

テーマ

開発

BOOK



Six Apart 編集

Movable Type .net のベーステーマは使えるけど…

- ☒ 「既存のテーマじゃ物足りない！」
- ☒ 「MTタグって難しそう～」
- ☒ 「自由にテンプレートをカスタマイズしたい！」



そんな方に…

知識

と

実践

で

サンプルサイトを作りながら

MTタグを習得!



トフでもできる!?テーマ開発講座

もくじ

この冊子はウェブサイト「トフでもできる!?テーマ開発講座」(<https://developer.movabletype.net/themeguide/>)を編集したものです。ウェブサイトでは、冊子で省略された内容を含め、全てのコンテンツが閲覧できます。

導入ー学習をはじめよう!

MovableType.net テーマの学習をはじめよう!	4	テーマってなに?	4
-------------------------------------	---	----------------	---

知識編

1. テーマはどんな風にできているの?

テーマの構成	6
MTタグとは?	6
テーマをつくるテンプレート、モジュール	7

2. 開発者向け! 管理画面の使い方

テンプレート	9
ファイルマネージャ	9
カスタムフィールド	9

実践編

0. 準備

どんなテーマを作るか考える	12
サンプルテーマをダウンロードしよう	12
テーマを書き換えよう	13
ファイルをアップロードしよう	15

1. MT化するーメインページ編

メインページを作る・その1-MTタグを使う	16
メインページを作る・その2-ナビ編	19
メインページを作る・その3-内容編	25

2. ファイルマネージャを使おう

ファイルマネージャにあるファイルを表示させる	32
------------------------------	----

3. モジュールの活用

モジュールを作って効率的なサイトに!	35
--------------------------	----

4. MT化するーウェブページ、ニュースページ編

ウェブページテンプレートを作る!	38
ニュースページを作る・その1-記事ページ編	40
ニュースページを作る・その2-news.html編	43
ニュースページを作る・その3-アーカイブ編	47

5. カスタムフィールドを使う

カスタムフィールドを設定する	51
画像を表示させるカスタムフィールドを作る	51

6. MTタグを使いこなそう

configモジュールとは?	55
configモジュールを使ってみよう	56

7. MT化するー404ページ編

404ページを作ろう!	57
-------------------	----

8. テーマを書き出す・読み込む

テーマを書き出す	58
テーマを読み込む	58

ex. カスタマイズしてみよう

MTタグでサイトの設定をしよう	59
-----------------------	----

さらにMTタグの学習をしよう!

カスタマイズの参考になるサイト	61
-----------------------	----

いっしょに
がんばろう!

導入 - 学習をはじめよう!



MovableType.net テーマの学習をはじめよう!

MovableType.net を使ったサイトを運営しているとき、「表示させる記事の数を増やしたい」と思ったことはありませんか? カスタマイズをしてみようとマニュアルを見ても、「MTタグ」「テンプレート」「テーマ」など、聞きなれない言葉ばかりでお手上げ……

本書はそんな方に向けた MovableType.net を基礎から学習できる教本です。

構成は、**知識編**→**実践編**の2ステップです。**知識編**では MovableType.net のつくりを学習します。部分的なカスタマイズをしたい場合も、まずは「**知識編**」を理解してから、「**逆引き操作リスト**」や「**マニュアル**」へ進むとスムーズでしょう。**実践編**では実際にサイトをテーマ化 (MovableType.net で運用するテーマにしていくこと) をしながら、MovableType.net の構造やMTタグを学習していきます。

❖ 本書は、こんな方にオススメです!

既存のテーマを使ってサイトを運営しているけど、部分的にカスタマイズしたい!

MovableType.net を使って自分でサイトを作りたい!

HTML、CSSは理解しているので、次はMTタグの勉強をしたい!

テーマってなに?

