/* {# Light Gray | Ex. "color: {{ accentColor2 }};" #} */

{% set accentColor3 = "#ddd" %} /* {# Lightest Gray | Ex. "color:
{{ accentColor3 }};" #} */

/* Typography */

{% set baseFontFamily    = "Arial, Sans-Serif" %} /* {# Used on 'body' in
'Base' | Ex. "font-family: {{ baseFontFamily }};" #} */

{% set baseFontSize      = "13px" %} /* {# Used on 'body' in 'Base' |
Ex. "font-size: {{ baseFontSize }};" #} */

```

```

{% set baseLineHeight = "1.5" %} /* {# Used on 'body' in 'Base' | Ex.
"line-height: {{ baseLineHeight }};" #} */

{% set webFont1 = "Helvetica, Arial, Sans-serif" %} /* {# Ex. "font-family:
{{ webFont1 }};" #} */

/* Containers */

{% set maxWidth = "960px" %} /* Used on '.page-center' in 'Base' and in the 'Portrait
tablet to landscape and desktop' media query */

/*
=====
=====

Macros

=====
===== */

/**
 * Create jinja macros for CSS3 properties that need browser prefixes.
 *
 * Apply the style as shown below on it's own CSS property line.
 * Ignore the jinja comment tags.
 *
 * {# {{ variableName("values") }} #}

```



```
*/

{% macro transition(value) -%}

-webkit-transition: {{ value }};

-moz-transition: {{ value }};

-o-transition: {{ value }};

-ms-transition: {{ value }};

transition: {{ value }};

{%- endmacro %}

/**

* Example of how to apply the above macro (ignore the jinja comment tags):

*

* a {

*   color: red;

*   font-size:12px;

*   {{ transition("all .2s linear") }}

* }

*/
```

```

/*
=====
=====

Base

=====
===== */

/**
 * Box Sizing
 *
 * Applies a natural box layout model to all elements so that width and padding
 * are essentially combined, making responsive styles easier to manage.
 */

*, *:before, *:after {
    -moz-box-sizing: border-box; -webkit-box-sizing: border-box; box-sizing: border-box;
}

html, body {
    min-height: 100%;
    margin: 0;
    padding: 0;
}

```

```
/* Website background color and default font styles */
body {
  background: #fff;
  color: {{ baseColor }};
  font-family: {{ baseFontFamily }};
  font-size: {{ baseFontSize }};
  line-height: {{ baseLineHeight }};
}

/**
 * Page Center
 *
 * Apply .page-center to full-width modules.
 */

.container-fluid .row-fluid .page-center {
  float: none;
  max-width: {{ maxWidth }};
  margin: 0 auto;
}

a { color: {{ linkColor }}; }
a:hover, a:focus { color: {{ linkHover }}; }
```

```
hr {
  color: #ccc;
  background-color: #ccc;
  height: 1px;
  border: none;
}

/* Highlighted Text */
::-moz-selection {
  color: #fff;
  background: {{ baseColor }};
  text-shadow: none;
}
::selection {
  color: #fff;
  background: {{ baseColor }};
  text-shadow: none;
}

/* Corrects width issues in table elements created via the wysiwyg editor */
table tr td img {
  max-width: initial;
}
```

```
/*  
=====
```

```
=====  
Typography  
=====
```

```
===== */  
  
/* Basic text */  
  
p {}  
small {}  
strong {}  
em {}  
cite {}  
code {}  
pre {}  
  
sup, sub {  
    position: relative;  
    font-size: 75%;  
    line-height: 0;
```

```
vertical-align: baseline;
}

sup { top: -0.5em; }
sub { bottom: -0.25em; }

/* Headings */
h1, h2, h3, h4, h5, h6 {}
h1 a, h2 a, h3 a, h4 a, h5 a, h6 a {}

h1 {}
h2 {}
h3 {}
h4 {}
h5 {}
h6 {}

/* Lists */
ul, ol {}
ul ul, ul ol, ol ol, ol ul {}
li {}

ul.unstyled, ol.unstyled {
```

```
list-style: none;
margin: 0;
padding: 0;
}

/* Quotes */
blockquote {}
blockquote p {}
blockquote small {}
blockquote small:before {}

q:before, q:after, blockquote:before, blockquote:after {}
```

```
/*
```

```
=====
=====
```

```
2.COS STRUCTURE
```

```
=====
===== */

/*
=====

Structure

=====
===== */

/* The outer wrappers of your website */
.header-container-wrapper,
.body-container-wrapper,
.footer-container-wrapper {}

/* The inner wrappers of your website */
.header-container,
.body-container,
.footer-container {}
```



```
/*  
=====
```

```
=====  
Header  
  
=====
```

```
===== */  
  
.header-container-wrapper {}  
.header-container {}  
  
/* Logo styling */  
.header-container .hs_cos_wrapper_type_logo {}  
.header-container .hs_cos_wrapper_type_logo a {}  
.header-container .hs_cos_wrapper_type_logo a img {}
```

```
/*
=====
Content
===== */

.body-container-wrapper {}
.body-container {}

/* Unique Pages
===== */

/* Homepage */

/* Contact Us */

/* About Us */

/* Modular & Reusable Components
```

```
=====
===== */

/* Callouts */

/* Social Media Custom Icons */

/* Additional Menus aside from the Main Nav */

/*
=====
=====
Footer
=====
===== */

.footer-container-wrapper {}

.footer-container {}
```

```
/*
```

```
=====
```

3.MAIN NAVIGATION

```
=====
```

```
===== */
```

```
/*
```

```
=====
```

Custom Menu Primary

```
=====
```

```
===== */
```

```

{% set ulColor      = "" %} /* Set ul background color */
{% set liColor      = "" %} /* Set li background Color */
{% set aColor       = "" %} /* Set link Color */
{% set aColorHover  = "" %} /* Set link Hover Color */

/* Parent List */

.custom-menu-primary .hs-menu-wrapper > ul{
  background:{{ ulColor }};
}

.custom-menu-primary .hs-menu-wrapper > ul > li{
  background:{{ liColor }};
}

.custom-menu-primary .hs-menu-wrapper > ul > li > a{
  color:{{ aColor }};
}

.custom-menu-primary .hs-menu-wrapper > ul > li > a:hover{
  color:{{ aColorHover }};
}

/* Child List */

.custom-menu-primary .hs-menu-wrapper > ul ul{}
.custom-menu-primary .hs-menu-wrapper > ul ul li{
  background:{{ liColor }};
}

```

```

}

.custom-menu-primary .hs-menu-wrapper > ul ul li a{
  color:{{ aColor }};
}

.custom-menu-primary .hs-menu-wrapper > ul ul li a:hover{
  color:{{ aColorHover }};
}

/* Override max width on menu links */
.custom-menu-primary .hs-menu-wrapper > ul li a,
.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-item-has-children ul.hs-menu-
children-wrapper li a {
  overflow: visible !important;
  max-width: none !important;
  width: auto !important;
}

/* Fix menu disappearing on desktop after toggling mobile menu */
@media screen and (min-width:768px) {
  .custom-menu-primary .hs-menu-wrapper {
    display:block !important;
  }
}

```

```

/*
=====
=====

Mobile Menu - Hubspot Standard Toggle Menu

=====
===== */

/**
 * Special Note
 *
 * When the menu is open, a class of .mobile-open is applied to the body. You can
 * use this for custom styling on any element when the menu is in the open position.
 */

.mobile-trigger, .child-trigger{
    display: none; /* Hide button on Desktop */
}

@media (max-width: 767px){

```

```
/* Variables
```

```
=====
===== */
```

```
{% set menuColorMobile = "#000000" %} /* Set Mobile Menu Background Color */
```

```
{% set aColorMobile = "#ffffff" %} /* Set Link Color */
```

```
{% set aColorHoverMobile = "#ffffff" %} /* Set Link Hover Color */
```

```
/*
```

```
 * Menu Reset
```

```
 *
```

```
 * Remove styling from desktop version of custom-menu-primary. Place any
```

```
 * additional CSS you want removed from the mobile menu in this reset
```

```
 */
```

```
.custom-menu-primary,
```

```
.custom-menu-primary .hs-menu-wrapper > ul,
```

```
.custom-menu-primary .hs-menu-wrapper > ul li,
```

```
.custom-menu-primary .hs-menu-wrapper > ul li a{
```

```
 display: block;
```

```
 float: none;
```

```
 position: static;
```

```
 top: auto;
```



```
right: auto;
left: auto;
bottom: auto;
padding: 0px;
margin: 0px;
background-image: none;
background-color: transparent;
border: 0px;
-webkit-border-radius: 0px;
-moz-border-radius: 0px;
border-radius: 0px;
-webkit-box-shadow: none;
-moz-box-shadow: none;
box-shadow: none;
max-width: none;
width: 100%;
height: auto;
line-height: 1;
font-weight: normal;
text-decoration: none;
text-indent: 0px;
text-align: left;
color: {{aColorMobile}};
```

```
}
```

```
/* Toggle Button
```

```
===== */
```

```
.mobile-trigger{
```

```
display: inline-block important; /* Show button on mobile */
```

```
cursor: pointer; /* Mouse pointer type on hover */
```

```
position: absolute; /****** */
```

```
top: 0px; /* Position Button at right of screen */
```

```
right: 10px; /****** */
```

```
width: auto; /* Button width */
```

```
height: auto; /* Button height */
```

```
padding: 7px 10px 8px 10px;
```

```
background: #ffffff; /* Background color */
```

```
border: 1px solid {{ menuColorMobile }};
```

```
font-size: 16px;
```

```
font-weight: normal;
```

```
text-align: left;
```

```
text-transform: uppercase;
```

```
-webkit-border-radius: 4px;
```

```

-moz-border-radius: 4px;
border-radius: 4px;
color: {{ menuColorMobile }};
}

.mobile-trigger:hover{
text-decoration: none; /* Removes link text underline on button */
color:{{ aColorHoverMobile }};
background-color: {{ menuColorMobile }};
border-color: transparent;
}

/* Change button when menu is open */
.mobile-open .mobile-trigger{
color:{{ aColorHoverMobile }};
background-color: {{ menuColorMobile }};
border-color: transparent;
}

/* Toggle Button Icon
=====
===== */

```

```
.mobile-trigger i{
  display: inline;
  position: relative;
  top: -4px;
}

.mobile-trigger i:before, .mobile-trigger i:after{
  position: absolute;
  content: "";
}

.mobile-trigger i, .mobile-trigger i:before, .mobile-trigger i:after{
  width: 22px; /* Icon line width */
  height: 2px; /* Icon line height */
  -webkit-border-radius: 1px;
  -moz-border-radius: 1px;
  border-radius: 1px;
  background-color: {{ menuColorMobile }}; /* Icon color */
  display: inline-block;
}

.mobile-trigger i:before{
  top: -6px; /* Position top line */
}

.mobile-trigger i:after{
  top: 6px; /* Position bottom line */
}
```

```

}

.mobile-trigger:hover i, .mobile-trigger:hover i:before, .mobile-trigger:hover i:after,
.mobile-open .mobile-trigger i, .mobile-open .mobile-trigger i:before, .mobile-
open .mobile-trigger i:after{

    background-color: {{ aColorHoverMobile }}; /* Icon color */
}

```

```

/* Child Toggle Button

```

```

=====
===== */

```

```

.child-trigger{

    display: block !important; /* Hide button on Desktop */
    cursor: pointer; /* Mouse pointer type on hover */
    position: absolute;

    top: 0px;

    right: 0px;

    width: 55px !important; /* Button width */
    min-width: 55px !important;

    height:45px !important; /* Button height */
    padding: 0 !important;

    border-left: 1px dotted rgba(255, 255, 255, .20);
}

```

```

}

.child-trigger:hover{
    text-decoration: none;
}

.child-trigger i{
    position: relative;
    top: 50%; /* Centers icon inside button */
    margin: 0 auto !important;
}

.child-trigger i:after{
    position: absolute;
    content: "";
}

.child-trigger i, .child-trigger i:after{
    width: 10px; /* Icon line width */
    height: 1px; /* Icon line height */
    background-color:{{ aColorMobile }}; /* Icon color */
    display: block;
}

.child-trigger i:after{
    -webkit-transform: rotate(-90deg);
    -ms-transform: rotate(-90deg);
    transform: rotate(-90deg);
}

```

```
}  
.child-trigger.child-open i:after{  
  -webkit-transform: rotate(-180deg);  
  -ms-transform: rotate(-180deg);  
  transform: rotate(-180deg);  
}
```

```
.child-trigger.child-open i{  
  -webkit-transform: rotate(180deg);  
  -ms-transform: rotate(180deg);  
  transform: rotate(180deg);  
}
```

```
/* Menu Styles on Mobile Devices
```

```
===== */
```

```
.custom-menu-primary.js-enabled{  
  position: relative;  
  padding-top: 44px; /* Makes room for button */  
  margin: 10px 0 10px 0;  
}
```

```

/* Hide menu on mobile */
.custom-menu-primary.js-enabled .hs-menu-wrapper,
.custom-menu-primary.js-enabled .hs-menu-children-wrapper{
    display: none;
}

/* Make child lists appear below parent items */
.custom-menu-primary ul.hs-menu-children-wrapper{
    visibility: visible !important;
    opacity: 1 !important;
    position: static !important;
    display: none;
}

/* Mobile Menu Styles */
.custom-menu-primary.js-enabled .hs-menu-wrapper{
    width: 100%;
    position: relative; /*******/
    top: 0;           /* Positions the menu to drop from the very top of the screen */
    left: 0;         /*******/
    padding: 0;
}
.custom-menu-primary .hs-menu-wrapper{

```



```

background-color:{{ menuColorMobile }}; /* Menu background color set off global
menuColorMobile variable */

width: 100%; /* Full screen width */
}

/* Level 1 Menu List Styles */

.custom-menu-primary .hs-menu-wrapper > ul > li{
position: relative;
}

.custom-menu-primary .hs-menu-wrapper > ul > li a{
font-size: 22px; /* Font size of top level list items */
line-height: 45px;
overflow: visible;
}

/* Level 1 and Higher Menu List Styles */

.custom-menu-primary .hs-menu-wrapper > ul li{
border-top: 1px dotted rgba(255, 255, 255, .35); /* Adds transparent dark highlights to
top of top level list items */
}

.custom-menu-primary .hs-menu-wrapper > ul li a{
padding: 0 10px;
color:{{ aColorMobile }}; /* link color set by global mobile-aColor variable */
}

```

```

.custom-menu-primary .hs-menu-wrapper > ul li a:hover{
    color:{{ aColorHoverMobile }}; /* link hover color set by global mobile-aColorHover
variable */
}

/* Level 2 and Higher Menu List Styles */
.custom-menu-primary .hs-menu-wrapper > ul ul li{
    background-color: rgba(255, 255, 255, .08);
}
.custom-menu-primary .hs-menu-wrapper > ul ul li a{
    text-indent: 10px; /* Indent Child lists */
    font-size: 16px; /* Font size of child lists */
}

/* Level 3 and Higher Menu List Styles */
.custom-menu-primary .hs-menu-wrapper > ul ul ul li a{
    text-indent: 30px; /* Indent Child lists */
}
.custom-menu-primary .hs-menu-wrapper > ul ul ul ul li a{
    text-indent: 50px; /* Indent Child lists */
}
}

```

```
/*
```

```
=====
```

```
=====
```

4.TEMPLATE MODULE SELECTORS

```
=====
```

```
===== */
```

```
/* Common Template Selectors
```

```
=====
```

```
===== */
```

```
body.one-column {}
```

```
body.two-column {}
```

```
body.three-column {}
```

```
.content-wrapper {}  
.main-content {}  
.sidebar{}  
.sidebar.right{}  
.sidebar.left{}  
  
/* Common One Column Selectors */  
body.site-page.one-column,  
body.landing-page.one-column,  
body.error-template.one-column,  
body.password-template.one-column,  
body.subscription-template.one-column {}  
  
body.site-page.one-column .main-content,  
body.landing-page.one-column .main-content,  
body.error-template .main-content,  
body.password-template .main-content,  
body.subscription-template .main-content {}  
  
/* Common Two Column Selectors */  
body.site-page.two-column,  
body.landing-page.two-column,  
body.blog {}
```

```

body.site-page.two-column .main-content,
body.landing-page.two-column .main-content,
body.blog .blog-content {}

/* Common Three Column Selectors */
body.site-page.three-column,
body.landing-page.three-column {}

body.site-page.three-column .main-content,
body.landing-page.three-column .main-content {}

/* Site Pages
=====
===== */

/* One Column Template */
body.site-page.one-column {}
body.site-page.one-column .content-wrapper {}
body.site-page.one-column .main-content {}

/* Two Column Template */

```

```

body.site-page.two-column {}

body.site-page.two-column .content-wrapper {}

body.site-page.two-column .main-content {}

body.site-page.two-column .sidebar.right {}

/* Three Column Template */

body.site-page.three-column {}

body.site-page.three-column .content-wrapper {}

body.site-page.three-column .main-content {}

/* Landing Pages

=====
===== */

/* One Column Template */

body.landing-page.one-column {}

body.landing-page.one-column .content-wrapper {}

body.landing-page.one-column .main-content {}

/* Two Column Template */

body.landing-page.two-column {}

body.landing-page.two-column .content-wrapper {}

```

```

body.landing-page.two-column .main-content {}
body.landing-page.two-column .sidebar.right {}

/* Three Column Template */
body.landing-page.three-column {}
body.landing-page.three-column .content-wrapper {}
body.landing-page.three-column .main-content {}

/* System Pages
=====
===== */

/* Error Templates 404, 500 */
body.error-template.one-column {}
body.error-template.one-column .content-wrapper {}
body.error-template.one-column .main-content {}

/* Password template */
body.password-template.one-column {}
body.password-template.one-column .content-wrapper {}
body.password-template.one-column .main-content {}

```

```
/* Email Subscription templates */  
body.subscription-template.one-column {}  
body.subscription-template.one-column .content-wrapper {}  
body.subscription-template.one-column .main-content {}
```

```
/*  
=====
```

5.COS COMPONENTS

```
===== */
```



```
/*
=====
=====

Blog

=====
===== */

/* Blog Post and Listing Body Class */
body.blog {}

/* Blog Post */
body.hs-blog-post {}

/* Blog Listing */
body.hs-blog-listing {}

/* Blog Columns Wrapper */
.blog .content-wrapper {}

/* Blog Content Column

=====
===== */
```

```
.blog .blog-content {}

/* Use this to style the main wrapper for the blog listing pages */
.blog-listing-wrapper {}

/* Use this to style the main wrapper for the blog post pages */
.blog-post-wrapper {}

/* Blog Title */
.hs-blog-header h1 {}
.hs-blog-header h1 a {}

/* Blog Post Title */
.post-header h2 {}
.post-header h2 a {}

/* Blog Author and Avatar */
#hubspot-author_data {}
#hubspot-author_data a.author-link {}
#hubspot-author_data .hs-author-avatar {}
#hubspot-author_data .hs-author-avatar img {}
```

```
/* Use this to set up different styling for your blog posts */  
  
.blog-listing-wrapper .post-listing .post-item {}  
  
.blog-listing-wrapper .post-listing .post-item .post-body {}  
  
.blog-post-wrapper .post-body {}  
  
/* Post Featured Image on Listings Page */  
  
.hs-featured-image-wrapper {}  
  
.hs-featured-image-wrapper .hs-featured-image-link {}  
  
.hs-featured-image-wrapper .hs-featured-image-link img.hs-featured-image{  
  
border-width: 0;  
  
max-height: 170px;  
  
max-width: 210px;  
  
height: auto;  
  
width: auto;  
  
display: inline-block !important;  
  
float: right;  
  
margin: 20px 0 20px 20px;  
  
}  
  
/* Read More Link */  
  
a.more-link {}  
  
/* Blog Topics List */
```

```
#hubspot-topic_data {}

#hubspot-topic_data a.topic-link {}

/**
 * Blog Social Sharing icons
 *
 * Overrides defaults to better align icons and prevent overlapping
 */

.hs-blog-social-share {
padding: 10px 0 0 0;
border-top: 1px solid #cccccc;
border-bottom: 1px solid #cccccc;
height: auto;
}

/**
 * Blog Comments
 *
 * Custom styles are listed below, as the COS does not provide default styles.
 */

#comments-listing {
border-bottom: 1px solid #ccc;
```

```

margin-bottom: 30px;
margin-top: 30px;
}
.comment {
margin: 10px 0;
border-top: 1px solid #ccc;
padding: 15px 0;
}
.comment-date { font-size: 13px; text-align: right; }
.comment-from { font-weight: bold; }
.comment-body { padding: 10px 0; }

/* Pagination */
.blog-pagination {}
.blog-pagination a {}
.blog-pagination a.previous-posts-link {}
.blog-pagination a.all-posts-link {}
.blog-pagination a.next-posts-link {}

/**
 * Blog Author Page
 *
 * ALWAYS be sure to set styles to these selectors, as the COS does not provide
 default styles

```

```

*/

.hs-author-profile {}

.hs-author-profile h2.hs-author-name {}

.hs-author-profile .hs-author-bio {}

.hs-author-profile .hs-author-avatar {}

.hs-author-profile .hs-author-avatar img {}

h3.hs-author-listing-header {}

/**
 * Blog All Posts Page
 *
 * ALWAYS be sure to set styles to these selectors, as the COS does not provide
default styles
 */

body.hs-content-path-blog-all {}

div.post-listing-simple {}

div.post-listing-simple .post-item {}

h2.post-listing-simple {}

h2.post-listing-simple a {}

/* Blog Sidebar

```

```
===== */
=====

.blog .blog-sidebar {}

/* Blog Modules */

.blog .blog-sidebar .widget-span {}

/*
=====
=====

Forms

=====
===== */

/* All Global Forms

=====
===== */

/* Form Field */
```

```
body .hs-form-field {}

/* Descriptions */
body .hs-field-desc {}

/* Labels */
body label {}
body .hs-form-field > label {}

/* One Line Inputs */
body input[type="text"],
body input[type="password"],
body input[type="datetime"],
body input[type="datetime-local"],
body input[type="date"],
body input[type="month"],
body input[type="time"],
body input[type="week"],
body input[type="number"],
body input[type="email"],
body input[type="url"],
body input[type="search"],
body input[type="tel"],
```



```
body input[type="color"],
body input[type="file"],
body textarea,
body select {
    width: 100%;
    font-family: {{ baseFontFamily }};
    font-size: {{ baseFontSize }};
}

body input[type="text"]:focus,
body input[type="password"]:focus,
body input[type="datetime"]:focus,
body input[type="datetime-local"]:focus,
body input[type="date"]:focus,
body input[type="month"]:focus,
body input[type="time"]:focus,
body input[type="week"]:focus,
body input[type="number"]:focus,
body input[type="email"]:focus,
body input[type="url"]:focus,
body input[type="search"]:focus,
body input[type="tel"]:focus,
body input[type="color"]:focus,
```

```
body input[type="file"]:focus,  
body textarea:focus,  
body select:focus {}  
  
/* Separate Styles for Multiple Line Inputs */  
body textarea {}  
body textarea:focus {}  
  
/* Separate Styles for Drop Downs */  
body select {}  
body select:focus {}  
  
/* Multiple Selection Inputs */  
body form.hs-form .hs-form-field ul.inputs-list {  
    list-style: none;  
    padding: 0;  
}  
body form.hs-form .hs-form-field ul.inputs-list li input {}  
body input[type="radio"] {}  
body input[type="checkbox"] {}  
  
/* Required */  
body input:focus:required:invalid,
```

```
body textarea:focus:required:invalid,
body select:focus:required:invalid {}

body input:focus:required:invalid:focus,
body textarea:focus:required:invalid:focus,
body select:focus:required:invalid:focus {}

/* Placeholder Text */
::-webkit-input-placeholder { /* Webkit Browsers */
  /* color: #555;
  opacity:1; */
}
:-moz-placeholder { /* Firefox 18- */
  /* color: #555;
  opacity:1; */
}
::-moz-placeholder { /* Firefox 19+ */
  /* color: #555;
  opacity:1; */
}
:-ms-input-placeholder { /* IE10 */
  /* color: #555;
  opacity:1; */
}
```

```

/* Form Module

=====
===== */

/* Form Field */
.hs_cos_wrapper_type_form .hs-form-field {}

/* Descriptions */
.hs_cos_wrapper_type_form .hs-field-desc {}

/* Labels */
.hs_cos_wrapper_type_form label {}
.hs_cos_wrapper_type_form .hs-form-field > label {}

/* One Line Inputs */
.hs_cos_wrapper_type_form input[type="text"],
.hs_cos_wrapper_type_form input[type="password"],
.hs_cos_wrapper_type_form input[type="datetime"],
.hs_cos_wrapper_type_form input[type="datetime-local"],
.hs_cos_wrapper_type_form input[type="date"],
.hs_cos_wrapper_type_form input[type="month"],

```

```
.hs_cos_wrapper_type_form input[type="time"],
.hs_cos_wrapper_type_form input[type="week"],
.hs_cos_wrapper_type_form input[type="number"],
.hs_cos_wrapper_type_form input[type="email"],
.hs_cos_wrapper_type_form input[type="url"],
.hs_cos_wrapper_type_form input[type="search"],
.hs_cos_wrapper_type_form input[type="tel"],
.hs_cos_wrapper_type_form input[type="color"],
.hs_cos_wrapper_type_form input[type="file"],
.hs_cos_wrapper_type_form textarea,
.hs_cos_wrapper_type_form select {}

.hs_cos_wrapper_type_form input[type="text"]:focus,
.hs_cos_wrapper_type_form input[type="password"]:focus,
.hs_cos_wrapper_type_form input[type="datetime"]:focus,
.hs_cos_wrapper_type_form input[type="datetime-local"]:focus,
.hs_cos_wrapper_type_form input[type="date"]:focus,
.hs_cos_wrapper_type_form input[type="month"]:focus,
.hs_cos_wrapper_type_form input[type="time"]:focus,
.hs_cos_wrapper_type_form input[type="week"]:focus,
.hs_cos_wrapper_type_form input[type="number"]:focus,
.hs_cos_wrapper_type_form input[type="email"]:focus,
.hs_cos_wrapper_type_form input[type="url"]:focus,
```

```
.hs_cos_wrapper_type_form input[type="search"]:focus,  
.hs_cos_wrapper_type_form input[type="tel"]:focus,  
.hs_cos_wrapper_type_form input[type="color"]:focus,  
.hs_cos_wrapper_type_form input[type="file"]:focus,  
.hs_cos_wrapper_type_form textarea:focus,  
.hs_cos_wrapper_type_form select:focus {}  
  
/* Separate Styles for Multiple Line Inputs */  
.hs_cos_wrapper_type_form textarea {}  
.hs_cos_wrapper_type_form textarea:focus {}  
  
/* Separate Styles for Drop Downs */  
.hs_cos_wrapper_type_form select {}  
.hs_cos_wrapper_type_form select:focus {}  
  
/* Multiple Selection Inputs */  
.hs_cos_wrapper_type_form form.hs-form .hs-form-field ul.inputs-list li input {}  
.hs_cos_wrapper_type_form input[type="radio"] {}  
.hs_cos_wrapper_type_form input[type="checkbox"] {}  
  
/* Blog Comment Form
```

```
===== */
/* Label */
#comment-form .hs-form-field > label {}
#comment-form label {}

/* Inputs */
#comment-form input[type="text"] {}
#comment-form input[type="text"]:focus {}

#comment-form textarea {}
#comment-form textarea:focus {}

#comment-form input[type="checkbox"] {}

/* Error */
body .hs-common-error-message {}

/* Blog Subscription Form
===== */
```

```

/* Label */

#hs_form_target_blog_subscription .hs-form-field > label {}

#hs_form_target_blog_subscription label {}

/* Inputs */

#hs_form_target_blog_subscription input[type="email"] {}

#hs_form_target_blog_subscription input[type="email"]:focus {}

#hs_form_target_blog_subscription input[type="radio"],
#hs_form_target_blog_subscription input[type="checkbox"] {

width: auto;

margin-right: 10px;

}

/* Google Search Form

=====
===== */

/* Label */

.widget-type-google_search label {}

/* Input */

```



```

.widget-type-google_search input[type="text"] {}
.widget-type-google_search input[type="focus"] {}

/* Multi Column Form
=====
===== */

body .hs-form fieldset.form-columns-1 {}
body .hs-form fieldset.form-columns-1 .hs-form-field {}

body .hs-form fieldset.form-columns-2 {}
body .hs-form fieldset.form-columns-2 .hs-form-field {}

body .hs-form fieldset.form-columns-3 {}
body .hs-form fieldset.form-columns-3 .hs-form-field {}

/* Multiple Selection Inputs */
body form.hs-form .hs-form-field ul.inputs-list li input {}
body input[type="radio"] {}
body input[type="checkbox"] {}
body form.hs-form fieldset { max-width: none; }
body form.hs-form fieldset .hs-form-field { padding: 0 10px 0 0; }

```

```

body form.hs-form fieldset[class*="form-columns"] .input { margin-right: 0; }

body form.hs-form fieldset[class*="form-columns"] .hs-input { width: 100%; }

body form.hs-form fieldset[class*="form-columns"] .hs-input[type="checkbox"] { width:
auto; }

body form.hs-form fieldset.form-columns-3 .hs-form-field { width: 33.33%; }

@media (max-width: 400px) {

  body form.hs-form fieldset .hs-form-field { padding: 0; }

  body form.hs-form fieldset.form-columns-3 .hs-form-field,

  body form.hs-form .form-columns-2 .hs-form-field .hs-input,

  body form.hs-form .form-columns-3 .hs-form-field .hs-input { width: 100%; }

  body form.hs-form .form-columns-3 .hs-form-field .hs-input[type="checkbox"] { width:
auto; }

}

/*
=====
=====

Buttons

=====
===== */

```

```
/* All Global Buttons (Excluding CTAs)
```

```
=====
===== */
```

```
body .hs-button.primary,
body input[type="submit"],
body input[type="button"] {
  cursor: pointer;
  -webkit-appearance: none;
  font-family: {{ baseFontFamily }};
  font-size: {{ baseFontSize }};
  text-decoration: none !important;
}
```

```
body .hs-button.primary:hover,
body input[type="submit"]:hover,
body input[type="button"]:hover {}
```

```
body .hs-button.primary:focus,
body input[type="submit"]:focus,
body input[type="button"]:focus {}
```

```
/* Form Module Button
```

```
=====
===== */
```

```
.hs_cos_wrapper_type_form .hs-button.primary.large {}
```

```
.hs_cos_wrapper_type_form .hs-button.primary.large:hover {}
```

```
.hs_cos_wrapper_type_form .hs-button.primary.large:focus {}
```

```
/* Blog Comment Button
```

```
=====
===== */
```

```
#add-comment-button.hs-button.primary {}
```

```
#add-comment-button.hs-button.primary:hover {}
```

```
#add-comment-button.hs-button.primary:focus {}
```

```
/* Blog Subscription Button
```

```
=====
===== */
```

```
#hs_form_target_blog_subscription .hs-button.primary.large {}
```

```
#hs_form_target_blog_subscription .hs-button.primary.large:hover {}
```

```
#hs_form_target_blog_subscription .hs-button.primary.large:focus {}
```

```
/* Google Search Button
```

```
===== */
```

```
.widget-type-google_search .hs-button.primary {}
```

```
.widget-type-google_search .hs-button.primary:hover {}
```

```
.widget-type-google_search .hs-button.primary:focus {}
```

```
/* CTA Buttons
```

```
===== */
```

```
/**
```

```
* Override global CTA styles by wrapping the CTA module in a module group  
* that has an ID, and use that ID to prefix your CTA selector. Example shown below.
```

```
*
```

```
* All overriding properties require !important tags.
```

```
*/
```

```
#module-group-ID .cta_button {}
```

```
#module-group-ID .cta_button:hover {}
```

```
#module-group-ID .cta_button:focus {}
```

```
/*
```

```
=====
```

```
=====
```

6.MISCELLANOUS

```
=====
```

```
===== */
```

```
/**
```

* Styles that do not fit into available sections above can go here.

* For example, one would place 3rd party plugin overriding styles here.

*

* This section is NOT for 'Custom' styles.All custom styles must go in the

* proper sections above.

```
*  
* For instance, custom header and footer styles must go in the HEADER and  
* FOOTER sections. Custom styles for anything in-between the header and footer  
* (such as, styles that are unique to the HOMEPAGE only), must go in the  
* CONTENT section.
```

```
*/
```

```
/*
```

```
=====
```

7.MOBILE MEDIA QUERIES

```
=====
```

```
===== */
```

```
/* Screen sizes smaller than site maxWidth (.page-center)
```

```
===== */
```

```
@media (max-width: {{maxWidth}}) {}
```

```
/* Landscape tablet and smaller devices
```

```
===== */
```

```
@media (max-width: 767px) {}
```

```
/* Landscape phones and smaller devices
```

```
===== */
```

```
@media (max-width: 479px) {}
```

CompanyModules.css

CompanyModules.css は、より難しいモジュールのスタイル設定に対応するファイルです。このファイルには、画像/メディアをレスポンシブにするためのコードや、HubSpot のレスポンシブなスライダーモジュールのスタイルを設定するためのコード

などが含まれています。CompanyStyle.css とは異なり、この CSS ファイルにはページの外観に影響を与える完全な CSS 定義が含まれています。このコードは、同じファイルに貼り付けたり、HubL の include トークンを使用してインクルードしたり、別のファイルに貼り付けてテンプレートに添付したりできます。

CompanyModules.css を CompanyStyle.css と同じファイルに貼り付ける場合は、CompanyStyle.css の先頭の を CompanyModules.css のコードに置き換える必要があります。

```
/* Images */  
  
img {  
    max-width: 100%;  
    height: auto;  
    border: 0;  
    -ms-interpolation-mode: bicubic;  
    vertical-align: bottom; /* Suppress the space beneath the baseline */  
}  
  
img.alignRight{  
    margin: 0 0 5px 15px;  
}  
  
img.alignLeft{  
    margin: 0 15px 5px 0;  
}
```

```

/* Reduces min-height on bootstrap structure */
.row-fluid [class*="span"] { min-height: 1px; }

/* Blog Social Sharing */
.hs-blog-social-share .hs-blog-social-share-list {
  margin: 0;
  list-style: none !important;
}

.hs-blog-social-share .hs-blog-social-share-list .hs-blog-social-share-item {
  height: 30px;
  list-style: none !important;
}

@media (max-width: 479px){
.hs-blog-social-share .hs-blog-social-share-list .hs-blog-social-share-item:nth-child(2) {
  float: none !important;; /* Stack social icons into two rows on phones */
}
}

.hs-blog-social-share .hs-blog-social-share-list .hs-blog-social-share-item .fb-like >
span{
  vertical-align: top !important;; /* Aligns Facebook icon with other social media icons */
}

/* Captcha */

```

```
#recaptcha_response_field {  
    width: auto !important;  
    display: inline-block !important;  
}
```

```
/* Videos */
```

```
video {  
    max-width: 100%;  
    height: auto;  
}
```

```
/* Date Picker */
```

```
#calroot {  
    width: 202px !important;  
    line-height: normal;  
}
```

```
#calroot,
```

```
#calroot *,
```

```
#calroot *:before,
```

```
#calroot *:after {
```

```
    -webkit-box-sizing: content-box;
```

```
    -moz-box-sizing: content-box;
```

```
    box-sizing: content-box;
```

```

}

#calroot select {
  min-height: 0 !important;
  padding: 1px 2px !important;
  font-family: "Lucida Grande", "Lucida Sans Unicode", "Bitstream Vera Sans",
  "Trebuchet MS", Verdana, sans-serif !important;
  font-size: 10px !important;
  line-height: 18px !important;
  font-weight: normal !important;
}

#caldays {
  margin-bottom: 4px;
}

/* Embed Container (iFrame, Object, Embed) */
.hs-responsive-embed {
  position: relative;
  height: 0;
  overflow: hidden;
  padding-top: 0;
  padding-left: 0;
  padding-right: 0;
}

```

```
}  
  
.hs-responsive-embed iframe,  
.hs-responsive-embed object,  
.hs-responsive-embed embed {  
    position: absolute;  
  
    top: 0;  
  
    left: 0;  
  
    width: 100%;  
  
    height: 100%;  
  
    border: 0;  
}  
  
.hs-responsive-embed,  
.hs-responsive-embed.hs-responsive-embed-youtube,  
.hs-responsive-embed.hs-responsive-embed-wistia,  
.hs-responsive-embed.hs-responsive-embed-vimeo {  
    padding-bottom: 56.25%;  
}  
  
.hs-responsive-embed.hs-responsive-embed-instagram {  
    padding-bottom: 116.01%;  
}  
  
.hs-responsive-embed.hs-responsive-embed-pinterest {  
    height: auto;  
  
    overflow: visible;
```

```
padding: 0;
}
.hs-responsive-embed.hs-responsive-embed-pinterest iframe {
    position: static;
    width: auto;
    height: auto;
}
iframe[src^="http://www.slideshare.net/slideshow/embed_code/"] {
    width: 100%;
    max-width: 100%;
}
@media (max-width: 568px) {
    iframe {
        max-width: 100%;
    }
}

/* Forms */
textarea,
input[type="text"],
input[type="password"],
input[type="datetime"],
input[type="datetime-local"],
```

```
input[type="date"],
input[type="month"],
input[type="time"],
input[type="week"],
input[type="number"],
input[type="email"],
input[type="url"],
input[type="search"],
input[type="tel"],
input[type="color"]
select {
    display: inline-block;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

/* System Template Forms */
#email-prefs-form .email-edit {
    width: 100% !important;
    max-width: 507px !important;
}

#hs-pwd-widget-password {
```

```
height: auto !important;
}

/* Menus */
.hs-menu-wrapper ul {
padding: 0;
}

/* Horizontal Menu
=====
===== */

.hs-menu-wrapper.hs-menu-flow-horizontal ul {
list-style: none;
margin: 0;
}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul {
display: inline-block;
}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul:before {
content: " ";
display: table;
}
```



```
.hs-menu-wrapper.hs-menu-flow-horizontal > ul:after {  
  content: " ";  
  display: table;  
  clear: both;  
}  
  
.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-menu-depth-1 {  
  float: left;  
}  
  
.hs-menu-wrapper.hs-menu-flow-horizontal > ul li a {  
  display: inline-block;  
}  
  
.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-item-has-children {  
  position: relative;  
}  
  
.hs-menu-wrapper.hs-menu-flow-horizontal.flyouts > ul li.hs-item-has-children ul.hs-  
menu-children-wrapper {  
  visibility: hidden;  
  opacity: 0;  
  -webkit-transition: opacity 0.4s;  
  position: absolute;  
  z-index: 10;  
  left: 0;  
}
```

```

.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-item-has-children ul.hs-menu-
children-wrapper li a {

  display: block;

  white-space: nowrap;
}

.hs-menu-wrapper.hs-menu-flow-horizontal.flyouts > ul li.hs-item-has-children ul.hs-
menu-children-wrapper li.hs-item-has-children ul.hs-menu-children-wrapper {

  left: 100%;

  top: 0;
}

.hs-menu-wrapper.hs-menu-flow-horizontal.flyouts > ul li.hs-item-has-children: hover >
ul.hs-menu-children-wrapper {

  opacity: 1;

  visibility: visible;
}

.row-fluid-wrapper:last-child .hs-menu-wrapper.hs-menu-flow-horizontal > ul {

  margin-bottom: 0;
}

.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch {

  position: relative;
}

.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch > ul {

  margin-bottom: 0;
}

```

```
.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch > ul li.hs-item-  
has-children {  
  
    position: static;  
  
}  
  
.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch > ul li.hs-item-  
has-children ul.hs-menu-children-wrapper {  
  
    display: none;  
  
}  
  
.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch > ul li.hs-item-  
has-children.active-branch > ul.hs-menu-children-wrapper {  
  
    display: block;  
  
    visibility: visible;  
  
    opacity: 1;  
  
}  
  
.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch > ul li.hs-item-  
has-children.active-branch > ul.hs-menu-children-wrapper:before {  
  
    content: " ";  
  
    display: table;  
  
}  
  
.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch > ul li.hs-item-  
has-children.active-branch > ul.hs-menu-children-wrapper:after {  
  
    content: " ";  
  
    display: table;  
  
    clear: both;  
  
}
```

```

.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch > ul li.hs-item-
has-children.active-branch > ul.hs-menu-children-wrapper > li {

    float: left;

}

.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch > ul li.hs-item-
has-children.active-branch > ul.hs-menu-children-wrapper > li a {

    display: inline-block;

}

/* Vertical Menu

=====
===== */

.hs-menu-wrapper.hs-menu-flow-vertical {

    width: 100%;

}

.hs-menu-wrapper.hs-menu-flow-vertical ul {

    list-style: none;

    margin: 0;

}

.hs-menu-wrapper.hs-menu-flow-vertical li a {

    display: block;

}

.hs-menu-wrapper.hs-menu-flow-vertical > ul {

```

```

margin-bottom: 0;
}

.hs-menu-wrapper.hs-menu-flow-vertical > ul li.hs-menu-depth-1 > a {
width: auto;
}

.hs-menu-wrapper.hs-menu-flow-vertical > ul li.hs-item-has-children {
position: relative;
}

/* Flyouts */

.hs-menu-wrapper.hs-menu-flow-vertical.flyouts > ul li.hs-item-has-children ul.hs-menu-
children-wrapper {
visibility: hidden;
opacity: 0;
-webkit-transition: opacity 0.4s;
position: absolute;
z-index: 10;
left: 0;
}

.hs-menu-wrapper.hs-menu-flow-vertical > ul li.hs-item-has-children ul.hs-menu-
children-wrapper li a {
display: block;
white-space: nowrap;
}

```

```
.hs-menu-wrapper.hs-menu-flow-vertical.flyouts > ul li.hs-item-has-children ul.hs-menu-children-wrapper {  
    left: 100%;  
    top: 0;  
}  
  
.hs-menu-wrapper.hs-menu-flow-vertical.flyouts > ul li.hs-item-has-children:hover > ul.hs-menu-children-wrapper {  
    opacity: 1;  
    visibility: visible;  
}  
  
@media (max-width: 767px) {  
    .hs-menu-wrapper, .hs-menu-wrapper * {  
        -webkit-box-sizing: border-box;  
        -moz-box-sizing: border-box;  
        box-sizing: border-box;  
        display: block;  
        width: 100%;  
    }  
  
    .hs-menu-wrapper.hs-menu-flow-horizontal ul {  
        list-style: none;  
        margin: 0;  
        display: block;
```

```

}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul {
  display: block;
}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-menu-depth-1 {
  float: none;
}

.hs-menu-wrapper.hs-menu-flow-horizontal > ul li a,
.hs-menu-wrapper.hs-menu-flow-horizontal > ul li.hs-item-has-children ul.hs-menu-
children-wrapper li a,
.hs-menu-wrapper.hs-menu-flow-horizontal.hs-menu-show-active-branch > ul li.hs-
item-has-children.active-branch > ul.hs-menu-children-wrapper > li a {
  display: block;
}

/* Stacking Horizontal Nav for Mobile */

.hs-menu-wrapper.hs-menu-flow-horizontal>ul li.hs-item-has-children ul.hs-menu-
children-wrapper {
  visibility: visible !important;
  opacity: 1 !important;
  position: static !important;
}

.hs-menu-wrapper.hs-menu-flow-horizontal ul ul ul {
  padding: 0;
}

```

```
.hs-menu-wrapper.hs-menu-flow-horizontal>ul li.hs-item-has-children ul.hs-menu-children-wrapper li a {
```

```
    white-space: normal;
```

```
}
```

```
/* Stacking Vertical Nav for Mobile */
```

```
.hs-menu-wrapper.hs-menu-flow-vertical.flyouts > ul li.hs-item-has-children ul.hs-menu-children-wrapper {
```

```
    position: static;
```

```
    opacity: 1;
```

```
    visibility: visible;
```

```
}
```

```
}
```

```
.hs-menu-wrapper.hs-menu-flow-vertical.no-flyouts .hs-menu-children-wrapper {
```

```
    visibility: visible;
```

```
    opacity: 1;
```

```
}
```

```
.hs-menu-wrapper.hs-menu-flow-horizontal.no-flyouts > ul li.hs-item-has-children ul.hs-menu-children-wrapper {
```

```
    display: block;
```

```
    visibility: visible;
```

```
    opacity: 1;
```

```
}
```



```
/* Space Module */

.widget-type-space {
  visibility: hidden;
}

/* Blog Author Section */

.hs-author-social-links {
  display: inline-block;
}

.hs-author-social-links a.hs-author-social-link {
  width: 24px;
  height: 24px;
  border-width: 0px;
  border: 0px;
  line-height: 24px;
  background-size: 24px 24px;
  background-repeat: no-repeat;
  display: inline-block;
  text-indent: -99999px;
}

.hs-author-social-links a.hs-author-social-link.hs-social-facebook {
  background-image: url("//static.hubspot.com/final/img/common/icons/social/facebook-24x24.png");
}
```

```
.hs-author-social-links a.hs-author-social-link.hs-social-linkedin {
    background-image: url("//static.hubspot.com/final/img/common/icons/social/linkedin-24x24.png");
}

.hs-author-social-links a.hs-author-social-link.hs-social-twitter {
    background-image: url("//static.hubspot.com/final/img/common/icons/social/twitter-24x24.png");
}

.hs-author-social-links a.hs-author-social-link.hs-social-google-plus {
    background-image:
url("//static.hubspot.com/final/img/common/icons/social/googleplus-24x24.png");
}

/* Fix for CTA border box issue */

.hs-cta-wrapper a {
box-sizing: content-box;
-moz-box-sizing: content-box;
-webkit-box-sizing: content-box;
}

/*
* jQuery FlexSlider v2.0
* http://www.woothemes.com/flexslider/
*
* Copyright 2012 WooThemes
```

```

* Free to use under the GPLv2 license.
* http://www.gnu.org/licenses/gpl-2.0.html
*
* Contributing author: Tyler Smith (@mbmuffin)
*/

/* Wrapper */
.hs_cos_wrapper_type_image_slider {
    display: block;
    overflow: hidden
}

/* Browser Resets */
.hs_cos_flex-container a:active,
.hs_cos_flex-slider a:active,
.hs_cos_flex-container a:focus,
.hs_cos_flex-slider a:focus {outline: none;}
.hs_cos_flex-slides,
.hs_cos_flex-control-nav,
.hs_cos_flex-direction-nav {margin: 0; padding: 0; list-style: none;}

/* FlexSlider Necessary Styles
***** /
.hs_cos_flex-slider {margin: 0; padding: 0;}

```

```

.hs_cos_flex-slider .hs_cos_flex-slides > li {display: none; -webkit-backface-visibility:
hidden; position: relative;} /* Hide the slides before the JS is loaded.Avoids image
jumping */