❖ テーマ作成のメリット

MovableType.net にはさまざまな運用に合わせたテーマが用意されています。既存のテーマだけでは物足りない! という場合は、より使いやすいサイトにするためにデザインをカスタマイズしたり、一から設計することも可能です。**ぜひカスタマイズ方法をマスターして、オリジナルのサイトのテーマを作ってみましょう!**

作成したサイトは、テーマとして書き出し、読み込むことができます。これを**テーマ化**と呼びます。

テーマ化をすると、一度作ったデザインと似たサイトを作る際に読み込んだテーマを流用できるため、**作成の時間や手間が少なくなります**。MovableType.net を使ったクライアントワークを考えている方は、ひな形になるテーマを複数用意し、その中からクライアントに選んでもらったものをベースに制作するといった運用も考えられます。



知識編

テーマはどんな仕組みでできているのか？
や、開発者向け管理画面の使い方を解説
します。実践編の前にこちらを理解して
おきましょう。

テーマはどんな風にできているの？

テーマの構成

実際のテーマとはどのようにできているのでしょうか？
テーマのファイルの構成を見てみましょう。各ページに使われる「**テンプレート**」がページの役割分だけあり、スタイルを指定する「**CSS**」、テーマ内で使われる「**画像**」、それと「**テンプレート・モジュール**」が入っています。



(例)「Simple Corporate」のテーマの構成

MTタグとは？

テンプレートの中身を見てみましょう。
<html><body><div>...など、他のサイトでも見られる一般的なHTMLタグが書かれています。よく見ると、それ以外に<mt: ～>というタグが出てきます。これが Movable Type 特有のタグ、**MTタグ**です。

```
1 <!DOCTYPE html><$mt:Include module="config"
2 <html lang="<$mt:BlogLanguage$>" itemscope i
3 <head>
4 <meta charset="<$mt:PublishCharset$>">
5 <title><$mt:BlogName encode_html="1"></titl
6 <meta name="description" content="<$mt:BlogD
7 <meta name="keywords" content="<$mt:GetVar n
8 <meta name="viewport" content="width=device-
```

MTタグは、Movable Type のテンプレートに使われる独自タグで、**テーマ内で使われてる文字列を探して表示**させたり、**機能を設定**するために使います。

例えば「ブログのタイトルを表示させたい」ときや「ここにニュース記事を3つ表示させたい」といったときです。MTタグにはたくさんの種類があり、さらに識別子を組み合わせているんなカスタマイズができます。

テーマをつくるテンプレート、モジュール

テンプレートは、インデックス、記事、ウェブページ、404など…サイトの中で必要になる**ページのひな形**です。テンプレートには以下の3つの種類があります。

インデックス・テンプレート	
+ 新規	
☐ 名前	出力ファイル
☐ ニュース TOP 🔗	news.html
☐ 最新記事のフィード 🔗	atom.xml
☐ スタイルシート 🔗	styles.css
☐ インデックスページ 🔗	index.html
sitemap.xml 🔗 は自動的に生成されるため、インデックステンプレートとして作成することはできません。	
削除	

(例) 「Simple Corporate」 デザイン > テンプレート

インデックス・テンプレート

index.html、news.htmlなど1つのサイトに1つにしかないページのテンプレート。

アーカイブ・テンプレート

記事、ウェブページなど、管理画面から投稿すると増えていくページのテンプレート。

システムテンプレート

システム上必要なページのテンプレート。

❖ モジュール

正しくは**テンプレート・モジュール**といいます。インデックス・テンプレートやアーカイブ・テンプレートの中で何度も使われている記述をパーツのようにしてまとめ、必要な場所で呼び出して使いまわしています。

この講座では、テンプレート・モジュールのことを、単に**モジュール**と呼ぶこともあります。

テンプレート・モジュール	
+ 新規	
☐ 名前	
☐ common_head_js	
☐ config	
☐ footer	
☐ blog-sidemenu	
☐ webpage-breadcrumbs	

(例) 「Simple Corporate」 デザイン > テンプレート

例えば、ヘッダーやナビゲーションはこのページにもありますよね。それをモジュール化し、テンプレートの中で該当のモジュールを呼び出せば、同じタグを何度も書かなくて良いというわけです。 **モジュールの呼び出しには、そのためのMTタグを使います。**