.hs_cos_flex-slider .hs_cos_flex-slides img {width: 100%; display: block; border-radius:
0px;}

.hs_cos_flex-pauseplay span {text-transform: capitalize;}

/* Clearfix for the .hs_cos_flex-slides element */

.hs_cos_flex-slides:after {content: "."; display: block; clear: both; visibility: hidden; line-
height: 0; height: 0;}

html[xmlns] .hs_cos_flex-slides {display: block;}

* html .hs_cos_flex-slides {height: 1%;}

/* No JavaScript Fallback */

/* If you are not using another script, such as Modernizr, make sure you
* include js that eliminates this class on page load */

/* FlexSlider Default Theme
*****/

.hs_cos_flex-slider {margin: 0 0 60px; background: #fff; border: 0; position: relative; -
webkit-border-radius: 4px; -moz-border-radius: 4px; -o-border-radius: 4px; border-
radius: 4px; zoom: 1;}

.hs_cos_flex-viewport {max-height: 2000px; -webkit-transition: all 1s ease; -moz-
transition: all 1s ease; transition: all 1s ease;}

.loading .hs_cos_flex-viewport {max-height: 300px;}

.hs_cos_flex-slider .hs_cos_flex-slides {zoom: 1;}

```

```

.carousel li {margin-right: 5px}

/* Direction Nav */

.hs_cos_flex-direction-nav {*height: 0;}

.hs_cos_flex-direction-nav a {width: 30px; height: 30px; margin: -20px 0 0; display:
block; background: url(http://cdn2.hubspotqa.com/local/hub/124/file-52894-
png/bg_direction_nav.png) no-repeat 0 0; position: absolute; top: 50%; z-index: 10;
cursor: pointer; text-indent: -9999px; opacity: 0; -webkit-transition: all .3s ease;}

.hs_cos_flex-direction-nav .hs_cos_flex-next {background-position: 100% 0; right: -
36px; }

.hs_cos_flex-direction-nav .hs_cos_flex-prev {left: -36px;}

.hs_cos_flex-slider:hover .hs_cos_flex-next {opacity: 0.8; right: 5px;}

.hs_cos_flex-slider:hover .hs_cos_flex-prev {opacity: 0.8; left: 5px;}

.hs_cos_flex-slider:hover .hs_cos_flex-next:hover, .hs_cos_flex-
slider:hover .hs_cos_flex-prev:hover {opacity: 1;}

.hs_cos_flex-direction-nav .hs_cos_flex-disabled {opacity: .3!important;
filter:alpha(opacity=30); cursor: default;}

/* Direction Nav for the Thumbnail Carousel */

.hs_cos_flex_thumbnavs-direction-nav {

margin: 0px;

padding: 0px;

list-style: none;

}

.hs_cos_flex_thumbnavs-direction-nav {*height: 0;}

```

```

.hs_cos_flex_thumbnavs-direction-nav a {width: 30px; height: 140px; margin: -60px 0 0;
display: block; background: url(http://cdn2.hubspotqa.com/local/hub/124/file-52894-
png/bg_direction_nav.png) no-repeat 0 40%; position: absolute; top: 50%; z-index: 10;
cursor: pointer; text-indent: -9999px; opacity: 1; -webkit-transition: all .3s ease;}

.hs_cos_flex_thumbnavs-direction-nav .hs_cos_flex_thumbnavs-next {background-
position: 100% 40%; right: 0px; }

.hs_cos_flex_thumbnavs-direction-nav .hs_cos_flex_thumbnavs-prev {left: 0px;}

.hs-cos-flex-slider-control-panel img { cursor: pointer; }

.hs-cos-flex-slider-control-panel img:hover { opacity:.8; }

.hs-cos-flex-slider-control-panel { margin-top: -30px; }

/* Control Nav */

.hs_cos_flex-control-nav {width: 100%; position: absolute; bottom: -40px; text-align:
center;}

.hs_cos_flex-control-nav li {margin: 0 6px; display: inline-block; zoom: 1; *display:
inline;}

.hs_cos_flex-control-paging li a {width: 11px; height: 11px; display: block; background:
#666; background: rgba(0,0,0,0.5); cursor: pointer; text-indent: -9999px; -webkit-border-
radius: 20px; -moz-border-radius: 20px; -o-border-radius: 20px; border-radius: 20px;
box-shadow: inset 0 0 3px rgba(0,0,0,0.3);}

.hs_cos_flex-control-paging li a:hover { background: #333; background:
rgba(0,0,0,0.7); }

.hs_cos_flex-control-paging li a.hs_cos_flex-active { background: #000; background:
rgba(0,0,0,0.9); cursor: default; }

.hs_cos_flex-control-thumbs {margin: 5px 0 0; position: static; overflow: hidden;}

```

```

.hs_cos_flex-control-thumbs li {width: 25%; float: left; margin: 0;}

.hs_cos_flex-control-thumbs img {width: 100%; display: block; opacity: .7; cursor:
pointer;}

.hs_cos_flex-control-thumbs img:hover {opacity: 1;}

.hs_cos_flex-control-thumbs .hs_cos_flex-active {opacity: 1; cursor: default;}

@media screen and (max-width: 860px) {

.hs_cos_flex-direction-nav .hs_cos_flex-prev {opacity: 1; left: 0;}

.hs_cos_flex-direction-nav .hs_cos_flex-next {opacity: 1; right: 0;}
}

.hs_cos_flex-slider .caption {
background-color: black;
position: static;
font-size: 2em;
line-height: 1.1em;
color: white;
padding: 0px 5% 0px 5%;
width: 100%;
top: 40%;
text-align: center;
}

.hs_cos_flex-slider .superimpose .caption {

```

```
color: white;

font-size: 3em;

line-height: 1.1em;

position: absolute;

padding: 0px 5% 0px 5%;

width: 90%;

top: 40%;

text-align: center;

background-color: transparent;
}

@media all and (max-width: 400px) {

  .hs_cos_flex-slider .superimpose .caption {

    background-color: black;

    position: static;

    font-size: 2em;

    line-height: 1.1em;

    color: white;

    width: 90%;

    padding: 0px 5% 0px 5%;

    top: 40%;

    text-align: center;

  }
}
```



```
}

.hs_cos_flex-slider h1,
.hs_cos_flex-slider h2,
.hs_cos_flex-slider h3,
.hs_cos_flex-slider h4,
.hs_cos_flex-slider h5,
.hs_cos_flex-slider h6,
.hs_cos_flex-slider p {
    color: white;
}

/* Thumbnail only version of the gallery */
.hs-gallery-thumbnails li {
    display: inline-block;
    margin: 0px;
    padding: 0px;
    margin-right: -4px;
}

.hs-gallery-thumbnails.fixed-height li img {
    max-height: 150px;
    margin: 0px;
    padding: 0px;
```

```
border-width: 0px;
}

/* responsive pre elements */

pre {
    overflow-x: auto;
}

/* responsive pre tables */

table pre {
    white-space: pre-wrap;
}

/* adding minimal spacing for blog comments */

.comment {
    margin: 10px 0 10px 0;
}

/* make sure lines with no whitespace don't interfere with layout */

.hs_cos_wrapper_type_rich_text,
```

```
.hs_cos_wrapper_type_text,  
.hs_cos_wrapper_type_header,  
.hs_cos_wrapper_type_section_header,  
.hs_cos_wrapper_type_raw_html,  
.hs_cos_wrapper_type_raw_jinja,  
.hs_cos_wrapper_type_page_footer {  
    word-wrap: break-word;  
}  
  
@media all and (max-width: 400px) {  
    /* beat recaptcha into being responsive, !important and specificity are necessary */  
    #captcha_wrapper,  
    #recaptcha_area,  
    #recaptcha_area table#recaptcha_table,  
    #recaptcha_area table#recaptcha_table .recaptcha_r1_c1  
    { width: auto !important; overflow: hidden; }  
    #recaptcha_area table#recaptcha_table .recaptcha_r4_c4  
    { width: 67px !important; }  
    #recaptcha_area table#recaptcha_table #recaptcha_image  
    { width:280px !important; }  
}
```

これら 2 つの CSS ファイルに加えて、[このチュートリアルに従ってモバイルナビゲーションを実装](#)できます。

HubDB

HubDB - HubSpot CMS

HubDB は、スプレッドシートのようにテーブルの行、列、セルで表現されるリレーショナルデータストアです。テーブル内のデータは、ニーズに応じてさまざまな方法で取得できます。外部から [HubDB API](#) を使用してデータを照会するか、または HubSpot の HubL マークアップタグを使用してデータを CMS に取得することができます。

注：CMS アドオンを使用している HubSpot のすべてのお客様は HubDB を利用できませんが、HubDB は HubSpot マーケティング E メールのパersonナライズには使用できません。

HubDB へのアクセス権の取得

ユーザーが HubDB を使用できるようにするためにユーザーに割り当てることができる 2 種類のアクセス権があります。いずれの設定もユーザーの権限領域内にあります。アクセス権を割り当てるには、[設定] > [ユーザーとチーム] に移動し、ユーザーの上にカーソルを置いて [アクション] メニューから [編集] を選択します。

Settings

The screenshot displays the 'Users' tab in the Salesforce Settings interface. The main content area shows a table of users with columns for 'NAME', 'TEAM', and 'ROLES'. An 'Actions' dropdown menu is open for the first user, with the 'Edit' option highlighted. The left sidebar contains a list of settings categories, including 'Account Defaults', 'Contacts & Companies', 'Conversations', 'Domains & URLs', 'Import & Export', 'Integrations', 'Marketing', 'Private Content', 'Properties', 'Reports', 'Sales', 'Service', 'Users & Teams', and 'Website'. At the top right, there are buttons for 'Manage paid users' and 'Create user'.

設定は編集権限ペインの以下のタブに表示されます。

- [マーケティング] タブ > HubDB [公開、読み取り、書き込み]:
 - 公開:ユーザーが既存の HubDB テーブルを公開/更新できるようにします。
 - 読み取り:ユーザーが HubDB テーブルの内容を表示できるようにします。
 - 書き込み:ユーザーが HubDB テーブルを編集できるようにします。
- [管理者] タブ > [HubDB テーブルを編集] [切り替え]:
 - オンに切り替えると、ユーザーがテーブルを作成、複製、非公開、削除できるようにになります。

HubDB の技術的な制限事項

- HubDB テーブルあたりの行数 : 10,000 行

- `hubdb_table_rows()`の1回の呼び出しとして定義されている CMS ページあたりのテーブルスキャン数: 10
- [HubSpot の API に関する一般制限事項](#)

最初のテーブルの作成

[HubDB エディターアプリ](#)を開き、[テーブルを作成] ボタンをクリックします (または API を使用して [テーブルエンドポイントを作成します](#))。

*注:*作成された新しいテーブルのステータスはすべて、下書きに設定されます。テーブルを公開するまでは、HubL または API でデータを出力するためにテーブルを使用することはできません。

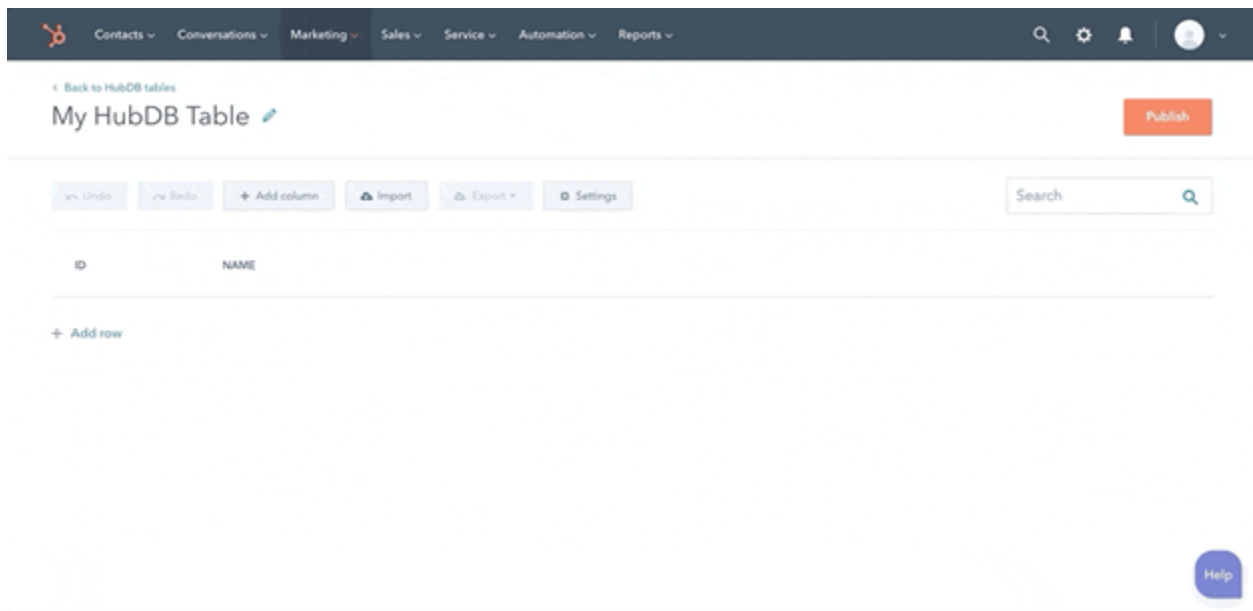


テーブルの権限設定

HubDB テーブル設定を管理するには、テーブル情報の上にある設定ボタンをクリックします。以下の設定が利用可能です。

- **[パブリック API アクセスを許可]:** [既定ではオン] このオプションをオフに切り替えると、テーブルの内容はアプリまたは認証済み API でのみ表示できます。オンに切り替えると、認証されていない API からでもデータを照会できます。

- [行データを使用して動的ページの作成が可能]:動的ページを使用すると、HubDB ではテーブルの各行に対応するページを作成できます。[動的ページの詳細についてはこちらをご覧ください。](#)このオプションをオンに切り替えると、メタディスクリプション、サムネイル画像、および正規 URL のソースとして使用される HubDB テーブルの列を選択できます。
 - 動的ページを使用する場合は、[子テーブル] も選択できます。[HubDB を使用したマルチレベルの動的ページの作成の詳細についてはこちらをご覧ください。](#)



API からデータを取得するだけの場合には、動的ページを有効にする必要がありますか？

いいえ。API からテーブルデータを取得するだけの予定である場合は、この設定を有効に必要なはありません。

HubDB アーキテクチャ

テーブル

テーブルとは、行と列からなる 2 次元構造です。作成されたテーブルには、グローバル一意 ID が付与されます。この ID を使用してテーブルを識別できます。

行

行は、テーブルの横方向のスライスです。通常、1 つの行のすべての値は 1 つのプライマリ ID に関連しています。スプレッドシートアプリケーションでは、行は 1 から始まる番号で表されます。各テーブル行には、作成時にグローバル一意 ID が指定されます。

各 HubDB テーブルには最大 10,000 行を格納できます。

列

列はテーブルの縦方向のスライスです。各列にはタイプがあります。スプレッドシートアプリケーションでは、列は A、B、C などの英字で表されます。次の 2 種類のテーブル列があります。

組み込み列

テーブル内の各行には、2 つの組み込み列があります。

- **hs_id**: この行に自動的に割り当てられた数値のグローバル一意 ID。
- **hs_created_at**: この行が作成された時点のタイムスタンプ。
- **hs_path**: 動的ページで使用される場合、この文字列はページの URL パスの最後のセグメントです。
- **hs_name**: 動的ページで使用される場合、これはページのタイトルです。

カスタム列

テーブルに追加できるカスタム列の数に制限はありません。カスタム列は、テキスト、リッチテキスト、数値、通貨、日付、時刻、画像、動画、選択、または場所 (緯度と経度) にできます。

作成された列には、テーブル内で固有な 1 から始まる数値 ID が割り当てられます。列 ID は増加しますが、必ずしも連続していません。列 ID は再利用できないので、テーブルに 1 と 2 という 2 列がある場合、2 番目の列を削除すると、次に作成される列は 3 になります。

セル

セルは列と行が交差する位置であり、値が格納されます。セルの読み取りと更新は、セルごとに個別に実行するか、または行の一部として実行します。セルの値を `Null` に設定する操作は、セルの値を削除する操作に相当します。

HubL を使用した HubDB データへのアクセス

行の取得

HubL からテーブルを照会し、結果を繰り返し処理できます。

テーブルの行をリストするには `hubdb_table_rows()` を使用します。

```
{% for row in hubdb_table_rows(<tableId>, <filterQuery>) %}  
  the value for row {{ row.hs_id }} is {{ row.<column name> }}
```

```
{% endfor %}
```

<filterQuery>の構文は、HTTP APIと同じです。たとえば hubdb_table_rows(123, "employees__gt=10&orderBy=count")は、"employees"列が 10 よりも大きい行を、"count"列に基づいて並べ替えた行のリストを返します。<filterQuery>のすべての任意パラメーターのリストについては[こちらをご覧ください](#)。

注：異なる<filterQuery>パラメーターを指定した複数の行クエリを使用する代わりに、1つのクエリを実行して、`selectattr()` または `rejectattr()` フィルターを使用して行を絞り込みます。

```
{% set all_cars = hubdb_table_rows(<tableId>) %}

{% set cars_with_windows = all_cars|selectattr('windows') %}

{% set teslas = all_cars|selectattr('make','equalto','tesla') %}
```

1つの行を取得するには、`hubdb_table_row()`を使用します。

```
{% set row = hubdb_table_row(<tableId>, <rowId>) %}
the value for {{ row.hs_id }} is {{ row.<column name> }}
```

組み込み列とカスタム列の名前では大文字と小文字は区別されません。HS_ID と hs_id は同じです。

行の属性

- `{{ row.hs_id }}` - 行のグローバル一意 ID。
- `{{ row.hs_path }}` - 動的ページを使用する場合、この文字列は [ページのパス] 列の値であり、ページの URL パスの最後のセグメントです。
- `{{ row.hs_name }}` - 動的ページを使用する場合、この文字列は行の [ページタイトル] 列の値です。

- `{{ row.hs_created_at }}` - 行が作成された時点の Unix タイムスタンプです。
- `{{ row.hs_child_table_id }}` - 動的ページを使用する場合、これは行のデータを提供する別のテーブルの ID です。
- `{{ row.column_name }}`または`{{ row["column_name"] }}` - 列名を使用してカスタム列の値を取得します。

注：HubDB テーブルあたりの行数は 10,000 行に制限されています。

テーブル情報の取得

テーブルの名前、列、最終更新日などの情報を取得するには、`hubdb_table()`関数を使用します。

```
{% set table_info = hubdb_table(<tableId>) %}
```

テーブルの属性

前述のコードで `hubdb_table()`が割り当てられている変数に関連する属性を以下に示します。実際に使用する変数が異なる場合があります。

注:使いやすくするため、変数にこれを割り当てることをお勧めします。割り当てない場合は、

`{{ hubdb_table(<tableId>).attribute }}`を使用できません。

- `{{ table_info.id }}` - テーブルの ID。
- `{{ table_info.name }}` - テーブルの名前。
- `{{ table_info.columns }}` - 列情報のリスト。
 - [forループ](#)を使用して、この属性で使用可能な情報を繰り返し処理できます。

- `{{ table_info.created_at }}` - テーブルが初めて作成された時点のタイムスタンプ。
- `{{ table_info.published_at }}` - このテーブルが公開された時点のタイムスタンプ。
- `{{ table_info.updated_at }}` - このテーブルの最終更新時点のタイムスタンプ。
- `{{ table_info.row_count }}` - テーブルの行数。

列情報の取得

テーブルの列に関する情報 (ラベル、タイプ、オプションなど) を取得するには、`hubdb_table_column()`関数を使用します。

```
{% set table_info = hubdb_table_column(<tableId>, <columnId or column name>) %}
```

列の属性

前述のコードで `hubdb_table_column()`が割り当てられている変数に関連する属性を以下に示します。実際に使用する変数が異なる場合があります。

注:使いやすくするため、変数にこれを割り当てることをお勧めします。割り当てない場合は、

`{{ hubdb_table_column(<tableId>,<columnId or column name>).attribute }}`を使用できません。

- `{{ table_info.id }}` - 列の ID。
- `{{ table_info.name }}` - 列の名前。
- `{{ table_info.label }}` - 列に使用するラベル。
- `{{ table_info.type }}` - この列のタイプ。

- `{{ table_info.options }}` - select 列タイプの場合、これは `optionId` から `option` 情報へのマップです。
- `{{ table_info.foreignIds }}` - `foreignId` 列タイプの場合、`foreignId` のリストです (`id` プロパティと `name` プロパティが含まれる)。

列のメソッド

- `getOptionByName("<option name")` - select 列タイプの場合、オプション名に基づいてオプションの情報を取得します。

HubDB REST API の使用

HubDB データを外部で管理するには、[こちらにある HubSpot の REST API](#) を使用できます。

コンテンツステージング

HubSpot CMS - コンテンツステージング

コンテンツステージングは、本番サイトにページを公開する前にステージングされたページの再設計や作成を行うためのアプリ内開発環境です。コンテンツステージングは、Website アドオン (サイトページ) がある HubSpot アカウントで利用可能です。

コンテンツステージングの使用方法に関する完全なチュートリアルについては、[ウェブ](#)
[サイトユーザーガイドのこちらの記事](#)をご覧ください。

HubSpot FTP

FTP で HubSpot CMS に接続する 方法

HubSpot では、FTP を使用したテンプレートファイル、スタイルシート、JavaScript ファイル、ファイルマネージャーアセットのアップロードとダウンロードがサポートされています。これは、廃止されたアップローダーツールに代わる機能です。新しいデザインマネージャーでは、FTP を使用してカスタムモジュールを作成、編集できます。

2 要素認証 (2FA) が有効になっているユーザーは FTP を使用できません。[ベータ] CLI では 2FA は機能しないので、強力なセキュリティのために OAuth2 を使用することが推奨されます。

一般的な手順

HubSpot FTP サービスは、ホスト名 `ftp.hubapi.com`、ポート 3200 で実行されます。初回接続はクリアテキストで行われますが、ログインパスワードの送信前に、接続を TLS にアップグレードする必要があります。クライアントは、明示的な認証を使用して FTP-SSL 対応で構成されている必要があります。ログインとパスワードは、

app.hubspot.com のログインと同じです。パッシブモードを有効にする必要があります。

あるいは、ポート 3201 で暗黙的 SSL を使用して接続できます。HubSpot FTP サービスは、コントロールポートの他に一連のデータポートを使用します。転送のたびに、アップロードまたはダウンロード用のデータポートがランダムに割り当てられます。企業のファイアウォールのデータポートをオープンする必要がある場合、HubSpot のデータポートは 3203 から 3299 です。

セキュリティで保護された FTP サービスを提供するには、FTP (SSL を使用する FTP) と SFTP の 2 種類の方法があります。HubSpot では FTPS のみがサポートされています。

FTP クライアントを使用した接続

FTP クライアントに接続するには、以下の手順に従います。スクリーンショットは、[Fetch](#)(Mac OS X の FTP クライアント) のものです。[他の FTP クライアントの例は以下で紹介します。](#) [cURL](#) を使用して接続することもできます。

1. 接続設定を編集する

接続が **TLS/SSL** を使用するように設定されていることを確認します。

2. サーバーを入力する

[ホスト] または [サーバー] フィールドに **ftp.hubapi.com** と入力します。

3. ポートを入力する

ポートとして **3200** を入力します。暗黙的 SSL を使用する場合は、ポートを 3201 に設定する必要があります。

4. ユーザー資格情報を入力する

HubSpot のユーザー名とパスワードを入力します。FTP で接続するには、アカウントの HubSpot ユーザーである必要があります。

5. 接続する

サーバーに接続します。正常に接続すると、"portals" というフォルダーが表示されます。

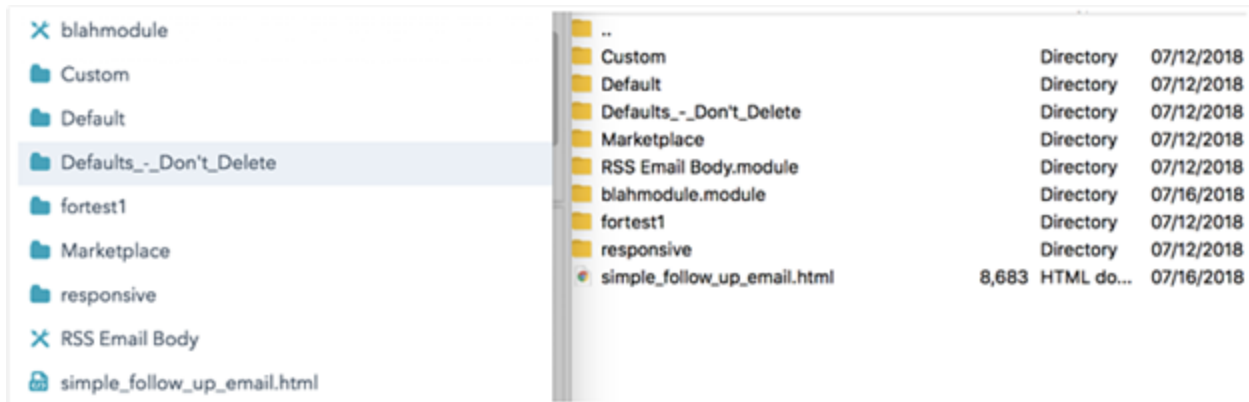
ディレクトリ構造

"portals" フォルダーには、ユーザーが認証されている各ポータル HubID ディレクトリが含まれています。各 HubID ディレクトリには 3 つのサブフォルダーがあります。

- **files** - ファイルマネージャーにアップロードされたアセット。
- **designs** - デザインマネージャーのテンプレート、CSS、JavaScript コードファイル。

▼ 158228	0 B 12/31/69, 7:00 PM
▼ content	0 B 12/31/69, 7:00 PM
▶ files	109 B 12/31/69, 7:00 PM
▶ templates	1 B 12/31/69, 7:00 PM

"designs" フォルダーのアセットは、[finder](#) でのデザインマネージャーのフォルダー構造に一致します。FTP での個々のアセットまたはフォルダー構造に対する変更は、デザインマネージャーに反映されます。



新しいコード化モジュールを作成するには、.module 拡張子を持つ新しいディレクトリを作成します。これにより新しいカスタムモジュールが自動的に作成されます。FTP で作成された新しいモジュールには、既定でページテンプレートとブログテンプレートとの互換性があります。

現時点では、designs フォルダでは [ドラッグ&ドロップテンプレート](#) はサポートされていません。

その他の FTP クライアントの例

Cyberduck

FTP-SSL (Explicit AUTH TLS)

Server: ftp.hubapi.com Port: 3200

URL: <https://yourlogin@yourcompany.com@ftp.hubapi.com:3200/>

Username: yourlogin@yourcompany.com

Password: ●●●●●●●●

Anonymous Login
 Add to Keychain

[?](#)

▼ **More Options**

Path:

Connect Mode: Default

Encoding: Default

Use Public Key Authentication
 No private key selected

Filezilla

General Advanced Transfer Settings Charset

Host: ftp.hubapi.com Port: 3200

Protocol: FTP - File Transfer Protocol

Encryption: Require explicit FTP over TLS

Logon Type: Normal

User: yourlogin@yourcompany.com

Password: ●●●●●●●●

Account:

Comments:

Transmit



Coda

Connect to FTP Server

Server: ftp.hubapi.com

User Name: yourlogin@yourcompany.com

Password:

Initial Path: optional

Port: 3200

FTP

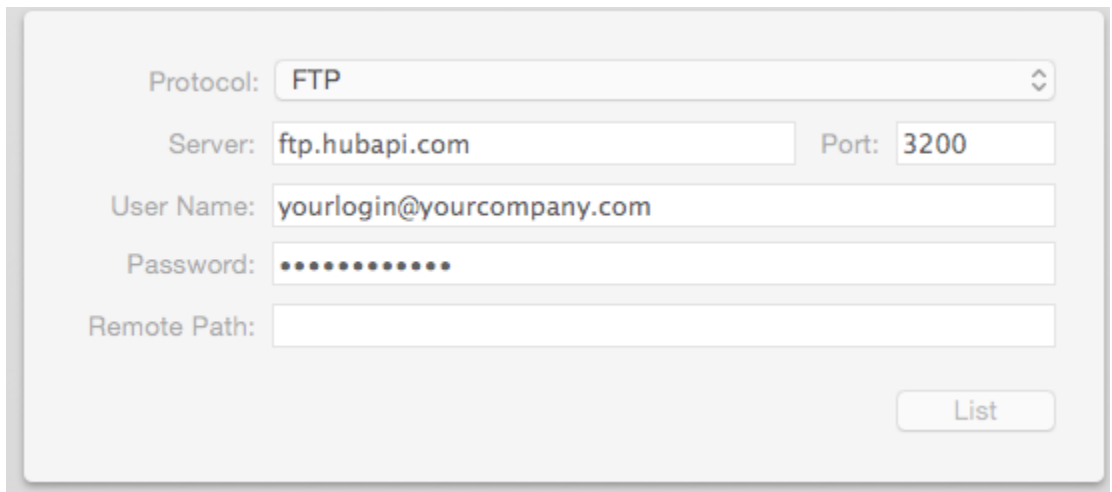
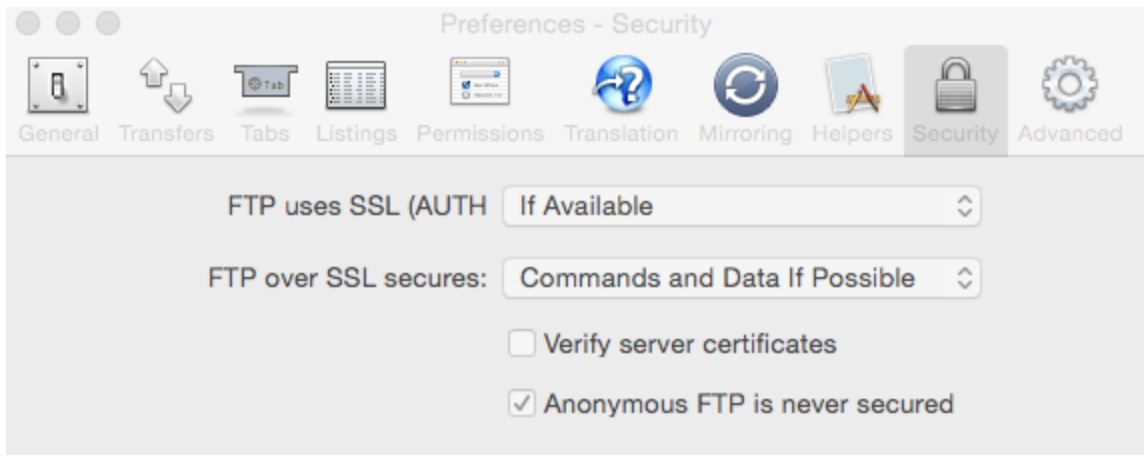
FTP with Implicit SSL

FTP with TLS/SSL

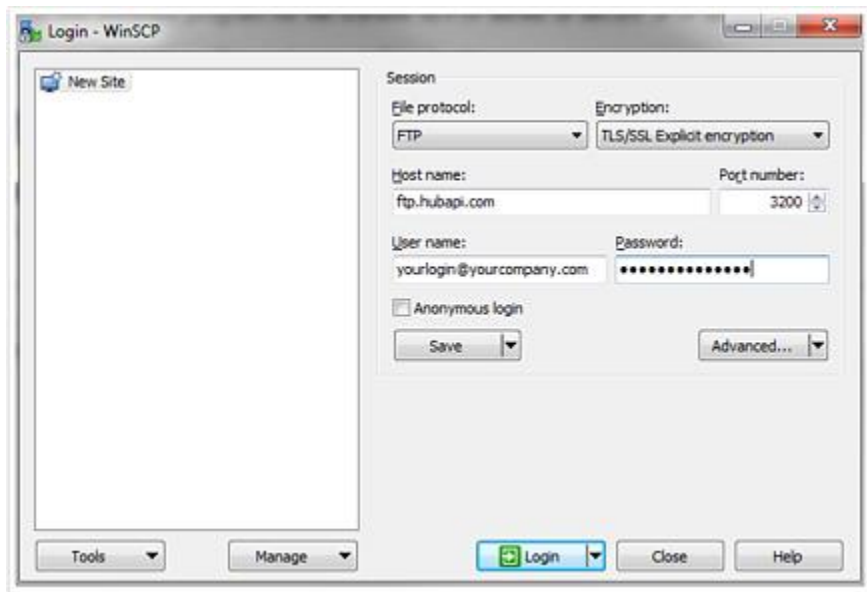
Connect

Interarchy

[Preferences] メニューで [Security] を選択し、**[FTP uses SSL (AUTH)]** で [Never] を [If Available] に変更します。



WinSCP



CuteFTP

Site Properties for: My HubSpot Site

General Actions Type Options

Label:
My HubSpot Site

Host address:
ftp.hubapi.com

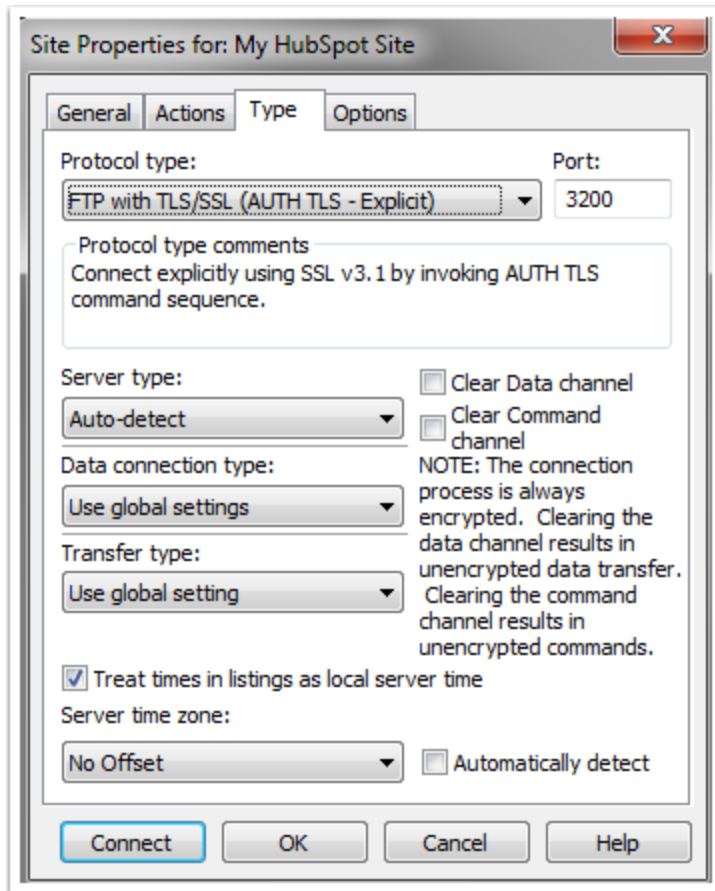
Username:
yourlogin@yourcompany.com

Password:
●●●●●●●●●●

Login method:
 Normal
 Anonymous
 Double

Comments:

Connect OK Cancel Help



Sublime SFTP

```
hubspot ftps x
1 {
2   // The tab key will cycle through the settings when first created
3   // Visit http://wbond.net/sublime_packages/sftp/settings for help
4
5   // sftp, ftp or ftps
6   "type": "ftps",
7
8   "sync_down_on_open": true,
9   "sync_same_age": true,
10
11   "host": "ftp.hubapi.com",
12   "user": "yourlogin@yourcompany.com",
13   //"password": "password",
14   "port": "3200",
15
16   "remote_path": "/portals/",
17   //"file_permissions": "664",
18   //"dir_permissions": "775",
19
20   //"extra_list_connections": 0,
21
22   "connect_timeout": 30,
23   //"keepalive": 120,
24   //"ftp_passive_mode": true,
25   //"ftp_obey_passive_host": false,
26   //"ssh_key_file": "~/.ssh/id_rsa",
27   //"sftp_flags": ["-F", "/path/to/ssh_config"],
28
29   //"preserve_modification_times": false,
30   //"remote_time_offset_in_hours": 0,
31   //"remote_encoding": "utf-8",
32   //"remote_locale": "C",
33   //"allow_config_upload": false,
34 }
35
```

cURL で接続する

このコマンドを使用して cURL で接続することもできます。

```
curl -v ftp://ftp.hubapi.com:3200/ --ftp-ssl --user yourlogin@yourcompany.com
```

ローカル HubL サーバー+ FTP

ローカル HubL サーバーと FTP クライアントを使用して、ローカルとリモートの両方の 'custom' ディレクトリを同期できます。このワークフローでは、テンプレートと CSS ファイルをローカルで開発し、HubSpot アカウントにアップロードできます。詳細は、[こちらの記事をご覧ください](#)。

トラブルシューティングのヒント

FTP で接続できない場合は、クライアントのログメッセージを確認してください。ご使用の設定が上記の説明にある設定と一致しており、接続が一貫してタイムアウトする状況では、ほとんどの場合は接続がネットワークファイアウォールによってブロックされています。HubSpot の FTP で使用されるポートがファイアウォールによりブロックされていないことを確認してください。

HubSpot CMS のローカル開発についても っと詳しく

HubSpot を使用してローカルに開発を行うための CLI がありますが、この CLI には同じ 2FA の制約事項がなく、カスタムモジュールの作成などの一般的なタスクを実行するためのヘルパーがあります。

このトピックに関するおすすめのスレッドがフォーラムにあります [📄 "Can you set up local development on HubSpot CMS?"](#)

CMS デベロッパーSlack でコミュニティに質問することもできます。

ローカル HubL サーバー

ローカル HubL サーバー

ローカル HubL サーバーでは、マークアップと HubL テンプレート作成言語を使い慣れているウェブデベロッパーが、[HubSpot CMS](#) テンプレートをローカルに作成、テストできます。このサーバーは外部の依存関係を使用せずに 100 %ローカルで稼働します。稼働したら、HubSpot インターフェイスにテンプレートを読み込むことなく、テンプレートを作成、テストできます。

このツールは (OS X、Windows、または Linux) シェル コマンドを使い慣れているデベロッパーを対象としており、実行するには [Java 8 \(またはこれ以降の\) 開発キット](#) がインストールされている必要があります。また、このツールはコンテンツをローカルにレンダリングできるようにすることだけを目的としています。HubSpot には接続せず、また HubSpot システムにファイルを読み込みません。ただし、[FTP](#) と組み合わせて使用するか、またはスタンドアロンで使用することができます。

ローカル開発向けのツールにも新しいベータ版があります。[ローカル開発ツールの概要の詳細についてはこちらをご覧ください。](#)

インストール

以下の手順を実行する前に、[Java 8 \(またはこれ以降の\) 開発キット](#) がインストールされていることを確認してください。以下の手順に従ってローカル HubL サーバーをセットアップします。

1. パッケージをダウンロードする

パッケージの最新バージョンをダウンロードします。

2. パッケージを解凍する

ファイルを解凍します。これにより **local-hubl-server** ディレクトリが作成されます。このフォルダーを、選択したディレクトリに移動できます。

3. ディレクトリを変更する

ターミナルまたはコマンドプロンプトを開き、**local-hubl-server** ディレクトリに移動します。ターミナルでのコマンドは次のとおりです：

```
cd local-hubl-server
```

ファイルのパスは、ディレクトリを解凍して移動した場所に応じて異なります。あるいは、Finder やエクスプローラーでサーバーフォルダーに移動できます。

4. サーバーを実行する

ターミナルまたはコマンドプロンプトで以下のコマンドを実行します。