[illegible]

(例)「Simple Corporate」 デザイン>テンプレート>インデックスページ
編集画面右に、使っているモジュールが表示されています。

ベーステーマとして提供している「Simple Corporate」を参考に具体的な使い方を見ていきましょう。

「Simple Corporate」のウェブページ編集画面では、チェックボックスのカスタムフィールドが使われています。「グローバルナビに表示」をチェックすると、情報が出力されてテンプレート側でカスタムフィールドの値を判別します。この場合、グローバルナビゲーションにページへのリンクが表示されました。(右図参照)

グローバルナビに表示

☒ グローバルナビに表示する場合にチェックを入れます。

「Simple Corporate」ウェブページ編集画面 > 会社概要 のカスタムフィールドにチェック!



グローバルナビゲーションに、このページへのリンクが表示されました。

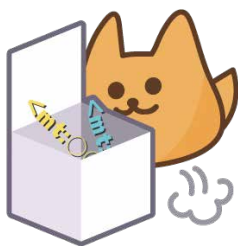
こんな仕組みも作れるのが**カスタムフィールド**です。組み合わせ次第でいろいろな活用方法があるので、上手に使っていきましょう! 詳しい使い方は「**実践編**」で扱っていきますので、ここではこんな機能があるということを覚えておいてくださいね。



ここまでで、知識編は終了です。MTテーマがテンプレート、MTタグ、カスタムフィールドでつくられていることが理解できたでしょうか?

まだピンとこないかもしれませんが、テーマを作ってみると各場所の動きが掴めると思います!

次からは「**実践編**」で実際に、テーマを制作していきましょう!!



実践編

サンプルサイトを使って実際にMTタグを使いながら、主要なタグの使い方や基本知識として押さえておきたいポイントをご紹介します。

どんなテーマを作るか考える

テーマを作成する前に、**どんなテーマを作るのか？どんなものが必要か**を考えます。サイトのデザインも、先に決めておく必要があるでしょう。



サンプルテーマをダウンロードしよう

❖ ダウンロード

今回は練習として、スタンダードなサイトが作れるサンプルファイルを用意しています。ファイルは、右のURLからダウンロードできます。また、今回作成するテーマのサンプルサイトも確認しておきましょう。

ダウンロード用サンプルファイル

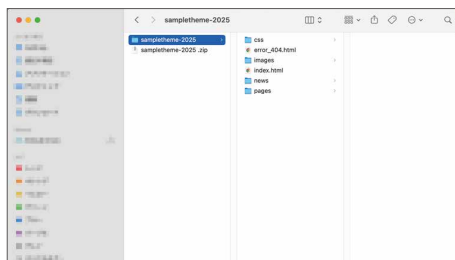
<https://developer.movabletype.io/themeguide/.assets/sampletheme-2025.zip>

サンプルサイト

<https://developer.movabletype.io/themesample202503/>

❖ サイトの構成を確認しよう

では、ダウンロードしたzipファイルの中身を見ていきましょう。次ページのファイルが入っています。フォルダに入っているファイルと、対応するテンプレートを考えてみましょう。



index.html

インデックス・テンプレート>メインページ
サイトの**メインページ**になるテンプレートです。

news/index.html

インデックス・テンプレート>ニュース TOP
ニュースや、**お知らせ記事のページ**になるテンプレートです。

news/archive.html

アーカイブ・テンプレート>〇〇別ニュース記事リスト
記事のアーカイブになるテンプレートです。

news/entry.html

アーカイブ・テンプレート>記事
1つの記事ページそのものになるテンプレートです。

pages/index.html

アーカイブ・テンプレート>ウェブページ
ウェブページになるテンプレートです。

css/styles.css

インデックス・テンプレート>スタイルシート
スタイルシートのテンプレートです。これを編集するだけでもオリジナリティが増します。

error_404.html

システムテンプレート> 404 エラー
存在しないページ (404)をリクエストした場合に表示されるページです。

images

使用する画像が入ったフォルダです。

テーマを書き換えよう

❖ 新しいサイト・ブログを用意

テーマを作るには、これからテンプレートを編集してもよいサイト、またはブログが必要です。
[MovableType.net](https://movabletype.net) のダッシュボードから、「**新しくウェブサイトを**作る」か「**新しくブログを**作る」をクリックし、テーマ作成用の環境を用意しましょう。