```
./bin/local-hubl-server
```

[Java 8 開発キット](#) がインストールされていない場合は、ダウンロードしてインストールする指示が表示されます。正常に完了すると、ブラウザーが開き 127.0.0.1:8080 が表示されます。

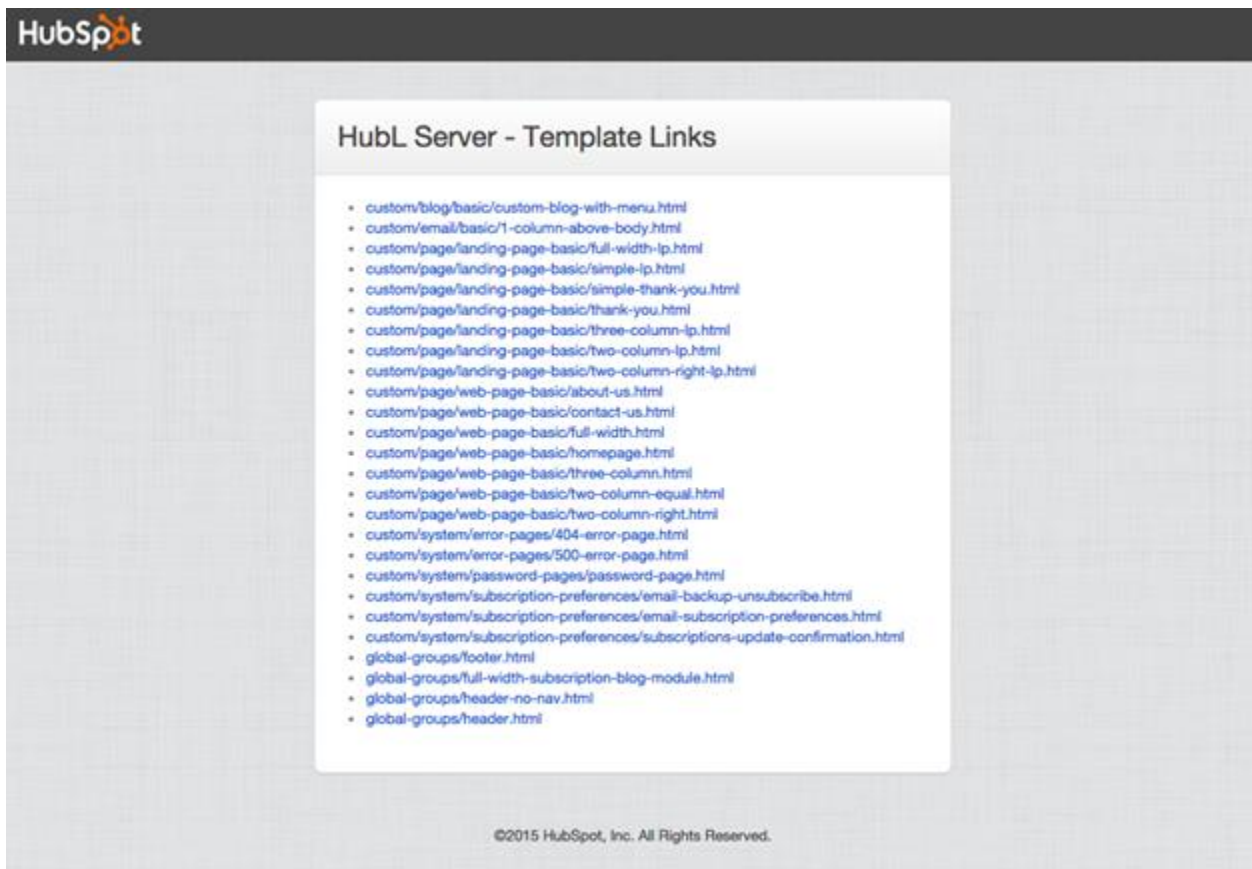
あるいは、Finder またはエクスプローラーから local-hubl-server ディレクトリにアクセスしている場合は、**'bin'** ディレクトリにある local-hubl-server 実行可能ファイルを実行します。

Last login: Fri May 15 11:03:15 on ttys000

→ ~ █

I

5.既定のテンプレートコンテンツを使ってみる



HubSpot

HubL Server - Template Links

- [custom/blog/basic/custom-blog-with-menu.html](#)
- [custom/email/basic/1-column-above-body.html](#)
- [custom/page/landing-page-basic/full-width-lp.html](#)
- [custom/page/landing-page-basic/simple-lp.html](#)
- [custom/page/landing-page-basic/simple-thank-you.html](#)
- [custom/page/landing-page-basic/thank-you.html](#)
- [custom/page/landing-page-basic/three-column-lp.html](#)
- [custom/page/landing-page-basic/two-column-lp.html](#)
- [custom/page/landing-page-basic/two-column-right-lp.html](#)
- [custom/page/web-page-basic/about-us.html](#)
- [custom/page/web-page-basic/contact-us.html](#)
- [custom/page/web-page-basic/full-width.html](#)
- [custom/page/web-page-basic/homepage.html](#)
- [custom/page/web-page-basic/three-column.html](#)
- [custom/page/web-page-basic/two-column-equal.html](#)
- [custom/page/web-page-basic/two-column-right.html](#)
- [custom/system/error-pages/404-error-page.html](#)
- [custom/system/error-pages/500-error-page.html](#)
- [custom/system/password-pages/password-page.html](#)
- [custom/system/subscription-preferences/email-backup-unsubscribe.html](#)
- [custom/system/subscription-preferences/email-subscription-preferences.html](#)
- [custom/system/subscription-preferences/subscriptions-update-confirmation.html](#)
- [global-groups/footer.html](#)
- [global-groups/full-width-subscription-blog-module.html](#)
- [global-groups/header-no-nav.html](#)
- [global-groups/header.html](#)

©2015 HubSpot, Inc. All Rights Reserved.

サーバーが稼働している場合には、IP アドレス 127.0.0.1:8080 では、作業ディレクトリに作成したテンプレートのリストが表示されます。既定のディストリビューションには、試してみることができる公開 hubtheme が付属しています。

'`conf/config.yaml`'が'`work/hubthemes/vast` as a baseDir'を指し示すようにセットアップされていることがわかります。これは、すべての既定のテンプレートが公開される場所です。baseDir を変更するには、`config.yaml` ファイルを編集する必要があります。

`work/hubthemes/vast` ディレクトリのフォルダー構造はデザインマネージャーのフォルダー構造に似ています。作業ディレクトリで HTML ファイルを作成すると、127.0.0.1:8080 で表示されるリンクのリストにこれらのファイルが自動的に追加されます。

ローカル HubL サーバー+ FTP

ローカル HubL サーバーと FTP クライアントを使用して、ローカルとリモートの両方の'`custom`'ディレクトリを同期できます。詳細は、[こちらの記事をご覧ください](#)。

HubSpot CMS のローカル開発についても っと詳しく

このトピックに関するおすすめのスレッドがフォーラムにあります [📄 "Can you set up local development on HubSpot CMS?"](#)

CMS デベロッパーSlack でコミュニティに質問することもできます。

ローカル HubL サーバー+ FTP

ローカル HubL サーバーと FTP の 使用

ローカル HubL サーバーと FTP を組み合わせて使用すると、開発者はテンプレートファイルと CSS ファイルをローカルに開発し、開発したファイルの本番環境の HubSpot アカウントと同期できます。[新しいデザインマネージャー](#)を使用している場合は、[FTP を使用したコンテンツの作成](#)についてこのドキュメントを参照してください。

一般情報

既定では、ローカルサーバーの ('conf'ディレクトリにある) config.yaml ファイルで、ベーステンプレートディレクトリとして **work/hubthemes/vast** が指定されています。

このベーステンプレートディレクトリの中の"custom"ディレクトリには、ストック HTML テンプレートとスタイルシートがあります。また、このディレクトリは、本番環境の HubSpot アカウントのディレクトリをミラーリングします。本番環境の"custom" フォルダーは **portals/hubID/content/templates** の中にあります。

ローカル"custom"ディレクトリのサブフォルダー内で、HTML、CSS、または JS ファイルを編集、作成し、対応するフォルダーに FTP でアップロードできます。

さまざまな [FTP クライアント](#)を使用して [HubSpot](#) に接続し、チームとあなた自身が利用できるワークフローを開発できますが、このチュートリアルでは OSX 向け FTP

クライアントである [Transmit](#) を使用します。Transmit の優れた同期機能を使用すれば、ローカルディレクトリをリモートディレクトリと容易に同期できます。

1.ローカル HubL サーバーをインストールする

ローカル HubL サーバーをまだセットアップしていない場合は、最新バージョンの [ローカル HubL サーバー](#) をダウンロードしてインストールする必要があります。セットアッププロセスの詳細な手順については、[こちらの記事](#)をご覧ください。

2.ローカル"custom"ディレクトリでテンプレートやスタイルシートを編集または作成する

`local-hubl-server` パッケージを解凍します。

このベーステンプレートディレクトリ (`local-hubl-server/work/hubthemes/vast`) の中の `"custom"` ディレクトリには、ストック HTML テンプレートとスタイルシートがあります。サブフォルダー内で HTML ファイルまたは CSS ファイルを編集または作成します。ローカル HubL サーバーに付属のストックテーマファイルを変更することもできます。

3.Transmit を起動する

ローカルコンピューターにサーバーがインストールされたら、FTP で HubSpot に接続できます。このチュートリアルでは FTP クライアントとして [Transmit](#) を使用します。

FTP で認証を行うには、以下の情報を入力します。

- サーバー: ftp.hubapi.com
- HubSpot ユーザー資格情報
- ポート 3200
- [FTP with TLS/SSL] を選択する

FTP での HubSpot への接続の詳細については、[この記事をご覧ください](#)。

4. ローカルサーバディレクトリでローカルの "custom" フォルダを見つける

[Transmit](#) のローカルペインでローカルの "custom" フォルダに移動します。

5. HubSpot がホストする "custom" フォルダを見つける

[Transmit](#) のリモートペインで "custom" フォルダに移動します。ローカルディレクトリはリモートディレクトリをミラーリングしていることがわかります。

6. グローバルグループをアップロードする

ローカル HubL サーバーのストックテンプレートなど、グローバル.html ファイルが含まれているテンプレートを使用している場合は、他のファイルを同期する前に、グローバル'**system/global**'ディレクトリをアップロードする必要があります。

含まれている.html ファイルを最初にアップロードしないと、これらのグループを含むテンプレートファイルを適切にアップロードできません。テンプレートファイルにグローバルグループが含まれていない場合は、このステップをスキップできます。

ローカルペインでシステムディレクトリを開き、'**global**'フォルダーをリモートペインのシステムフォルダーにドラッグします。

7. [Sync] をクリックする

ローカルの'**custom**'ディレクトリに戻ります。両方の custom ディレクトリが [Transmit](#) で開いている状態で **[Sync]** をクリックします。

同期設定の調整

両方の custom ディレクトリが **[Sync]** に設定されていることを確認します。**[File Size]** を選択し、ファイルが変更されているかどうかを確認します。

8. ファイルを同期する

[Synchronize] をクリックします。ローカルサーバーにあるがリモートサーバーにないファイルはすべて、HubSpot と同期されます。同期正常完了メッセージが表示され、同期されたファイルのレポートを確認できます。

9. デザインマネージャーでテンプレートを 表示する

同期されたファイルは、デザインマネージャーの'**Coded Files > Custom**'ディレクトリ
にあります。

HubSpot CMS のローカル開発についても っと詳しく

このトピックに関するおすすめのスレッドがフォーラムにあります👉 ["Can you set up local development on HubSpot CMS?"](#)

[CMS デベロッパーSlack](#) でコミュニティに質問することもできます。

マーケットプレイスプロバイダーアカウント

ローカル開発ツール

ローカル開発を開始する

Local Development Tooling [ベータ]:はじめに

これはベータ版リリースです。当社の [Developer Beta Terms](#) (デベロッパーベータ版利用規約) をご覧ください。サポートが必要な場合には、[デザイナーおよびデベロッパー Slack](#) の #local-dev-beta チャンネルにご参加ください。

Local Development Tooling ベータによりローカルワークフローが HubSpot に接続されるので、HubSpot CMS での開発時に、バージョン管理、お気に入りのテキストエディター、さまざまなウェブ開発テクノロジーを利用できます。

Local Development Tooling では、[モジュール](#)、ドラッグ&ドロップテンプレート、HTML、CSS、および JavaScript ファイルのローカル プロジェクトがサポートされています。このプロジェクトを HubSpot デザインツールと同期できます。また、このローカルプロジェクトを監視し、ローカルファイルの変更内容を HubSpot アカウントに自動的に同期できます。これにより、ご使用のコードエディターと HubSpot がシームレスに統合され、迅速な開発が実現します。

このガイドでは、基本機能をインストールし、HubSpot アカウントからプロジェクトを取得し、変更をデザインツールに同期する方法を詳しく説明します。このワークフローを最大限に活用するには、[コマンドライン](#)に関する実用的な知識が必要です。コマンドとローカルファイル形式については、[Local Development Tooling リファレンス](#)をご覧ください。

i 無料の CMS デベロッパーサンドボックスを開発環境として利用できることはご存知ですか？ アカウントを今すぐ作成するには、[ここで](#) [アカウントの作成] をクリックし、ユーザーを選択し、[新規アカウントを作成] を選択してください。 **i**

依存関係のインストール

ローカルツールを使用するには、以下の手順に従う必要があります。

1. [Node.js](#) (ローカルツールを有効にする JavaScript ランタイム環境) をインストールします。バージョン 8.9.1 以降の Node がサポートされていますが、長期サポート (LTS) バージョンが推奨されます。
2. HubSpot ツールをグローバルにインストールするには、コマンドラインで `npm install -g @hubspot/cms-cli` を実行します。ツールを現在のディレクトリにインストールするには、`npm install @hubspot/cms-cli` を実行します。

注:必要に応じて [Yarn](#) も利用できます。 [Homebrew](#) は必須ではありませんが、インストールを容易にするために推奨されます。Yarn を使用している場合は、コマンドは `yarn` プレフィックス付きで実行されます。

1.作業ディレクトリを作成する

ローカルプロジェクトのフォルダーを作成します。たとえば、コマンドラインで `mkdir local-cms-dev` を実行すると、ディレクトリが作成されます。次に `cd local-cms-dev` を実行して、作成したディレクトリに移動します。

2. 設定ファイルをセットアップする

新しいディレクトリで `touch hubspot.config.yml` を実行して、`hubspot.config.yml` ファイルを作成します。このファイルには、ローカルプロジェクトが HubSpot アカウントを操作できるようにするための認証資格情報が含まれています。ローカルツールでコマンドを実行すると、ツールは HubSpot アカウントへのアクセスを認証するためにこのファイルを参照します。このファイルはプロジェクトファイルよりも上位のディレクトリに含まれています。

ローカル開発環境で HubSpot による認証を行う方法は 2 種類あります。

1. OAuth2 (安全性が高い)

1. **非公開デベロッパーアプリを作成します。**注:アプリの権限で"Content"スコープを超えるスコープを選択すると、このツールでアプリが機能しなくなります。アプリはスコープが選択されていない状態で機能します。
2. コマンドラインで `npx hs auth oauth2` を実行します。これにより、一連のコマンドラインプロンプトが表示されます。
3. CMS ポータル ID を入力します。
4. 非公開デベロッパーアプリからのクライアント ID とクライアントシークレットを入力します。これにより、既定のブラウザが開き、認証が要求されます。
5. 認証ページで CMS ポータルを選択します。
6. 統合権限の要求: 非公開アプリから、アカウントデータへのアクセス権が要求されます。[アクセス権を付与] をクリックします。

7. 正常に完了すると [認証が成功しました (Authorization Succeeded)] が表示され、hubspot.config.yml ファイルが更新されます。

[OAuth2](#) についてもっと詳しく。

2. HubSpot API キー

1. ポータルの API キーをセットアップします。
2. 設定ファイルにエントリーを追加します。

hubspot.config.yml は以下のようになります。

```
defaultPortal: 'DEV'

portals:

# Sample apikey entry, manually entered

- name: 'DEV'

  portalId:123

  authType: 'apikey'

  apiKey: 'xxxxxx-xxxxx-xxxxx-xxxxx-xxxxx'

# Sample oauth2 entry, generated from running npx hscms auth oauth2

- name: 'PROD'

  portalId:345

  authType: oauth2

  auth:

  clientId: xxx-xxx-xxx-xxx

  clientSecret: xxx-xxx-xxx-xxx
```

```
scopes:
- content

tokenInfo:

expiresAt: '2019-07-01T16:42:44.311Z'

refreshToken: xxx-xxx-xxx-xxx

accessToken: >-

xxx-xxx-xxx-xxx-xxx-xxx-xxx-xxx-xxx-xxx-xxx-xxx-xxx-xxx
```

上記の例では、`hubspot.config.yml` ファイルで複数のエントリーがサポートされています。必要に応じてアカウントエントリーに `name` を追加できます。コマンドを実行するときに、`--portal=` オプションで `name` または `portalId` を使用できます (例: 上記の例の場合、`--portal=123` と `--portal=DEV` はいずれもアカウント ID 123 と連携します)。最上位の `defaultPortal` を設定する場合、コマンドで `--portal=` オプションを省略できますが、この場合コマンドはこの既定のポータルを参照します。


3. アカウントからプロジェクトを取得する

`fetch` コマンドを使用して `npx hs fetch --portal=<name> <src> <dest>` を実行し、プロジェクトフォルダツリー全体を HubSpot アカウントからローカル環境に取得します。`src` は HubSpot デザインツールのプロジェクトパスであり、`dest` はファイルを入れるローカルディレクトリのパス (現在の作業ディレクトリを基準とした相対パス) です。現在の作業ディレクトリにプロジェクトを取得するには、`<dest>` 引数を省略できます。

HubSpot アカウントからプロジェクトを取得する例を以下に示します。

```
$ local-cms-dev npx hs fetch --portal=DEV cms-project
```

これは、ローカル環境に取得したデザインツールのプロジェクトです。

 cms-project

 javascript

 scripts.js

 modules

 my_module

 styles

 style.css

 templates

 base.html

 partials

 footer.html

 header.html

LOCAL-CMS-DEV

- cms-project
 - javascript
 - scripts.js
 - modules
 - my_module.module
 - _locales
 - fields.json
 - meta.json
 - module.css
 - module.html
 - module.js
 - styles
 - style.css
 - templates
 - partials
 - footer.html
 - header.html
 - base.html
 - hubspot.config.yml

4.変更を行い、HubSpot アカウントと同期する

`upload` コマンドを使用して `npx hs upload --portal=<name> <src> <dest>` を実行し、ローカル環境のプロジェクトを HubSpot アカウントにアップロードします。これにより `<src>` の内容がコピーされ、`<dest>` の HubSpot アカウントにアップロードされます。デザインツールで新しいプロジェクトを作成する場合には、`<dest>` を HubSpot の新規フォルダーにできます。

サンプルプロジェクトを HubSpot アカウントにアップロードする例を以下に示します。

```
$ local-cms-dev npx hs upload --mode=draft cms-project cms-project
```

これにより変更内容のドラフトバージョンがアップロードされるので、デザインツールでこの下書きバージョンをプレビューできます。この同期コマンドから変更を公開する場合は、`--mode=publish` オプションを使用するか、または `--mode` オプションを省略します。

5.ローカルプロジェクトを監視し、変更内容を自動的にアップロードする

`npx hs watch --portal=<name> <src> <dest>` を実行してローカルディレクトリを監視し、ファイルの保存時に変更内容を HubSpot アカウントに自動的にアップロードします。

`watch` コマンドを使用してこのコマンドを実行する例を以下に示します。

```
$ local-cms-dev npx hs watch --mode=draft cms-project cms-project
```

その他のコマンドについては、[Local Development Tooling: リファレンス](#)をご覧ください。

ローカル開発リファレンス

Local Development Tooling [ベータ]:リファレンス

これはベータ版リリースです。当社の [Developer Beta Terms](#) (デベロッパーベータ版利用規約) をご覧ください。サポートが必要な場合には、[デザイナーおよびデベロッパー Slack](#) の #local-dev-beta チャンネルにご参加ください。

ここでは、HubSPot の CMS Local Development Tooling で使用できるコマンドとファイルの形式について概説します。これらの新しいツールの詳細については、[はじめに](#)をご覧ください。

注:必要に応じて [Yarn](#) も利用できます。[Homebrew](#) は必須ではありませんが、インストールを容易にするために推奨されます。Yarn を使用している場合は、コマンドは `yarn` プレフィックス付きで実行されます。

i 無料の CMS デベロッパーサンドボックスを開発環境として利用できることはご存知ですか? アカウントを今すぐ作成するには、[ここで](#) [アカウントの作成] をクリックし、ユーザーを選択してから [新規アカウントを作成] を選択してください。 **i**

インストールまたはアップグレード

HubSpot ツールをグローバルにインストールするには、コマンドラインで `npm install -g @hubspot/cms-cli` を実行します。ツールを現在のディレクトリにインストールするには、`npm install @hubspot/cms-cli` を実行します。

ローカルツールのバージョンをアップグレードするには、`npm install -g @hubspot/cms-cli@latest` を実行します。

コマンド

help

すべてのコマンドを表示します。

```
npx hs help
```

fetch

ファイル、またはディレクトリとその子フォルダーおよびファイルをパス別に取得します。

これにより、ファイルが HubSpot アカウントからローカル環境に取得されます。`<src>`は HubSpot デザインツールのパスです。`<dest>`はファイルの追加先ローカルディレクトリです。

```
npx hs fetch --portal=<name> <src> [dest]
```

必須の引数

src

HubSpot デザインツールのパス

オプションの引数

dest

ファイルを保存するローカルディレクトリのパス (現在の作業ディレクトリを基準にした相対パス)。この引数を省略すると、既定で現在の作業ディレクトリになります。

オプション

--portal

取得元の portalId または名前を指定します。

--overwrite

取得したファイルで既存のファイルを上書きします。

--mode

HubSpot からファイルの下書きバージョンと公開バージョンのいずれを取得するかを指定します。

upload

新規ローカルアセットを HubSpot アカウントにアップロードします。変更内容は即時に公開されます。

このコマンドの `<src>` は、現在の作業ディレクトリを基準にしたローカルファイルの相対パスです。 `<dest>` は HubSpot アカウントのファイルのパスです。新規フォルダーを指定できます。

```
npx hs upload --portal=<name> <src> <dest>
```

必須の引数

src

ローカルファイルのパス (現在の作業ディレクトリを基準にした相対パス)。

dest

HubSpot デザインツールのパス。新規パスを指定できます。

オプション

--portal

取得元の portalId または名前を指定します。

--mode

アップロードされたファイルを HubSpot で公開するかどうかを指定します。

sync

このコマンドはリリース 0.0.21 では廃止されています。代わりに `upload` を使用してください。

watch

ローカルディレクトリを監視し、ファイルの保存時に変更内容を HubSpot アカウントに自動的にアップロードします。**変更内容は即時に公開されます。**

このコマンドの `<src>` は、現在の作業ディレクトリを基準にしたローカルファイルの相対パスです。`<dest>` は HubSpot アカウントのファイルのパスです。新規フォルダーを指定できます。

注：`watch` プロセスでは、確認された削除済みファイルは HubSpot アカウントから自動的に削除されます。

```
npx hs watch --portal=<name> <src> <dest>
```

必須の引数

src

ファイルが含まれているローカルディレクトリのパス (現在の作業ディレクトリを基準にした相対パス)。

dest

HubSpot デザインツールのパス。新規パスを指定できます。

オプション

--portal

取得元の portalId または名前を指定します。

--mode

ローカルの変更内容を HubSpot で公開するかどうかを指定します。

create

新規アセットのフォルダー/ファイルを作成します。サポートされている type は `module` と `template` です。

```
npx hs create <type> <name> [dest]
```

必須の引数

type

アセットのタイプ。module または template のいずれかでなければなりません。

name

新規アセットの名前

オプションの引数

dest

新規アセットの宛先フォルダー (現在の作業ディレクトリを基準にした相対パス)。この引数を省略すると、既定で現在の作業ディレクトリになります。

remove

HubSpot アカウントからファイル、またはフォルダーとその中のファイルを削除します。このコマンドには `rm` というエイリアスがあります。

```
npx hs remove --portal=<name> <dest>
```

必須の引数

dest

HubSpot デザインツールのパス

オプション

--portal

ファイルを削除する portalId または名前を指定します。

auth

[OAuth2](#) を使用して HubSpot の認証を生成します。これにより `hubspot.config.yml` ファイルが更新されます。

```
npx hs auth oauth2
```

ローカルテンプレートの形式

HubSpot テンプレートアセットのオプションは、`.html` ファイルの上部にアノテーションを使用して定義されています。

アノテーション	説明	値
---------	----	---

templateType	ファイルのテンプレートタイプを指定します。	page, email, blog, error_page, password_prompt_page, email_subscription_preferences_page, email_backup_unsubscribe_page, email_subscriptions_confirmation_page, search_results_page
isAvailableForNewContent	コンテンツ作成プロセスでテンプレートを選択できるかどうかを指定します。	true, false

例:

```
<!--
  templateType: page
  isAvailableForNewContent: false
-->
<!doctype html>
<html>
...
```

ローカルモジュールの形式

ローカルモジュールは一連のファイルで形式設定されており、各ファイルではモジュールの各種コンポーネントが指定されます。ローカルでのモジュール開発については、[ローカルモジュール開発リファレンス](#)をご覧ください。

.hsignore

プロジェクトに .hsignore ファイルを含めることで、CLI を使用する際にトラッキングしないファイルを指定できます。upload または watch コマンドを使用するときに、.hsignore で指定されたパターンと一致するファイルは HubSpot にアップロードされません。

モード

--mode オプションを使用すると、ローカルの変更内容を HubSpot のアップロードするときに公開するかどうかを決定できます。このオプションは各コマンドで使用でき、hubspot.config.yml ファイルで既定として設定できます。

--mode には --mode=draft と --mode=publish の 2 つのオプションがあります。

--mode の設定の優先順を以下に示します。

1. コマンドで --mode を使用すると、その他のすべてのオプションがオーバーライドされます。
2. hubspot.config.yml ファイルで各ポータルを defaultMode を設定すると、各コマンドで --mode を使用する必要がなくなります。これにより最上位の設定がオーバーライドされます。

3. `hubspot.config.yml` ファイルの最上位レベルで `defaultMode` を設定すると、すべてのポータルに対して既定の `--mode` が設定されます。これにより既定の動作がオーバーライドされます。
4. `--mode` の既定の動作は `publish` です。

認証

`hubspot.config.yml` ファイルには、HubSpot アカウントの資格情報が含まれています。セットアップについては、[はじめに](#)をご覧ください。

```
defaultPortal: 'DEV'  
  
defaultMode: 'draft'  
  
portals:  
  - name: 'DEV'  
    portalId: 123  
    defaultMode: 'publish'  
    authType: 'apikey'  
    apiKey: 'xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx'
```

最上位レベルのパラメータ

これらのオプションパラメータは、コマンドを簡略化する目的で使用され、ファイル内のすべてのポータルに適用されます。

名前	説明
----	----

defaultPortal	操作するポータルを指定する--portal=を使用してコマンドを実行する必要がなくなります。コマンドの実行時に--portal=が指定されていない場合には defaultPortal が使用されます。
allowUsageTracking	HubSpot にローカル開発ツールの使用状況をトラッキングする権限を許可するかどうかを指定します。既定では true です。すべてのトラッキングを無効にするには false を設定します。

portals: パラメータ

これらのパラメータは、特定のポータルの認証設定を設定します。

名前	説明
portalId	認証をセットアップするポータルのアカウント ID。
authType	特定エントリーで使用される認証の形式。api キーを使用して認証を手動で入力する場合は、'apikey'を設定します。OAuth2 認証エントリーを自動的に作成する方法の詳細については、 auth コマンド を参照してください。

apikey	<u>認証をセットアップするポータル</u> の apikey 。
name	(オプション) アカウント名を指定するためのパラメータです。コマンドが操作するアカウントを指定するときに使用できます。たとえば、上記の例では--portal=123 と--portal=DEV はいずれもアカウント ID 123 を操作します。

ローカルモジュール開発

Local Development Tooling [ベータ]:ローカルモジュール開発

これはベータ版リリースです。当社の [Developer Beta Terms](#) (デベロッパーベータ版利用規約) をご覧ください。サポートが必要な場合には、[デザイナーおよびデベロッパー Slack](#) の#local-dev-beta チャンネルにご参加ください。

ここでは、モジュールをローカルで開発するために利用可能な属性とフィールドマークアップの概要を説明します。利用可能なコマンドについては、[ローカルモジュール開発](#)

[リファレンス](#)をご覧ください。これらの新しいツールの詳細については、[はじめに](#)をご覧ください。

i 無料の CMS デベロッパーサンドボックスを開発環境として利用できることはご存知ですか？ アカウントを今すぐ作成するには、[ここで](#) [アカウントの作成] をクリックし、ユーザーを選択してから [新規アカウントを作成] を選択してください。 **i**

ローカルモジュールのファイル構造

ローカル[モジュール](#)は一連のファイルで形式設定されており、各ファイルではモジュールの各種コンポーネントが指定されます。これらのファイルは、.module ディレクトリに保存されます。[作成コマンド](#)を使用してモジュール全体の雛形を作成できます。

```
▼ local_module.module
  {} fields.json
  {} meta.json
  # module.css
  <> module.html
  JS module.js
```

E メール対応モジュールの場合、module.css と module.js は無効になります。

ファイル名	内容
module.html	モジュールの HTML と HubL。

module.css	モジュールの CSS。
module.js	モジュールの JavaScript。
meta.json (必須)	モジュールに関するメタ情報を格納する JSON オブジェクト。
fields.json (必須)	モジュールフィールドを格納する JSON オブジェクト。
_locales	言語別のローカルサブフォルダーを格納するフォルダー。サブフォルダーごとに、モジュールのフィールドの翻訳を含む messages.json ファイルが格納されます。

meta.json

meta.json ファイルには、モジュールの配置、モジュールに付加されたアセット、利用可能な機能に関する情報が含まれています。

名前	データタイプ	説明	既定
css_assets	list	モジュールで使用されている CSS アセットオブジェクトの番号付きリスト。マーケットプレイスからモジュールが配信されるときに一緒にコピーされます。	null
external_js	list	モジュールに追加された外部 JS ファイルのリスト。	null

global	boolean	ブール値。true に設定されている場合、このモジュールがデザインツールでグローバルモジュールとして分類されていることを意味します。	false
host_template_types	list	ホストテンプレートのタイプのリスト。既定では、モジュールを E メール、ブログ、ページとして使用できることを意味する"ANY"に設定されます。	["ANY", "PAGE", "BLOG_POST", "BLOG_LISTING", "EMAIL"]

js_assets	list	JavaScript アセットオブジェクトの番号付きリスト。マーケットプレイスからモジュールが配信されるときに一緒にコピーされます。	null
other_assets	list	js アセットと css アセットと同様。配布用のモジュールにファイルマネージャーアセットを組み込むために使用されます。	null

smart_type	string	<p>文字列。スマート機能を使用してモジュールが作成されているかどうかを指定します。非グローバルモジュールでは "NOT_SMART" 以外には設定できません。</p>	"NOT_SMART"
tags	list	<p>テンプレートエディターおよびマーケットプレイス内でモジュールを分類するために使用されるタグのリスト。</p>	null

is_available_for_new_content	boolean	HubSpot のモジュールエディターの右上隅にあるトグルの値。モジュールをコンテンツで使用可能かどうかを指定します。	false
------------------------------	---------	---	-------

fields.json

コンテンツ作成者がコンテンツエディター内で編集可能なフィールドを生成するモジュールフィールドは、モジュールの `fields.json` ファイル内で設定されます。

すべてのモジュールフィールドの属性

以下の属性はすべてのモジュールフィールドで使用できます。

属性名	データタイプ	説明	既定値
type	string	フィールドのタイプ (コード名)。すべてのフィールドタイプのドキュメントについては、以下の フィールドタイプ をご覧ください。	

id	string	フィールドの固有 ID。これは HubSpot によって生成されます。	ローカルで開発する場合は必須ではありません。
label	string	フィールドのラベル。これはコンテンツエディターに表示されます。	リッチテキストフィールド、日付フィールドなど。
locked	boolean	フィールドがコンテンツエディターで編集可能かどうかを指定します“true”の場合、フィールドはコンテンツエディターに表示されません。	false
helptext	string	コンテンツ作成者を支援するためにエディターに表示されるテキスト。	none

name	string	フィールド値の変数名。 HTML、CSS、JavaScript の各セクションで、 module.name を使用して アクセスできます。	richtext_field、 date_field など。
required	boolean	エディターでフィールドを 空白のままにできるかどう かを指定します。true の場 合、フィールドに値を入力 しなければコンテンツを公 開できません。	false
visibility	array	フィールドの表示条件を指 定します。	
default	varies	フィールドの既定値。以下 のフィールドスキーマを参 照。	

リピーターの使用

ローカルで開発するフィールドでリピーターオプションを使用するには、`occurrence` 属性を含めます。

属性名	データタイプ	説明	既定値
min	number	繰り返しフィールドの最小出現回数。	null
max	number	繰り返しフィールドの最大出現回数。	null
sorting_label_field	string	リピーターオプションを使用したフィールドグループでは、この属性により、エディターに表示されるフィールド値が決まります。	null
default	number	初めてモジュールをコンテンツに追加する際にフィールドが出現する既定の回数。	null

以下は、`occurrence` 属性の例です。

```
"occurrence" : {  
  "min" :1,  
  "max" :5,  
  "sorting_label_field" : null,  
  "default" : null  
}
```

フィールドタイプ

日付

"date"

カレンダーの日付を選択します。

```
{  
  "name" : "date_field",  
  "label" : "Date field",  
  "help_text" : "Select a date.",  
  "sortable" : false,  
  "required" : false,  
  "locked" : false,  
  "step" :1,  
  "type" : "date",  
  "default" :1566360000000  
}
```

日付と時刻

"datetime"

カレンダーの日付と時刻を選択します。

属性名	データタイプ	説明	既定値
step	number	時刻を前後に切り替える際の既定の間隔（分数）。	30

```
{
  "name" : "datetime_field",
  "label" : "Date and time field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "step" : 30,
  "type" : "datetime",
  "default" : 1566360000000
}
```

リンク

"link"

<a>タグ内にリンクを生成します。

属性名	データタイプ	説明	既定値
-----	--------	----	-----

supported_types	list	<p>コンテンツエディターから追加できるリンクのタイプを指定します。既定では、すべてのタイプが有効にされます。リストからタイプを削除すると、そのタイプは無効になります。</p>	<p>"EXTERNAL"、"CONTENT"、"FILE"、"EMAIL_ADDRESS"、"BLOG"</p>
-----------------	------	--	---

```

{
  "name" : "link_field",
  "label" : "Link field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "supported_types" : [ "EXTERNAL", "CONTENT", "FILE", "EMAIL_ADDRESS",
"BLOG" ],
  "type" : "link",
  "default" : {
    "url" : {
      "type" : "EXTERNAL",
      "href" : "www.hubspot.com",
      "content_id" : null
    },
    "open_in_new_tab" : false,
    "no_follow" : false
  }
}

```

数値

"number"

数値を選択する機能を追加します。

属性名	データタイプ	説明	既定値
min	number	フィールドの最小値を設定します。	none
max	number	フィールドの最大値を設定します。	none
default	number	フィールドの既定値を設定します。	none
step	number	上下のボタンをクリックする際に値を増減する単位を指定します。	none

```
{
```

```
"name" : "number_field",
"label" : "Number field",
"sortable" : false,
"required" : false,
"locked" : false,
"display" : "text",
"min" : 1,
"max" : 5,
"step" : 1,
"type" : "number",
"default" : 1
}
```

リッチテキスト

"richtext"

リッチテキストの WYSIWYG 領域。テキストフィールドとは異なり、リッチテキストフィールドではさまざまなテキストスタイルを使用できます。また、各種フォームのコンテンツ (CTA、画像、動画など) を挿入するオプションもあります。

```
{
  "default": "<h1>Hello, world!</h1>",
  "label": "Rich text field",
  "locked": false,
  "name": "text",
  "required": false,
  "sortable": false,
  "type": "richtext",
```

```
}
```

シンプルメニュー

"simplemenu"

コンテンツエディター内で一からメニューを作成できます。

```
{  
  "name" : "simplemenu_field",  
  "label" : "Simple menu field",  
  "sortable" : false,  
  "required" : false,  
  "locked" : false,  
  "type" : "simplemenu",  
  "default" : [ {  
    "isPublished" : true,  
    "pageLinkId" : 123456789,  
    "pageLinkName" : "My page",  
    "isDeleted" : false,  
    "categoryId" : 1,  
    "subCategory" : "site_page",  
    "contentType" : "site_page",  
    "state" : "PUBLISHED_OR_SCHEDULED",  
    "linkLabel" : "This is a page",  
    "linkUrl" : null,  
    "linkParams" : null,  
    "linkTarget" : null,  
    "type" : "PAGE_LINK",  
    "children" : [ ]  
  } ]  
}
```

```
}
```

テキスト

"text"

シンプルなテキスト入力。

属性名	データタイプ	説明	既定値
placeholder	string	コンテンツ作成者のエディターには表示される一方、ページには出力されないテキスト。	none
allow_new_line	boolean	テキストフィールド入力に新しい行を追加できるかどうかを指定します。	false

show_emoji_picker	boolean	アプリ内のリッチテキストツールバーに絵文字ピッカーを表示するかどうかを指定します。	false
-------------------	---------	---	-------

```

}
  "name" : "text_field",
  "label" : "Text field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "validation_regex" : "",
  "placeholder" : "isn't it neat",
  "allow_new_line" : false,
  "show_emoji_picker" : false,
  "type" : "text",
  "default" : "This is default text"
}

```

URL

"url"

<a>タグの外部に完全な URL を生成します。

属性名	データタイプ	説明	既定値

supported_types	list	コンテンツエディターから追加できるリンクのタイプを指定します。既定では、すべてのタイプが有効にされます。リストからタイプを削除すると、そのタイプは無効になります。	"EXTERNAL"、"CONTENT"、"FILE"、"EMAIL_ADDRESS"、"BLOG"
-----------------	------	---	--

```
{
  "name" : "url_field",
  "label" : "URL field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "supported_types" : [ "EXTERNAL", "CONTENT", "FILE", "EMAIL_ADDRESS",
"BLOG" ],
  "type" : "url",
  "default" : {
    "type" : "CONTENT",
    "href" : null,
    "content_id" : 123456789
  }
}
```

ブール値

"boolean"

true または false の切り替え。ブール値フィールドの既定値は false です。

```
{
```



```
"name" : "boolean_field",
"label" : "Boolean field",
"sortable" : false,
"required" : false,
"locked" : false,
"type" : "boolean",
"default" : false
}
```

選択

"choice"

コンテンツエディターで一連のオプションのいずれかを選択できるフィールド。

属性名	データタイプ	説明	既定値
display	string	"select"または"radio"に設定できます。モジュールエディターでの選択フィールドの表示方法を指定します。"select"を設定するとドロップダウンメニューになり、"radio"を設定するとラジオボタンのオプションになります。	select

choices	list	フィールドオプションに入力される一連の値とラベル。ラベルにはHTML マークアップを含めることができます。	[["value 1", "Label 1"], ["value 2", "Label 2"]]
---------	------	---	--

```
{
  "name" : "choice_field",
  "label" : "Choice field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "display" : "select",
  "choices" : [[ "value 1", "Label 1" ], [ "value 2", "Label 2" ]],
  "type" : "choice",
  "default" : "value 2"
}
```

ブログ

"blog"

アカウントからブログを選択します。このフィールドはブログ ID を生成します。

```
{
  "name" : "blog_field",
  "label" : "Blog field",
  "sortable" : false,
  "required" : false,
```

```
"locked" : false,  
"type" : "blog",  
"default" :1234567890  
}
```

色

"color"

色を選択します。16 進コードと不透明度の値が返されます。

```
{  
  "name" : "color_field",  
  "label" : "Color field",  
  "sortable" : false,  
  "required" : false,  
  "locked" : false,  
  "type" : "color",  
  "default" : {  
    "color" : "#ffffff",  
    "opacity" :100  
  }  
}
```

CTA

"cta"

Call-to-Action (CTA) を選択します。アクション GUID が返されます。

```
{  
  "name" : "cta_field",
```

```
"label" : "CTA field",
"sortable" : false,
"required" : false,
"locked" : false,
"type" : "cta",
"default" : "fb9c0055-6beb-489d-8dda-3e1222458750"
}
```

E メールアドレス

"email"

アカウントからユーザーの E メールアドレスを選択します。

```
{
  "name" : "email_field",
  "label" : "Email address field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "type" : "email",
  "default" : [ "develop@hubspot.com", "design@hubspot.com" ]
}
```

ファイル

"file"

HubSpot ファイルマネージャーからファイルを選択します。

属性名	データタイプ	説明	既定値
picker	string	選択するファイルのタイプ。オプションは、"file"、"image"、"document"です。	file

```
{
  "picker" : "file",
  "name" : "file_field",
  "label" : "File field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "type" : "file",
  "default" : "https://cdn2.hubspot.net/hubfs/file.pdf"
}
```

フォローアップEメール

"followupemail"

ポータルからフォローアップEメールを選択します。カスタムモジュールで完全なフォームをレンダリングする場合は、このフィールドが特に役立ちます。

```
{
  "name" : "followupemail_field",
  "label" : "Followup email field",
```

```
"sortable" : false,  
"required" : false,  
"locked" : false,  
"type" : "followupemail",  
"default" :1234567890  
}
```

フォント

font

モジュールで使用する Google フォントを選択します。

```
{  
  "name" : "font_field",  
  "label" : "Font field",  
  "sortable" : false,  
  "required" : false,  
  "locked" : false,  
  "load_external_fonts" : true,  
  "type" : "font",  
  "default" : {  
    "size" :12,  
    "size_unit" : "px",  
    "color" : "#000",  
    "styles" :{  
      "text-decoration" : "underline"  
    },  
    "font" : "Alegreya",  
    "fallback" : "serif",  
    "variant" : "regular",  
  },  
}
```

```
"font_set" : "GOOGLE"  
  }  
}
```

フォーム

"form"

HubSpot アカウントからフォームを選択します。

```
{  
  "name" : "form_field",  
  "label" : "Form field",  
  "sortable" : false,  
  "required" : false,  
  "locked" : false,  
  "type" : "form",  
  "default" : {  
    "form_id" : "9aa2e5f3-a46d-4774-897e-0bc37478521c",  
    "response_type" : "redirect",  
    "redirect_url" : "http://www.hubspot.com",  
    "redirect_id" : null,  
    "gotowebinar_webinar_key" : null,  
    "form_type" : "HUBSPOT"  
  }  
}
```

HubDB テーブル

"hubdbtable"

HubSpot アカウントから HubDB テーブルを選択します。テーブル ID が返されます。

```
[
{
"default": "<h1>Hello, world!</h1>",
"label": "Rich text field",
"locked": false,
"name": "text",
"required": false,
"sortable": false,
"type": "richtext",
"validation_regex": ""
}
]
```

アイコン

"icon"

FontAwesome アイコンを選択します。

```
{
  "name" : "icon_field",
  "label" : "Icon field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "type" : "icon",
  "default" : {
```



```

"name" : "align-center",
"unicode" : "f037",
"type" : "SOLID"
}
}

```

画像

"image"

HubSpot ファイルマネージャーから画像を選択します。

属性名	データタイプ	説明	既定値
responsive	boolean	レスポンシブなサイズ変更オプションをコンテンツエディターに表示するかどうかを指定します。	false
resizable	boolean	画像のサイズコントロールを HubSpot エディターに表示するかどうかを指定します。	true

```

{
  "name" : "image_field",
  "label" : "Image field",

```

```
"sortable" : false,  
"required" : false,  
"locked" : false,  
"responsive" : false,  
"resizable" : true,  
"type" : "image",  
"default" : {  
"src" : "https://cdn2.hubspot.net/hubfs/image.jpeg",  
"alt" : "an_image",  
"width" :100,  
"height" :100  
}  
}
```

ロゴ

"logo"

設定からロゴをレンダリングします。フィールドの各インスタンスで既定のロゴをオーバーライドすることもできます。

```
{  
  "name" : "logo_field",  
  "label" : "Logo field",  
  "sortable" : false,  
  "required" : false,  
  "locked" : false,  
  "resizable" : true,  
  "type" : "logo",  
  "default" : {  
    "override_inherited_src" : true,  

```

```
"src" : "https://cdn2.hubspot.net/hubfs/logo.png",  
"alt" : "best_logo_ever",  
"width" :100,  
"height" :100  
}  
}
```

ミーティング

"meeting"

HubSpot ミーティングリンクを選択します。このフィールドをモジュールに追加するには、HubSpot アカウントにセールス権限が割り当てられている必要があります。

```
{  
  "name" : "meeting_field",  
  "label" : "Meeting field",  
  "sortable" : false,  
  "required" : false,  
  "locked" : false,  
  "type" : "meeting",  
  "default" : "https://app.hubspot.com/meetings/developers-r-kewl"  
}
```

メニュー

"menu"

[設定] > [ウェブサイト] > [ナビゲーション] に組み込まれたメニューを選択します。

```
{
  "name" : "menu_field",
  "label" : "Menu field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "type" : "menu",
  "default" : 123456789
}
```

ページ

"page"

HubSpot アカウントからウェブサイトまたはランディングページを選択します。

```
{
  "name" : "page_field",
  "label" : "Page field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "type" : "page",
  "default" : 1234567890
}
```

Salesforce キャンペーン

"salesforcecampaign"

Salesforce キャンペーンを選択します。このフィールドを使用して、既定のフォームモジュールの機能を複製できます。このフィールドは、HubSpot アカウントで Salesforce 統合を使用できる場合にのみ有効になります。

```
{
  "name" : "salesforcecampaign_field",
  "label" : "Salesforce campaign field",
  "sortable" : false,
  "required" : false,
  "locked" : false,
  "type" : "salesforcecampaign",
  "default" : "7016A0000005S0tQAE"
}
```

タグ

"tag"

HubSpot ブログのタグを選択します。

属性名	データタイプ	説明	既定値
tag_value	string	タグセレクターの戻り値タイプを選択します。オプションは"ID"または"Name"です。	Name

```
{
  "name" : "tag_field",
```

```
"label" : "Tag field",
"sortable" : false,
"required" : false,
"locked" : false,
"tag_value" : "ID",
"type" : "tag",
"default" : 1234567890
}
```

ローカルモジュール開発の詳細については、[こちらの](#) HubSpot CMS ボイラープレートモジュールをご覧ください。

チュートリアル

HubSpot CRM データを CMS ページに取り込む

CMS ページへの HubSpot CRM データの取り込み

HubSpot CMS では、[`crm_object HubL` 関数](#)または [`crm_objects HubL` 関数](#)を使用して、HubSpot CRM オブジェクトからウェブサイトページに情報を直接取り込むことができ

まず、*crm_object* 関数は 1 つの CRM オブジェクトを取り込みます。この関数はページあたり最大 10 回まで呼び出すことができ、コンタクト、製品、取引、チケット、または見積もりの各オブジェクト情報をページに取り込みます。取り込む必要がある CRM オブジェクトの数が 11 以上の場合は、*crm_objects* 関数で最大 100 個までオブジェクトを取り込むことができます。この関数は、ページあたり最大 10 回まで呼び出すことができ、最大で 1000 CRM オブジェクトを取り込みます。

セキュリティ上の目的から、誰でもアクセスできるページでは Product オブジェクトだけを取得できます。その他のオブジェクトタイプは、パスワードで保護されているかまたは CMS メンバーシップログインを必要とするページでホストする必要があります。

登録を必須とするプライベートページを作成し、*crm_object* 関数を使用して、ログインしたコンタクトまたは指定されたコンタクトのデータを取り込む例を以下に示します。

1. ページを作成する

2. 新規モジュールを作成する

コンタクトレコードデータを表示するロジックを格納する新規モジュールを作成します。

Set up your new module



[← Back to file types](#)

New file type

✕ Module

Where would you like to use this module? *

Page templates

Blog templates

Email templates

Module content scope

Local module Global module ⓘ

File name *

Contact Data Module

File location

 (in root folder)

[Change](#)

Create

Cancel

crm_object HubL 関数に、データのレンダリング対象として指定されたコンタクトを渡すか、またはこの HubL 関数がログインコンタクトのデータをレンダリングできます。この例のモジュールでは、コンタクトレコードが、特定のコンタクトまたはログインコンタクトのいずれのレコードであるかを、マーケターが指定できます。

このモジュールには次の2つのフィールドがあります。

1. `requesting_contact` (ブーリアン) - 表示するコンタクトデータが、ログインコンタクトまたは指定のコンタクトのデータであるかどうかを指定します。このフィールドは必須であり、既定ではオンになっています。
2. `specified_contact_email` (テキスト) - ログインコンタクトのデータを表示しない場合に、データを表示するコンタクトをそのEメールで指定します。このフィールドは、条件 (`requesting_contact` が `false` である場合) に基づいて表示される必要があります。

ページエディターでの選択に基づいてデータをレンダリングするロジックは、以下のようになります。

```
{% if module.requesting_contact %}
  {% set membership_contact = crm_object('contact', request.contact.contact_vid,
'firstname,lastname') %}
  Logged-in contact name: {{ membership_contact.firstname }}
  {{ membership_contact.lastname }}
{% else %}
  {% set query = 'email=' ~ module.specified_contact_email %}
  {% set specified_contact = crm_object("contact", query, "firstname,lastname") %}
  Specified contact name: {{ specified_contact.firstname }}
  {{ specified_contact.lastname }}
{% endif %}
```

3.モジュールをページに追加する

テンプレートレベルまたはページレベルで、新規に作成したモジュールを追加します。条件文をモジュールのフィールドに追加した場合は、編集作業は以下のようになります。

Contact Data Module

 Options

Cont Specifies if the contact data to display is for the logged-in contact, or a specified contact.

Requesting Contact *

Save and go to module list

Apply

訪問するコンタクトに対してデータを表示する目的でページを作成する場合は、表示するコンタクトデータを選択するためにログインコンタクトを使用することをお勧めします。指定コンタクトを使用する場合、ページにアクセスできるコンタクトが他のコンタクトのデータを閲覧できる可能性があります。

注:指定のコンタクトに一致するコンタクトが HubSpot CRM 内にない場合、使用可能なコンタクトデータがありません。

注:公開中のページでのみデータを表示でき、ページのプレビューやページエディターでは表示できない場合は、ログインコンタクトを使用します。

4. ページのオーディエンスアクセスルールを設定する

[「ページへのオーディエンスアクセスを制御」](#)の説明に従い、このページを [非公開 - 登録が必要] にします。注：これにより、このページへのアクセス権が付与されているすべてのコンタクトに登録メールが送信されます。

複数のブログをリストに統合する

複数のブログ記事を 1 つのリスト に統合する

デザイナーは、[HubL 関数を使用して一連の記事を取得し、ページテンプレートに記事の変数を印刷する](#)ことができます。複数のブログがある場合は、一連の記事を取得し、1 つの並べ替えたリストに組み込むことができます。Wordpress のようなプラットフォームで複数の記事タイプを管理し、記事をフィードに集約することに慣れている開発者は、以下の手法を使用して同じような機能を実現できます。

この手法はページ番号に対応していないことに注意してください。HubSpot のページ番号設定は、標準のブログリストにのみ適用されます。HubL 関数で表示されるコンテンツにページ番号を付ける必要がある場合は、JavaScript を使用して静的な記事セットにページ番号を付ける必要があります。

ここでは、2 つの異なるブログに投稿された 5 件の最新記事を 1 つのリストに統合します。この手法は任意のページテンプレートで使用できます。この手法を実装するには、次の手順を実行します。

- 変数を定義して[記事セット](#)を保存します
- 2つ目の記事セットを保存する2つ目の変数を定義します
- 2つの変数を統合する3つ目の変数を定義し、[並べ替えフィルター](#)を適用します。以下の例では公開日で並べ替えています。 「name」などの他のプロパティに基づいて並べ替えることもできます。
- 統合変数内の記事をループ処理します
- ループ内で[ブログ変数を印刷](#)します*

```

<!-- sets a variable for a the 5 most recent posts of my default CMS blog -->
{% set blog_one_posts = blog_recent_posts('default', limit=5) %}

<!-- sets a variable for a the 5 most recent posts of my blog id 47082700. You can find
the blog id in the URL of the blog dashboard for a particular blog-->
{% set blog_two_posts = blog_recent_posts('47082700', limit=5) %}

<!-- combines the two variables into a single variable and sorts them by publish
date. Please note the sort filter requires all three parameters -->
{% set all_posts = (blog_one_posts + blog_two_posts) | sort(true, false,
'publish_date') %}

<!-- loop through posts in custom all posts variable -->
{% for post in all_posts %}
  <div class="post-item">
    {% if not simple_list_page %}
      <div class="post-header">
        <h2><a href="{{post.absolute_url}}">{{ post.name }}</a></h2>
        <div id="hubspot-author_data" class="hubspot-editable" data-hubspot-
form-id="author_data" data-hubspot-name="Blog Author">
          Posted by
          {%if post.blog_post_author %}

```

```

        <a class="author-link" href="
{{ post.parent_blog.absolute_url }}/author/{{ post.blog_post_author.slug }}">{{ post.blog_
post_author.display_name }}</a> on {{ post.publish_date_localized }}
    {% endif %}
</div>
    {% blog_social_sharing "blog_social_sharing" overrideable=False,
label='Blog Social Sharing' %}
</div>
<div class="post-body clearfix">
<!--post summary-->
{% if post.post_list_summary_featured_image %}
<div class="hs-featured-image-wrapper">
    <a href="{{post.absolute_url}}" title="" class="hs-featured-image-
link">
        
    </a>
</div>
{% endif %}
    {{ post.post_list_content|safe }}
</div>
{% if content_group.show_summary_in_listing %}
<a class="more-link" href="{{ post.absolute_url }}">Read More</a>
{% endif %}
<div class="custom_listing_comments">
    {% set comments_number = post.comment_list|length %}
    {% set comments_label = "Comment" if comments_number == 1 else
"Comments" %}
    {{ comments_number }} {{ comments_label }} <a
href="{{post.absolute_url}}#comments-listing">Click here to read/write comments</a>
</div>

```

```

    {% if post.topic_list %}
    <p id="hubspot-topic_data"> Topics:
    {% for topic in post.topic_list %}
        <a class="topic-link" href="
    {{ post.parent_blog.absolute_url }}/topic/{{ topic.slug }}</a>{% if not loop.last %},{%
    endif %}
    {% endfor %}
    </p>
    {% endif %}
    {% else %}
        <h2 class="post-listing-simple"><a
    href="{{post.absolute_url}}">{{ post.name }}</a></h2>
    {% endif %}
</div>
{% endfor %}

```

*for ループ内のコードは、さまざまな変数を独自の集約された記事セットに関連付けるため、[HubSpot の標準的なリスト](#)の「content」を「post」に置き換えたものです。

HubDB で動的ページを作成する方法

HubDB で動的ページを作成する

動的ウェブサイトページとは、エンドユーザーが要求する URL のパスに基づいてコンテンツが変化する CMS ページです。HubDB ではすでに、データの保存、絞り込み、HubSpot ウェブサイトページでの表示が可能です。HubDB では、動的ページを使用してテーブルのすべての行のページを作成できます。各動的ページには独自の SEO 対応

URL があり、ページ固有のアナリティクスが提供されます。詳細な HubDB ドキュメントは[ここで参照](#)できます。

以下のものがが必要です。

- [HubSpot CMS](#)
- ページをカスタマイズするには、HubSpot の CMS、HTML、および CSS に関する予備知識が必要です。

動的ページを使用開始する手順を以下に示します。

1. HubDB テーブルを作成する

[HubDB](#) に移動し、[テーブルを作成] ボタンをクリックします。テーブルに分かりやすい名前を付けます。[設定] ボタンをクリックし、[行データを使用して動的ページを作成できるようにする (Enable creation of dynamic pages using row data)] を有効にします。

Manage settings



Allow public API access

Enable creation of dynamic pages using row data

[Learn how to use HubDB to create dynamic pages](#)

Meta data

Meta data for the dynamic pages created by this table will be sourced from the columns you choose below.

Meta description column

Select

Featured image column

Select

Canonical URL column

Select

Child tables

Allow use of child tables

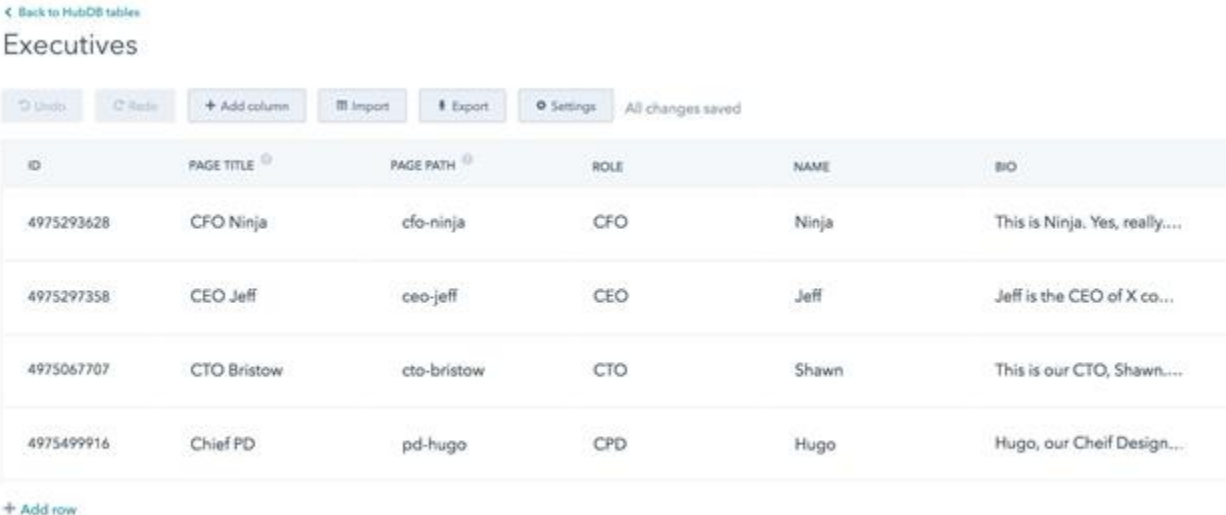
Automatically create listing pages for child tables

Update

[保存] をクリックします。テーブルに [ページタイトル] と [ページのパス] という 2 つの追加列が表示されます。また、個別の動的ページのサムネイル画像、メタディスクリプション、正規 URL の値を指定するための列を選択できます。

タイトルは、HTML title タグに指定されているこのページの名前です。パスは、テーブルの各行により作成される動的ページの URL の最後のセグメントです。URL のその他の部分は後で付加できます。サムネイル画像、メタディスクリプション、正規 URL の列を選択しない場合、または特定の行のセルをブランクのままにする場合には、動的ページは親ページからこれらの値を継承します。

テーブルには必要なあらゆるデータを追加できます。経営幹部のテーブルの例を以下に示します。



The screenshot shows a table management interface for a table named "Executives". At the top left, there is a link "Back to HubDB tables". Below the table name, there are several action buttons: "Undo", "Redo", "+ Add column", "Import", "Export", and "Settings". A status message "All changes saved" is displayed on the right. The table itself has six columns: "ID", "PAGE TITLE", "PAGE PATH", "ROLE", "NAME", and "BIO". There are four rows of data:

ID	PAGE TITLE	PAGE PATH	ROLE	NAME	BIO
4975293628	CFO Ninja	cfo-ninja	CFO	Ninja	This is Ninja. Yes, really....
4975297358	CEO Jeff	ceo-jeff	CEO	Jeff	Jeff is the CEO of X co...
4975067707	CTO Bristow	cto-bristow	CTO	Shawn	This is our CTO, Shawn....
4975499916	Chief PD	pd-hugo	CPD	Hugo	Hugo, our Cheif Design...

At the bottom left of the table, there is a "+ Add row" button.

この例では、“jane”、“jeff”、“justine”、および“gregory”で終わるパスを持つ動的ページが作成されます。

次のステップに進む前に、テーブルを必ず公開してください。

2. テンプレートを作成する

デザインマネージャーに移動し、新しい"HTML & HUBL テンプレート"を作成します。

Set up your new HTML & HUBL template



[< Back to file types](#)

New file type

 HTML & HUBL

What are you building?

Template Template partial 

Template type

Page 

File name *

dynamic_page

.html

File location

 (in root folder)

[Change](#)

Create

Cancel

名前は自由に付けることができます。

ブログテンプレートをリスティングテンプレートおよび記事詳細テンプレートの両方として使用できることと同様に、このテンプレートはすべての行の一覧を表示するリスティングページであり、かつ各行の詳細ページでもあります。動的ページがこのテンプレートを使用するように設定されており、URL の終わりがパス列と一致する場合は、`dynamic_page_hubdb_row` および `dynamic_page_hubdb_table_id` 変数がテンプレートで

使用できます。行の詳細を表示するコードは非常にシンプルです。直前に以下のコードを挿入します。

```
{% if dynamic_page_hubdb_row %}
  <h1>{{ dynamic_page_hubdb_row.hs_name }}</h1>
  <h2>{{ dynamic_page_hubdb_row.name }}</h2>
  <h3>{{ dynamic_page_hubdb_row.role }}</h3>
{% endif %}
```

H1 タグによりページのタイトルが表示されます。これはまた、HTML title タグに自動的に挿入されます。

H2 タグにより経営幹部の名前が表示されます。

H3 タグにより役職が表示されます。

上記のテーブルの例とは異なる名前を列に指定している場合は、これに合わせて属性名を調整する必要があります。

次に、(テーブルからの追加パスなしで) 動的ページが読み込まれた場合のケースが処理されます。通常これは、HubDB の行のページへのリンクをリストするリストページとして使用されます。コードを以下の内容に置き換えます。

```
{% if dynamic_page_hubdb_row %}
  <h1>{{ dynamic_page_hubdb_row.hs_name }}</h1>
  <h2>{{ dynamic_page_hubdb_row.name }}</h2>
  <h3>{{ dynamic_page_hubdb_row.role }}</h3>
{% elif dynamic_page_hubdb_table_id %}
  <ul>
    {% for row in hubdb_table_rows(dynamic_page_hubdb_table_id) %}
      <li><a href="{{ request.path }}/{{ row.hs_path }}">{{ row.hs_name }}</a></li>
    {% endfor %}
  </ul>
{% endif %}
```

```
{% endfor %}
</ul>
{% endif %}
```

elif ブロック内のコードは、経営幹部のテーブルのすべての行を繰り返し処理し、リストの各エントリとその一意のパスへのリンクをリストに表示します。

このテンプレートは、ページのコンテンツを使用して `dynamic_page_hubdb_row` 変数または `dynamic_page_hubdb_table_id` 変数を設定するため、このテンプレートのプレビューには何も表示されません。

テンプレートをテストするには、コードの上に以下の行を一時的に追加します。テンプレートを公開する前に、追加した行を必ず削除してください。

```
{% set dynamic_page_hubdb_table_id = <YOUR_HUBDB_TABLE_ID> %}
```

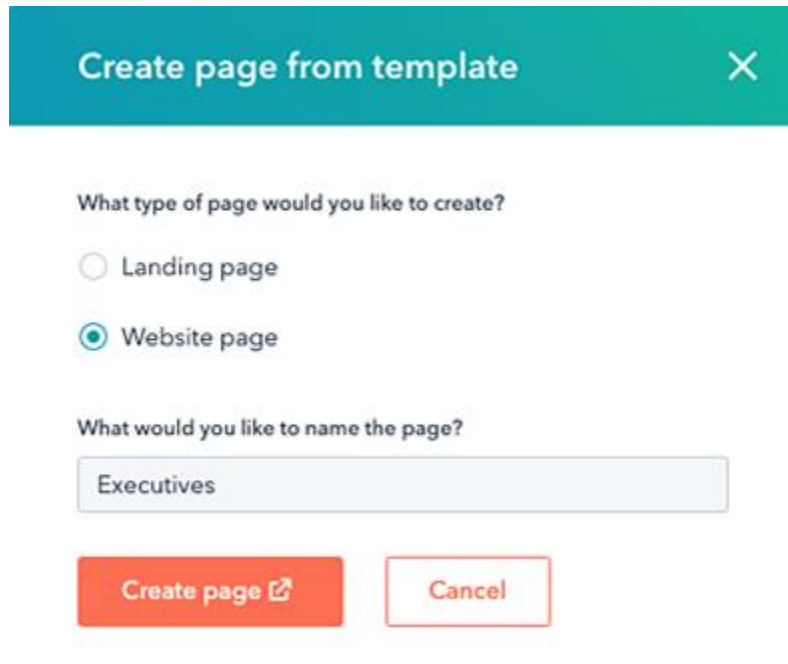
プレビューには次のような内容が表示されます。

- CFO Ninja
- CEO Jeff
- CTO Bristow
- Chief PD

テスト行を追加している場合はテスト行を削除します。最後にテンプレートを公開して、[このテンプレートを新しいコンテンツで利用できるようにする] がオンになっていることを確認します。

3.動的ページを作成する

テンプレートから動的ページを作成する最も簡単な方法は、デザインマネージャーを使用する方法です。テンプレートを選択した状態で、[アクション]メニューから [ページを作成] を選択し、ウェブサイトページにして名前を付けます。



Create page from template X

What type of page would you like to create?

Landing page

Website page

What would you like to name the page?

Executives

Create page ↗

Cancel

新しいページで [設定] タブをクリックします。ページのタイトルと URL を指定します。この URL は動的ページのベース URL になります。

下にスクロールして [詳細オプション] をクリックします。[動的ページ用のテーブル] ドロップダウンメニューで、以前に作成した HubDB テーブルを見つけて選択します。

✓ Advanced Options

Expire page on a specific date

Control audience access for page

Control the privacy of this page and how visitors can access it

- Public (default)
- Private - Password required ⓘ
- Private - Registration required ⓘ

Additional code snippets

Head HTML ⓘ

1

Footer HTML ⓘ

1

HubDB

Table for dynamic pages ⓘ

Executives ▼

ページの設定 UI でベースページの詳細を入力します。ベースページのページタイトル値とメタディスクリプションを入力する必要があります。完了したら [公開] をクリックします。これでページを表示できます。

4. 動的ページのパフォーマンスを確認する

HubDB テーブルで定義されているように、新しい動的ページとそのすべてのパスを訪問できます。

ページの URL を <http://your.company.com/executives> に設定した場合、URL が <http://your.company.com/executives> のページに、経営幹部の一覧が表示されます。URL が <http://your.company.com/executives/jane> のページには、CEO である Jane の詳細が表示されます。

これらの URL は、(アナリティクス処理完了後に) ウェブサイトページダッシュボードの分析セクションに表示されます。個別の URL と訪問数がリストに表示され、タイトルをクリックして詳細情報とアナリティクスを確認できます。レポート設定で自分の IP を除外している場合に、ページが分析に表示されるようにするため、開発ネットワークの外部からページにアクセスする必要があります。

HubDB で動的チームメンバーページを作成する方法

HubDB を使用して動的チームメンバーページを作成しよう

このチュートリアルでは、HubDB を使用して HubSpot の CMS で動的チームメンバーページを作成する方法を説明します。詳細な HubDB ドキュメントは[ここで](#)参照できます。

以下のものがが必要です。

- CMS アドオンがある HubSpot アカウント
 - *Website Starter* があるポータル、または *Website* アドオンがないポータルでは、HubDB はサポートされていません。
 - [ここからサインインして](#)、ポータルに CMS アドオンがあるかどうかを確認できます。
- 約 3 時間
- ページをカスタマイズするには、HubSpot の CMS、HTML、および CSS に関する予備知識が必要です。


このチュートリアルを完了すると、次の例に示すように、リクエストしているページの URL に基づく実用的な動的チームメンバーページが完成します：[動的チームメンバーの例](#)

1. HubDB チームメンバーテーブルを作成する

このチュートリアルは、リクエストされた CMS ページの URL に基づいてチームメンバーのページを動的に作成するためのロジックを追加することに重点を置いています。HubDB を初めて使う方や、最初の HubDB チームメンバーテーブルの生成についてサポートが必要な方は、こちらのチュートリアルをご覧ください：[静的チームメンバーページ](#)

HubDB テーブルの作成に慣れている場合は、次のファイルを使用して、この例を実行するためのテストデータをインポートできます：[team-member.csv](#)。この例に沿って作業を進めるため、このファイルを HubDB にアップロードします。

完了したら、必ずテーブルを公開し、テーブルの ID (URL の最後の番号) をメモしてください。これで、データを使い始める準備が整います。



app.hubspot.com/hubdb/Portal ID/table/1058586

2. カスタムモジュールを作成する

さて、すばらしいチームメンバーを紹介するための HubDB テーブルが生成されました。

最初に、デザインマネージャーで新しいテンプレートを (まだ作成していない場合は) 作成し、新しいカスタムモジュールを作成する必要があります。「Dynamic Team Member Module」のような名前を付けることができます。カスタムモジュールを生成するときは、ニーズに応じて、複数のテンプレートで簡単に使用できるようにしたい場

合と、カスタムフィールドを作成できるようにしたい場合があります。この例では、2つのカスタムフィールドを組み込んで、各ページで表示されるテーブルに異なるスタイルを適用できるようにします（後ほど詳しく説明します）。必ず新しいカスタムモジュールを必要なテンプレートに追加してください。

新しいカスタムモジュールの中に以下のコードを追加します。このコードは、チームメンバーのセクションを動的にするロジックを適用するためのフレームワークを生成しています。

```
{# Get Team Members from HubDB #}
{% set table = hubdb_table_rows([Insert YOUR Table ID], queryparam)%}

{% if table == [] %}
  <p class='align-center'>Sorry, no one found for that Search.Try changing your filter
  and search again.</p>
{% else %}
  {% for row in table %}
<div class="team-member-card-container {{ widget.cards_in_row }}">
  <div class="team-member-card">
    <div class="image-container">
      
row.image.height %}class="landscape"{% endif %} alt="{{ row.name }}" Headshot">
    </div>
    <div class="team-member-info">
      <h3>{{ row.name }}</h3>
      <h4>{{ row.title }}</h4>
      <p>{{ row.description }}</p>
    </div>
  </div>
</div>
```

```
</div>
{% endfor %}
{% endif %}
```

テンプレートのプレビューをクリックすると、シンプルなチームリストが表示されます。何も表示されない場合は、テーブル ID が正しいこと、テーブルが公開されていること、および変数名が HubDB テーブル内で設定された名前と一致することを再確認してください。

ここで行われている処理を詳しく見てみましょう。「queryparam」については、まだ気にしないでください。これは、チュートリアルの後半でリクエストしているページの URL パスに基づいてチームメンバーテーブルを絞り込むときに必要になります。

まず、何らかの理由でテーブルから空のオブジェクトが返された場合にエラーを表示するための小さなロジックを追加しています。次に、チームメンバーの行をマッピングしています。正常に完了すれば、プレビューにチーム全体が表示されるはずです。必要に応じて、HTML 構造を自由に変更してください。

ここでは、ドット表記を使用して HubDB の行の値にアクセスしていることに注意してください。「角括弧」表記を使用するように調整することも可能です。


3. カスタムモジュールにカスタムフィールドを追加する

この例では、表示されるチームメンバーページの機能をさらに強化するため、モジュール内でカスタムフィールドを使用します。カスタムモジュールは最終的に複数のページ



に適用されるため、たとえば、ある拠点には従業員が 2 人しかいないが、別の拠点には 6 人いる場合などに、テーブルの HTML 構造を簡単に調整する必要があります。

HTML に適用されるカスタムクラスに基づいてチームメンバーのコンテナのサイズを制御する CSS を取り込むことで、これに対応できます。これは、以下のようにモジュールに選択フィールドを追加することで実現できます。

✕ > # of members in flow

of members in flow 

Choice field


 









HubL variable name Copy ▾

▼ Content options


Default

▾

Choices 

 full	<input type="text" value="1"/>	
 half	<input type="text" value="2"/>	
 third	<input type="text" value="3"/>	
 quarter	<input type="text" value="4"/>	

[+ Add another choice](#)

Placeholder 

このフィールドがウィジェットの表示レイアウトに影響を及ぼすためには、これを処理する CSS を生成する必要があります。作業を簡単にするため、すぐに使用できる CSS を用意しました：[team-member-card.css](#)

既定では、このモジュールはコンテナに「half」クラスを設定します。このため、各カードは 50%の幅で表示されます。このモジュールを使用してページを生成し、表示されるチームメンバーが少ないことがわかっている場合は、CSS を動的に調整してテーブルをより適切に表示することができます。

必ずこの CSS をカスタムモジュールに含めてください。更新されたモジュールは次のようになります。

```
{# Load Card Stylesheet #}
{{ require_css([insert link to CSS]) }}

{# Get Team Members from HubDB #}
{% set table = hubdb_table_rows([Insert YOUR Table ID], queryparam)%}

{% if table == [] %}
  <p class='align-center'>Sorry, no one found for that Search. Try changing your filter and search again.</p>
{% else %}
  {% for row in table %}
<div class="team-member-card-container {{ widget.cards_in_row }}">
  <div class="team-member-card">
    <div class="image-container">
      
row.image.height %}class="landscape"{% endif %} alt="{{ row.name }} Headshot">
    </div>
    <div class="team-member-info">
      <h3>{{ row.name }}</h3>
      <h4>{{ row.title }}</h4>
      <p>{{ row.description }}</p>
  </div>
  </div>
  </div>
  </div>
```

```
</div>
</div>
</div>
{% endfor %}
{% endif %}
```

4. ページの URL による動的なフィルタリングを処理するロジックを追加する

チームメンバーが表示されたので、次はリクエストしているページの URL パスに基づいてテーブルの表示を処理するロジックを記述する必要があります。

これは、ページに返される行を制御する「queryparam」変数を HubDB 関数に渡すいくつかの [if request](#) を使用して実現できます。カスタムモジュールの CSS インクルードとメインの HubDB テーブルコードの間に、以下のコードを挿入します。

```
{# sets the different query parameters using request path for hubdb query #}
{% set queryparam = "" %}
{% if request.path == "/team-members/location/ma" %}
  {% set queryparam = queryparam ~ "&location=1" %}
{% endif %}
{% if request.path == "/team-members/location/nh" %}
  {% set queryparam = queryparam ~ "&location=2" %}
{% endif %}
{% if request.path == "/team-members/location/pa" %}
```

```
{% set queryparam = queryparam ~ "&location=3" %}  
{% endif %}
```

このチュートリアルの冒頭で触れた queryparam 変数を覚えていますか？このコードはそれを利用しています。CMS ページで request オブジェクト（特に path）を使用することで、返されるチームメンバーを制御できます。ページのパスに応じて、「queryparam」を HubDB テーブルの location フィールドの特定の値に設定します。この値がテーブルのフィルターオプションとして追加されると、チームメンバーページが該当するメンバーのみを反映して動的に更新されます。

5. チームメンバーページを生成する

動的チームメンバーモジュールが設定され、準備が完了しました。あとはページを生成するだけです。この例では、4つのページを使用しています。メインのチームメンバーの「親」ページが1つと、子ページが3つです。ページを生成するときは、パスの例（スラッグ）を自由に使用してください。それ以外の場合は、必ず訪問者がアクセスするパスを使用するようにカスタムモジュールを更新する必要があります。

このチュートリアルは非常にシンプルなロジックを使用しているため、ここに記載したコードを各自のニーズに合わせて自由に変更してください。

こちらは、このチュートリアルに基づいて生成された動的チームメンバーディレクトリの例です：[チームメンバーの例](#)

HubDB でロケーションマップを作成する方法

HubDB を使用してマップを含むページを作成する

CMS で HubDB によって実現することの 1 つに、場所を使用して訪問者が求める情報を見つけることができるようにすることがあります。HubDB はフィールドタイプとして場所をネイティブにサポートしており、建物の場所に基づくページを簡単に作成できるようにする機能が多数あります。

このチュートリアルでは、HubSpot の CMS でマップを含むページを作成する方法を説明します。HubDB を使用します。詳細な HubDB ドキュメントは[ここで参照](#)できます。

以下のものが必要です。

- CMS アドオンがある HubSpot アカウント
 - *Website Starter* があるポータル、または *Website* アドオンがないポータルでは、HubDB はサポートされていません。
 - [ここからサインインして](#)、ポータルに CMS アドオンがあるかどうかを確認できます。
- 約 1 時間
- ページをカスタマイズするには、HubSpot の CMS、HTML、および CSS に関する予備知識が必要です

場所とは？

HubDB テーブルに「場所」タイプの列を作成すると、この列は、地球上の場所を指し示す座標の格納先として指定されます。

場所は、座標 (10 進数値のペア) として表されることがよくあります (例: 42.36, -71.11)。1 番目の数値は緯度 (赤道からの北緯または南緯) です。2 番目の数値は経度 (本初子午線からの度数) です。本初子午線は、北極からグリニッジ (英国)、フランス、スペイン、アルジェリア、ガーナ、南極に延びる仮想の線です。この線の東側の緯度は正数、西側の緯度は負数です。180 度の地点で正数の度数と負数の度数が一致します。

1. 訪問者の場所を検出する

[グローバルポジショニングシステム](#) (GPS) により、サイトの訪問者の所在地の検出が容易になりました。最近の携帯電話端末には、GPS 信号を受信できる GPS チップが搭載されています。端末は、周辺の Wi-Fi ネットワークや移動体通信網の通信塔を使用して所在地を判別することもできます。ラップトップとデスクトップコンピューターでは通常、Wi-Fi を使用してその所在地が判別されます。

ウェブブラウザに読み込まれたページ内で、JavaScript を使用して端末の所在地を要求できます。

```
<script>
navigator.geolocation.getCurrentPosition(function(position) {
  console.log("I'm at " + position.coords.latitude + ", " + position.coords.longitude);
});
</script>
```

最初に、ページから訪問者に対し、訪問者の所在地を取得してもよいかどうかを尋ねられ、その後座標が報告されます。これにより以下のような内容がコンソールに出力されます。

```
I'm at 42.370354299999995, -71.0765916
```

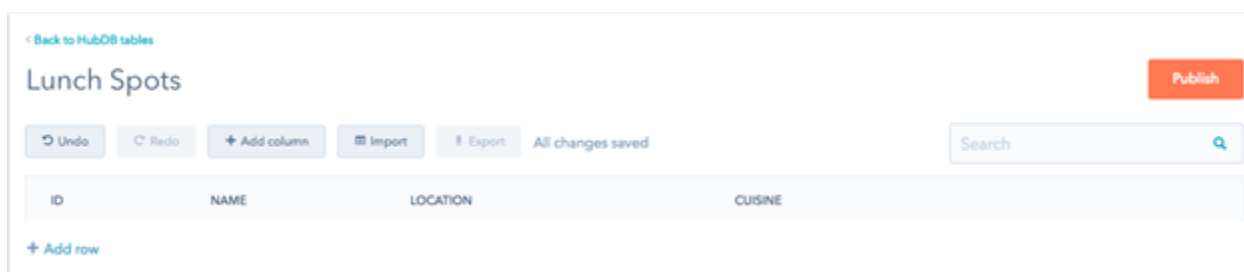
これで、訪問者の所在地が判明しました。

2. ランチスポットのデータベースを作成する

訪問者の所在地が判明したので、次にその周辺に関する情報を提供できます。この例では、ランチスポットのデータベースを作成します。

HubDB で新しいテーブルを作成し、2 つの列を追加します。追加する列は 'location' (タイプ: 場所) と、詳しい情報を示す 'cuisine' (タイプ: テキスト) です。

これは以下ようになります。



The screenshot shows the HubDB interface for a table named "Lunch Spots". At the top left, there is a link "Back to HubDB tables". The table name "Lunch Spots" is displayed in the center, with a "Publish" button on the right. Below the name, there are several action buttons: "Undo", "Redo", "+ Add column", "Import", and "Export", followed by the text "All changes saved". A search bar with a magnifying glass icon is on the right. The table structure is shown with four columns: "ID", "NAME", "LOCATION", and "CUISINE". At the bottom left, there is a "+ Add row" button.

地元のレストランをまとめた独自のデータベースを作成するか、またはマサチューセッツ州ケンブリッジにある HubSpot 本社付近のランチスポットのデータをインポートで

きます。このデータをインポートするには、[lunch-spots.csv](#)を使用します。ファイルを HubDB にアップロードし、以下に示すように列をマッピングします。

Map fields to HubDB table columns

Select the column that each field should be imported into. If a field is not mapped it will not be imported.

CSV COLUMN	VALUE FROM FIRST ROW	HUBDB TABLE COLUMN
Name	2nd St Cafe	<input type="text" value="Name"/> × ▼
location	42.3751,-71.1056	<input type="text" value="location"/> × ▼
cuisine	Cafe	<input type="text" value="cuisine"/> × ▼

インポートが完了すると、以下のようなレストランリストが完成します。

ID	NAME	LOCATION	CUISINE
2899852502	2nd St Cafe	42.3751, -71.1056	Cafe
2899852507	Desfina	42.3684, -71.0791	Greek
2899852512	Boca Grande Taquer	42.3662, -71.1132	Tex-Mex
2899852517	Similans	42.3663, -71.1132	Thai
2899852522	Sweet Touch Cafe	42.3711, -71.1149	Cafe
2899852527	Olive Cafe	42.3717, -71.1145	Cafe

テーブルを公開し、テーブル ID (URL の最後の番号) をメモしたら、データを使用開始できます。

3. シンプルなリストページを作成する

このデータを使って最初にできることは、HubL と HubDB 関数を使用してシンプルなリストを作成することです。

デザインマネージャーでコードエディターを使用して新しいテンプレートを作成します。テンプレートに "lunchspots.html" のような名前を付けます。

前にメモしたテーブル ID を使用して、`<body>` タグの直後に以下のコードを追加します。

```
{% set table_id = YOUR_TABLE_ID %}

<table>
  {% for row in hubdb_table_rows(table_id) %}
    <tr>
      <td>{{ row.name }}</td>
      <td>{{ row.cuisine }}</td>
    </tr>
  {% endfor %}
</table>
```

クリックしてテンプレートのプレビューを表示すると、レストランとそのジャンルの簡単なリストが表示されます。何も表示されない場合は、テーブル ID が正しく、テーブルが公開されているかどうかを再確認してください。

4. シングルリストに絞り込む

短いロジックを追加して、リストページを詳細ページにすることもできます。レストラン固有のページにそのレストランの名前をリンクしてみましょう。この例と、このページのその他のすべてのコード例では、`set table_id...`行をそのままにしておいてください。

```
<table>
  {% for row in hubdb_table_rows(table_id) %}
    <tr>
      <td><a
href="?{{ request.query }}&row_id={{ row.hs_id }}">{{ row.name }}</a></td>
      <td>{{ row.cuisine }}</td>
    </tr>
  {% endfor %}
</table>
```

`row_id` (この行の一意の ID) のクエリ文字列パラメータにより、各行が同一ページにリンクされます。現在の URL で既存のクエリ文字列パラメータを維持するため、`request.query` が使用されていることに注意してください。

次に、以下の詳細を追加します。

```

{% if request.query_dict['row_id'] %}
  {% set row = hubdb_table_row(table_id, request.query_dict['row_id']) %}
  <h1>{{ row.name }}</h1>
  <h3>{{ row.cuisine }}</h3>
  <p>{{ row.location['lat'] }}, {{ row.location['lon'] }}</p>
{% else %}
  <table>
    {% for row in hubdb_table_rows(table_id) %}
      <tr>
        <td><a
href="?{{ request.query }}&row_id={{ row.hs_id }}">{{ row.name }}</a></td>
        <td>{{ row.cuisine }}</td>
      </tr>
    {% endfor %}
  </table>
{% endif %}

```

if 文 (`{% if request.query_dict['row_id'] %}`) により、詳細情報を表示するか、または else ブロックまでスキップしてリストを表示するかが決定します。詳細情報ページにはあまり情報が表示されませんが、機能します。これは以下のようになります。

Tahaza Hummus Kitchen

Mediterranean

42.369766, -71.07696

5.訪問者からの距離

次に、訪問者の所在地を要求します。これにより、訪問者に対し各レストランからの距離を表示できます。コードを以下のように変更します。

```
{% if request.query_dict['row_id'] %}
  {% set row = hubdb_table_row(table_id, request.query_dict['row_id']) %}
  <h1>{{ row.name }}</h1>
  <h3>{{ row.cuisine }}</h3>
  <p>{{ row.location['lat'] }}, {{ row.location['lon'] }}</p>
{% elif request.query_dict['lat'] %}
  <table>
    {% for row in hubdb_table_rows(table_id) %}
      <tr>
        <td><a
href="?{{ request.query }}&row_id={{ row.hs_id }}">{{ row.name }}</a></td>
        <td>{{ row.cuisine }}</td>
        <td>{{ row.location|geo_distance(request.query_dict['lat'],
request.query_dict['lon'], "mi")|round(3) }} mi away</td>
```



```

        </tr>
    {% endfor %}
</table>
{% else %}
    Please allow us to read your location so we can show you the closest lunch spots.
<script>
    navigator.geolocation.getCurrentPosition(function(position) {
        window.location = window.location.href.split('?')[0] + "?lat=" +
position.coords.latitude + "&lon=" + position.coords.longitude;
    });
</script>
{% endif %}

```

このコードでは `if` 文が使用されています。既定では、訪問者に対して所在地を共有するかどうかを尋ねます。訪問者が同意した場合は、クエリパラメータ `lat` と `lon` を使用してページがリダイレクトされます。2 回目に読み込まれたページに表示されるリストには、提供された所在地から計算された距離を示す 3 番目の列が追加されています。

`row.location` はレストランの座標を返すだけであるため、距離の計算に `geo_distance` フィルターを使用します。このフィルターは訪問者の緯度、経度、および距離の単位をとります。既定では単位はメートル ("M") ですが、キロメートル ("KM")、マイル ("MI")、およびフィート ("FT") も受け入れられます。最後に、読みやすくするため数値を丸めます。

6. リストを並べ替える

このリストで十分ですが、リストがかなり長くなる場合などは、最も近い位置にあるレストランから順に表示すると便利です。このためには、並べ替えオプションを `hubdb_table_rows` 関数に追加できます。このセクションを次のように変更します。

```
...
{% elif request.query_dict['lat'] %}
    {% set params = "orderBy=geo_distance(location," ~ request.query_dict['lat'] ~ "," ~
request.query_dict['lon'] ~ ")" %}
    <table>
    {% for row in hubdb_table_rows(table_id, params) %}
        <tr>
...

```

これで、リストが `location` 列からユーザーの座標までの距離に基づいて並べ替えられます。重要なポイントは、`geo_distance` 並べ替え関数を作成することです。最終的にはこれは `orderBy=geo_distance(42.37,-71.076)` のようになります。リストを逆の順序(最も遠いレストランを最初に表示)で並べ替えるには、`geo_distance` の前に `-` (マイナス記号) を付加します。

7. 距離による絞り込み

このコードは、場所の数が少ないリストでは適切に機能しますが、数百または数千件の場所がある場合にはどうすればよいでしょうか。かなり大きなリストになります。このような場合には、リストを距離で絞り込むこともできます。`geo_distance` をクエリパラメータとして使用し、1 マイル以内にあるすべてのレストランを検出します。

```

...

{% elif request.query_dict['lat'] %}
    {% set params = "orderBy=geo_distance(location," ~ request.query_dict['lat'] ~ "," ~
request.query_dict['lon'] ~ ")" %}

    {% set params = "geo_distance(location," ~ request.query_dict['lat'] ~ "," ~
request.query_dict['lon'] ~ ",mi) __lt=1.0&" ~ params %}

    <table>
    {% for row in hubdb_table_rows(table_id, params) %}
        <tr>
...

```

これにより、hubdb クエリに対するもう 1 つのパラメータが作成されます。これは `geo_distance(location, 42.37, -71.076, mi) __lt=1.0` のようになります。

`geo_distance` クエリフィルターは、4 つの引数をとります。1 番目の引数は、絞り込み条件の場所列です。この例では"location"という名前が付けられています。その他の 3 つのパラメータは分かりやすいものです。これは緯度、経度、および距離単位です。

`geo_distance` 演算子の後に、フィルター演算子"less than"または"lt"を指定します。最後に、等号の後に関数の結果と比較する値が指定されています。

8. マップに表示する

このリストでも十分ですが、マップなどを使ってさらに視覚的にすることをお勧めします。これは、[Google Maps API](#)を使用することで実に簡単に実現します。2 番目の if 条

件 (`elif request.query_dict['lat']`) 内部のコードブロックを以下の内容に置き換えます。

```
<style>
  #map {
    width: 100%;
    height: 1200px;
  }
</style>
<div id="map" height="500"></div>
<script>
  var map;
  function initMap() {
    map = new google.maps.Map(document.getElementById('map'), {
      center: {lat:42.3665171, lng: -71.0820328}, zoom:16
    });

    var marker_origin = new google.maps.Marker({
      map: map,
      title: "you are here",
      icon: { scaledSize: {width:50, height:50}, url:
"http://maps.google.com/mapfiles/kml/paddle/blu-blank.png"},
      position: {lat: {{ request.query_dict['lat'] }}, lng: {{ request.query_dict['lon'] }}}
    });

    {% for row in hubdb_table_rows(table_id) %}
    var marker_{{ row.hs_id }} = new google.maps.Marker({
      map: map,
      position: {lat: {{ row.location["lat"] }}, lng: {{ row.location["lon"] }}},
```

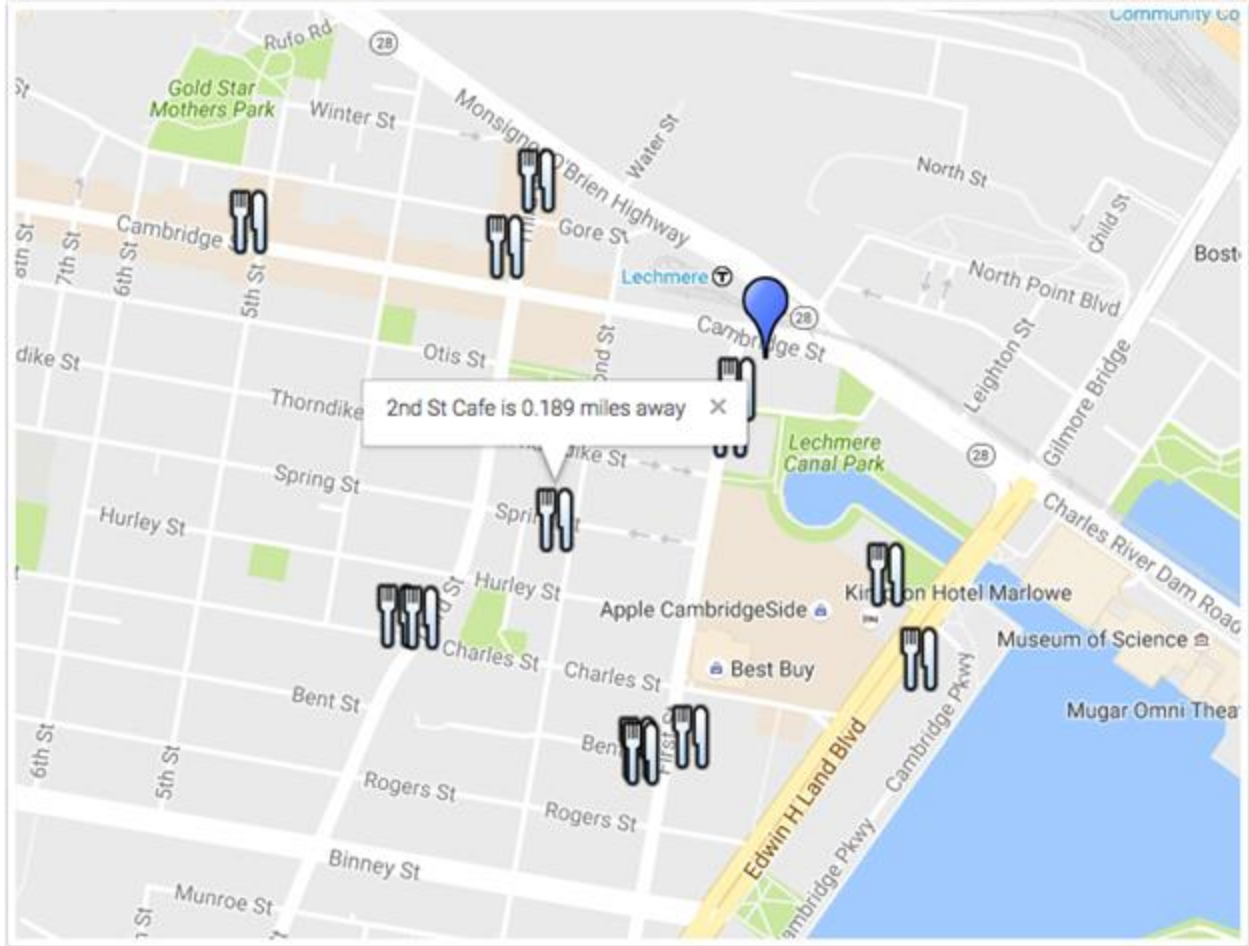
```

    title: '{{ row.name }}',
    icon: { scaledSize: {width:40, height:40}, url:
"http://maps.google.com/mapfiles/kml/shapes/dining.png"}
  });

  marker_{{ row.hs_id }}.addListener('click', function() {
    new google.maps.InfoWindow({
      content: '<div> {{ row.name }} is
{{ row.location|geo_distance(request.query_dict['lat'], request.query_dict['lon'],
"mi")|round(3) }} miles away</div>'
    }).open(map, marker_{{ row.hs_id }});
  });
{% endfor %}
}
</script>
<script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_GOOGLE_API_KEY&callba
ck=initMap" async defer></script>

```

テーブル ID と [Google API キー](#) を置き換えると、リストの代わりにインタラクティブマップが表示されます。マップは以下のようになります (どこにいても、青色の現在地マーカーが表示されます)。



9. 完了

これで完成しました。訪問者の所在地を使用して近くのレストランを探すページが完成しました。重要なポイントは、距離の絞り込み、並べ替え、計算に使用できる `geo_distance` 関数です。このシンプルなページを改善するため、リストビューとマップビューの切り替え、マップ、詳細ページの情報、および各レストランの Web サイトへのリンクなどを追加できます。

ただし、訪問者のブラウザから所在地が分からない場合や、別のエリアのスポットを探したい場合にはどうしたらよいでしょうか？ [Google Maps Geocoding API](#) を使用して、訪問者が場所を入力して座標を取得できるようにできます。

いただきます！

HubDB で不動産リストを作成する方法

HubDB を使用して不動産物件情報を作成する

このチュートリアルでは、HubDB を使用して HubSpot の CMS で不動産物件情報を作成する方法を説明します。詳細な HubDB ドキュメントは [ここで](#) 参照できます。

以下のものがが必要です。

- CMS アドオンがある HubSpot アカウント
 - *Website Starter* があるポータル、または *CMS アドオン* がないポータルでは、HubDB はサポートされていません。
 - [ここからサインインして](#)、ポータルに CMS アドオンがあるかどうかを確認できます。
- 約 3 時間
- ページをカスタマイズするには、HubSpot の CMS、HTML、および CSS に関する予備知識が必要です。

1. 物件情報データベースを作成する

HubDB で新しいテーブルを作成し、列を追加します。この例では、いくつかの列を使用してデータベースを作成します。このデータベースで物件情報が生成されます。この例では以下の列とタイプを使用します。

- listing_image (画像)
- listing_price (テキスト)
- listing_address (テキスト)
- type (選択)
- listing_bedrooms (数値)
- listing_baths (数値)
- listing_sq_ft (数値)
- listing_acres (数値)
- listing_url (url) -- 詳細情報ビューにリンクする場合

独自の物件情報データベースを作成するか、またはこの例のウォークスルーのためのテストデータをインポートできます。このデータをインポートするには、[real-estate-listings.csv](#) を使用します。作業を進めるためファイルを HubDB にアップロードします。

テーブルを公開し、テーブル ID (URL の最後の番号) をメモしたら、データを使用開始できます。

 app.hubspot.com/hubdb/Portal ID/table/1058586

2. シンプルな物件情報ページを作成する

このデータを使って最初にできることは、HubL と HubDB 関数を使用してシンプルな物件情報を作成することです。


```
        </ul>
    </div>
    <div class="button"><a href="{{ row["listing_url"] }}">Details</a></div>
    </div>
</div>
</div>
{% endfor %}
{% endif %}
```

クリックしてテンプレートのプレビューを表示すると、シンプルな不動産物件情報が表示されます。何も表示されない場合は、テーブル ID が正しく、テーブルが公開されているかどうかを再確認してください。

この処理の内容を説明します。この時点では'queryparam'について心配する必要はありません。'queryparam'は、このチュートリアルの後半で、訪問者の検索入力内容に基づいてテーブルを絞り込み、物件情報を絞り込むときに必要となります。

最初に、何らかの理由でテーブルから空のオブジェクトが返される場合にエラーをレンダリングするロジックを追加します。次に、行を物件情報ページにマッピングします。マッピングが正しく行われると、プレビューには物件情報データベースがレンダリングされます。必要に応じて HTML 構造を自由に変更できます。

角括弧表記を使用して HubDB の行の値にアクセスしていることに注意してください。ドット表記を使用するように調整することも可能です。

3. 物件情報ページにフィルターを追加する

ここまで問題なく進んできましたが、ここで1つの課題があります。訪問者が常にすべての物件情報結果を参照するわけではありません。多くの場合、訪問者は特定のタイプの物件情報のみに関心を持ちます。また、郵便番号や市町村に基づいて検索することがあります (この機能は後で追加します)。

したがって、訪問者を支援するため、シンプルなフィルターを物件情報ページに追加します。この例では、HubDB の **type** 列にフィルターを追加します。

カスタムモジュールの上部に、シンプルなフィルターを生成する以下のコードを追加します。

```
<!-- set the filter by drop down, search bar, and submit button -->

<div>
<form id="form_id" method="get">

  <div>
    <h4>FILTER BY LISTING TYPE: </h4>
    <select name="type" form="form_id" onChange="this.form.submit()">
      <option value="show-all">Show All</option>
      {% set types = hubdb_table_column({Insert YOUR Table ID}, "type").options %}
      {% for choice in types %}
        {% set type_list = type_list~choice.id|list%}
        {% if choice.id == request.query_dict.type%}
          <option selected="selected"
value="{{ choice.id }}">{{ choice.name }}</option>
        {% else %}
          <option value="{{ choice.id }}">{{ choice.name }}</option>
        {% endif %}
      {% endfor %}
    </select>
  </div>
</form>
</div>
```

```
        {% endfor %}
    </select>
</div>

</form>
</div>
```

フィルターを生成するため、HTML フォーム要素を作成します。子の要素は HubDB Type フィールドにマッピングされ、'types'という新しい変数を設定します。次に、[for ループ](#)を使用してフィールドの値をループ処理します。これはロジックの最終部分で使用され、フィールド選択オプションの値とラベルのペアを設定します。

4. クエリパラメーターフィルターを処理するロジックを追加する

ページにシンプルなフィルターが追加されたので、次にこの入力に基づいて物件情報ページのレンダリングを処理するロジックを作成する必要があります。

このためには、テーブルに渡されるフィルターパラメーターを設定する If リクエストを使用します。カスタムモジュールで、Form オブジェクトからメイン HubDB テーブルの間に以下のコードを挿入します。

```
<!-- sets the different query parameters using submitted input for hubdb query -->
{% set queryparam = "" %}
{% if request.query_dict.type in ["1", "2", "3", "4"] %}
```

```
{% set queryparam = queryparam ~ "&type=" ~ request.query_dict.type|urlencode %}
{% endif %}
{% if request.query_dict.type == "show-all" %}
    {% set queryparam = queryparam %}
{% endif %}
```

このチュートリアルの冒頭にあった `queryparam` 変数を覚えていますか？次にこの変数を使用します。このコードでは `if` 文が使用されています。CMS ページでリクエストオブジェクトを使用して、訪問者が選択したフィルターオプションが、配列に含まれているかどうかを判断できます。含まれている場合は、'queryparam'をそのオプションと同等に設定します。この値がテーブルのフィルターオプションとして付加されると、物件情報ページが更新され、お客様が求めている物件情報のみが表示されます。テンプレートからページを生成して、すべてが機能することを確認しましょう。

5. 物件情報に検索フィールドを追加する

この物件情報ページで十分ですが、リストがかなり長くなる場合には特に、訪問者に対し検索クエリに基づいて検索結果の物件情報を表示すると便利です。このためには、検索フィールドを追加し、ロジックを多少変更します。

</form>閉じタグの前に以下の HTML を追加します。

```
...
<div>
    <input name="listing_address" type="text" id="search-by" class="autocomplete"
placeholder="Search by City or Zip Code...">
</div>
```

```
<input id="submit-button" type="submit" value="search">
```

...

この入力フィールドにより、訪問者が"listing_address"列内で複数コンポーネントを対象に検索できるようになります。プレースホルダーテキストを使用すると、検索の目的を示すことができます。この場合、郵便番号または市町村の検索に基づいて結果が返されることをユーザーが理解できるようにします。

6. クエリパラメータロジックを完成させる

ユーザーの検索が適切に機能できるようにするには、ステップ4で定義したロジックを多少変更します。最初に生成したコードを以下の更新されたコードに置き換えます。

...