既存のサイトを使いたくない場合は**ブログの横にある+**ボタンをクリックし、新しいブログを作って作業しましょう。

MT化する - メインページ編

メインページを作る・その1 - MTタグを使う

❖ メインページを作る準備

設定>全般を開くと、名前の欄で先ほど設定したブログ名を確認できます。このブログ名を、サイトのメインページに表示していきます。

ではここから、MTタグを入れ込んでいきましょう！
まずは、「**メインページ**」をテンプレート化していきます。
デザイン>テンプレートから、index.htmlのテンプレートを開きましょう。



❖ MTタグを使ってみよう

サイト名を表示させる

「ここにサイト名」となっている箇所。ここにはサイト名を表示させたいですね。そんなときに使うのが**MTタグ**！ここには「**サイトのタイトルを表示させる**」ためのMTタグを入れましょう。



<{\$mt:BlogName\$}>

ブログの名前を表示します。

```
19 <body>
20 <header>
21 <div class="mt-container">
22 <a href="#">
23 <h1 class="mt-h1">ここにサイト名</h1>
24 </a>
25 <ul class="navi-list">
26 <li>
27 <a href="#">ページ1</a>
```

【インデックス・テンプレート「メインページ」】ここを…

```
19 <body>
20 <header>
21 <div class="mt-container">
22 <a href="#">
23 <h1 class="mt-h1"><{$mt:BlogName$}></h1>
24 </a>
25 <ul class="navi-list">
26 <li>
27 <a href="#">ページ1</a>
```

このように書きかえ!

これを保存し、サイトを見てみると…?

[デモサンプル](#)

[ページ1](#)

[サブメニュー](#)

[サブメニュー](#)

[サブメニュー](#)

[サブメニュー](#)

[ページ2](#)

[サブメニュー](#)

[サブメニュー](#)

[サブメニュー](#)

[サブメニュー](#)

[ページ3](#)

[サブメニュー](#)

[サブメニュー](#)

[サブメニュー](#)

[サブメニュー](#)

[ページ1](#)

[ページ2](#)

[ページ3](#)

サイト名が入りました!

MTタグを入れたところに、先ほど管理画面で設定したサイト名が入っています。このようにMTタグは**MT内で設定した文字列や、作ったページなどが表示できる**のです。では、どんどん作っていきましょう!

サイト/ブログのURLを表示させる

サイト名には、メインページに戻るためのリンクが貼ってあるといいですね。そんなときは**URLを表示させるMTタグ**を使いましょう。

<\$mt:BlogURL\$>

ブログのURL（サイトURL）をhttp://から始まる絶対URLで表示します。

<\$mt:BlogRelativeURL\$>

ブログのURLを、ホストからの相対URLで表示します。たとえば、ブログのURLがhttp://blog.example.com/first-weblog/の場合、/first-weblog/を表示します。このタグが「http://〜〜」に置き換わるので、このタグはa href="〇〇"に入れて使います。

```
18 </head>
19 <body>
20 <header>
21   <div class="mt-container">
22     <a href="#">
23       <h1 class="mt-h1"><$mt:BlogName$></h1>
24     </a>
25     <ul class="navi-list">
26       <li>
27         <a href="#">ページ1</a>
28       <ul class="navi-list__sub">
```



```
<a href="<$mt:BlogURL$>">
  <h1 class="mt-h1"><$mt:BlogName$></h1>
</a>
```