```
{% set queryparam = "" %}
{% if request.query_dict.type in ["1", "2", "3", "4"] and request.query_dict.listing_address == "" %}
    {% set queryparam = queryparam ~ "&type="~request.query_dict.type|urlencode %}
{% endif %}
{% if request.query_dict.type in ["1", "2", "3", "4"] and request.query_dict.listing_address != "" %}
    {% set queryparam =
queryparam~"&type="~request.query_dict.type|urlencode~"&listing_address__icontains="~request.query_dict.listing_address|urlencode %}
{% endif %}
{% if request.query_dict.type == "show-all" and request.query_dict.listing_address != "" %}
```

```
{% set queryparam =  
queryparam~"&listing_address__icontains="~request.query_dict.listing_address|urlencode %}  
{% endif %}  
...
```

訪問者の検索クエリをサポートするには、ロジックを多少追加する必要があります。このためには、これらの新しい If 文を使用します。各文で、送信される検索フィールドがあるかどうかをチェックする追加条件を追加します。まず、検索フィールドに何も渡されず、ページを絞り込む場合には、タイプフィールドに対するクエリだけを設定します。ロジックのそれ以降の部分では、検索語が存在する場合にリクエストの queryparam が変更されます。

このためには HubDB フィルター '__icontains' を使用します。これが、最初に物件情報の住所を含めるフィールドとして 'text' フィールドを使用した理由です。ユーザーが入力する郵便番号または市区町村がこの列に含まれている場合、HubDB テーブルに適切な行が返されます。

7. 完了

この時点で不動産物件情報ページはフルに動作しています。ユーザーが物件情報タイプに基づいて絞り込み、さらに物件情報を郵便番号や市区町村で検索できるようにする機能を追加します。

以下は、この例を使用して作成したサンプルページです。[不動産物件情報の例](#)

この例をさらに変更できます。HubDB の場所サービスを使用して物件情報をマップにプロットしてみたい場合はどうすればよいでしょうか？これは、HubDB の機能を使用

し、以下のチュートリアルを変更することで実現します。[HubDB を使用してマップを作成する](#)

複数の HubDB テーブルを結合する方法

複数の HubDB テーブルを結合する方法

[HubDB](#) は、スプレッドシートのようにテーブルの行、列、セルで表現されるリレーショナルデータストアです。HubDB テーブルは、[外国人 ID] 列タイプを使用して結合できます。これにより開発者は、複数テーブルのデータと情報を結合して、結合データをレンダリングできます。これは、複数のデータストア間で一部のデータが共有される場合に非常に役立ちます。これにより、1つの集中管理データテーブルでこの情報を管理でき、その他の複数の HubDB テーブルデータストアからこのテーブルにアクセスできます。

このチュートリアルでは、[外国人 ID] フィールドを使用して HubDB テーブルを結合する方法と、結合した HubDB データをレンダリングする HubL を作成する方法について詳しく説明します。以下のものがが必要です。

- **CMS アドオンがある HubSpot アカウント**
 - *Website Starter* があるポータル、または *CMS アドオン* がないポータルでは、HubDB はサポートされていません。

- [ここからサインインして](#)、ポータルに CMS アドオンがあるかどうかを確認できます。
- ページをカスタマイズするには、HubSpot の CMS、HTML、および HubL に関する予備知識が必要です。

このチュートリアルでは、結合する 2 つの HubDB テーブルをすでに作成していることを前提としています。HubDB に詳しくない場合、または初めて HubDB テーブルを作成する場合には、[HubDB ドキュメント](#)をご覧ください。

1. HubDB テーブルに [外国人 ID] 列を追加する

HubSpot ポータルで [HubDB](#) に移動し、別のテーブルと結合するテーブルを編集します。[列の追加] ボタンを選択し、[外国人 ID] 列タイプを選択します。

Add column



Column label

Expertise Table Join

Column name



expertise_table_join

Column type

Foreign ID



Select table

Table Joins | Areas of expertise



Select column

Name



Add column

[テーブルを選択] フィールドで、編集するテーブルに結合する HubDB テーブルを選択します。[列を選択] フィールドで、選択した結合対象テーブルから、[外国人 ID] フィールドに表示する列を選択します。

[列を選択] で選択した値は、HUBDB ユーザーインターフェイスの [外国人 ID] フィールドに表示される列値のみを示し、結合した HubDB テーブルを実際にレンダリングするときにはすべてのテーブル列が使用可能であることに注意してください。

2. テーブルの行に外部テーブルの行を追加する

[外国人 ID] HubDB 列が選択されたため、次に HubDB テーブルのすべての行に複数選択列フィールドを追加します。これにより、外部テーブルの行を選択できるようになります。前述のとおり、[列を選択] フィールドはこの複数選択フィールドで使用され、これにより外部テーブルから選択する行を確認しやすくなります。以下の例では、[専門分野テーブルの結合 (Expertise table join)] フィールドの複数選択値は外部 HubDB テーブルの [名前] 列です。

Table Joins | Names

ID	NAME	EXPERTISE TABLE JOIN
3276315385	Bill	Skiing × Fishing ×
3276315448	Will	Fishing ×
3276315465	William	Skiing Walking Biking
3276315514	Billiam	

+ Add row

[外国人 ID] 列の [列を選択] フィールドは安全に編集できる点に注意してください。この編集では、HubDB ユーザーインターフェイスに値が表示される列が更新されるだけです。

3.結合した HubDB テーブルデータをレンダリングする

前述のとおり、レンダリングのために、[列を選択] だけではなく、すべての外部テーブル行情報に HubL でアクセスできます。HubDB 外部行データにアクセスするには、[ネストした for ループ](#)を使用し、1つの行に関連付けられているすべての外部行をループ処理します。

外部行をレンダリングする for ループへは、[外国人 ID] 列で設定された内部 [カラム名] フィールド値からアクセスします。HubL は以下のようになります。

```
{% for row in hubdb_table_rows(<tableId>, <filterQuery>) %}
  the name for row {{ row.hs_id }} is {{ row.name }}
  {% for foreign_row in row.foreign_table %}
    the name for foreign row {{ foreign_row.hs_id }} is {{ foreign_row.name }}
  {% endfor %}
{% endfor %}
```

HubDB でマルチレベルの動的ページを作成する方法

HubDB でマルチレベルの動的ページを作成する方法

動的ウェブサイトページとは、エンドユーザーが要求する URL のパスに基づいてコンテンツが変化する CMS ページです。HubDB では、HotSpot ウェブサイトページのデータを格納、絞り込み、表示できます。マルチレベルの動的ページではこのコンセプトをさらに発展させ、1つの動的テンプレート内で最大5レベルのページを作成できます。

各動的ページには専用の一意的 SEO 対応 URL があり、ページ固有のアナリティクスを提供します。詳細な HubDB ドキュメントは [ここで](#) 参照できます。

- **\$300 Website Add-On がある HubSpot アカウント**
 - *Website Starter* があるポータル、または *Website アドオン* がないポータルでは、HubDB はサポートされていません。
 - [ここからサインインして](#)、ポータルに \$300 Website Add-On があるかどうかを確認できます。
- ページをカスタマイズするには、HubSpot の CMS、HTML、および HubL に関する予備知識が必要です。

このチュートリアルでは、複数の HubDB テーブルがすでに作成されていることを前提としています。HubDB に詳しくない場合、または初めて HubDB テーブルを作成する場合には、[HubDB ドキュメント](#) をご覧ください。

1. テーブルの設定で子テーブルを有効にする。

HubSpot ポータルで [HubDB](#) に移動し、他のテーブルの親にするテーブルを編集します。テーブル設定で [子テーブルを許可] と [子テーブル用のページを有効化] の両方が選択されていることを確認します。

Settings
×

- Use for dynamic pages
- Allow child tables ⓘ
 - Enable pages for child tables ⓘ
- Allow public API access

Update

これにより新しい「子テーブル」列が追加され、ポータルで公開されている他の HubDB テーブルを選択できるようになります。

2. 各行の子テーブルを選択する

親テーブルの各行を子テーブルにリンクできます。

← Back to HubDB tables
Revert to the last published version
✓ Saved

Update

Cuisine

Search
🔍

↶ Undo

↷ Redo

+ Add column

📁 Import

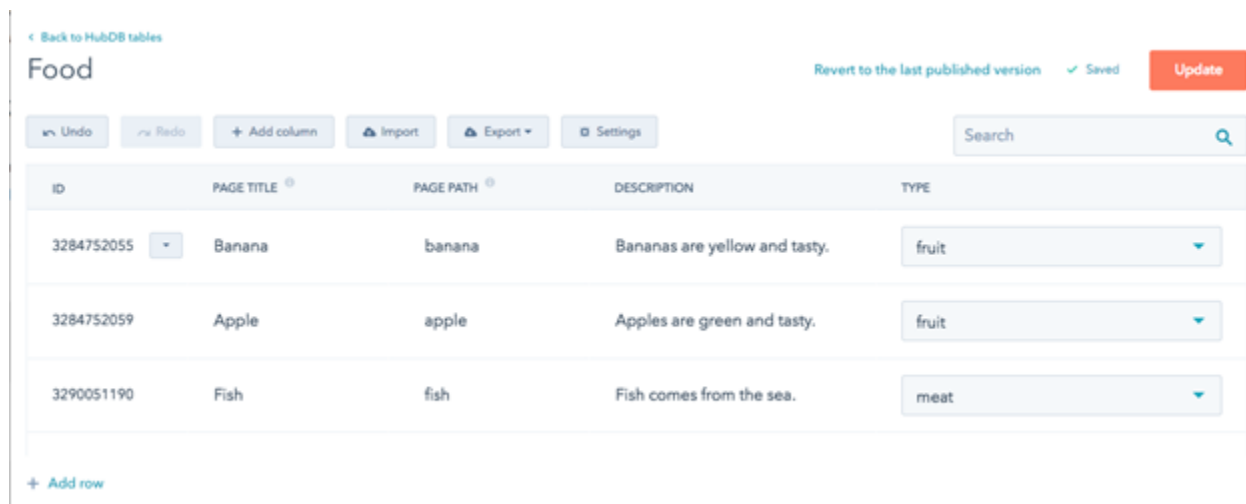
📄 Export ▾

⚙️ Settings

ID	PAGE TITLE ⓘ	PAGE PATH ⓘ	CHILD TABLE ⓘ
3284752046	Food	food	Food ▾
3284752048	Beverage	beverage	Beverage ▾

+ Add row

子テーブルには独自のプロパティを設定できます。



ID	PAGE TITLE	PAGE PATH	DESCRIPTION	TYPE
3284752055	Banana	banana	Bananas are yellow and tasty.	fruit
3284752059	Apple	apple	Apples are green and tasty.	fruit
3290051190	Fish	fish	Fish comes from the sea.	meat

マルチレベル動的ページを設定すると、子テーブルの各行のページパスが "parent_path/child_path" になります。たとえば "apple" 行のページパスは "page_path/food/apple" になります。[子テーブル用のページを有効化] を True に設定すると、中間パス ("page_path/food" と "page_path/beverage") も動的テンプレートに解決されるので、これらのサブタイプの個別リストページを作成できます。中間パスを解決せず 404 ページを返すようにするには、[子テーブル用のページを有効化] をオフにできます。

3. マルチレベルテンプレートを作成する

以前は、動的ページでサポートされていたレベルは、最上位リストページと、行から生成された動的ページの 2 レベルだけでした。子テーブルを使用することで、1 つの動的テンプレートで最大 5 つのレベルのページを作成できます。異なるテンプレートレベルを区別するには、dynamic_page_route_level HubL 変数を使用できます。この変数は 0 (最上位テンプレート) から始まり、テーブルレイヤーごとに増分します。

```
{% if dynamic_page_route_level == 0 %}
```

Top Level Template

```
{% elif dynamic_page_route_level == 1 %}
    Parent table template (/food /beverage)
{% elif dynamic_page_route_level == 2 %}
    Child table template (/food/banana etc., /beverage/soda etc.)
{% endif %}
```

4.最上位テンプレートに入力する

テンプレートレベルごとに独自の処理が行われることがあります。最上位テンプレートで子行をリストし、親カテゴリ別にグループ化します。最初に

`dynamic_page_hubdb_table_id` を使用してカテゴリ行を取得します。次に各カテゴリ行の `hs_child_table_id` プロパティを使用して、子テーブルのテーブル ID を取得できます。最後に、これらのテーブル ID を使用して各親カテゴリの子行をリストします。

注:子テーブルの列とその内部名を同一にすることが重要です。同一ではない場合には、特定のテーブルの一意のコンテンツをレンダリングするために条件文を使用する必要があります。

```
{% if dynamic_page_route_level == 0 %}
    <h1>Categories</h1>
    {% set rows = hubdb_table_rows(dynamic_page_hubdb_table_id) %}
    {% for row in rows %}
        <h2><a href="{{ request.path }}/{{ row.hs_path }}">{{ row.hs_name }}</a></h2>
        {% set childRows = hubdb_table_rows(row.hs_child_table_id) %}
        {% for childRow in childRows %}
            <li><a
href="{{ request.path }}/{{ row.hs_path }}/{{ childRow.hs_path }}">{{ childRow.hs_name }}
</a></li>
        {% endfor %}
    {% endfor %}
```



```
{% endif %}
```

5.動的レベルテンプレートに入力する

表示するテンプレートレベルごとにテンプレートを定義する必要があります。これは、子テンプレートで親行のデータにアクセスする際に役立ちます。たとえば

"food/banana"に解決するときには、`dynamic_page_hubdb_row` 変数が"banana"行に設定されますが、"food"行のデータにも引き続きアクセスする必要があります。親行を取得を取得するには、`dynamic_page_hubdb_row` で `hs_parent_row` 値を使用します。

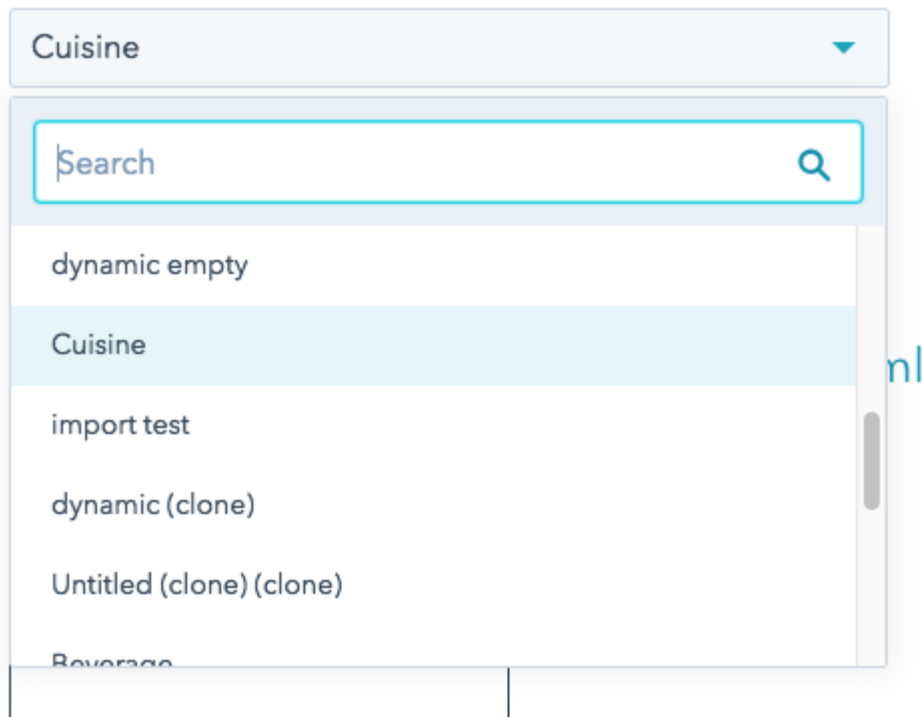
```
{% if dynamic_page_route_level == 1 %}
    <h1>Categories</h1>
    <h2>{{dynamic_page_hubdb_row.hs_name}}</h2>
    {% set rows = hubdb_table_rows(dynamic_page_hubdb_row.hs_child_table_id) %}
    {% for row in rows %}
        <li><a href="{{ request.path }}/{{ row.hs_path }}">{{ row.hs_name }}</a></li>
    {% endfor %}
{% elif dynamic_page_route_level == 2 %}
    <h1>Categories</h1>
    <h2>{{dynamic_page_hubdb_row.hs_parent_row.hs_name}}</h2>
    <h3>{{dynamic_page_hubdb_row.hs_name}}</h3>
{% endif %}
```

6.親テーブルをページにリンクする

最後に、マルチレベルテンプレートからページを作成し、ページエディターの [詳細オプション] セクションでこのページに最上位親テーブルをリンクします。[このチュートリアルで作成したテンプレートから生成されるネストした動的ページの例についてはこちらをご覧ください。](#)

HubDB

Table for dynamic pages ⓘ



動的ページに動画を追加する方法

HubDB の動的ページに動画を追加する方法

[動的ウェブサイトページ](#)とは、エンドユーザーによってリクエストされた URL のパスに基づいてコンテンツが変化する CMS ページです。HubDB ではすでに、データの保

存、絞り込み、HubSpot ウェブサイトページでの表示が可能です。マルチレベルの動的ページでは、この考え方をさらに推し進めて、1つの動的テンプレートで最大5つのレベルのページを作成できます。

各動的ページには独自の SEO 対応 URL があり、ページ固有のアナリティクスが提供されます。詳細な HubDB ドキュメントは[ここ](#)で参照できます。

- \$300 の Website アドオンがある HubSpot アカウント
 - *Website Starter* があるポータル、または *Website* アドオンがないポータルでは、HubDB はサポートされていません。
 - [ここからサインインして](#)、ポータルに\$300 の Website アドオンがあるかどうかを確認できます。
- HubSpot の CMS、HTML、および HubL に関する予備知識

このチュートリアルは、複数の HubDB テーブルがすでに作成されていることを前提としています。HubDB を使い慣れていない場合や、HubDB テーブルを初めて作成する場合は、[HubDB ドキュメント](#)をご覧ください。

1. テーブルに「VIDEO」列を追加します。

HubSpot ポータルの[\[HubDB\]](#)に移動し、他のテーブルの親にするテーブルを編集します。[Add new column (新規列を追加)]をクリックし、[動画]タイプの列を作成します。

Add column ×

Column label

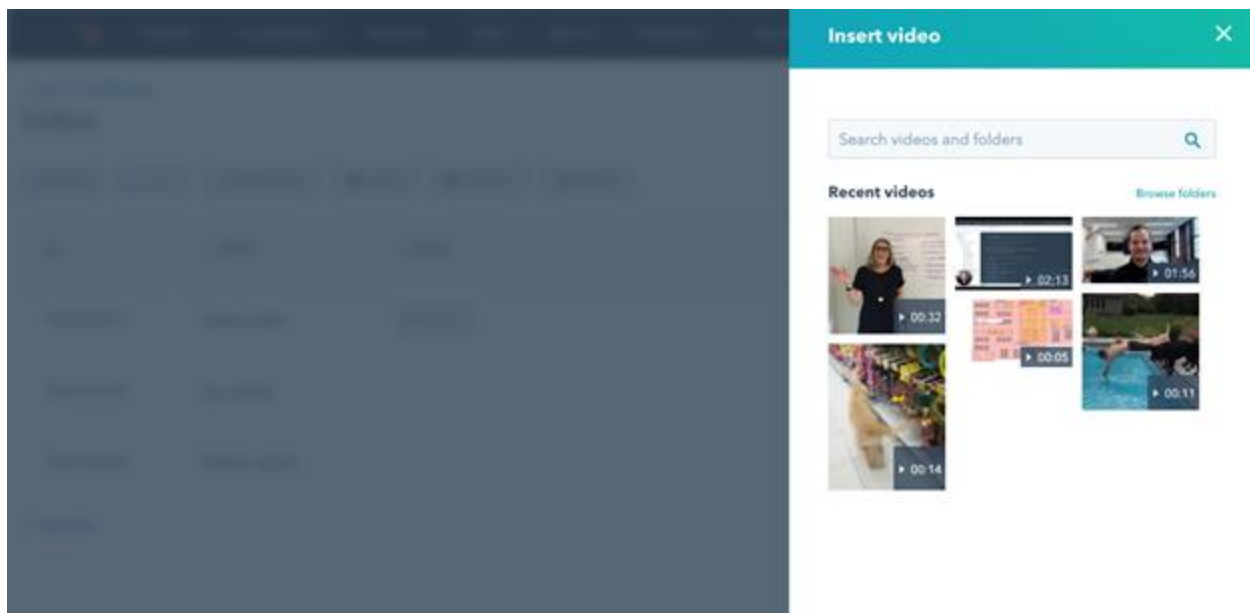
Column name ?

Column type

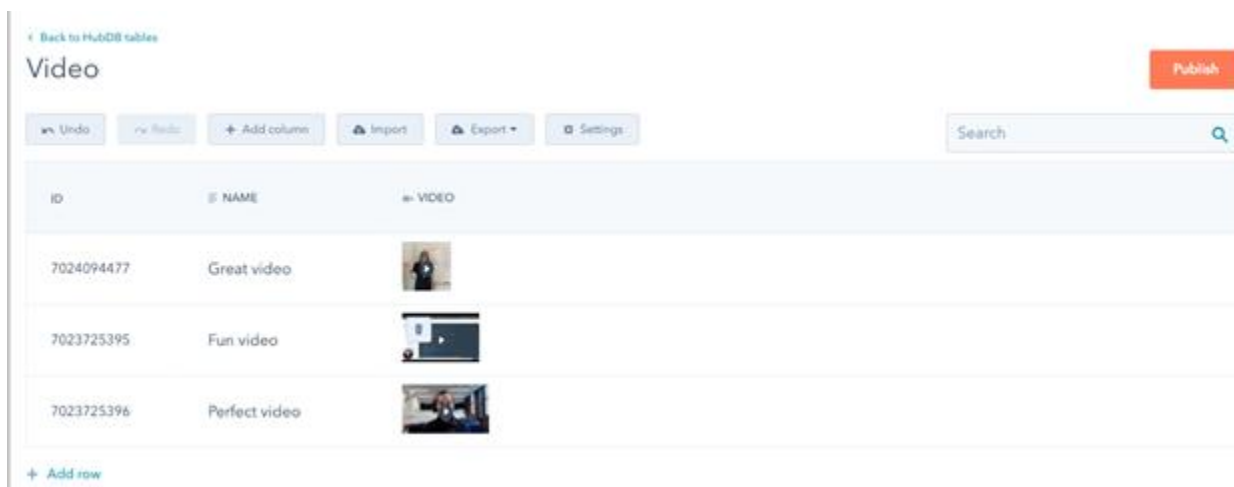
Add column

2.各行の動画を選択します。

Video 列の[選択]ボタンをクリックすると、行に動画を追加できます。



動画を選択すると、動画の「player_id」が列の値として行に保存されます。UIで動画を視覚的に表すため、ファイルのサムネイルが使用されます。



3. 動画プレイヤーウィジェットを動的テンプレートに追加します。

動的テンプレートで行データを参照して、「[video_player](#)」ウィジェットを作成できます。

```
{% if dynamic_page_hubdb_row %}
  {% video_player "embed_player" player_id="{{dynamic_page_hubdb_row.video}}" %}
{% endif %}
```

記事で特定のトピックを確認する方法

ブログ記事で特定のトピックを確認する方法

特定のトピックを扱っている記事では、異なるマークアップを出力するのが適切な場合もあります。たとえば、お勧めの記事やプロモーション記事に異なるマークアップを使用して、リスト内でこれらの記事を強調表示するようにスタイルを設定できます。

HubSpot では if 文と [マップフィルター](#) を使用して、特定のトピックの記事を絞り込むことができます。

各ブログ記事には、`topic_list` という 1 つのディクショナリが関連付けられています。ディクショナリ内の各項目には名前やスラッグなどの属性があります。特定のトピックがこのディクショナリ内に存在するかどうかを検査するには、`content.topic_list` にマップフィルターを適用する必要があります。フィルターのパラメーターとして属性 `name` または `slug` を指定する必要があります。

個々の記事のマークアップまたはブロー一覧の記事項目マークアップの中で、次のような if 文を記述して特定のトピック名を検査することができます。スラッグ属性を使ってマップする場合は、トピックスラッグでスペースの代わりにダッシュが使用されることに注意してください。

```
{% if 'Featured' in content.topic_list|map('name') %}
    記事のトピックとして「Featured」が設定されている場合に出力するマークアップ
{% endif %}
```

多言語テンプレートの用語集を作成する方法

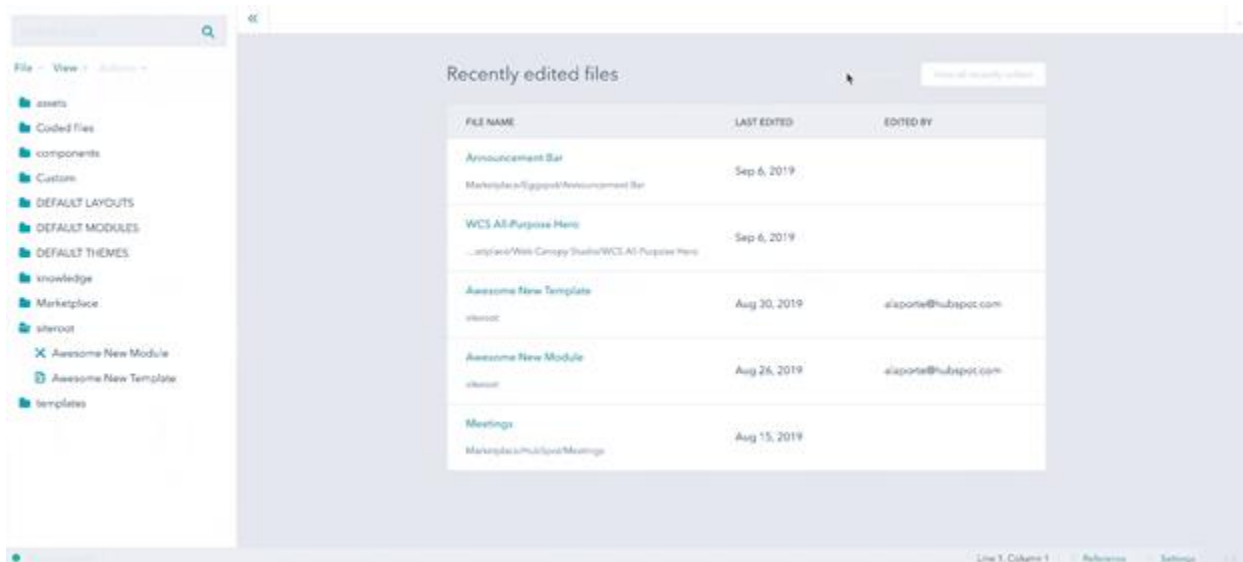
多言語テンプレート向けの用語集を作成する方法

HubSpot CMS は、世界中の企業のサイトをさまざまな言語でホストしています。大規模な多言語対応サイトを構築することは、多くの組織にとって課題となる場合があります。それを簡単にするために、HubSpot は [いくつかのサービスを統合](#)し、ページコンテンツを翻訳してページの複製バージョンに再注入できるようにしています。言語固有のテンプレートを作成することも可能ですが、この方法に加えて、HubL で中心的な用語集ファイルを作成すると、それを使ってサイトの URL に基づいてテンプレート用語を動的に更新することができます。

このチュートリアルでは、インポートされるデザインマネージャーHubL ファイルを作成します。これは、さまざまな言語のテンプレート文字列からなる用語集として機能します。ページの URL に基づきどの言語をレンダリングするかを決定するために、[If文](#)を使用します。その後、テンプレート全体でこれらの文字列変数を呼び出すことができます。この方法により、さまざまな言語のすべてのテンプレート文字列を1つのファイルに保持できます。このファイルを更新すると、テンプレート全体およびカスタムモジュールにそれが反映されます。

1..html ファイルの作成

デザインマネージャーに移動し、[新しいコードテンプレート](#)を作成します。必ずファイル拡張子.html を付けてそれを保存してください。.html を作成したら、事前入力されていたマークアップをすべて消去します。



2.URL 構造の変数を識別する

多言語サイトではさまざまな URL 構造を使用することがあります。したがって、どの言語のテンプレート用語を読み込むべきか決定するための変数を識別する必要があります。[コンテンツ変数](#)の2つの例、およびスペイン語検索結果ページ

<http://knowledge.hubspot.com/es/search> でのそれらの値を以下に示します。

- **content.slug** - es/search
- **content.domain** - <http://www.knowledge.hubspot.com>

複数言語 URL 構造でサブディレクトリを使用している場合は、**content.slug** を選択するのが適切でしょう。一方、複数の異なるドメインを使用している場合は、**content.domain** を使用するのが適切でしょう。いつでもページ上の[デベロッパー情報](#)を使用すると、テンプレートコンテキストにどんな値が保存されているかを確認できます。

3.条件分岐を追加する

URL 構造の検査に使用する変数を選択した後は、どの用語セットを読み込むかを判断するための if 文を書くことができます。

以下は、ドメインとサブディレクトリに関する if 文を示しています。サブディレクトリ手法では文ごとに2つの条件があります。なぜなら、サブディレクトリ

(knowledge.hubspot.com/es/search) 内の両方のパスで、およびパス末尾に「/」が含まれない場合 (help.hubspot.com/es) に、これが true と評価される必要があるためです。

```
{# 言語コードサブディレクトリを使用するサイト用の if 文 #}
{% if (content.slug is string_startingwith "es/") or (content.slug == "es") %}

{% elif (content.slug is string_startingwith "de/") or (content.slug == "de") %}
```

```

{% else %}

{% endif %}

{# 言語コードドメインを使用するサイト用の if 文 #}
{% if content.domain == "hubspot.es" %}

{% elif content.domain == "hubspot.de" %}

{% else %}

{% endif %}

```

4.if 文の中で変数を設定する

それぞれの if 文の中で、言語ごとに出力する用語の [文字列変数を定義](#) します。言語ごとに同じ変数名を再設定して、別の値を割り当てます。以下の例では、文字列変数「Hello」と「world」を保存します。

```

{# 言語コードサブディレクトリを使用するサイト用の if 文 #}
{% if (content.slug is string_startingwith "es/") or (content.slug == "es") %}
    {% set greeting = "Hola" %}
    {% set world = "mundo" %}
{% elif (content.slug is string_startingwith "de/") or (content.slug == "de") %}
    {% set greeting = "Hallo" %}
    {% set world = "welt" %}
{% else %}

```

```
{% set greeting ="Hello" %}
{% set world ="world" %}
{% endif %}

{# 言語コードドメインを使用するサイト用の if 文 #}
{% if content.domain == "hubspot.es" %}
    {% set greeting ="Hola" %}
    {% set world ="mundo" %}
{% elif content.domain == "hubspot.de" %}
    {% set greeting ="Hallo" %}
    {% set world ="welt" %}
{% else %}
    {% set greeting ="Hello" %}
    {% set world ="world" %}
{% endif %}
```

ファイルを公開し、[このテンプレートを新しいコンテンツで利用できるようにする]オプションをオフにします。このオプションをオフにすると、テンプレート検証で必須のテンプレートタグの検査がバイパスされます。

5. ファイルをテンプレートにインポートして変数を出力する

こうして用語集 HTML ファイルを作成した後は、それをテンプレートにインポートすることができます。インポートした後、テンプレート全体とカスタムモジュールで変数を出力することができます。変数は、リクエスト URL に基づいて適切な言語に自動的

に更新されます。たとえばリクエストが/es/サブディレクトリに到達した場合、スペイン語の挨拶付きでマークアップを出力します。

- HubL
- 出力例

```
{% import "custom/page/web_page_basic/glossary.html" as lang %}  
  
<h1>{{ lang.greeting }} {{ lang.world }}</h1>
```

変数セットを別のテンプレートにインポートする方法

変数セットをテンプレートにインポートする方法

HubL では、コードファイルから 1 つまたは複数の他のテンプレートのコンテキストに変数とマクロをインポートすることができます。この手法は複雑なサイトまたは複数言語サイトに非常に役立ちます。一連の情報を 1 つの中心的な場所で保存して編集することができ、それをテンプレート全体および含まれているカスタムモジュールで出力できます。

このチュートリアルでは、インポートタグの基本的な実装を示し、これが役に立つ実際の導入事例をいくつかご紹介します。

- シンプルな静的データベースの作成
- 複数言語の用語集の作成

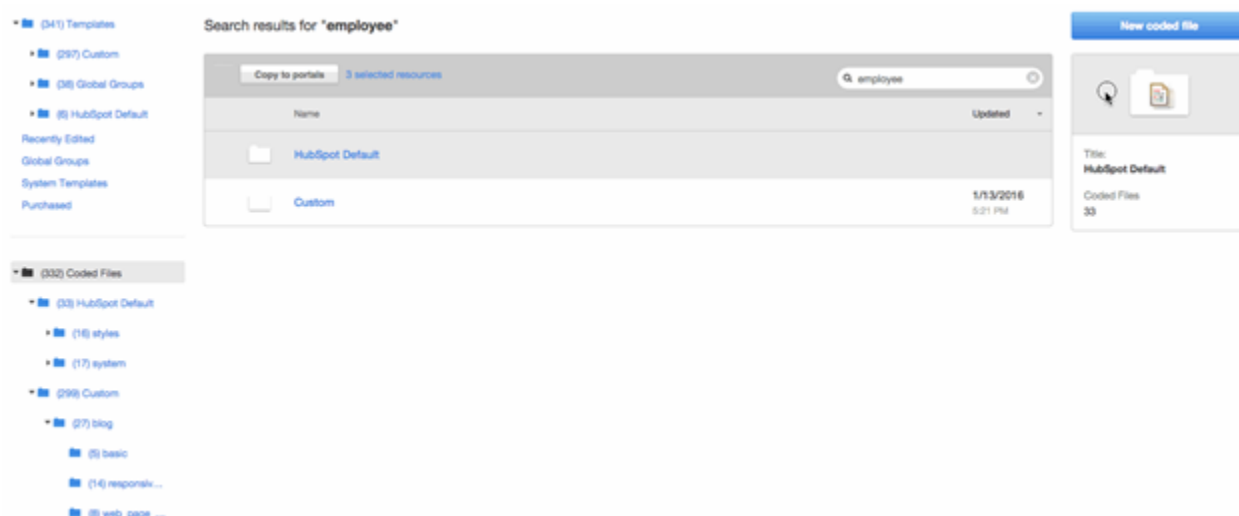
基本的な使用法のチュートリアル

以下のチュートリアルでは、2つの変数と1つのマクロをHTMLファイルで宣言して、それを別のテンプレートの中にインポートします。これらのインポートされた値を、マスターテンプレートおよび含まれているカスタムモジュールで出力できます。

なお、テンプレートおよび含まれているアセットとコンパイル順序が原因で、これらのインポートされた変数とマクロ値をグローバルモジュールやグローバルグループで引き継ぐことはできません（そのモジュールまたはグループにファイルが直接インポートされる場合は引き継ぎ可能です）。その場合でも、インポートタグを再宣言することなく、カスタムモジュールでそれら呼び出すことができます。

1. デザインマネージャーで新規.htmlファイルを作成する

デザインマネージャーに移動し、[新しいコードテンプレート](#)を作成します。必ず.htmlファイル拡張子を付けて保存してください。作成した.htmlで、事前入力されたマークアップをクリアします。



2. ファイルに変数やマクロを追加する

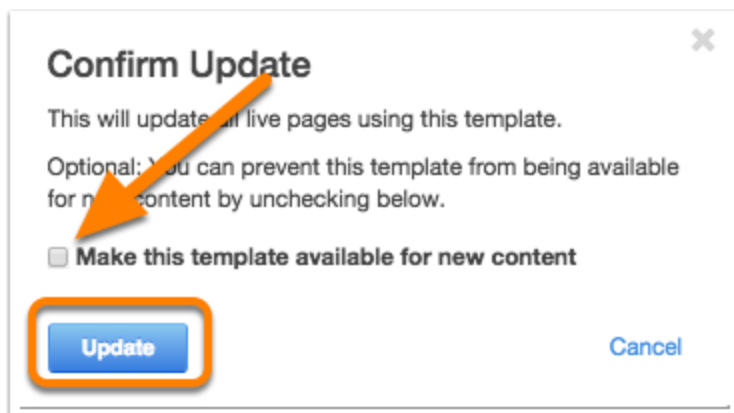
他のテンプレートで使用する変数を定義します。この例では、従業員名の2つの変数と1つのマクロを定義します。マクロは、定義済みクラスでカスタマイズ可能なタグに囲まれた役職名を出力します。変数とマクロの構文の詳細については、[こちら](#)をご覧ください。

変数として単語を1つ指定するか、複数の単語の場合はスペースの代わりに下線文字を使用します（ハイフン付きの変数はサポートされません）。

```
{% set ceo = "Brian Halligan" %}
{% set cto = "Dharmesh Shah" %}
{% macro job(tag, job_title) %}
  <{{ tag }} class="job-title">{{ job_title }} : </{{tag}}>
{% endmacro %}
```

3. ファイルを公開する

このファイルを公開するときには、[このテンプレートを新しいコンテンツで利用できるようにする]オプションを選択解除する必要があります。このオプションを使用した場合、必要なページ変数なしでファイルを保存できます。このファイルはコンテンツのマスターテンプレートとして使用されないため、コンテンツ作成者向けのオプションとしてこれを使用可能にする必要はありません。



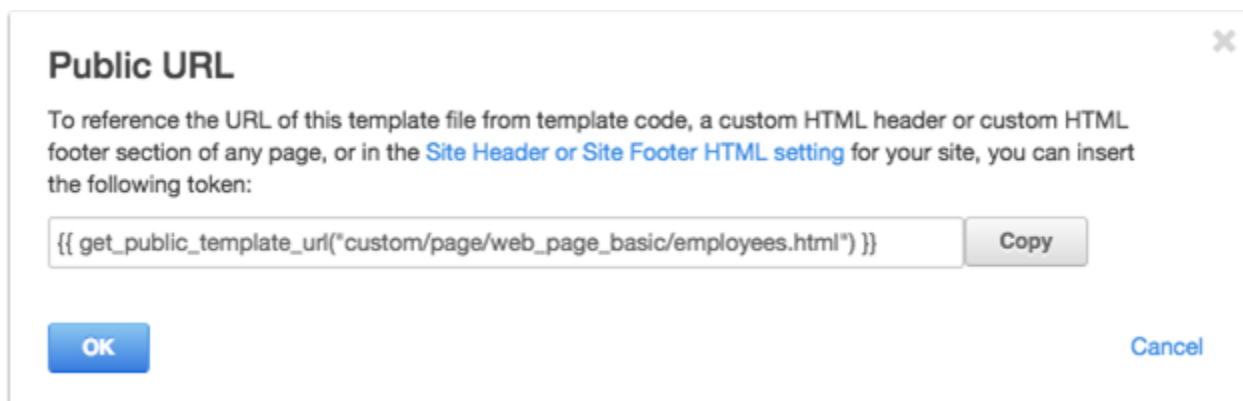
4. このファイルへのパスをコピーする

このファイルを別のテンプレートにインポートするには、デザインマネージャーにおけるファイルの相対パスをコピーする必要があります。パスをコピーするには、次のようになります。

- [アクション]>[公開 URL を取得]をクリックします。
- HubL タグをコピーする

デザインマネージャーパスは、このタグ内の引用符に囲まれたファイル構造です。必要なのはタグ全体ではなく、引用符に囲まれた部分だけです。たとえば次のとおりです。

"custom/page/web_page_basic/my_filename.html"



5. ファイルをテンプレートにインポートする

変数を別のテンプレートの中にインポートするには、次のようにします。

- これらの変数を使用する場所となるテンプレートレイアウトまたはコードテンプレートを開きます。
 - あらかじめコピーしたタグを、テンプレートの<head>の中に貼り付けます。テンプレートレイアウトでは、[編集]>[Head の編集]でこの操作を行うことができます。
 - 引用符に囲まれたパス (たとえば "custom/page/web_page_basic/my_filename.html") 以外のものをタグから削除してください
 - パスを含むインポートステートメントを追加します (下記の構文を参照) 。
 - インポートするファイルに変数名を付けます (下記の構文を参照) 。
- そうするには、ファイルパスの後に「as」を追加し、その後に変数名を指定します。以下の例では、インポートに「employees」という名前を付けます。

インポートタグの基本構文は次のとおりです。

```
{% import "path" as name_of_import %}
```

import の例

```
{% import "custom/page/web_page_basic/employees.html" as employees %}
```

このファイルを head タグ内にインポートすると、これらの変数をテンプレート全体で使用できるようになります。なお、テンプレート内で変数が呼び出される場所より下に import タグが存在する場合、その変数はレンダリングされないことに注意してください。

6. 変数またはマクロを呼び出す

これで、これらの変数をテンプレート内の他の場所で呼び出すことができます。含まれているファイルの変数の値を更新すると、マスターテンプレートが自動的に更新されます。また、上記のとおり、マスターテンプレートで使われるカスタムモジュール内でもこれらの変数がレンダリングされます。

- HubL
- 出力例

```
{{ employees.job('span', 'CEO') }} {{ employees.ceo }}  
{{ employees.job('span', 'CTO') }} {{ employees.cto }}
```

7. マスターテンプレートを公開する

最後に、マスターテンプレートを公開します。インポートされる html ファイル内の変数値を変更すると、変更後の値がマスターテンプレートとカスタムモジュールにも反映されます。

静的データベースの例

この手法を使用すると、簡単な静的データベースを作成してそれを反復処理できます。この例では、インポートされるファイルに、3つの製品からなるデータベースが含まれています。

インポートされるファイルでは、「in_stock」という変数がディクショナリのシーケンスとして定義されています。それぞれのディクショナリには、個別の製品に関する情報が含まれています。

```
{% set in_stock = [  
{'name': 'Toy', 'price': '20', 'size': 'Small', 'quantity': '3'},
```

```
{'name': 'Shirt', 'price': '15', 'size': 'XL', 'quantity': '10'},  
{'name': 'Bike', 'price': '200', 'size': 'Large', 'quantity': '6'}  
] %}
```

このディクショナリシーケンスを別のテンプレートにインポートした後、for ループを使ってこれらのカスタム変数を反復処理し、製品情報をマークアップの中に出力することができます。

コードの第 1 行の import タグでは、このファイルの内容が「products」という名前でインポートされることを指定します。このコード行を<head>の中、またはテンプレートの先頭付近に配置することをお勧めします。

次に、products.in_stock 内の各製品を反復処理する for ループを宣言します。それぞれの製品は、インポートされるファイル内の個別のディクショナリであるため、反復項目（たとえば product）に属性名（name、size、price など）を付加することで、任意の製品のプロパティにアクセスできます。

- . HubL
- . 出力例

```
{% import "custom/page/web_page_basic/products.html" as products %}  
{% for product in products.in_stock %}  
<h2>Name: {{ product.name }}</h2>  
<h3>Size: {{ product.size }}</h3>  
<h4>Price: ${{ product.price }}</h4>  
<h4>Number in stock: {{product.quantity }}</h4>  
{% endfor %}
```

この例をさらに拡張して、特定の条件を満たす製品のみを出力するよう、ループ内の if ステートメントをネストさせることができます。たとえば 30 ドル未満の製品を表示したい場合は、product.price を対象とする if ステートメントを使用できます。

- HubL
- 出力例

```
{% import "custom/page/web_page_basic/products.html" as products %}
{% for product in products.in_stock %}
{% if product.price < 30 %}
<h2>Name: {{ product.name }}</h2>
<h3>Size: {{ product.size }}</h3>
<h4>Price: ${{ product.price }}</h4>
<h4>Number in stock: {{product.quantity }}</h4>
{% endif %}
{% endfor %}
```

多言語の用語集の例

この手法は、ページの URL に基づいてテンプレート全体で読み込まれる多言語の文字列変数からなる用語集を作成するうえで役立ちます。[チュートリアル全体をここで確認](#)できます。

blog related posts HubL タグを使用する方法

related_blog_posts HubL タグを使用して関連するブログ記事のリストを作成する

`related_blog_posts` HubL タグを使用すると、さまざまなパラメーターに基づいて関連するブログ記事の動的なリストを作成できます。執筆者、投稿パス、および公開日による絞り込み機能を使用して、複数のブログにまたがる記事のリストを生成できます。開発者は、マクロを使用してモジュールの HTML 出力を指定することもできます。この HubL タグはブログ記事とページの両方で使用できます。このチュートリアルでは、`related_blog_posts` HubL タグのパラメーターと使用方法について説明します。

`related_blog_posts` HubL タグは、記事/ページレベルで編集可能なモジュールを生成せず、HubL で完全に設定されることに注意してください。

パラメーター

記事のリストは、以下のパラメーターに一致する記事（この HubL タグが表示された記事に関連する記事）と設定されたパラメーター値を比較して得られた関連性スコアから生成されます。必須のパラメーターはありませんが、パラメーターを指定することで、返される記事を細かく制御できます。カンマ区切りパラメーターの場合は、設定する値が多いほど返されるリストの範囲が広がります。*post_formatter* パラメーターを使用すると、モジュールの HTML 出力を生成する [マクロ](#) を指定できます。パラメーターの完全なリストと既定の HTML 出力の例については、[related_blog_posts](#) の仕様を参照してください。

パラメーター名	タイプ	説明	既定値
limit	number	リストするブログ記事の最大数。	3
blog_ids	「default」または	リストする記事を含むブログの ID。	なし

ブログ
ID

tags	string	記事に関連性があるかどうかを判定するために使用するタグ（カンマ区切り）。ブログ記事にこれらのタグのいずれかまたは類似のタグがある場合、記事の関連性が高くなり、リストでのランキングが向上します。	なし
------	--------	--	----

blog_authors	string	リストする記事の執筆者の名前（カンマ区切り）	なし
--------------	--------	------------------------	----

blog_post_ids	string	リストする関連ブログ記事を見つけたときに使用するブログ記事の ID（カンマ区切り）。このパラメーターは、ブログ記事に表示されるウィジェットをページにも表示する場合にのみ使用してください。既定値はウィジェットが表示される記事です。	なし
---------------	--------	--	----

post_formatter	string	返されるブログ記事を表示するためのカスタムマクロの名前。このマクロには、フォーマットするブログ記事オブジェクト、ブログ記事の繰り返し回数、および結果に含まれるブログ記事の合計数という3つのパラメーターが渡されます。指定されていないか、「default」に設定された場合は、組み込みのフォーマッターを使用して各記事がフォーマットされます。	なし
path_prefixes	string	リストする記事のURLパスまたはサブディレクトリ（カンマ区切り）。ブログ記事のパスに類似するプレフィックスがある場合は、記事の関連性が高くなり、リストでのランキングが向上します。	なし
start_date および end_date	date/time	期間内に公開された記事に絞り込むことができます。	なし
blog_post_override	string	他のパラメーター値やフィルターに関係なく、返されたリストに常に表示されるブログ記事のID（カンマ区切り）。	なし

1つの記事に対して `related_blog_posts` HubL タグを使用する場合は、複数のブログ記事の場合のように `blog_post_ids` パラメーターを指定することはできません。既定値はウィジェットが表示される記事です。

以下は、`related_blog_posts` HubL タグの使用例です。

特定の執筆者の投稿を複数のブログにまたがって表示する

この例では、指定された3人の `blog_authors` のいずれかによって書かれた記事のリストを、2つの異なるブログにまたがって生成します。