リンクのアドレスが**ブログのURL**に指定されました！
リンク付きのサイト名が完成！

ブログのURLを表示させる方法がわかったところで、読み込まれているCSSのアドレスなども**相対パス**で呼び出してみましょう。

テーマサンプル

ページ1
サブメニュー
サブメニュー
サブメニュー
サブメニュー
ページ2
サブメニュー
サブメニュー
サブメニュー
サブメニュー
サブメニュー
ページ3
サブメニュー
サブメニュー

【サンプルサイト】見た目ではわかりませんが…リンク先アドレスに注目！

```

9   <link rel="stylesheet" type="text/css" href="//cdnjs.clo
10  <link href="/css/styles.css" rel="stylesheet">
11  <link
12  rel="stylesheet"

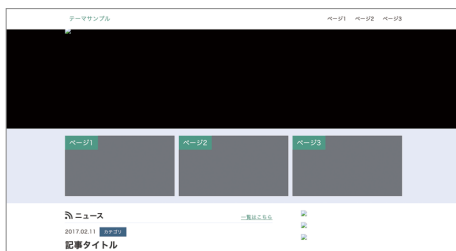
```



```
<link href="<$mt:BlogRelativeURL$>styles.css" rel="stylesheet">
```

CSSはサンプルで用意してあるので、正しく読み込まれると、**サイトにスタイルが当たります**。スタイルシートが適用されました。サイトの形になりましたね！

今回のサンプルサイトには使われていませんが、javascriptなどの外部ファイルを使用する場合も同じように設定できます。



メインページを作る・その2-ナビ編

❖ ナビゲーションの仕組み

続いて、ナビゲーションをMT化していきます。今表示しているページ1、2…というのは、HTMLで書いているものなので、[MovableType.net](http://movabletype.net) で作ったウェブページが表示されているわけではありません。これを、**MovableType.net 上で作ったページのタイトルが並ぶように**します。

まずは管理画面、ウェブページ>新規から、**ウェブページをいくつか作りましょう**。

作ったら、再びテンプレート編集画面に戻ります。ウェブページ関連のMTタグをご紹介します。ウェブページ一覧を表示させるには、範囲を `<mt:Pages> ~ </mt:Pages>` で囲み、その中にウェブページのタイトルを表示させる `<$mt:PageTitle$>`、ウェブページのアドレスを表示させる `<$mt:PagePermalink$>` を使います。



ナビ部分です。



わかりやすいタイトルをつけてあげましょう。

<mt:Pages>

ウェブページの一覧を表示するためのMTタグです。ブロックタグと呼ばれる種類に属します。タグの種類についてはこのあと解説します。

<\$mt:PageTitle\$>

ウェブページのタイトルを表示します。

<\$mt:PagePermalink\$>

現在のページのアーカイブへの絶対URLを表示します。

❖ タグの種類

上で、<mt:Pages>～</mt:Pages>で「囲む」、また「ブロックタグ」と解説しました。実はMTタグには種類があり、その種類によって使い方が異なります。

例えば、ページ名やページのアドレスを取得したい場合は、<mt:Pages>～</mt:Pages>のブロックタグで囲んであげると、そのブロック内のページの名前が、ファンクションタグ<\$mt:PageTitle\$>で取得できるようになります。