```
{% related_blog_posts blog_ids="3241539189,3261083894", limit=6,
blog_authors="John Smith,Joe Smith,Frank Smith" %}
```

「sales enablement」タグを含む記事を、公開日の期間を限定して表示する

この例では、特定のブログ記事に関連し、「sales enablement」タグを含む10件の記事のリストを、公開日の期間を限定して作成します。この例では、`blog_post_ids` パラメーターをページで使用するために指定します。

```
{% related_blog_posts blog_post_ids="3267910554", limit=10, tags="sales
enablement", start_date="2018-02-05", end_date="2018-06-10" %}
```


共通のパスを持つ記事を表示し、マクロを使用して HTML 出力を制御する

この例では、`path_prefixes` パラメーターを使用して共通のパスを持つ 5 件の記事のリストを生成します。また、マクロを使用して記事リストの HTML 出力を制御するため、`post_formatter` パラメーターを指定します。

```
{% macro blog_post_formatter(blog_post, index, count) %}
<div class="related-blog-item">
  <span>Related Post {{ index }}/{{ count }}</span><br>
  <a class="related-blog-title" href="{{blog_post.url}}">
    <span>
      {{ blog_post.name }}
    </span>
  </a>
  <div class="hs-related-blog-byline">
    by
    <span class="related-blog-author">{{ blog_post.blogAuthor }}</span>
    <span class="related-blog-posted-at"> posted on </span>
    <span class="related-blog-date">{{ blog_post.publishDate|datetimeformat('%B %e, %Y') }}</span>
  </div>
  <p class="related-blog-post-summary">
    {{ blog_post.post_summary }}
    <a href="{{ blog_post.url }}">Read more</a>
  </p>
  <div class="related-blog-tags">
    {% if blog_post.tagList|length > 0 %}Tags: {% endif %}{{ blog_post.tagList|join(", ") }}
  </div>
</div>
```

```
</div>
{% if blog_post.featured_image %}
  
{% endif %}
</div>
{% endmacro %}

{% related_blog_posts limit=5, path_prefixes="/marketing,/sales"
post_formatter="blog_post_formatter" %}
```

**HubSpot CMS エディターで CSS/JS が正常に動作している
ことを確認する**

HubSpot CMS エディターで CSS/JS が正常に動作しているこ とを確認する

HubSpot CMS エディターではコンテンツ作成者がインライン編集を行って、WYSIWYG (見たとおりのものが結果に反映される) 形式でコンテンツを作成、編集できるようになっています。そのために、このエディターでは iframe を使用して、モジュールとテンプレートのコードと HubSpot アプリケーションの CSS/JS の両方を含むウェブページのプレビューを読み込みます。けれどもテンプレートまたはモジュールの

CSS/JS が原因で、エディターで問題が発生することがあります。この記事では、こうした問題を回避するためのベストプラクティスについて説明します。

エディター内でテストする

重要な点として、開発したアセットを配信する前に、HubSpot のコンテンツエディターでテストしてください。エディター内でテストすると、エディターと対立するスタイル設定を特定するにも、シームレスなユーザーエクスペリエンスを実現するにも効果的です。

以下に、エディター内で実行できるテストをいくつか取り上げます。

- [スタイル] ドロップダウンメニューのオプション (フォント、テキストサイズ) とツールバーのオプション (位置合わせ、色) を使用してテキストの書式を設定する
- リッチテキストツールバーのオプション (埋め込み、リンク、画像、パーソナライズトークン) をクリックして挿入する
- インラインエディター内でリッチテキスト要素のプラグイン (埋め込み、リンク、画像、パーソナライズトークン) をクリックし、UI が正常に動作して期待通りに表示されることを確認する

具体的にすること

大まかに言うと、カスタムの JavaScript または CSS が十分に具体化されていないと、コンテンツエディターで問題が発生する可能性があります。この問題は、コードの作成方法によって異なる形で表われます。また、カスタムコードがエディターを干渉するリスクもあります。一例として、次の関数を見てください。

```
$('#body').click(function(event){  
  
  $('#blog-listing-wrapper .post-listing .post-  
item').siblings().find('.right').removeClass('social-active');  
  
  event.stopPropagation();  
  
});
```

この関数は、body 要素がクリックされると実行されます。このコードは `event.stopPropagation()` を呼び出すため、イベントの吹き出しはドキュメントまたはウィンドウにポップアップされません。これらの要素にイベントリスナーが関連付けられている場合、イベントリスナーのコードは実行されません。上記のコードでの問題は、要素がクリックされるたびに click イベントハンドラーが実行されることです。インラインエディターは React を介して click イベントハンドラーを window 要素に追加するため、ハンドラーが実行されるとインラインエディターで問題が発生します。すべてのクリックをリッスンするのではなく、必要なときにだけ click イベントハンドラーを追加したほうが適切です。たとえば、ユーザーがサイドメニューを開くためのボタンをクリックした時点で click イベントハンドラーを追加し、サイドメニューが開いた時点でハンドラーを削除します。

また、`label` のような汎用のセレクターを使用する場合も、CSS で問題が発生することがあります。可能であれば、ウェブページの一部だけに固有のセレクター (`.hs-form label` など) を使用するようになしてください。

!important タグの使用を回避すること

`!important` は、スタイルルールを他のものより優先させるために使用するタグです。ウェブコンテンツを作成するときに、`label` や `input[type="text"]` のような基本的な要素セレクターで `!important` を使用するのはいくつかのベストプラクティスではありません。けれども、選択したスタイルルールよりも具体的なものによってそのルールがオーバーライドされる場合、通常はこのタグが使用されます。`!important` を使用すると、選択したルールが優先され、そのルールが他よりも具体性に欠けるとしても所定のアセットに適用されます。

`!important` を使用するのではなく、セレクターの具体性を優先させて、このタグを使用しなくても済むようにしてください。そうすることにより、ターゲット要素以外にも適用される包括的なルールを設定せずに、スタイルを設定する対象の要素を特定できます。また、HotSpot の [ボイラープレート CSS ファイル](#) を利用して、CSS の塗り足し問題を回避するために使用すべきセレクターを確認することもできます。

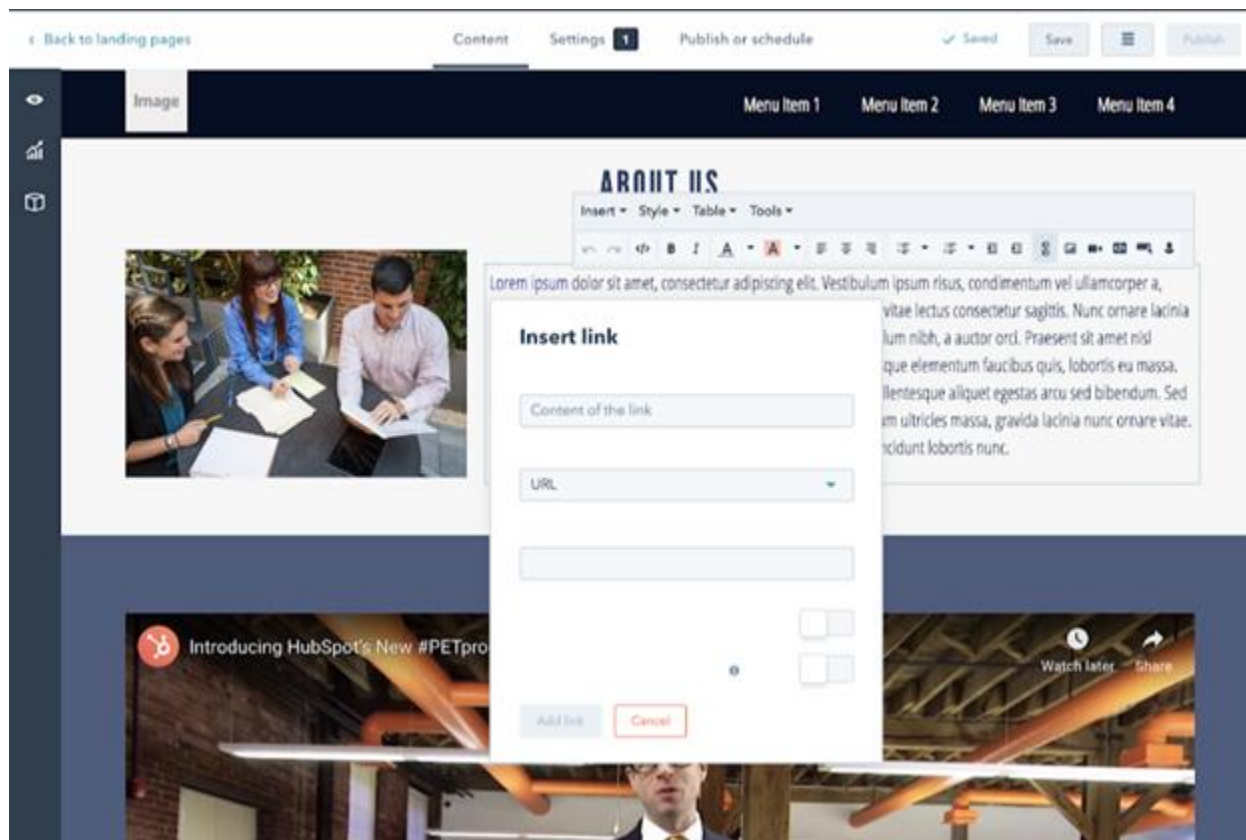
!important の使用を回避する方法

場合によっては、次のように CSS で `<label>` 要素で `!important` が使用されることがあります。この場合の目的は、フォーム内の `<label>` 要素のすべてが白になるようにするためです。

```
label, legend {  
  color: white !important;  
}
```

その目的に反し、これではコンテンツエディター内のすべての `<label>` タグが白になってしまいます。要素をスタイルルールのセレクターとして使用するのはいくつかの広義であるためです。このスタイルルールは、リンク挿入プラグインで使用されている

<label>要素を除外しないことから、プラグインでフィールドのラベルが欠落しているかのように見えることになります。



以下のスタイルルールのように、より具体的なセレクタ-を使用すると、白を追加する対象のフォームモジュールのラベルだけに白を追加できるため、コンテンツエディター内の他のラベルは影響を受けません。フォームのスタイルを設定する際に使用できるセレクタ-について理解を深めるには、開発者向けの[この優れたリソース](#)をご覧ください。

```
.hs-form label, .hs-form .hs-form-field > label, .hs-form .hs-field-desc {  
  color: white;  
}
```

.hs-inline-edit と window.hsInEditor を使用 すること

エディターでは `iframe` を使用して、コンテンツのプレビューを HubSpot のコンテンツエディターに読み込みます。この `.hs-inline-edit` という `iframe` は、`<html>` に追加されるクラスであり、`window.hsInEditor` は「true」に定義されます。インラインエディターを初期化するときに、開発者はこの 2 つの項目を手掛かりに、コンテンツが HubSpot コンテンツエディターに読み込まれるかどうかを確認できます。

`.hs-inline-edit` を使用すると、ウェブページがエディターに読み込まれるときに適用されない CSS を作成して、CSS の塗り足しを回避できます。インライン編集に影響する問題が JavaScript にある場合は、`window.hsInEditor` 変数を使用して、エディター内でコードが実行されるときに問題のあるコードの部分を無効にすることができます。

.hs-inline-edit を使用する方法

エディター内でのみルールを適用するセレクターの前に `.hs-inline-edit` を追加します。

たとえば、次の CSS では疑似セレクター `:not()` を利用しています。

```
:not(.hs-inline-edit) label {  
  
  color: white;  
  
}
```

別の方法として、HotSpot 固有のクラスを利用することもできます。

たとえば、ページ内に固定ヘッダーがあり、エディター内のリッチテキストツールバーがその固定ヘッダーの下に隠れてしまう場合、ウェブサイトの UI レイヤーを最上位の z-index に配置していることが考えられます。通常、この配置はベストプラクティスではありません (どうしてもそうしなければならない場合を除き)。これは、エディター内では特に大きい問題になります。コンテンツがインラインコントロールの上にレンダリングされる可能性があるためです。最善の結果を得るために、ウェブサイトの UI レイヤーを [X] z-index の下に維持するようにしてください。それには、セレクター `.hs-inline-edit` の後に固定ヘッダーのセレクター (`header`) を続けて、z-index の値を 2147483599 にします。

```
.hs-inline-edit .header {  
  
    z-index: 2147483599;  
  
}
```

window.hsInEditor を使用する方法

エディターが読み込まれた後に JavaScript 関数が起動されるようにするには、`window.hsInEditor` を使用して何らかの処理を行うかどうかを決定する条件を含めると役立ちます。以下に記載するように、jQuery ドキュメントの ready ハンドラーを利用することを検討してください。

```
jQuery(document).ready(function($) {  
    if (window.hsInEditor) {  
        return;  
    }  
});
```



```
}  
// so other stuff  
$('.some_widget').fancyThing();  
});
```

ブログ記事のセットを出力する

ページテンプレートでのブログ記事セットの出力

ブログ記事は通常、ブログテンプレートリストまたは個別の投稿レイアウトで表示されますが、ブログ以外のテンプレートで記事セットを出力する場合があります。たとえば、サイトのホームページに最新記事のセットを出力することがあります。

ブログ以外のテンプレートで記事を出力するには、RSS モジュールまたは HubL 関数を使用します。HubSpot の標準ブログコンテンツモジュールと標準ブログコンテンツマークアップについての詳細は、[こちらの記事をご覧ください](#)。

RSS モジュール

[RSS リストモジュール](#)は、HubSpot または外部 RSS フィードからの記事の情報を出力する標準モジュールです。RSS モジュールはテンプレートレイアウト内で使用でき、

また [HubL モジュール文](#) でコードテンプレートに RSS モジュールを追加できます。HubSpot でホストされている RSS フィードの RSS モジュールでは、サムネイル画像がサポートされています。

HubL 関数

HubL 関数を使用して、最新記事、特定のトピックに関する最新記事、特定の執筆者による最新記事、最も人気の高い記事、および関連ブログ記事のセットを取得できます。変数に記事のシーケンスを格納することで、一連の記事を繰り返し、カスタムマークアップで各記事の変数を出力できます。

記事のセットを取得する関数を以下に示します。

- [blog_popular_posts](#)
- [blog_recent_author_posts](#)
- [blog_recent_posts](#)
- [blog_recent_topic_posts](#)
- [related_blog_posts](#)

この手法ではページ番号がサポートされていないことに注意してください。HubSpot のページ番号設定は、標準ブログリストにのみ適用されます。HubL 関数でレンダリングされたコンテンツにページ番号を付ける場合は、JavaScript を使用して記事の静的セットにページ番号を付けます。

```
{% set posts = blog_recent_posts('default', 5) %}
{% for post in posts %}
  <h3><a href="{{ post.absolute_url }}">{{ post.name }}</a></h3>
  
{% endfor %}
```

上記の例では、5 件の最新記事が既定のブログから取得され、posts という名前の変数に格納されます。繰り返しごとにカスタム記事マークアップを出力するため [for ループ](#) が使用されています。

ループ内では任意の [blog_post 変数](#) を使用できますが、"content" をループ項目の名前に置き換える必要があります。上記の例では content.name が post.name になります。

また、異なるブログからの複数の記事セットをまとめて 1 つのリストにすることもできます。詳細は、[こちらの記事をご覧ください](#)。

スニペット

カスタムモジュールでフォームモジュールを作成する

カスタムモジュールでフォームモジュールを作成する方法

新しいデザインマネージャーでは、既定のフォームモジュールが改良され、開発者がより細かく制御できるようになりました。たとえば、必ずしもすべてのフォームを Salesforce に接続する必要はなく、必ずしもすべてのマーケターが通知用の E メールアドレスを上書きできる必要もありません。この新しい環境では、既定のモジュールは 1 つの HubL タグの完全なレンダリングではありません。つまり、既定のフォームモジュールは単なるフォームではなく、フィールドのコレクションです。

フォームフィールドとフォームモジュール の違い

フォームフィールドには以下の項目が含まれています。

- フォームセレクトターのドロップダウン
- 送信後のアクションのラジオボタン
- サンキューメッセージのフィールド
- フォームのプレースホルダー

上記の項目に加えて、既定のフォームモジュールには次の項目が含まれています。

- タイトル
- 新しい E メールアドレスに通知を送信するためのブール値
- E メールアドレスのセレクトター
- ワークフローのブール値
- ワークフローのセレクトター
- フォローアップ E メール of ブール値
- フォローアップ E メール of セレクトター
- Salesforce キャンペーンのフィールド

フォームのフィールドは非常にシンプルのため、チームのニーズに合わせてカスタマイズできます。

カスタムモジュールで完全なフォームを作成するには、既定のフォームモジュールを複製する方法と手動でフォームフィールドをカスタムモジュールに追加する方法の2つがあります。

方法 1：既定のモジュールを複製する

この方法は、手動でフォームモジュールを作成せずに始められる簡単な方法です。

1. デザインマネージャーで「フォーム」を検索します
2. 既定のモジュールを複製します
3. HTML/CSS/フィールドをモジュールに追加します

方法 2：手動でフォームフィールドを追加する

この方法は、各モジュールフィールドがどのように連携して動作するかを理解したい場合に適しています。

1. デザインマネージャーで [新しいカスタムモジュールを作成](#) します。

2. 既定のフォームモジュールからベースとなる HubL をコピーします。

```

{% module "form"
    path="@hubspot/form"
    form_key='{{ name }}',
    form_to_use='{{ module.form.form_id }}',
    title='{{ module.title }}',
    follow_up_type_simple='{{ module.follow_up_type_simple }}',
    simple_email_for_live_id='{{ module.simple_email_for_live_id }}',
    follow_up_type_automation='{{ module.follow_up_type_automation }}',
    form_follow_ups_workflow_id='{{ module.form_follow_ups_workflow_id }}',
    response_response_type='{{ module.form.response_type }}',
    response_redirect_id='{{ module.form.redirect_id }}',
    response_redirect_url='{{ module.form.redirect_url }}',
    response_message='{{ module.form.message }}',
    notifications_are_overridden='{{ module.notifications_are_overridden }}',

    notifications_override_email_addresses='{{ module.notifications_override_email_addresses }}'

    gotowebinar_webinar_key='{{ module.form.gotowebinar_webinar_key }}'
    sfdc_campaign='{{ module.sfdc_campaign }}'
%}

```

3. このコードを新しいカスタムモジュールに貼り付けます。

GoToWebinar インテグレーションがない場合は、gotowebinar の行を削除できます。

4. それぞれの内部 (変数) 名を使用して以下のフィールドを作成します。

フィールドタイプ	変数名	目的
パス	@hubspot/form	これにより、フォームのベースが作成され、名前フィールドとフォーム ID フィールドにデータが入力されます。
テキストフィールド	title	フォームにタイトルが追加されます。このフィールドは任意です。フォームで使用されている部分を削除してください。
ブール値	follow_up_type_simple	フォームのフォローアップ E メールを送信するかどうかを切り替えます
フォローアップ E メール	simple_email_for_live_id	フォローアップ E メール用の E メールを選択します

ブール値	follow_up_type_automation	コンタクトをワークフローに登録するかどうかを切り替えます
ワークフロー	form_follow_ups_workflow_id	コンタクトに登録するワークフローを選択します
ブール値	notifications_are_overridden	フォーム通知の既定のアドレスを上書きするかどうかを切り替えます
Eメールアドレス	notifications_override_email_addresses	既定値を上書きするEメールアドレス
Salesforce キャンペーン	sfdc_campaign	コンタクトを関連付けるSalesforceキャンペーンを選択します。

これで、フォームモジュールのすべての機能を利用できます。エディターにはフィールド名が表示されるため、マーケターにとってわかりやすいフィールド名を付けてください。

ドラッグ&ドロップエリアを作成する

デザインマネージャー

UI の概要

UI の概要

デザインマネージャーは、ファインダー、エディター、およびインスペクターという 3 つのセクションで構成されています。各セクションが連携して動作するので、ツールのすべての機能とオプションに簡単かつ直感的にアクセスできます。以前のデザインマネージャーをよく利用されていた場合には、可能な限り多くのメニューとダイアログが排除されたことがお分かりになると思います。

以下のセクションでは、UI の各エリアについて簡単に説明します。その後、UI の各エリアについてアセットの作成と編集の点から詳しく説明します。

ファインダー

Search assets 

New ▾ View ▾ Actions ▾

 Coded files

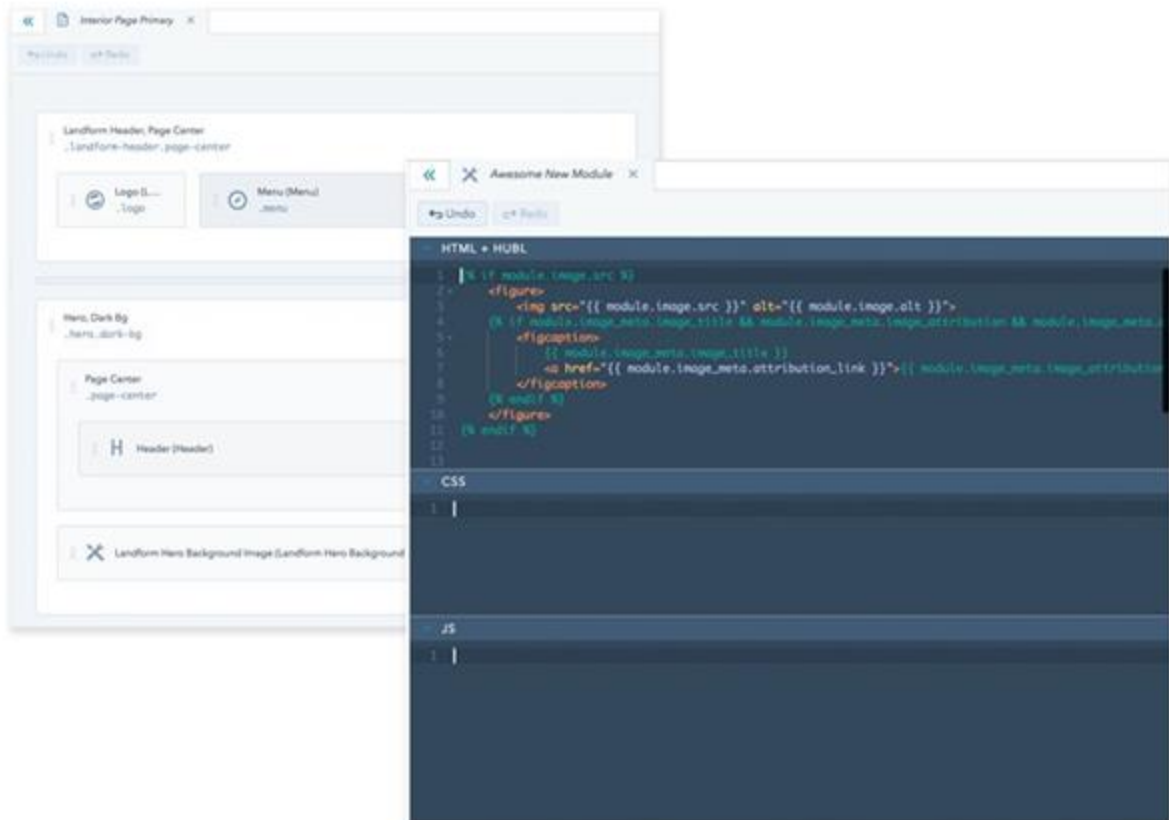
 Custom modules

 Global content

 Templates

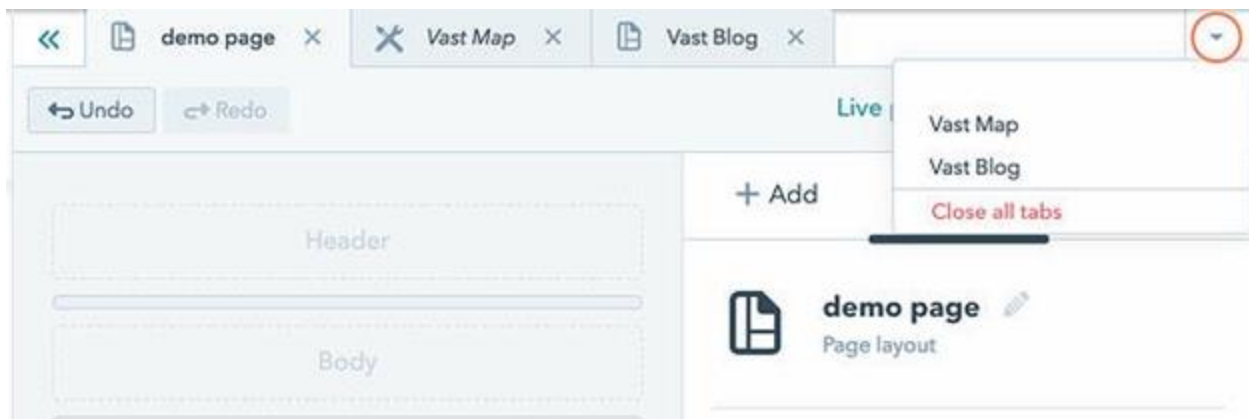
ファインダーは、コンピューターの OS に相当するアセット操作機能を提供します。作成/削除や検索/ソートなどの一般的なファイルやフォルダーの操作は、ファインダーメニューから使用するか、または各アセットでコンテキストメニューを右クリックして使用することができます。

エディター



エディターはデザインマネージャーの中心的なセクションであり、作業の大部分をエディターで行います。処理するアセットに基づいて、ドラッグ&ドロップインターフェイスまたはコードエディターのいずれかになります。

エディターのすぐ上にあるタブ形式のインターフェイスでは、開いているアセットを切り替えることができます。これらのタブをドラッグして並べ替えることができます。また、右端にあるタブメニューでは、すべてのタブを閉じる操作や、特定のタブを選択する操作を実行できます。

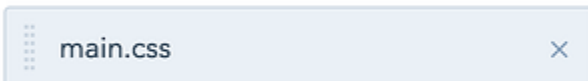


インスペクター



✓ HEAD AND BODY OPTIONS

Linked stylesheets [Add stylesheet](#) ▾



Domain stylesheets



Additional <head> markup



Body classes

インスペクターでは、テンプレートやモジュールのコンポーネントを追加して編集できます。インスペクターは、実行しているタスクとファイルのタイプのコンテキストに依存します。例：

- ドラッグ&ドロップエディターでコンポーネントが選択されている場合、インスペクターにはそのコンポーネントのすべての機能とオプションが表示されます。
- ドラッグ&ドロップエディターでコンポーネントが選択されていない場合、インスペクターにはテンプレート自体のオプションが表示されます。
- モジュールを編集する場合、インスペクターでフィールドの編集と追加ができます。
- HTML、CSS、または JS ファイルを編集するときには、インスペクターは表示されません。

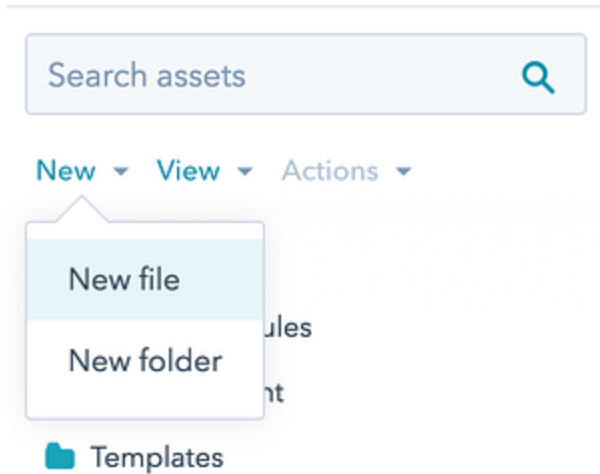
ファインダーの使用

ファインダーを使用する

ファインダーをすぐに使いこなせるよう、CMS で最もよく使われるアセットのタイプとして、ドラッグ&ドロップテンプレートを作成する手順を説明します。

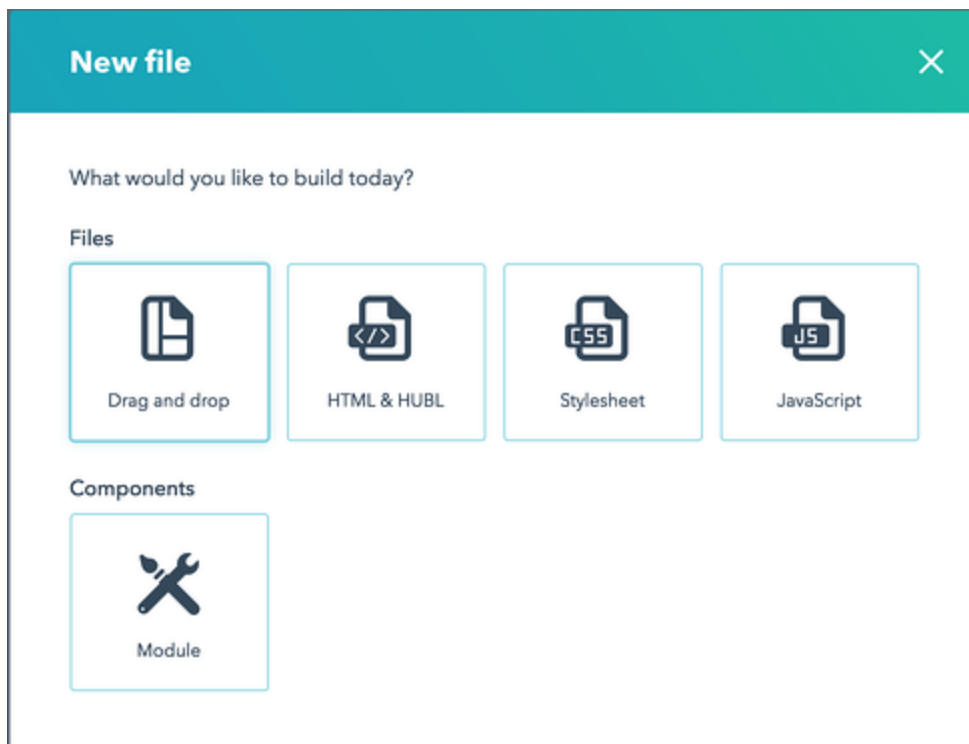
1. ファイルを作成する

新しいファイルを作成するには、[新規作成] ボタンをクリックし、表示されるドロップダウンから [新規ファイル] を選択します。



2. ファイルタイプを選択する

ダイアログボックスで [ドラッグ&ドロップ] を選択します。このダイアログボックスに示されるファイルタイプの詳細については、[リファレンスページ](#)をご覧ください。



3.テンプレートを設定する

次のダイアログでは、[テンプレートタイプ] ドロップダウンで [ページ] が選択された状態で、テンプレートの名前を入力します。このダイアログボックスで選択可能なテンプレートタイプの詳細については、[リファレンスページ](#)をご覧ください。

Set up your new drag and drop template ×

[← Back to file types](#)

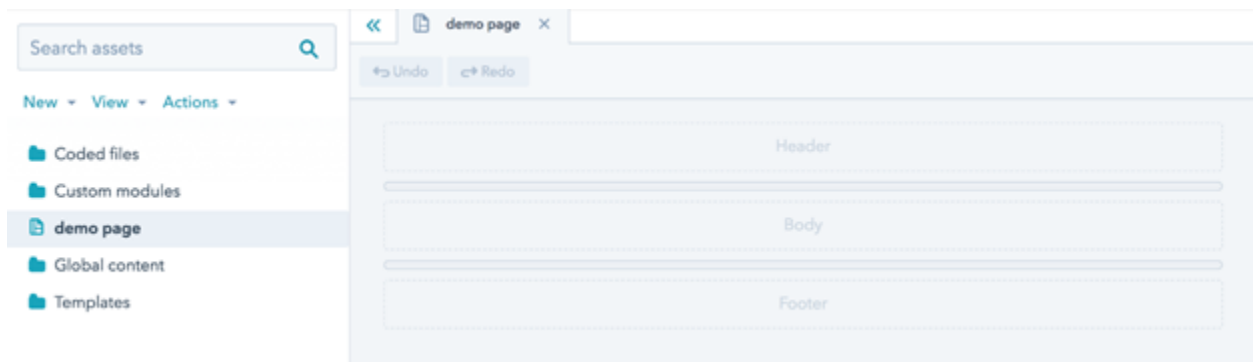
New file type
📁 Drag and drop

Template type
Page ▼

File name *
demo page

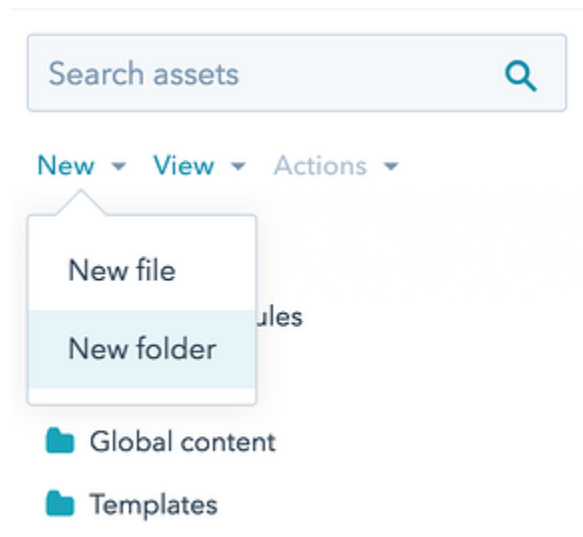
Create **Cancel**

[作成] をクリックすると、新しいドラッグ&ドロップページテンプレートが作成され、以下のようにファイnderにアルファベット順で登録されます。

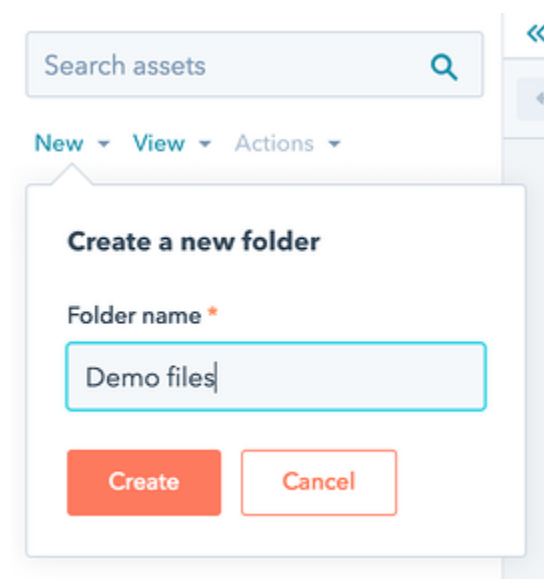


4.新しく作成したテンプレート用のフォルダーを作成する

コンピューターのオペレーティングシステムに含まれるファイルシステムと同様に、デザインマネージャーでは任意の方法でファイルを整理できます。新しいフォルダーを作成するには、[新規作成] ボタンをクリックし、表示されるドロップダウンから [新規フォルダー] を選択します。









表示されるダイアログで、新しく作成するフォルダーの名前を入力します。ファインダーでアセットを右クリックすることで、いつでも簡単にファイルやフォルダーの名前を変更できます。



5.新しいテンプレートを新しいフォルダーにドラッグする

New ▾ View ▾ Actions ▾

-  Coded files
-  Custom modules
-  Demo files
 -  demo page
-  Global content
-  Templates

インスペクターおよびエディターの使用

インスペクターとエディターを使用する

インスペクターとエディターでは、すべてのテンプレートとコンポーネントのオプションに簡単にアクセスできます。このセクションでは、テンプレートの設定を説明してから、コンポーネントを追加および編集する手順を説明します。このページの手順で例と

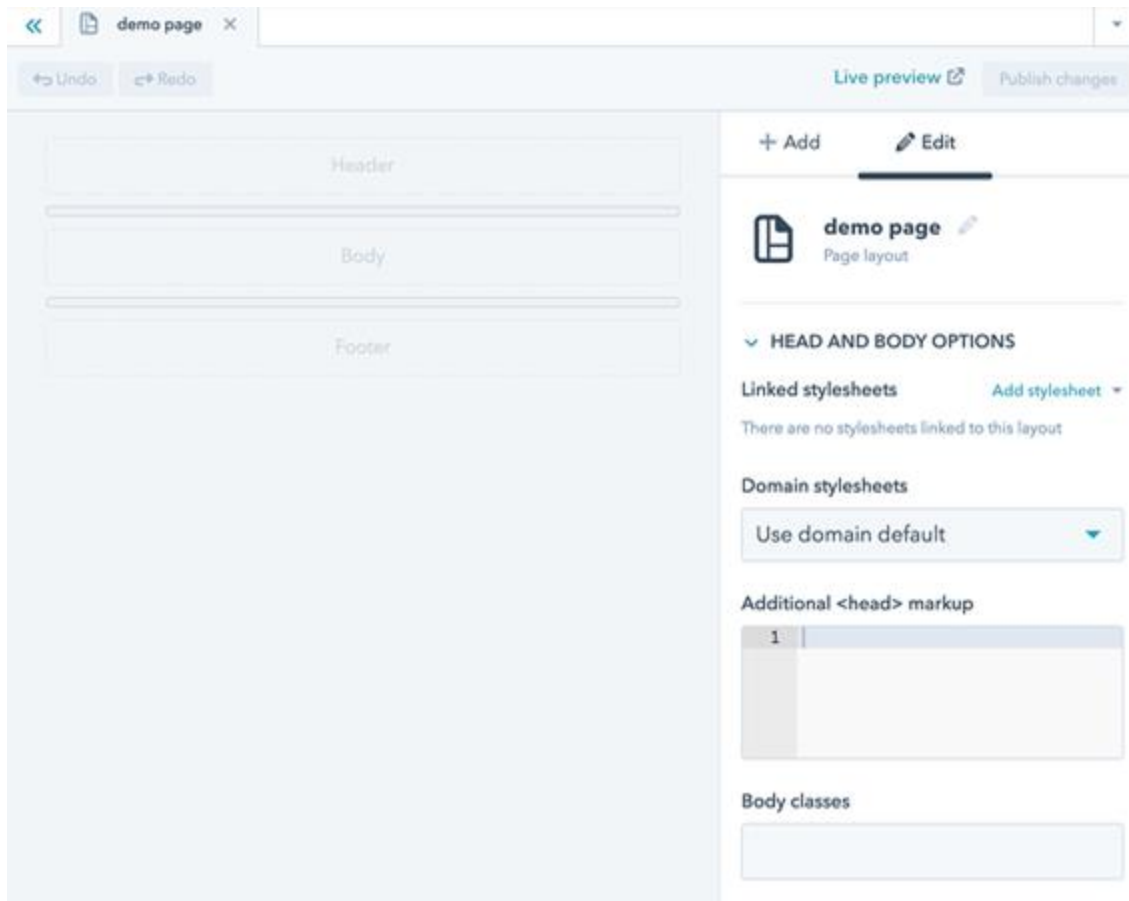
して使用するテンプレートは、[ファインダーを使用する](#)のページで説明している手順に従って作成できます。

テンプレートの設定を編集する

インスペクターはコンテキストに依存しています。ドラッグ&ドロップテンプレートの場合、インスペクターにはエディターで現在選択されているコンポーネントに関連するオプションが表示されます。コンポーネントが選択されていない場合、インスペクターにはテンプレートのオプションが表示されます。

以下のスクリーンショットに、このページで使用するデモテンプレートが示されています。まだコンポーネントを1つも追加していないので、エディターは空です。インスペクターには、テンプレートのオプションが表示されます。

具体的には、インスペクターではリンク済み CSS/JS に簡単にアクセスできるだけでなく、本文タグ、見出しマークアップ、ドキュメントの一番下のマークアップにもアクセスできます。また、ここからファイル名を変更することもできます。

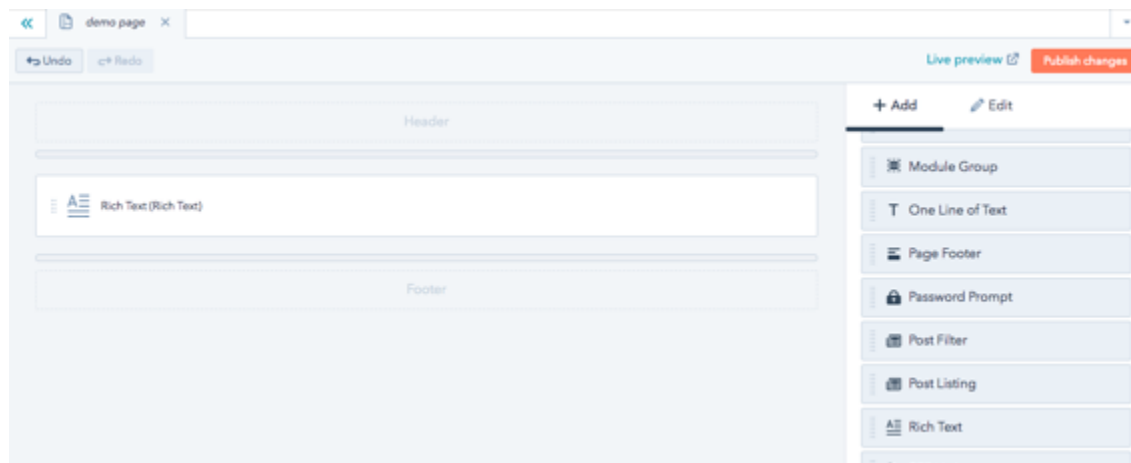


コンテンツを追加して編集する

このセクションでは、エディターとインスペクターの使い方を理解するために、コンポーネントを追加して編集するプロセスを説明します。使用できるコンポーネントは多数あるので、ここでは表面をかじるだけにすぎません。使用可能なコンポーネントの詳細については、[リファレンスページ](#)をご覧ください。

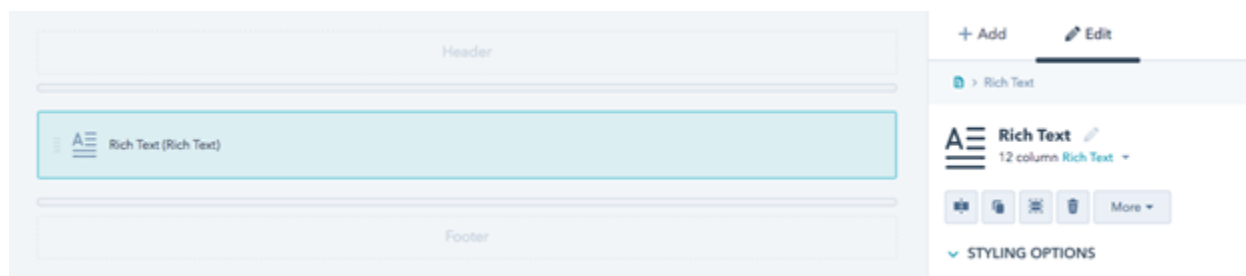
1. リッチテキストモジュールを追加する

インスペクターの [追加] カラムが選択された状態で、使用可能なコンポーネントのリストをスクロールダウンしてリッチテキストモジュールを見つけます。このモジュールをエディターの [本文] セクションにドラッグ & ドロップします。



2. エディターで新しく追加したリッチテキストモジュールを選択する

エディターで新しく追加したリッチテキストモジュールを選択します。インスペクターが編集モードに切り替わり、以下のようにこのコンポーネントに関連するオプションが表示されます。このスクリーンショットは、トップセクションを示すために意図的にトリミングされています。最初にトップセクションのオプションに注目しましょう。その後で、残りの部分を調べます。



トップセクションのオプションでは、このコンポーネントに名前を付けられる他、コンポーネントを分割、複製、グループ化、削除できます。[さらに表示...] ドロップダウンを使用すると、編集集中のコンポーネントのタイプに固有のオプションにもアクセスできます。

3. リッチテキストモジュールを分割する

[モジュールを分割] ボタンをクリックします。エディター内でリッチテキストモジュールが 2 つに分割されて横に並んだ状態になります。

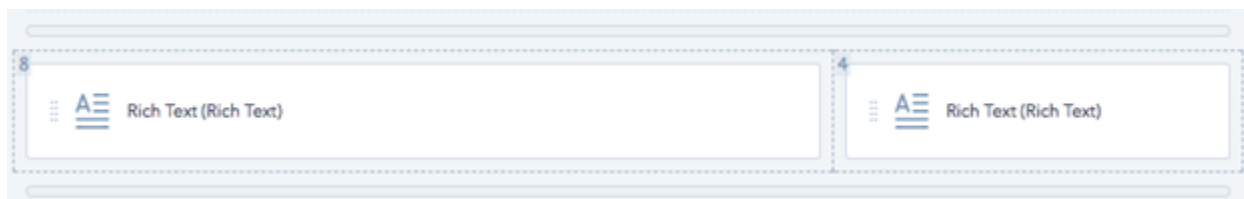
4.2 つのモジュールの間にマウスポインターを合わせる

エディターで、分割された 2 つのモジュールの間にマウスポインターを合わせます。それぞれのモジュールを囲む点線が表示され、各モジュールの左上隅に小さい 6 の数字が表示されます。この線は、12 列のグリッドでの列の幅を示します。この例の場合、それぞれのモジュールが 6 列にまたがります。つまり、各モジュールが幅の合計の半分を占めます。



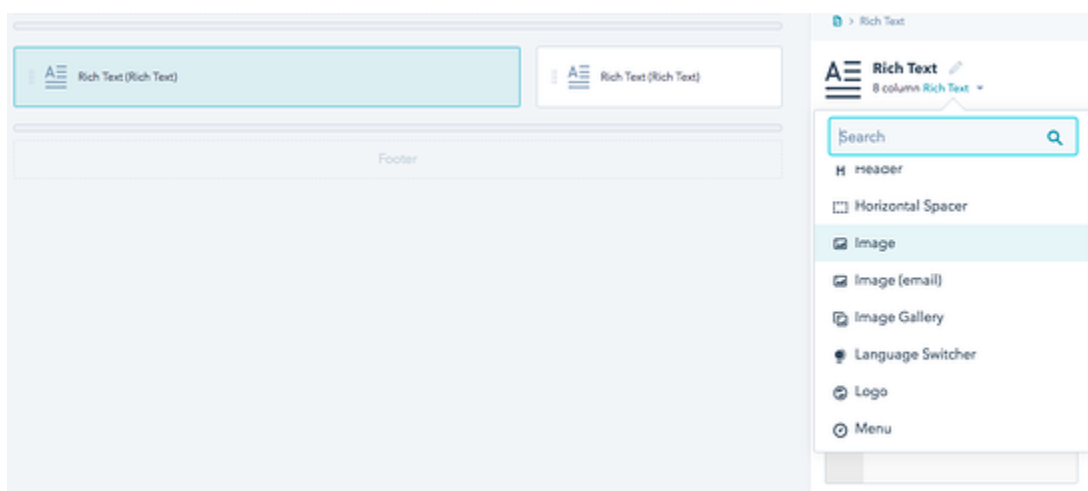
5. ドラッグして列の幅を変更する

2つの列の左上隅に示される数字がそれぞれ8と4になるまで、クリックした状態で右方向にドラッグします。これで、使用可能な幅の3分の2を左側のリッチテキストモジュールが占め、残りの3分の1を右側のリッチテキストが占めるようになります。



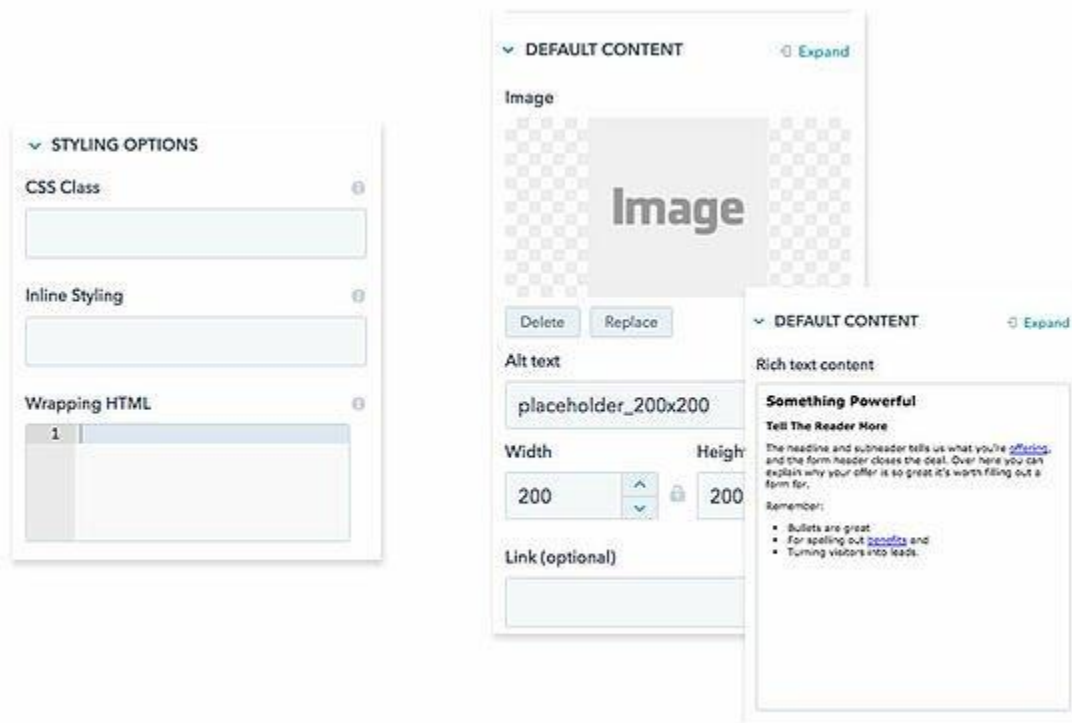
6. 左側のリッチテキストモジュールを画像モジュールに切り替える

左側のリッチテキストモジュールをクリックして選択します。このモジュールを画像モジュールに変更するために、インスペクターでコンポーネントタイプのドロップダウンをクリックし、[画像]を選択します。これで、左側のリッチテキストモジュールが画像モジュールに変更されます。



7. モジュールごとのインスペクターオプションを比較する

インスペクターでスクロールダウンして、これまでに説明したボタンの下にあるオプションを調べます。次に、エディターでリッチテキストモジュールを選択し、このモジュールのオプションを調べます。2つのモジュールはスタイルオプションを共有していますが、既定のコンテンツには各モジュールに固有のフィールドが反映されます。



スタイルオプション

[スタイルオプション] セクションでは、コンポーネントにレンダリングされるマークアップにクラスを追加したり、マークアップにインラインスタイルを追加したりできます。また、コンポーネントをカスタムHTMLでラップすることもできます。各フィー

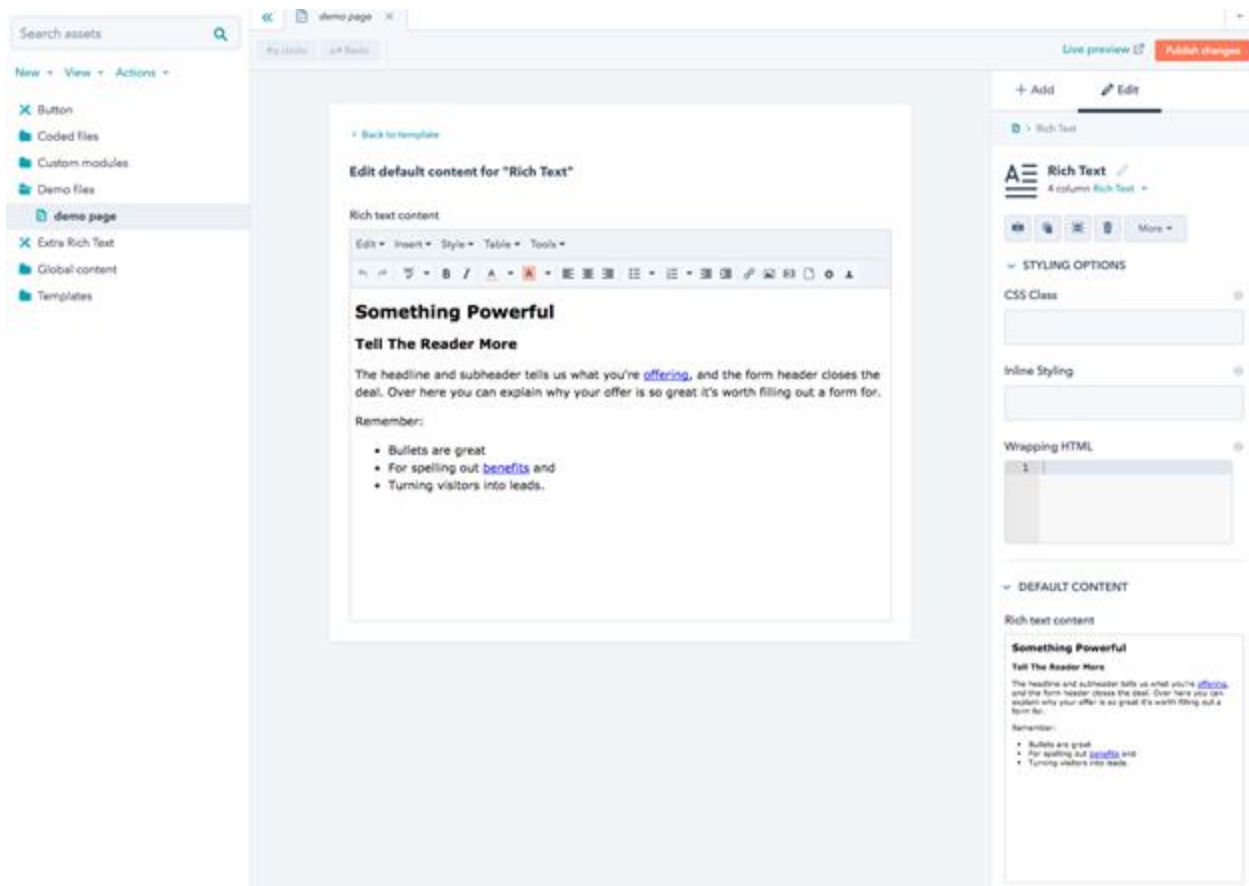
ルドの右上隅にある情報アイコンにマウスポインターを合わせると、詳細と構文を説明するヘルプが表示されます。

既定のコンテンツ

既定のコンテンツは、ユーザーがこのテンプレートを使用してページ作成するとエディターに表示されるコンテンツです。テンプレートとユーザーに応じて、このコンテンツは示唆的コンテンツ、指示的コンテンツ、またはフォールバックコンテンツになります。

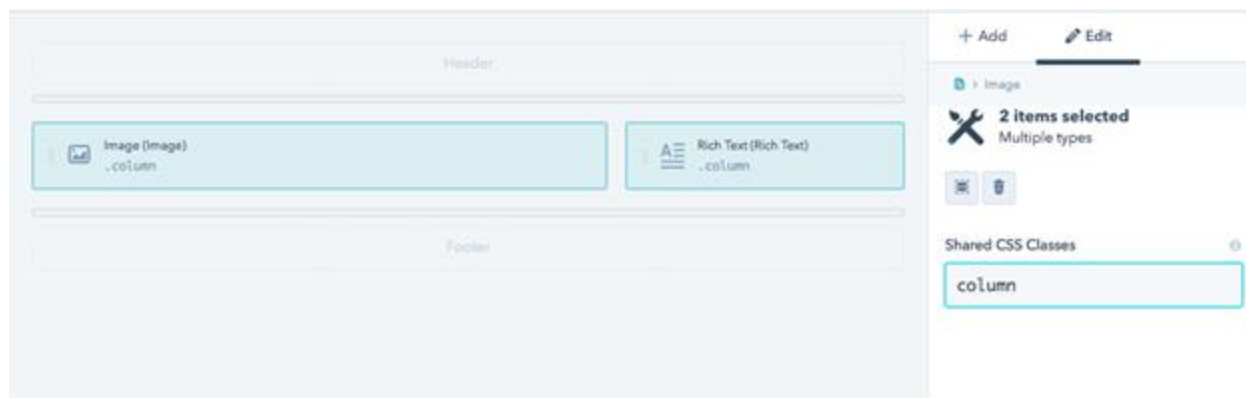
8. リッチテキストモジュールの [既定のコンテンツ] セクションを展開する

エディターでリッチテキストモジュールが選択された状態で、[既定のコンテンツ] セクションの右上隅にある [展開] ボタンをクリックします。インスペクターの [既定のコンテンツ] セクションがエディターのサイズ全体に拡大され、画面での表示面積が大きくなります。テンプレート全体のビューに戻るには、[テンプレートに戻る] ボタンを使用します。



9. エディターで両方のモジュールを選択して複数のモジュールオプションを表示する

エディターをテンプレート全体のビューに戻し、Command/Ctrl キーを使用してリッチテキストモジュールと画像モジュールの両方を選択します。インスペクターに、選択された複数のコンポーネントのオプションが表示されます。これにより、コンポーネントをグループ化したり、複数のコンポーネントを同時に削除したりできます。また、複数のコンポーネントに同時に CSS クラスを追加することもできます。



注:インスペクターの下部にある [エディターのオプション] セクションは意図的にガイドのこのページから省略しました。ロックモジュールコンテンツのオプションについては、コンテンツの範囲に関する一般情報で説明しています。

リファレンス

リファレンス

キーボードショートカット

コードエディターで使用できるキーボードコマンドのリストについては、[こちら](#)をご覧ください。注:リストの下部に、検索/置換に関連する多数のコマンドが記載されています。

デザインマネージャー全体で使用できる便利なコマンドもいくつかあります。該当するコマンドを、Mac/Windows の形式で以下にリストします。

ファイル編集アクション:

- **Command/Control + S:** ファイルを保存します。
- **Command/Control + Z:** 最後のファイル編集を元に戻します。
- **Command/Control + Shift + Z:** 最後のファイル編集をやり直します。
- **Command/Control + Shift + P:** ファイルを公開します。

ファインダーアクション:

- **Command/Control + P:** すべてのファイルとフォルダーを検索します。

アプリケーションアクション:

- **Delete/Backspace:** 選択した 1 つ以上のモジュールを削除します (ドラッグ&ドロップエディター)。
- **Command/Control + クリック:** 複数の項目を選択します (ファインダー、ドラッグ&ドロップエディター)。
- **右クリック:** アクションメニューを開きます (ファインダー、ドラッグ&ドロップエディター、タブ)。

コンテンツの範囲に関する一般情報

コンテンツ編集のフローとアクセスを制御するために使用できる多数のオプションがあります。これらのオプションの一部については別の場所で説明されています。このトピ

ックはテンプレートデザインを効果的なものにする上で非常に重要であるため、以下に記載するさまざまなオプションと併せて確認することをおすすめします。

- **コンテンツエディターでの編集を禁止する:**テンプレート内のモジュールとカスタムモジュール内のフィールドに使用できます。このオプションを使用すると、モジュール全体あるいはモジュール内の特定のフィールドをユーザーがエディターで編集できないようにすることができます。
- **グローバルグループ:**グローバルグループは、ポータル全体で共有されるコンポーネントのグループです。ページレベルでは編集できませんが、ページからリンク先に移動して、デザインマネージャー内でグローバルグループを編集できるようになっています。
- **グローバルモジュール:**グローバルモジュールは、ページレベルまたはテンプレートレベルでは編集できません。それ以外の点では、単なるモジュールです。

コンポーネント

ドラッグ&ドロップテンプレートを編集する際は、インスペクターの [追加] タブからポータル内の使用可能なすべてのコンポーネントに簡単にアクセスできます。ポータルは多様なアセットからなるライブラリで、アセットは以下のカテゴリに分類されています。

注:コンポーネントとモジュールという用語は同じ意味で使用されているように思えるかもしれませんが、この2つはまったく同じではありません。コンポーネントという用語は、モジュールとグループの両方を意味します。したがって、すべてのコンポーネントがモジュールであるわけではありませんが、すべてのモジュールはコンポーネントです。

既定のモジュール

HubSpot CMS には多数のモジュールが組み込まれています。画像や単行テキストといった単純なモジュールもあれば、イメージギャラリーのような複雑なモジュールもあります。モジュール自体はロックされており、編集することはできません。ただし、必要に応じて複製し、コピーを変更することはできます。

カスタムモジュール

既定のモジュールを複製して変更を加える場合、またはゼロからモジュールを作成する場合、そのモジュールはカスタムモジュールに分類されます。カスタムモジュールは、再利用可能な小さいスニペットにすることも、HTML、HubL、CSS、JS を組み合わせた複雑なモジュールにすることもできます。モジュールフレームワークのドキュメントについては、[こちら](#)をご覧ください。

構造モジュール

構造モジュールに分類されるモジュールは、グループとフレキシブルカラムの 2 つだけです。グループモジュールは、グループにまとめられたコンポーネント一式にすぎません。フレキシブルカラムモジュールは、エンドユーザーがエディターを使用して新しいコンポーネントをテンプレートに追加するために使用できる、特殊なコンテナタイプのモジュールです。

グローバルグループ

グローバルグループは、ポータル内の多数のページで共有される一連のコンポーネントをグループ化したものです。グローバルグループのコンテンツはさまざまな場所で繰り返し使用されるため、ページレベルでグローバルグループを編集することはできません。

グローバルモジュール

グローバルモジュールは、「グローバル」として指定された、ページレベルまたはテンプレートレベルでは編集できないモジュールです。この区別により、デザイナー/開発者はモジュールの機能を一定のコンテンツ形式に対して使用できます。

ファイルタイプ

CMS 内では、ファイルがいくつかのタイプ別に分けられます。コードファイルではタイプが混在する場合がありますが、ほとんどのところ、ファイルタイプのそれぞれは相互排他的です。

ドラッグ&ドロップ

ドラッグ&ドロップテンプレートは CMS の基本的なテンプレートです。このリファレンスガイドの後述のセクションで説明するように、使用できるテンプレートにはいくつかのタイプがあります。ドラッグ&ドロップテンプレートを使用すると、開発者はブートストラップグリッドで最もパワフルなレイアウトをすばやく作成できます。ドラッグ&ドロップインターフェイスも直観的で、コーディングの経験がなくてもレイアウトを変更できます。さらに、ゼロからレイアウトを作成することも可能です。

HTML & HUBL

マークアップをよりきめ細かく制御するには、ドラッグ&ドロップテンプレートの代わりに HTML&HubL テンプレートを使用できます。これらのテンプレートではインスペクターペインは使用されていません。HTML&HubL テンプレートファイルに CMS 機能を組み込むには、ビルダーで HubL ドキュメントを参照する必要があります。

スタイルシート

CSS ファイル。このファイルで HubL を使用できます。このコンテキストでは、変数とマクロがとりわけ役立ちます。CSS ファイルで使用される HubL は、ページビューで評価されるのではなく、ページの公開時に評価されます。

JavaScript

JavaScript ファイル。

モジュール

モジュールは、独自の HTML & HubL、CSS、JS を使用した再利用可能なコンポーネントです。モジュールの詳細については、[こちら](#)をご覧ください。

テンプレートのタイプ

ブログ

標準のブログテンプレートは、ページテンプレートとほぼ同一です。重要な違いは、ブログテンプレートには内部でフラグが立てられて、コンテキスト設定でブログテンプレートとして選択できますが、ページテンプレートとしては選択できない点にあります。ブログテンプレートに追加する機能は、開発者が完全に制御できます。

E メール

E メールテンプレートは、E メールツールで使用できるテンプレートです。E メールテンプレートは多種多様なメールクライアントで表示され、到達可能性を確保するためのベストプラクティスに従わなければならないため、これらのテンプレートには最も厳格な要件一式が適用されます。ドラッグ & ドロップテンプレートとコード化された E メールテンプレートはどちらもベースラインコンポーネントが事前に取り込まれます。

ページ

ページテンプレートは最も柔軟なテンプレートタイプです。これらのテンプレートは任意のウェブページまたはランディングページとしての役割を果たすことができます。ドラッグ&ドロップページテンプレートに事前に取り込まれるコンポーネントはありません。コードページテンプレートには、わずかなボイラープレートマークアップ（メタ情報、タイトル、必須のヘッダー/フッターインクルードに提案される HubL タグなど）が事前に取り込まれます。

システムページ

ブログページと同じように、システムページテンプレートには内部でその特定の目的に応じたフラグが立てられます。エラーページは、コンテンツ設定で 404 または 500 のいずれかのエラーページとして設定できます。パスワードプロンプトページもコンテンツ設定を使用して設定します。これらのページタイプはいずれも既定で空になっています。コンテンツ設定では E メールサブスクリプションテンプレートとして使用される 3 つのテンプレートにフラグが立てられていて、これらのテンプレートには適切なコンポーネントが事前に取り込まれます。

パーシャルテンプレート

パーシャルテンプレートは、他のコードファイルに組み込むことができるコードファイルです。パーシャルテンプレートを使用することで、開発者は複雑なドキュメントをモジュール方式で作成したり、複数のファイルの間でマークアップを共有したりできます。パーシャルテンプレートを HTML/HUBL ファイルに組み込むための構文は次のとおりです。