ブロックタグ

開始タグと終了タグがあり、ひとまとまりのテンプレートを囲みます。(HTMLタグのような使い方ですね。) <mt:Pages>は、これにあたります。

ファンクションタグ

単体での利用が可能です。どこでも利用可能なもの、特定のブロックタグの囲み内でのみ、利用可能なものがあります。

❖ ナビゲーションを表示させる

以上を踏まえて、**ウェブページのタイトルとリンク先をMTタグで取得**できるように記述してみます。一度、ナビに関するすべての記述を削除し、次ページのコードに書き換えましょう。

```
24 </a>
25 <ul class="navi-list">
26 <li>
27   <a href="#">ページ1</a>
28   <ul class="navi-list__sub">
29     <li><a href="#">サブメニュー</a></li>
30     <li><a href="#">サブメニュー</a></li>
31     <li><a href="#">サブメニュー</a></li>
32     <li><a href="#">サブメニュー</a></li>
33   </ul>
34 </li>
35 <li>
36   <a href="#">ページ2</a>
37   <ul class="navi-list__sub">
38     <li><a href="#">サブメニュー</a></li>
39     <li><a href="#">サブメニュー</a></li>
40     <li><a href="#">サブメニュー</a></li>
41     <li><a href="#">サブメニュー</a></li>
42   </ul>
43 </li>
44 </ul>
```



```
<ul class="navi-list">
<mt:Pages>
  <li>
    <a href="<mt:PagePermalink$>"><mt:PageTitle$></a>
  </li>
</mt:Pages>
</ul>
```

一旦、サンプルに実装されているホバーの仕組みをなくしてしまいます。(後でまたつけますので安心してくださいね。) ページの数だけを自動で取得してくれるので以前より短くなります。

これで保存してみると…ページが表示され、メニューになりました！



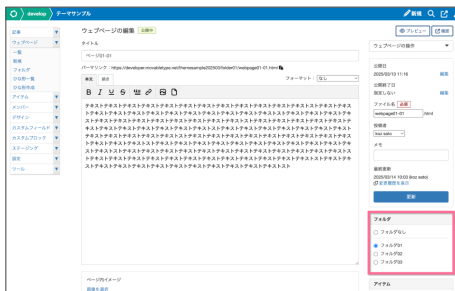
❖ フォルダ内のウェブページを取得する

さて、このナビゲーション。このままでいいのですが、さらに [MovableType.net](https://movabletype.net) の機能に合わせて改良していく必要があります。それはどこかというところ… **フォルダ！！**

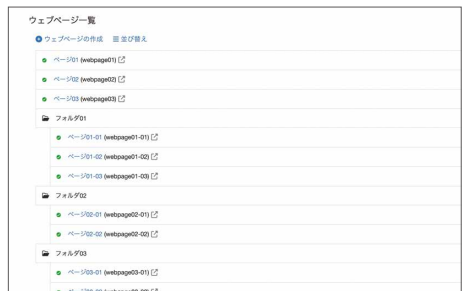
ウェブページはフォルダに入れることができます。ウェブページをフォルダ構造にしたら、ナビにはフォルダ名を表示させたいですね。ホバーの仕組みはCSSの方に用意してあるので、ここではフォルダ名と、中のウェブページ名を表示する方法を覚えましょう。

確認のため、管理画面の **ウェブページ > フォルダ > フォルダ作成** から新しいフォルダを作ります。新規でウェブページを作成し、右カラムのフォルダ選択フィールドで **ウェブページが属するフォルダを選択**してから、保存します。これで準備は完了です！

サブカラムの **ウェブページ > 一覧** から、フォルダ内の階層を確認することができます。



フォルダ選択フィールド



こんな感じになっていればOKです。

ウェブページの階層をホバーで表示する部分を再度作っていきます。削除したホバーのCSSに対応させたHTMLをもとに、先ほどのMTタグを入れたものが、次のコードです。

```
<ul class="navi-list">
  <mt:Pages>
    <li>
      <a href="<$mt:PagePermalink$"><$mt:PageTitle$"></a>
      <ul class="navi-list__sub">
        <li><a href="#">サブメニュー</a></li>
        <li><a href="#">サブメニュー</a></li>
        <li><a href="#">サブメニュー</a></li>
        <li><a href="#">サブメニュー</a></li>
      </ul>
    </li>
  </mt:Pages>
</ul>
```

これをもとに考えてみましょう。また、**フォルダの中にフォルダを置くこともできるので**（ここでは最上位のフォルダ、サブフォルダと呼びます）すべてのフォルダの中に入っているページが表示できるようにします。