```
{% include "/template_partial_filename.html" %}
```

モジュール

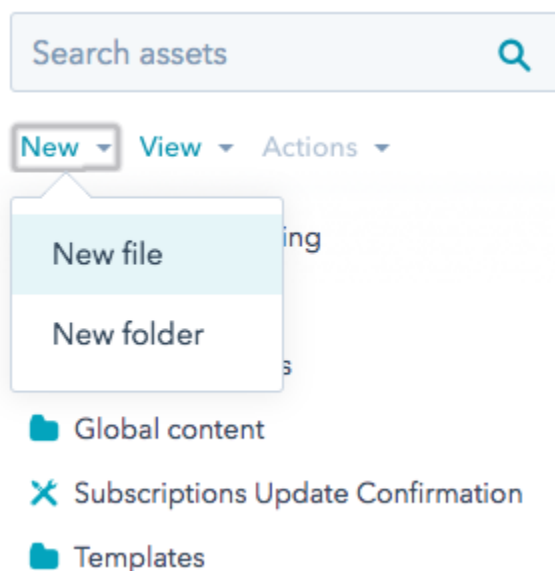
はじめに

モジュールの作成の基本

このセクションでは、デザインツールでモジュールを作成する基本的な手順について説明します。デザインツールにアクセスするには、[マーケティング]>[ファイルとテンプレート]>[デザインツール]の順に進みます。

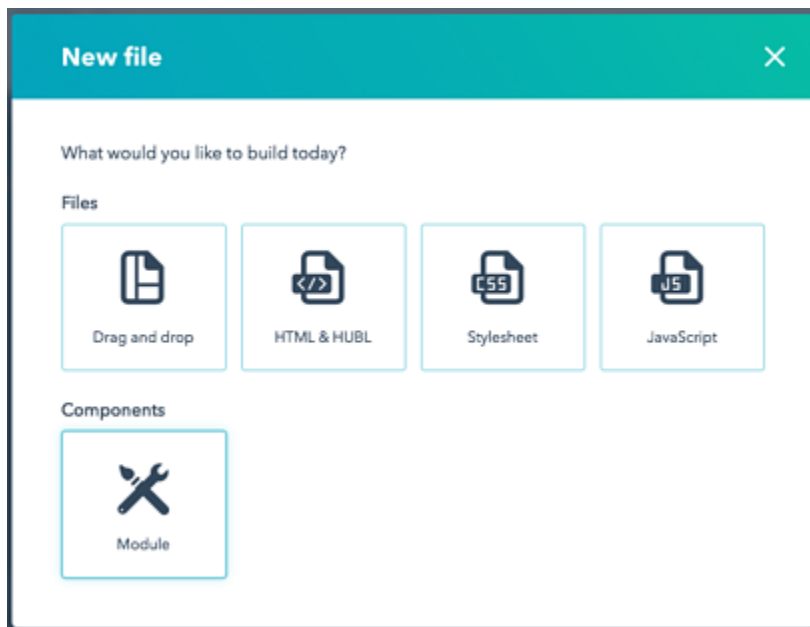
1.新規ファイルを作成する

モジュールを作成するには、画面左側の[ファインダー]パネルで[新規]ボタンをクリックし、[新規ファイル]を選択します。



2. ファイルタイプを選択する

表示されるダイアログボックスで、[モジュール]を選択します。



3. モジュールタイプを選択する

次のダイアログで、モジュールの名前を入力し、それをどこで使用するかを選択します。これらのモジュールタイプ間には技術的な違いがあり、互換性ないことに注意してください。詳細については、下記の[エディターセクション](#)を参照してください。この例では、ページおよびブログのテンプレート用のモジュールを作成します。ローカルモジュールを選択したままで、[作成]をクリックします。

Set up your new module ✕

[← Back to file types](#)

New file type
✕ Module

Where would you like to use this module? *

Page templates

Blog templates

Email templates

Module content scope

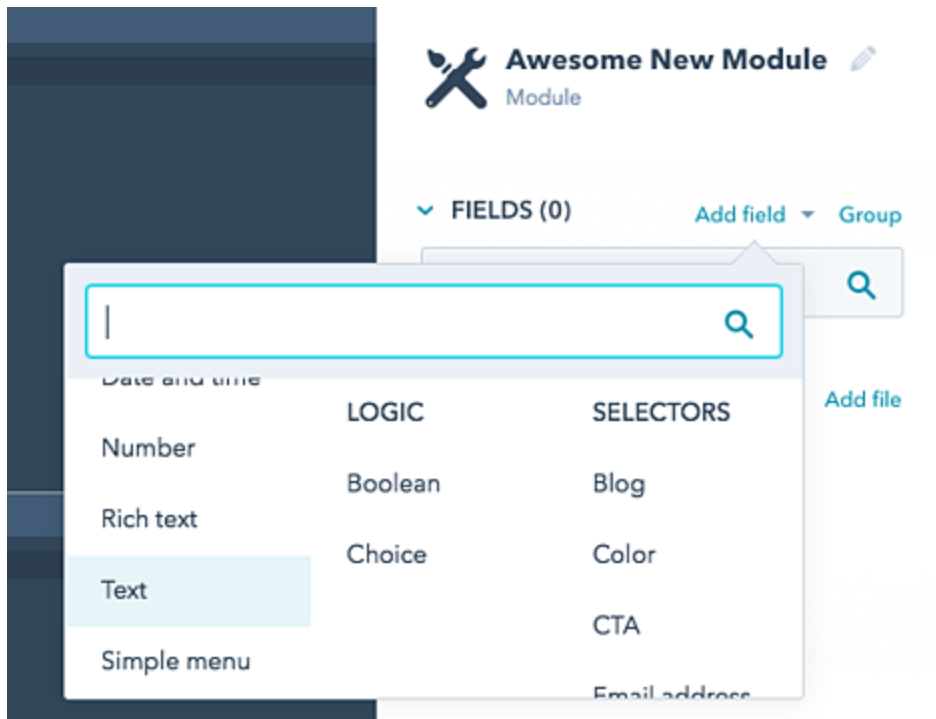
Local module Global module ⓘ

File name *

4. フィールドを追加する

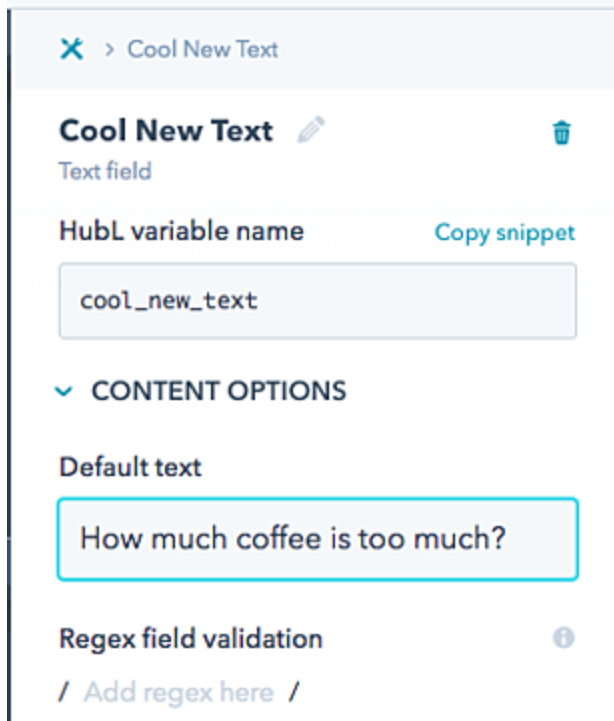
[ファインダー]パネルで新しいモジュールを選択し、画面右側の[インスペクター]セクションで[フィールドの追加]をクリックします。

新しいフィールドオプションが多数あり、このドキュメントの[参照セクション](#)でそれらを詳しく説明しています。この例では、[フィールドの追加]メニュー左側の[コンテンツ]列で[テキストフィールド]を選択します。



5.既定のコンテンツを追加する

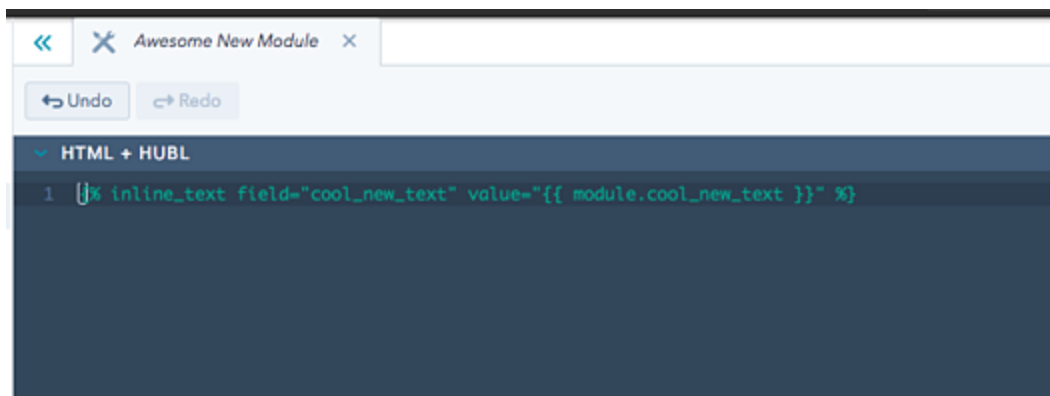
新しいフィールドに名前を付け、既定のテキストを追加します。



6. フィールドをモジュールマークアップに追加する

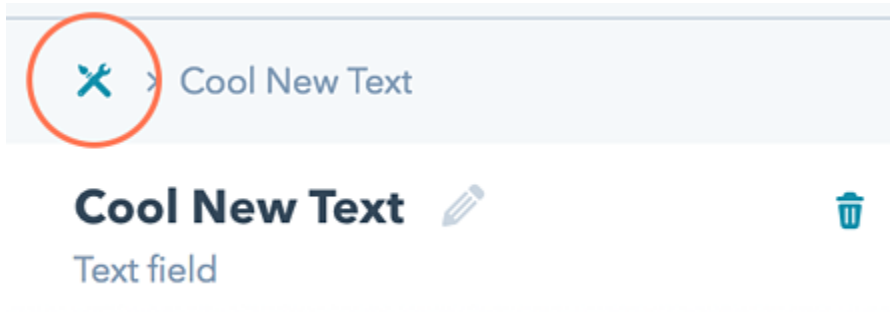
フィールドエディターの[HubL 変数名]セクションで、[スニペットをコピー]をクリックします。画面中央のエディターの[HTML +HUBL]セクションにスニペットを貼り付けます。

このエディターの詳細については、[こちらをご覧ください](#)。



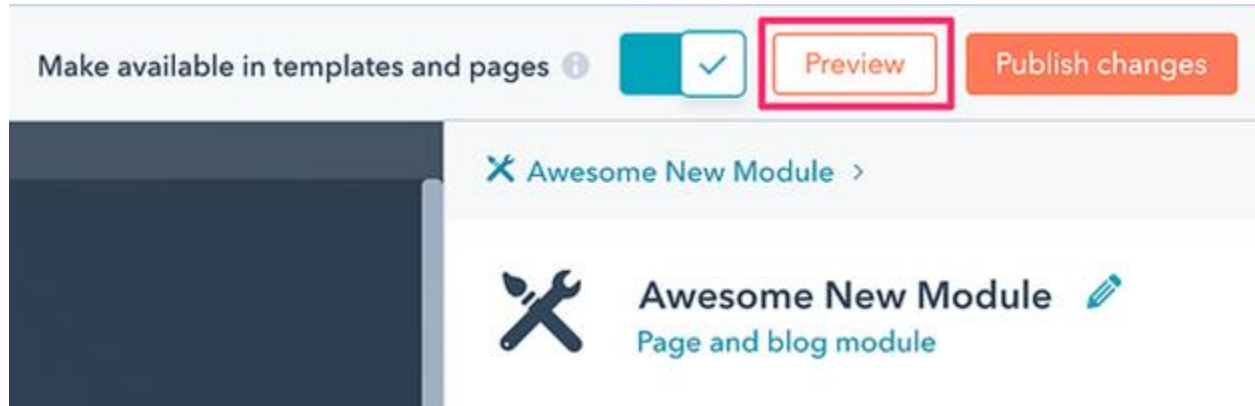
7. フィールドの編集を終了する

この新規フィールドの編集を終了してモジュール全体の編集に戻るには、インスペクターの上部にある階層リンク (パンくず) のモジュールアイコンをクリックします。



8. モジュールをプレビューする

画面右上隅の[プレビュー]ボタンをクリックします。モジュールのプレビューが新しいタブで開きます。このプレビューはエディターと同期され、作業の進行に合わせて自動的に更新されます。プレビューを新しいウィンドウやモニターに自由に移動させることができます。



コードエディターの詳細

コードエディターの詳細

新しいコードエディターには、ブログまたはページモジュールの編集時に使用する HTML + HubL、CSS、および JS 用の独立したセクションが含まれています。E メールモジュールを使用する場合は、エディターの HTML + HubL セクションのみが表示されます。このようにモジュールのタイプで区別するのは、以下で詳しく説明するように、表示されるページでの CSS と JS の処理方法が異なるためです。

以下では、エディターの各セクションの使用方法和制限事項の概要を示し、各セクションでのコードの処理方法について説明します。新しいモジュールフレームワークと互換性のないコードの詳細については、「[リファレンス](#)」ページの重要な既知の問題と制限される HubL に関するセクションも確認してください。

エディターのセクション


```
HTML + HUBL
1 |!-- HTML and HubL -->

CSS
1 |*
2 |   CSS only. No HubL.
3 |   Minified and included in head of doc.
4 |   Only included once, even if module is used multiple times on a page.
5 | */

JS
1 //
2 //   JS only. No HubL.
3 //   Minified and included in foot of doc.
4 //   Wrapped in a closure.
5 //   Only included once, even if module is used multiple times on a page.
6 //
```

- CSS と JS は、それぞれドキュメントの先頭と末尾に圧縮して挿入されます。
- モジュールの CSS と JS は、そのモジュールがページで複数回使用される場合でも、ドキュメントに一度だけ挿入されます。
- JS は名前空間用のクロージャで囲まれます。
- エディターの CSS セクションと JS セクションは HubL トークンを受け付けません ([ここで説明している module_asset_url 関数を除く](#))。フィールドデータやその他の CMS データをモジュールの CSS と JS に渡す必要がある場合は、以下のオプションがあります。
 - data や class などの適切な HTML 属性を使用してマークアップに HubL を出力します。
 - 必要に応じてインラインスタイルを使用します。
 - HTML + HubL セクションで CSS または JS を追加します。このコードを [require_css](#) タグと [require_js](#) タグでそれぞれ囲むことを強くお勧めします。注 : エディターの JS セクション内の JS は自動的にクロージャで

囲まれるため、HTML + HubL セクション内の JS とは名前空間が異なります。

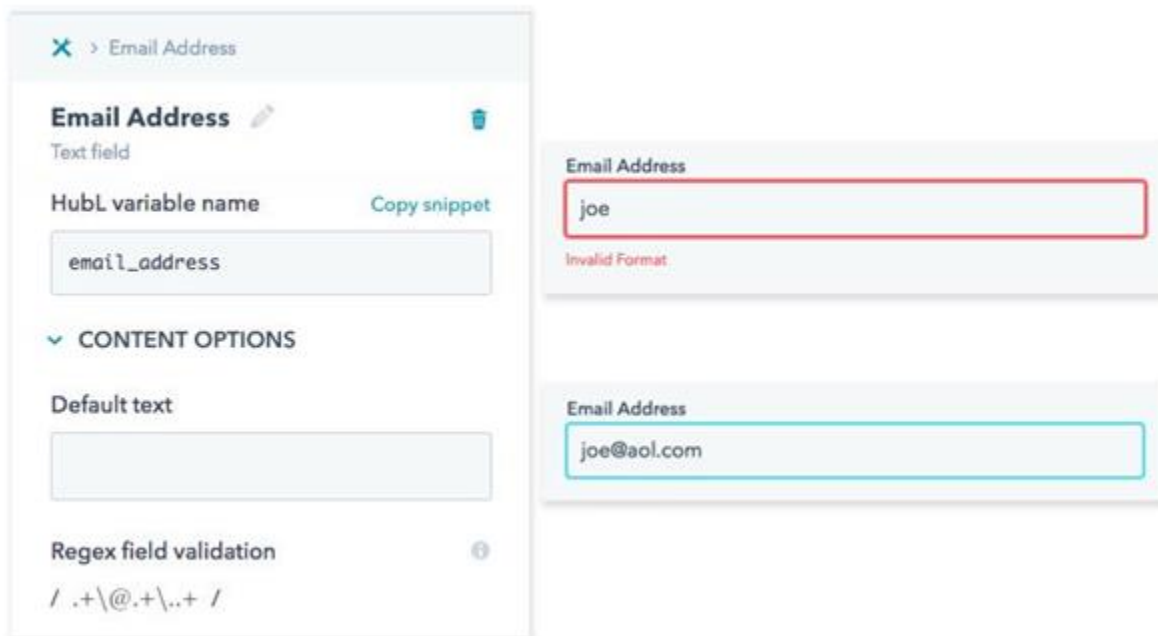
フィールドオプションと使用法

フィールドのオプション

モジュールに検証、エディターのオプション、表示条件、および繰り返しとグループ化の機能が追加されました。このセクションでは、これらの機能について紹介します。

正規表現によるフィールド検証

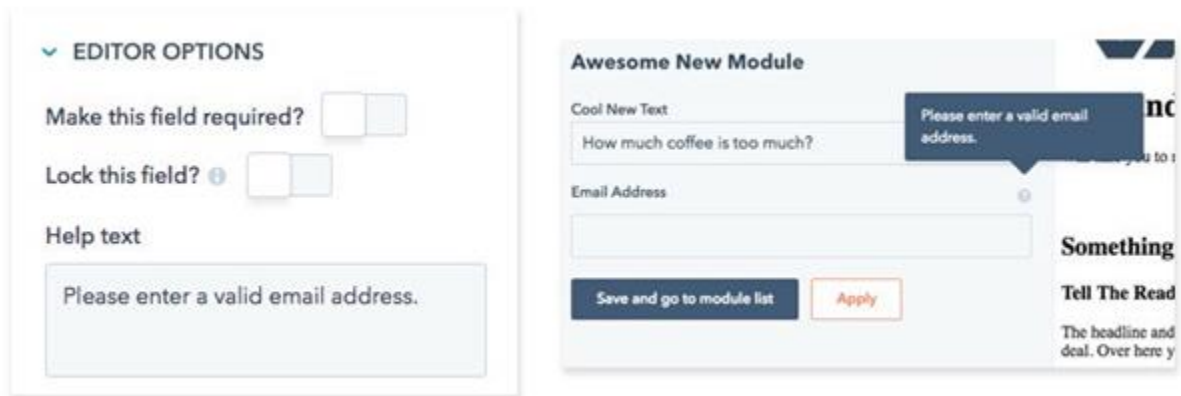
正規表現を使用してフィールドを検証します。以下は、シンプルな Eメールの検証式と、その結果得られるエディター動作の例です。



エディターのオプション


- このフィールドを必須にする：ユーザーがこのフィールドを空白のままにすることはできません。
- コンテンツエディターでの編集を禁止する：エンドユーザーが編集できないコンテンツを作成できます。このコンテンツはテンプレートレベルで使用できることに注意してください。これは、細部が異なる複数のモジュールを作成する代わりに、テンプレートごとに微調整できるモジュールを設計する場合に便利な機能です。
- ヘルプテキスト：ユーザーにコンテキストや指示を与えるヘルパーテキストを追加します。以下は、インスペクターのヘルプテキストの例と、エディターでの



表示例です。



表示条件

別のフィールドが特定の条件を満たしている場合にのみ、ユーザーにフィールドを表示したい場合があります。[表示条件]セクションでは、これらの条件を設定できます。以下の例では、[フルネーム]フィールドへの入力が完了したユーザーにのみ、[E メールアドレス]フィールドが表示されます。

 > Email Address

Email Address  
Text field

HubL variable name Copy snippet


email_address

▼ CONTENT OPTIONS

Default text

Regex field validation ?
/ .+\@.\+.\.+ /

> EDITOR OPTIONS

▼ DISPLAY CONDITIONS ? 

Condition

full_name ▼

is not empty ▼

Value or regex

> REPEATER OPTIONS ?

フィールドの繰り返しとループ

フィールドの繰り返しとループ

モジュールに追加された重要な新機能の1つは、フィールドやグループを繰り返す機能です。つまり、構造化され、しかも高度に拡張可能な方法でユーザーがコンテンツを追加または除去できるようなモジュールを開発者は作成できます。では、その仕組みを見ていきましょう。

1. 繰り返しフィールドを作成してオプションを設定する

インスペクターで新規フィールドを作成します。この例では、1つのイメージフィールドを使用します。インスペクターの最下部で、このフィールドの[リピーターオプション]スイッチをオンに切り替えます。オプションで、[オブジェクト数の上限]セクションで、このフィールドインスタンスが出現すべき最小回数または最大回数を設定します。

また、既定のオブジェクト数を設定することもできます。

✕ > Snapshots



Snapshots  


Image field

HubL variable name Copy snippet

snapshots

✓ CONTENT OPTIONS

Default image





Delete Replace

Alt text

View from the top

› EDITOR OPTIONS

› DISPLAY CONDITIONS 

✓ REPEATER OPTIONS 

Object count limits

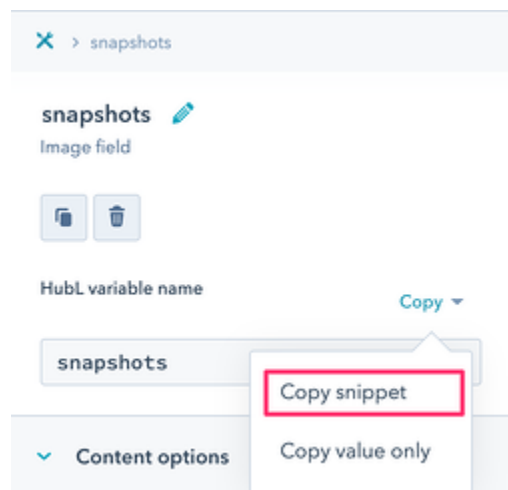
1 to No maximum

Default object count

No default

2.HubL でフィールドをループ処理する

任意の数のフィールドインスタンスをユーザーが追加できるようにするには、HubL マークアップで1つのタグの代わりにループ形式を使用します。コードエディターの [HTML + HUBL] セクションで、下記のスクリーンショットのように新規フィールドをループ処理します。また、[コピー] メニューの [スニペットをコピー] を選択して、for ループコードを取得することもできます。



この例では、snapshots という名前のイメージフィールドをループ処理し、各インスタンスの src 属性と alt 属性をイメージタグの中に取り入れます。どのフィールドを繰り返すかによって、このループで使われる属性は異なります。フィールドとその属性の詳細リストについては、[参考資料](#) ページをご確認ください。

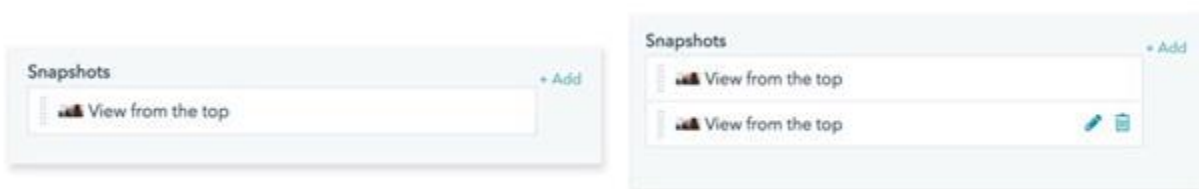
HubL ループに関する一般情報は、[こちらのドキュメント](#) に詳しく掲載されています。

```
HTML + HUBL
1  {% for img in module.snapshots %}
2
3      
4
5  {% endfor %}
```

繰り返しフィールドの編集作業

下記に、ユーザーがエディターを使用して編集する作業を示します。左側のスクリーンショットでは、ユーザーが最初にこのモジュールを編集し始めたとき、既定の1つのイメージが表示されます。[追加]ボタンをクリックすると既定のイメージが複製されます。

右側のスクリーンショットに示すように、複製されたイメージの上にマウスカーソルを置くと、ユーザーはイメージを削除したり、編集モードに入ってイメージを取り替えたり他の属性を設定したりできます。



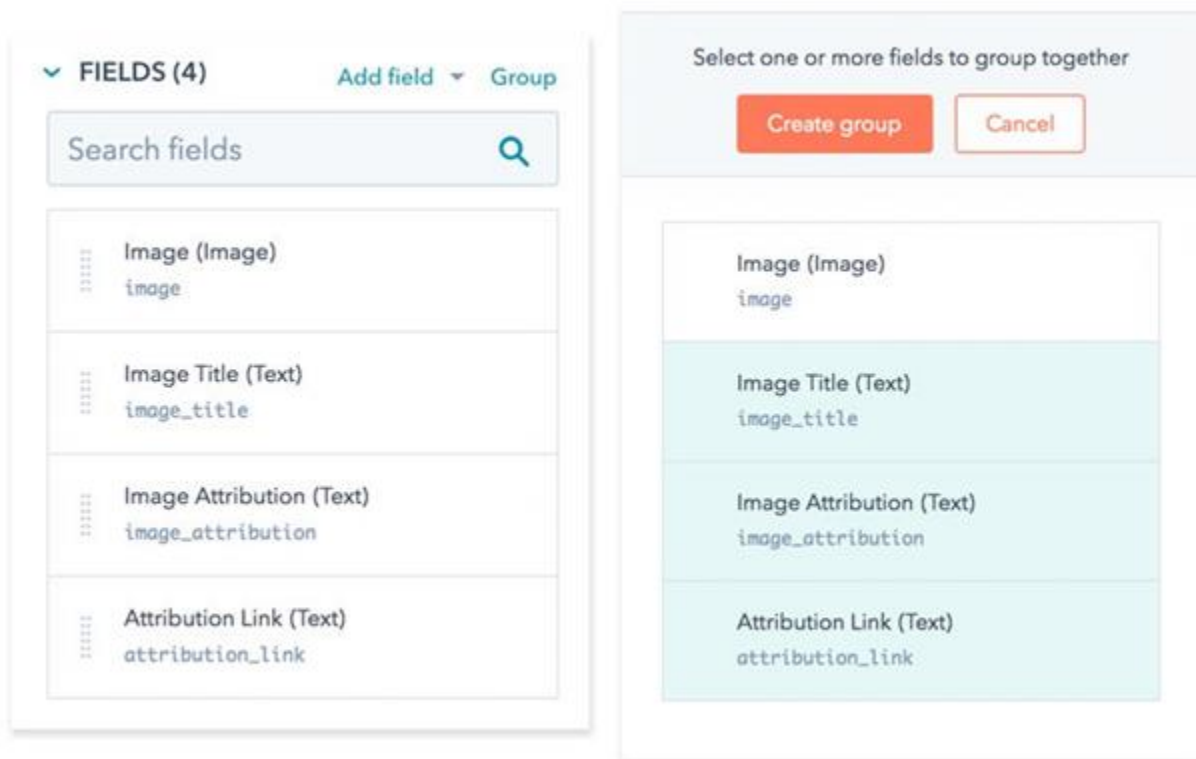
フィールドのグループ化とロジック

フィールドのグループ化とロジック

モジュールに追加されたもう1つの強力な新機能は、フィールドをグループ化する機能です。この機能を使用すると、ユーザーのために直観的な編集環境をカスタマイズできます。このセクションでは、グループ化されたフィールドの操作について説明し、エディターの HTML + HubL セクションで条件文を使用する方法を確認します。

1.新規グループを作成する

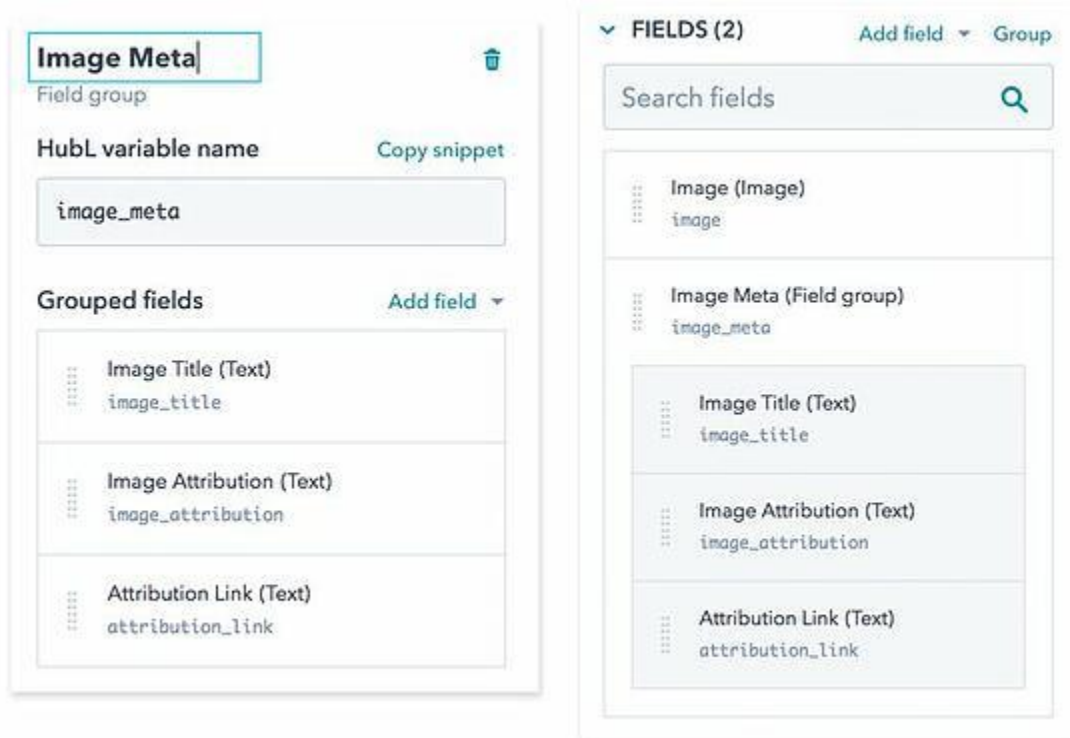
複数のフィールドをグループ化するには、インスペクターの右上にある[グループ]ボタンをクリックします。グループ化するフィールドを選択するように指示するメッセージが表示されます。ここでは、この画像のメタ情報を区分するためのグループを作成します。



2.グループに名前を付ける

新しいグループを作成すると、このグループの編集モードになります。このグループの名前を追加します。ユーザーがこのページを直観的に編集できるかどうかは、この名前にかかっています。トピックパスを使用してフィールドグループの編集モードを終了し

ます。右下の画像では、このグループがモジュール内の他のフィールドと並んで表示されています。



3. HubL でグループのデータにアクセスする

グループ化されたフィールドのデータはグループ内に保存されることに注意してください。以下のスクリーンショットは、上記の例の Image Meta にアクセスするための HubL を示しています。この場合、`image_title` はモジュールの直接の子ではなく、`image_meta` の子です。

```
{{ module.image_meta.image_title }}
```

4.条件文を追加する

フィールドをグループ化してより複雑なモジュールを作成できるようになると、ユーザーの編集環境の使い勝手が悪くなる可能性があるため、注意が必要です。上記の例の Image Meta のようなグループには、ユーザーが一部の画像でこのコンテンツを利用できない場合に備えて、条件文を追加する必要があります。

以下のスクリーンショットでは、画像が選択されていることを確認してから、figure タグと image タグを追加しています。その後、各 Image Meta フィールドに入力されていることを確認してから、フィールドのマークアップを追加しています。

```
HTML + HUBL
1  {% if module.image.src %}
2  <figure>
3    
4    {% if module.image_meta.image_title && module.image_meta.image_attribution && module.image_meta.attribution_link %}
5    <figcaption>
6      {{ module.image_meta.image_title }}
7      <a href="{{ module.image_meta.attribution_link }}">{{ module.image_meta.image_attribution }}</a>
8    </figcaption>
9  {% endif %}
10 </figure>
11 {% endif %}
12
```

モジュールによっては、表示条件、必須フィールド、および HubL ロジックを組み合わせることで、ユーザーにできるだけ柔軟で直観的なコンテンツを提供できる場合があります。この例は説明を簡単にするためのものです。条件文を含む HubL の詳細については、[こちらのドキュメント](#)をご覧ください。

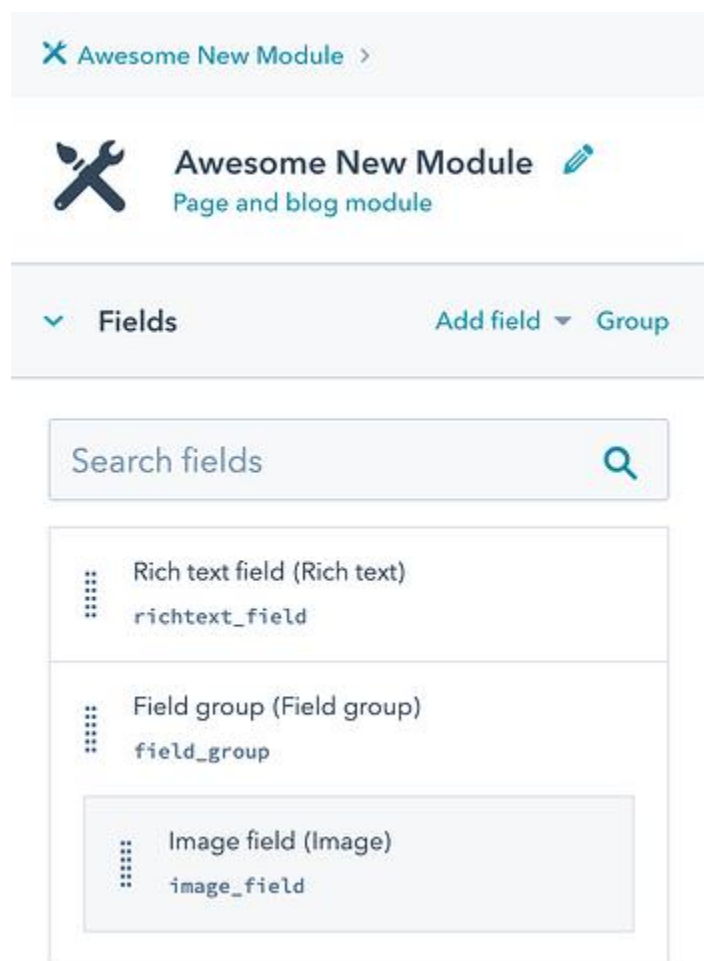
モジュールオプション

モジュールオプション

新しいモジュールフレームワークには、デザインマネージャー内でモジュールを整理してアクセスするための、シンプルながらも強力なオプションがいくつか用意されています。

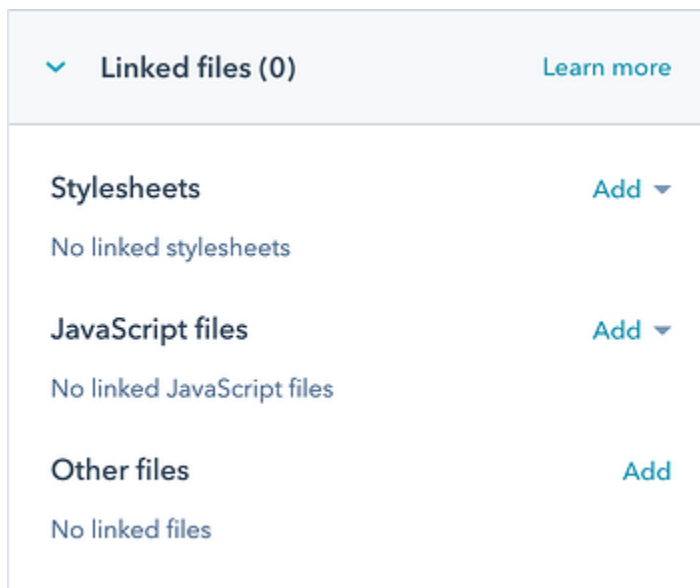
フィールド

この領域では、さまざまな[フィールドタイプ](#)をモジュールに追加して、必要に応じて[グループ化](#)できます。



リンク済みファイル

CSS および JS のライブラリとフレームワークといったアセットを関連付けるのに適切な場所です。モジュールを扱った経験がある場合は、サポートされなくなったメソッドに頼ることがないように、[リファレンスページ](#)を確認することをおすすめします。



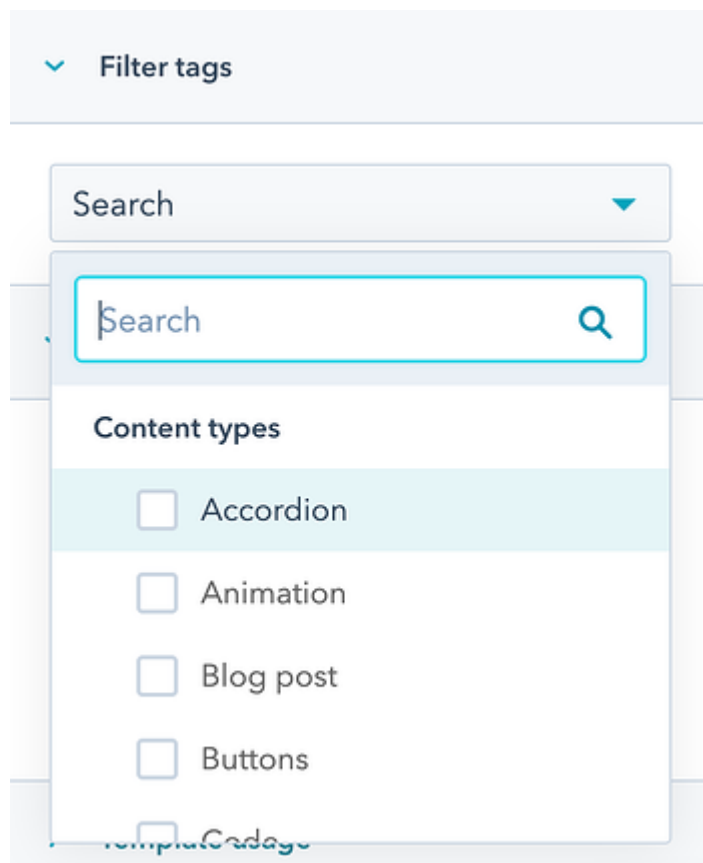
スタイルシートと JavaScript ファイルは、ドキュメントの先頭と末尾にそれぞれ自動的に読み込まれます。モジュールが移動/コピーされると、インクルードされる他のファイルにフラグが立てられます。必要に応じて、コードエディターでこれらのアセットをリンクする必要があります。ほとんどの場合、リンクには `module_asset_url` 関数を使用することをおすすめします。

`module_asset_url` 関数は、モジュールエディターの CSS と JS のセクションで受け入れられる唯一の HubL です。この関数は、リンク済みファイルの完全な URL を返します。この関数を取るパラメータは、アセットのファイル名のみです。以下に、この関数を使用して `arrow` という名前の png にリンクする例を示します。

```
.arrow{  
  background: url({{ module_asset_url('arrow.png') }});  
}
```

フィルタータグ

タグを利用すると、ユーザーが探しているモジュールを見つけやすくなります。モジュールデザイナーではコンテンツのタイプと目的に応じたタグを追加できます。こうしたタグを追加するだけで、混雑したポータルでモジュールを見失うことがなくなるため、クライアントだけでなく、将来的には自分自身にとっても役立ちます。



フィールドの翻訳

モジュールのフィールドラベル、既定のコンテンツ、ヘルプテキストを複数の言語に翻訳してください。ユーザーにはユーザーアカウントの既定の言語で翻訳されたモジュールのフィールドラベルが表示されます。

Spanish - Mexico translations

```
1 - {
2 -   "richtext_field.label": {
3 -     "message": "Campo de texto enriquecido"
4 -   },
5 -   "richtext_field.help_text": {
6 -     "message": ""
7 -   },
8 -   "richtext_field.default": {
9 -     "message": ""
10 -  },
11 -   "field_group.label": {
12 -     "message": "Grupo de campo"
13 -   },
14 -   "field_group.help_text": {
15 -     "message": ""
16 -   },
17 -   "field_group.image_ffield.label": {
18 -     "message": "Campo de imagen"
19 -   },
20 -   "field_group.image_ffield.help_text": {
21 -     "message": ""
22 -   },
23 -   "field_group.image_ffield.default.alt": {
24 -     "message": ""
25 -   }
26 - }
```

Awesome New Module >

field_group

Image field (Image)

image_ffield

> Linked files (0) [Learn more](#)

> Filter tags

▼ Field translations [Learn more](#)

Translations (2) [Add translation +](#)

Spanish - Mexico [×](#)

English (master) [Change](#)

> Template usage

テンプレートの使用

以前はメニューやモーダルの後ろに隠れていた要素が公開されるようにするには、テンプレートを使用するのが簡単な方法です。新しいモジュールの ID と、コード化されたテンプレート内でこのモジュールを使用するための適切な構文に簡単にアクセスできます。

▼ Template usage

This module can be added to any drag and drop template via the Add sidebar. It can be added to HTML/HubL templates using the following snippet.

Usage snippet

```
{% module
"module_156685400830185"
path="/siteroot/Awesome New
Module", label="Awesome New
Module" %}
```

Copy snippet

モジュールアイコン

モジュールビルダーでモジュールにアイコンを追加できるようになりました。アイコンを追加すると、ユーザーがモジュールを区別しやすくなります。また、ビルダーでモジュールに小さなブランドアイデンティティを注入できるようにもなります。モジュールアイコンを変更するには、現在のモジュールアイコンをクリックすればよいだけです。アイコンは SVG 形式でなければなりません。また、10kb 未満にする必要があります。

注:アイコンは単色で表示されます。その他の色は失われるため、最適な結果を得るには単色のアイコンをデザインしてアップロードすることをおすすめします。

リファレンス

モジュールリファレンス

ここで取り上げるモジュールの詳細と使用例を確認するには、[HubL でサポートされているタグ](#)のページをご覧ください。

全般的な変更

- モジュールに不変の数値 ID が割り当てられるようになりました。これにより、モジュールビルダーでモジュールの名前を変更する際に、テンプレートコードでモジュールへの参照を変更する必要がなくなりました。
- モジュールは自己完結型として設計されており、CSS テキスト、JS テキスト、ファイルアセットの参照がすべて組み込まれています。
- v1 モジュールは `span` にラップされますが、それとは異なり、v2 モジュールは `div` にラップされます。
- UI との整合性を確保するために、HubL の `widget` は `module` に名前変更されました。今後も両方の用語が使用されますが、すべてのドキュメントでは用語がモジュールに統一されます。

- CSS については、コードエディターでモジュール CSS に書き込むか、インスペクターの [\[リンク済みファイル\]](#) セクションを使用してリンクするようになりました。

既存のフィールドタイプへの追加

フィールド	新しい属性
選択	<ul style="list-style-type: none"> • 表示形式: 'select' (既定。プルダウンスタイル) または 'radio' • 選択肢: [value, label] 項目の配列。カテゴリーの場合は [categoryLabel, listOfChoices] 項目の配列。すべての値は一意でなければなりません。選択フィールドのラベルに HTML を含めることはできますが、JS はすべて除去する必要があります。HTML ラベルと選択カテゴリは、いずれも表示形式が 'select' の場合にのみ有効です。 • プレースホルダー: 表示形式が select に設定されている場合、ドロップダウンメニュー内でプレースホルダーとして表示される文字列。プレースホルダーに値が設定されていない場合、既定では「検索」 (UISelect での既定) に設定されます。

テキスト

- **正規表現フィールドの検証:**このフィールドの値が定義済みの正規表現に一致すると、フィールドは有効とみなされます。正規表現は、「/」でラップされていない単純な形式でなければなりません。また、正規表現は（部分文字列ではなく）文字列全体に一致する必要があり、大文字/小文字も区別されます。

画像

- **サイズコントロールを非表示:**画像のサイズを変更できるかどうか（実際には、サイズ変更コントロールを表示するかどうか）を決定するブール値。既定では `true` に設定されます。

新しく追加されたフィールドタイプ

フィールド	説明	属性
-------	----	----

数値フィールド

スピナースタイルのテキストフィールド。10進数または整数の値のみをサポートします。

- **タイプ:** 'number'
- **最小:** 最小値。既定なし
- **最大:** 最大値。既定なし
- **ステップ:** スピナー矢印をクリックしたときの増分値または減分値。既定では 1
- **既定の値:** 既定の整数値。値が複数回出現する場合は、既定の値の配列 (オプション)
- **表示形式:** 'text' (既定)。スライディングセレクターを表示する場合は 'slider'

日付フィールド

日付を選択します。エポックからその日付の真夜中 (UTC) までの経過ミリ秒数として値が保管されます。

- **タイプ:** 'date'
- **既定の日付:** 既定として使用する日付

日時フィールド

日付と、必要に応じて時刻を選択します。エポック (UTC) からの経過ミリ秒数として値が保管されます。

- **タイプ:** 'datetime'
- **既定の日時:** 既定として使用する日時
- **期間:** 時間選択ドロップダウンで使用する増分値 (分数)。既定では 30

CTA フィールド

CTA を選択します。CTA モジュールに必須です。

- **タイプ:** 'cta'
- **既定の CTA:**既定の CTA を選択するための CTA セレクター

ブログ フィールド

ポータルブログリストからブログを選択します。ブログ E メールサブスクリプション、投稿リスト、RSS リストの各モジュールに必須です。

- **タイプ:** 'blog'
- **既定のブログ:**使用する既定のブログを選択するためのブログセレクター
- **プレースホルダー:**表示形式が `select` に設定されている場合、ドロップダウンメニュー内でプレースホルダーとして表示される文字列。プレースホルダーに値が設定されていない場合、既定では「検索」(UISelectでの規定)に設定されます。

タグ フィールド

ポータルからブログタグを選択します。RSS リストモジュールに必須です。

- **タイプ:** 'tag'
- **既定のタグ:**使用する既定のタグを選択するためのタグセレクター。
- **戻り値タイプを選択:**タグセレクターから取得する値のタイプを選択します。選択肢はタグの「名前」または「ID」です。

フォームフィールド

HubSpot フォームを選択します。フォームモジュールに必須です。

- **タイプ:** 'form'
- **既定のフォーム:**使用する既定のフォームを選択するためのフォームセレクトター。
- **フォーム送信後に訪問者に何を表示しますか?:**ユーザーを別のページにリダイレクトするか、フォームの送信時にサンキューメッセージをインラインで表示できます。

色フィールド

色を選択します。

- **タイプ:** 'color'
- **既定の色:**既定の色とアルファ（不透明度）値を選択するための色のセレクトター。色を選択するには、16進値を入力するか、カラーピッカーを使用します。

ページビュー

公開済みのウェブサイトまたはランディングページを選択します。フォームモジュールに必須です。

- **タイプ:** 'page'
- **既定のページ:**既定のページを選択するためのページセレクトター。
- **プレースホルダー:**ページが選択されていない場合のプレースホルダーの値（この値で、ドロップダウンの「ページを選択」テキストが置き換えられます）。

ワークフローフィールド

HubSpot ワークフローを選択します。フォームモジュールに必須です。

- **タイプ:** 'workflow'
- **既定のワークフロー:**既定のワークフローを選択するためのワークフローセレクトター。新しいワークフローを作成するためのボタンもあります。
- **プレースホルダー:**ワークフローが選択されていない場合のプレースホルダーの値（この値で、ドロップダウンの「ワークフローを選択」テキストが置き換えられます）。

フォローアップ E メールフィールド

フォローアップ E メールを選択します。

- **タイプ:** 'followupemail'
- **既定のフォローアップ E メール:**既定のフォローアップ E メールを選択するためのフォローアップ E メールセレクター。
- **プレースホルダー:**フォローアップ E メールが選択されていない場合のプレースホルダーの値 (この値で、ドロップダウンの「フォローアップ E メールを選択」テキストが置き換えられます)。

フォントフィールド

Google フォントを選択します。

- **タイプ:** 'font'
- **フォント:**Google フォントを選択するためのドロップダウンセレクターと、フォントサイズ、色、スタイル、単位のオプションがあります。
- **既定の設定:**

```
{ color: "#000", font: "helvetica", size: 12, size_unit: "px", styles: {font-weight: "bold", font-style: "italic", text-decoration: "underline"} }
```

E メールアドレスフィールド

ポータルユーザーの E メールアドレスが自動入力されます。ただし、他の有効な E メールアドレスを使用することもできます。値は文字列のリストです。

- **タイプ:** 'email'
- **既定の E メールアドレス:** ポータルユーザーの E メールアドレスを選択するためのドロップダウンセレクターがあります。手動で E メールアドレスを入力することもできます。
- **プレースホルダー:** E メールアドレスが選択されていない場合のプレースホルダーの値 (この値で、ドロップダウンの「E メールアドレスを選択...」テキストが置き換えられます)。

ファイルフィールド

画像セレクターと似ていますが、ファイルマネージャーから他のファイルタイプを選択できます。リンク先の PDF や画像を選択する際に役立ちます。ピッカー属性を使用して、特定のタイプを選択できます。

- **タイプ:** 'file'
- **ファイルタイプ:** 'file' (既定。任意のファイル)、'image'、または 'document' (画像以外のファイル)
- **既定のファイル:** モジュールフィールドの既定のファイルを選択するために使用するファイルピッカー。

HubDB テーブルフィールド

このモジュールに関連付ける公開済み HubDB テーブルを選択します。

- **タイプ:** 'hubdbtable'
- **既定の HubDB テーブル:** 既定の HubDB テーブルを選択するためのドロップダウンセレクターがあります。
- **プレースホルダー:** HubDB テーブルが選択されていない場合のプレースホルダーの値 (この値で、ドロップダウンの「HubDB テーブルを選択」テキストが置き換えられます)。

シンプルメニューフィールド

ローカルのシンプルメニューを作成できます。

- **タイプ:** 'simplemenu'
- **既定のシンプルメニュー:** [メニューを編集] ボタンをクリックすると、HubSpot のメニュー作成インターフェイスを使用してシンプルメニューを作成できます。

メニューフ
ィールド

ポータルメニューからメ
ニューを選択します。

- **タイプ:** 'menu'
- **既定のメニュー:**既定のメ
ニューを選択するための
ドロップダウンセレクト
アがあります。
- **プレースホルダー:**メ
ニューが選択されてい
ない場合のプレースホ
ルダの値 (この値で、
ドロップダウンの「メ
ニューを選択」テキ
ストが置き換えられ
ます)。

ミーティ
ングフ
ィールド

ポータルミーティ
ングからミーティ
ングを選択します。

- **タイプ:** 'meeting'
- **既定のミーティング:**既定
のミーティングリンク
を選択するためのドロ
ップダウンセレクト
アがあります。
- **プレースホルダー:**ミー
ティングが選択され
ていない場合のプレ
ースホルダの値 (こ
の値で、ドロップ
ダウンの「ミーティ
ングを選択」テキ
ストが置き換えられ
ます)。

ロゴフィールド **ロゴを選択できます。**

- **タイプ:** 'logo'
- **既定のロゴを上書き:**特定のページのロゴを上書きするかどうかを選択できるトグル。
- **既定のロゴ:**コンテンツ設定でグローバルに既定となっているロゴから取得されたサイトロゴ。
- **ロゴが設定されていない場合は会社名を表示する:**ロゴが設定されていない場合にコンテンツ設定で使用されている会社名を表示するためのトグル。

アイコンフィールド **アイコンを選択できます。**

- **タイプ:** 'icon'
- **既定のアイコン:**アイコンを選択するためのドロップダウンセレクターがあります。
- **注:**選択できるアイコンはFontAwesome アイコンライブラリに基づきます。

URL フィールド

外部ページ、HubSpot コンテンツ、HubSpot でホストしているファイル、E メールアドレス、または HubSpot ブログを指す URL を選択または入力できます。

- **タイプ:** 'url'
- **既定の URL**
 - **リンク先:**リンクとして外部リンク、コンテンツリンク (HubSpot ページ)、ファイル (HubSpot ファイルマネージャー)、E メールアドレス、またはブログ (メインブログリスト) を選択するためのドロップダウンセレクターがあります。
- **サポートされる URL タイプ:**エントリでサポートされている URL のタイプを選択します。

リンクフィールド

外部ページ、HubSpot コンテンツ、HubSpot でホストしているファイル、E メールアドレス、または HubSpot ブログを指す URL を選択または入力できます。また、[リンクを新しいウィンドウで開く] と [検索エンジンにこのリンクをたどらないように指示する] のブール値も設定します。

- **タイプ:** 'link'
- **既定の URL**
 - **リンク先:**リンクとして外部リンク、コンテンツリンク (HubSpot ページ)、ファイル (HubSpot ファイルマネージャー)、E メールアドレス、またはブログ (メインブログリスト) を選択するためのドロップダウンセレクターがあります。
- **リンクを新しいウィンドウで開く:**リンクを新しいウィンドウで開けるように設定する
`target='_blank'` をリンク属性に追加するためのトグル。
- **検索エンジンにこのリンクをたどらないように指示する:**`rel='nofollow'` をリンク属性に追加するためのトグル。
- **サポートされる URL タイプ:**エントリでサポートさ

れている URL のタイプを
選択します。

制限された HubL

モジュールのレンダリングが複雑化しないよう、新しいモジュールフレームワークでは
ネストされたモジュールやカスタムモジュールソースのインクルードといった複雑な
HubL は使用できないようになっています。

以下のタグは使用できません。

- custom_module
- extends
- import
- include
- module
- raw_jinja
- raw_html
- widget_container

以下の関数は使用できません。

- get_public_template_url
- get_public_template_url_by_id
- footer_js
- head_css
- head_js
- include_attached_css
- include_default_custom_css
- include_css
- include_JavaScript
- include_targeted_definitions
- js_integration_head
- js_integration_body_start
- js_integration_body_end
- require_attached_css
- require_default_custom_css

上記のタグまたは関数を使用して作成された新しいモジュールを公開しようとするエラーが発生します。

古いデザインマネージャーで作成されたモジュールに対する HubL 検証はそれほど厳格ではありません。次のタグまたは関数によって検証警告と推奨が生成されますが、それでも v1 モジュールで公開することは可能です。

- Import
- include
- get_public_template_url
- フィールド内の句読点

- 文字列としてのブール値
- 選択フィールドの同じ値
- 重複するフィールド名

モジュールには[サポートされている HubL タグ](#)を含めることができます。この場合、そのモジュールは静的バージョンになります。ただし、`overrideable` 属性は `true` として指定されている場合でも常に `False` とみなされます。これは、タグによって生成されるサブモジュールは、コンテンツエディターで編集できないためです。同様に、`export_to_template_context` パラメータもカスタムモジュール内では使用できません。

モジュールの変数

モジュールのいくつかの変数が更新、追加されています。

サポートされなくなった既存の変数:

- `custom_widget_name`
- `parent_custom_widget_name`

新しく追加された変数:

- `portal_id`
- `is_live`
- `module` (後方互換性を確保するためのウィジェットとしても使用可能)
- `name` (以前の `parent_custom_widget_name`)
- `module_id`
- `language`
- `request`
- `messages`
- `js_assets`
- `css_assets`
- `other_assets`

V1 モジュールから V2 モジュールへの移行

V1 モジュールから V2 モジュールへの移行

このページの目的は、これまでカスタムモジュールを作成してきた開発者が、バージョン 1 のカスタムモジュールと新しいモジュールフレームワーク（それぞれ v1、v2 と呼びます）の違いに関してよくある質問の答えをすぐに見つけられるようにすることです。新しいモジュールフレームワークには、より堅牢で信頼性の高いオプションを使用して複雑な機能を作成できるよう、多数の機能が追加されています。これらの追加に伴って、v2 ではさまざまな点が変更されました。以降のセクションでは、これまで v1 を使用してきた開発者に影響する可能性の高い変更点を取り上げ、サポートされなくなった一般的な v1 パターンに対処するための推奨事項を説明します。

注: モジュールエディターでは引き続き v1 モジュールを編集、使用できます。v1 モジュールの複製を含め、新しいモジュールは [v2 モジュールフレームワーク](#) で作成されます。

inline_rich_text フィールドの使用

v1 でリッチテキストモジュールのインライン編集を行うには、カスタムモジュールに `{% rich_text %}`モジュールタグまたは`{% widget_block %}`タグを追加する必要があります。

v2 では、リッチテキストフィールドを`{% inline_rich_text %}`モジュールタグでラップできるようになっています。以下に例を示します。

```
{% inline_rich_text field="rich_text_field" value="{{ module.rich_text_field }}" %}
```

CSS/JS の使用

v1 モジュールでは、CSS と JS に対して非常に柔軟なアプローチを取っており、開発者はさまざまな手法を組み合わせることでコードを読み込んでいました。こうしたコードの読み込みに使用されていた、`require_js` と `require_css` をはじめとする多数の関数とタグはサポートされなくなっています。サポートされなくなったタグの全リストについては、[リファレンスセクション](#)をご覧ください。

v2 モジュールでは、CSS と JS に対してより体系的なアプローチを取っています。Eメール用のカスタムモジュールを除き、すべてのカスタムモジュールには CSS と JS のエディターセクションが追加されています。

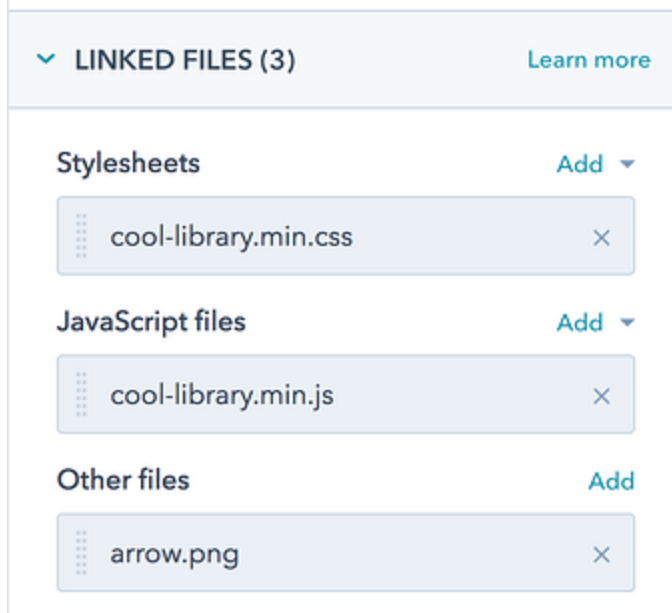
```
HTML + HUBL
1 |<!-- HTML and HubL -->

CSS
1 |*
2 |   CSS only. No HubL.
3 |   Minified and included in head of doc.
4 |   Only included once, even if module is used multiple times on a page.
5 | */

JS
1 | //
2 | //   JS only. No HubL.
3 | //   Minified and included in foot of doc.
4 | //   Wrapped in a closure.
5 | //   Only included once, even if module is used multiple times on a page.
6 | //
```

所定のモジュールに固有の CSS と JS は、エディターのこれらのセクションで記述する必要があります。記述したコードは圧縮されて、表示されるページの先頭と末尾に組み込まれます。JS は名前空間用のクロージャードでもラップされます。

モジュールにライブラリやその他の外部アセットが必要な場合は、インスペクターの [リンク済みファイル] セクションを使用する必要があります。



[リンク済みファイル] を使用して追加された CSS ファイルと JS ファイルは、そのモジュールを使用するページに自動的に読み込まれます。画像などの他のアセットは、モジュールとともにポータル間でコピーされますが、読み込まれるのはコードで参照されている場合のみです (CSS の背景イメージなど)。これらのファイルを参照するときは、以下に示すように `module_asset_url` 関数を使用してください。

```
.arrow{  
  background: url({{ module_asset_url('arrow.png') }});  
}
```

この関数は、モジュールエディターの CSS と JS のセクションで受け入れられる唯一の HubL であることにご注意ください。以降のセクションで、CSS セクションまたは JS セクションで HubL を使用する必要性を感じた場合のソリューションを説明します。

予期しない競合や依存関係のエラーを回避するために、CSS ファイルと JS ファイルの実行順序をよく理解しておいてください。

JavaScript のインクルード順は次のとおりです。

1. external_js
2. js_assets (順次)
3. module js
4. require_js ブロックに含まれる HubL マークアップ内の JS

CSS のインクルード順は次のとおりです。

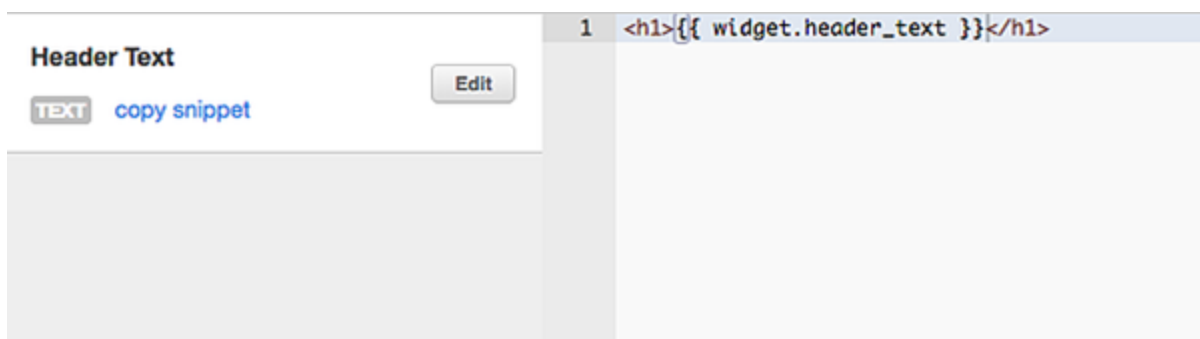
1. css_assets
2. モジュール CSS
3. require_css ブロックに含まれる HubL マークアップ内の CSS

モジュールのフィールドを参照するために 使用するモジュールとウィジェットの違い

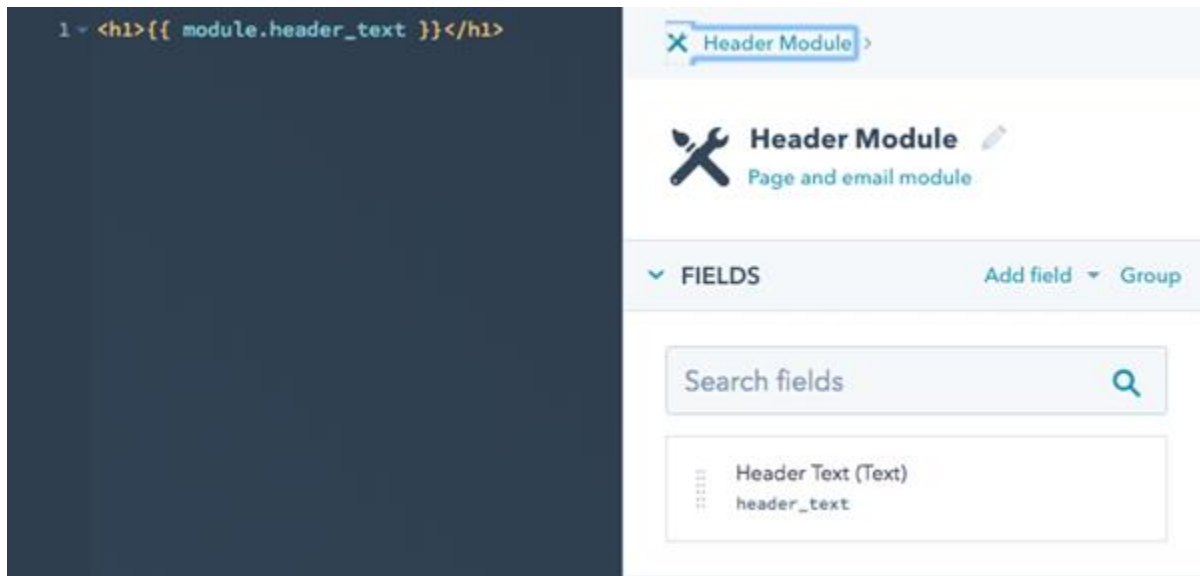
v1 モジュールフレームワークでは、`{{ widget.field }}`を使用してモジュールのフィールドを参照していました。v2 フレームワークでは、`{{ module.field }}`を使用してモジュールのフィールドを参照します。

ただし、v2 フレームワークで`{{ widget.field }}`を使用しても、引き続き正常に機能します。

v1



v2



JS プラグインの使用

上述の順序で JavaScript のインクルードを行うと、モジュール内でコードを整理しやすくなります。通常、サードパーティのスクリプトを使用する場合は、インスペクターの [リンク済みファイル] セクションを使用してスクリプトファイルをリンクします。これにより、コードエディターの JS セクションでプラグインをインスタンス化できるようになります。プラグインにパラメータを渡す必要がある場合、エディターの [HTML + HubL] セクションでデータ属性を使用してパラメータの値を保存できます。下のスクリーンショットに、疑似コードの例が示されています。


```
HTML + HUBL
1 - <div class="do-cool-js" data-width="{{ module.width }}">
2     <p>All kinds of stuff</p>
3 </div>
4
5
6
7
8
CSS
JS
1 - $(' .do-cool-js').each(function(){
2
3     var hublWidth = $(this).attr('data-width');
4
5 -     $(' .do-cool-js').addOnName({
6         width : hublWidth
7     });
8
9 });
10
11
12
```

注:HubL を直接スクリプトに渡すとしても、[HTML + HubL] セクションでモジュールをインスタンス化できます。ただし、エディターの JS セクション内のスクリプトはクロージャ-でラップされるため、[HTML + HUBL] セクションで記述されたスクリプトとは名前空間が異なることにご注意ください。

HTML/HubL の使用

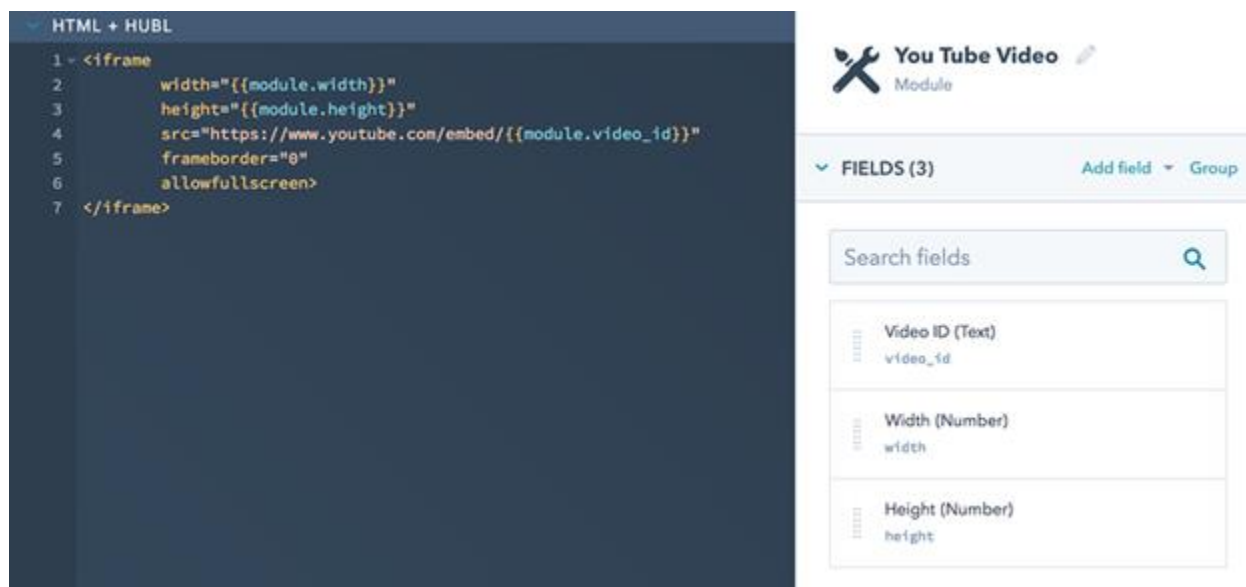
HTML モジュールと HubL モジュールは除去されました。これらのモジュールは、開発者にとって HTML および HubL のスニペットをテンプレートに簡単に追加する手段となっていました。けれども、それによって理解するにも編集するにも難しいテンプレートになりがちです。より予測しやすく、よりメンテナンスしやすいテンプレートにするための長期間にわたる取り組みの一環として、すべてのコードをカスタムモジュールのコンテキスト内に収めることを優先し、これらのモジュールを除去しました。

HTML をどうしてもウェブサイトのページレベルで追加しなければならない場合は、HTML モジュールを [HubSpot マーケットプレイス](#) からダウンロードできます。

推奨事項:HTML/JS の小さいスニペット

開発者が HTML または JS のスニペットが関与する一般的なシナリオのソリューションを前もって用意していないと、コンテンツ編集者が苦勞することがよくあります。埋め込みスクリプトなどのコードがテキストの間に貼り付けられていると、動画が多少なりとも正しくレンダリングされます。ただし正しくレンダリングされない場合、ほとんどのシナリオで推奨されるソリューションはカスタムモジュールを使用することです。

以下の例で、開発者が作成したモジュールに 3 つのフィールドがあります。そのうちの 2 つには、ユーザーがすばやく確実に YouTube 動画を埋め込めるよう、デフォルト値が設定されています。ユーザーに埋め込みコードをリッチテキストモジュールに貼り付けさせる方法とは対照的に、この方法では開発者がマークアップを完全に制御できます。



The image shows a code editor on the left and a module configuration interface on the right. The code editor displays the following HTML code:

```
HTML + HUBL
1 <iframe
2   width="{{module.width}}"
3   height="{{module.height}}"
4   src="https://www.youtube.com/embed/{{module.video_id}}"
5   frameborder="0"
6   allowfullscreen>
7 </iframe>
```

The module configuration interface on the right is titled "You Tube Video Module". It features a search bar labeled "Search fields" and a list of three fields:

- Video ID (Text) with the value `video_id`
- Width (Number) with the value `width`
- Height (Number) with the value `height`

下のスクリーンショットに、このモジュールの編集作業を示します。サイト編集者が YouTube 動画を埋め込む際は、必要なあらゆるオプションが提供されるため、曖昧さやエラーが発生したりする余地はありません。開発者がヘルプテキストとフィールド検

証を追加すれば、編集作業をさらに改善できます。これらの手法について詳しくは、[こちら](#)をご覧ください。

You Tube Video



 Options

Video ID

Width

Height

Save and go to module list

Apply

推奨事項:E メール内の HTML テーブル

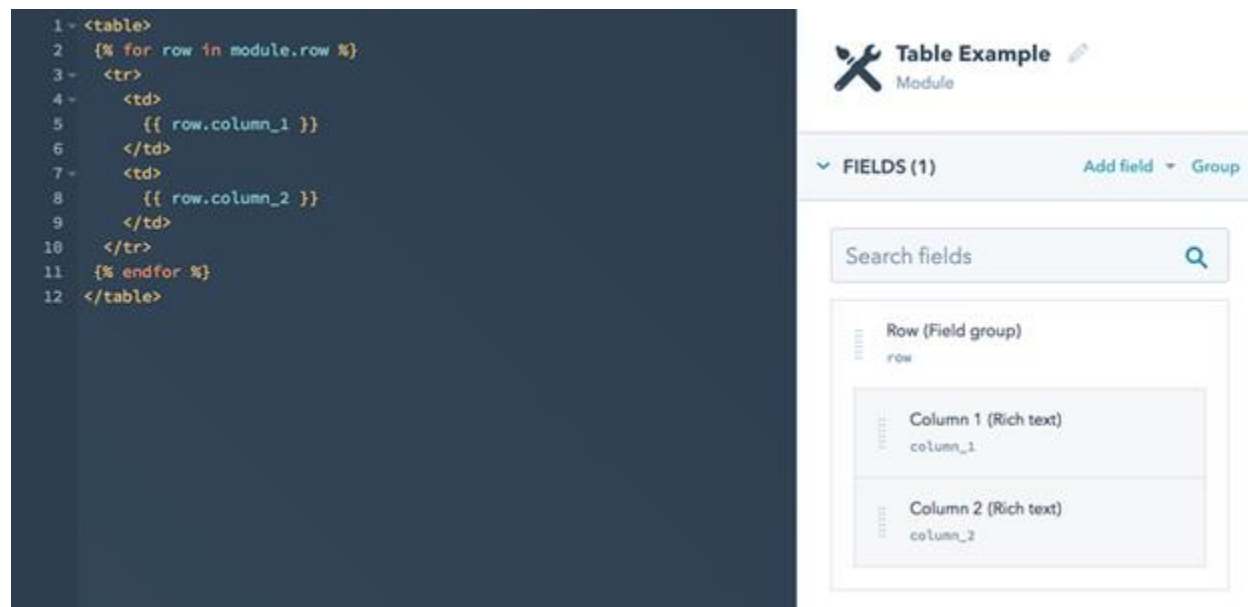
これまで、特定のマークアップを使用した展開可能なコンテンツ領域をクライアントに提供するのは簡単なことではありませんでした。特に、フレキシブルカラムを使用できない E メールとなると、その難易度はさらに高くなります。現在、新しいモジュールフレームワークを使用して、繰り返しフィールドとフィールドのグループを作成できるようになり、これまでは不可能だったことが今では容易な作業になっています。

以下の例では、開発者が Column 1 と Column 2 という名前の 2 つのリッチテキストフィールドを作成し、この 2 つをグループ化して Row というグループ名を付けています。その上で、Row グループのリピーターオプションを有効にしています。

このように設定されたフィールドで、開発者は各行をループ処理して列データをテーブルセルに挿入しています。この例はわかりやすくするためにかなり単純にしていますが、この手法を拡張して改良すれば、編集者が非常に複雑なレイアウトを広範囲にわたって制御できると同時に、開発者が引き続きマークアップのすべての側面を制御できます。

繰り返しとループ処理について詳しくは、[こちら](#)をご覧ください。

グループについて詳しくは、[こちら](#)をご覧ください。



The image displays a code editor on the left and a visual table structure on the right. The code editor shows the following HTML code:

```
1 <table>
2   {% for row in module.row %}
3   <tr>
4     <td>
5       {{ row.column_1 }}
6     </td>
7     <td>
8       {{ row.column_2 }}
9     </td>
10  </tr>
11  {% endfor %}
12 </table>
```

The visual representation on the right, titled "Table Example Module", shows a table structure with a search bar and a "FIELDS (1)" section. The table structure is as follows:

Row (Field group)
row
Column 1 (Rich text) column_1
Column 2 (Rich text) column_2

コードギャラリー