<mt:TopLevelFolders>

現在のフォルダに関わりなく、最上位のフォルダからサブフォルダを含めたリストを表示するブロックタグです。<mt:TopLevelFolders>～</mt:TopLevelFolders>で囲んで使います。

<mt:SubFolders>

現在のフォルダのサブフォルダを階層化して一覧表示するブロックタグです。

<\$mt:FolderLabel\$>

フォルダの名前を表示します。

<\$mt:SubFolderRecurse\$>

そのフォルダに属するサブフォルダの<mt:SubFolders>ブロックを表示します。

これをホバーができるHTMLに組み込みます。

```
23 <div class="navi-list">
24   <h1 class="mt-h1"><$mt:BlogName$></h1>
25   <ul class="navi-list">
26     <mt:Pages>
27       <li>
28         <a href="<$mt:PagePermalink$"><$mt:PageTitle$"></a>
29       </li>
30     </mt:Pages>
31   </ul>
32 </div>
33 </header>
```



```

<ul class="navi-list">
  <mt:TopLevelFolders>
    <li>
      <a href="#">
        <$mt:FolderLabel$>
      </a>
      <ul class="navi-list__sub">
        <mt:Pages>
          <li>
            <a href="<$mt:PagePermalink$>">
              <$mt:PageTitle$>
            </a>
          </li>
        </mt:Pages>
        <mt:SubFolders>
          <li class="navi-list__sub--title">
            <$mt:FolderLabel$>
          </li>
          <mt:Pages>
            <li>
              <a href="<$mt:PagePermalink$>">
                <$mt:PageTitle$>
              </a>
            </li>
          </mt:Pages>
          <$mt:SubFolderRecurse$>
        </mt:SubFolders>
      </ul>
    </li>
  </mt:TopLevelFolders>
</ul>

```

これでフォルダ名とフォルダに含まれたページが表示されました。最後に、フォルダに入っていないページを表示するための記述を追加します。

フォルダ01	フォルダ02	フォルダ03
	ページ01-01	
	ページ01-02	
	ページ01-03	

```

<mt:Pages no_folder="1">
  <li>
    <a href="<$mt:PagePermalink$>">
      <$mt:PageTitle$>
    </a>
  </li>
</mt:Pages>

```

を、<mt:TopLevelFolders>～</mt:TopLevelFolders>の後ろに追加しましょう。

no_folder="1"

フォルダに属さない、トップレベルに作成されたウェブページのみを表示する、**モディファイア**です。(モディファイアについては後ほど詳しく説明します。)

最終的なコードはこうになりました。

```
<ul class="navi-list">
  <mt:TopLevelFolders>
    <li>
      <a href="#">
        <$mt:FolderLabel$>
      </a>
      <ul class="navi-list__sub">
        <mt:Pages>
          <li>
            <a href="<$mt:PagePermalink$>">
              <$mt:PageTitle$>
            </a>
          </li>
        </mt:Pages>
        <mt:SubFolders>
          <li class="navi-list__sub--title">
            <$mt:FolderLabel$>
          </li>
          <mt:Pages>
            <li>
              <a href="<$mt:PagePermalink$>">
                <$mt:PageTitle$>
              </a>
            </li>
          </mt:Pages>
          <$mt:SubFolderRecurse$>
        </mt:SubFolders>
      </ul>
    </li>
  </mt:TopLevelFolders>
  <mt:Pages no_folder="1">
    <li>
      <a href="<$mt:PagePermalink$>">
        <$mt:PageTitle$>
      </a>
    </li>
  </mt:Pages>
</ul>
```

これで保存してみると…フォルダ名、フォルダに入っていないページの全てを表示できました！フォルダを増やしたりして、確認してみてくださいね。

フォルダ01	フォルダ02	フォルダ03	ページ01	ページ02	ページ03
--------	--------	--------	-------	-------	-------

メインページを作る・その3- 内容編

❖ ニュース記事の一覧を表示させる

次はニュース記事の表示をMT化していきます。管理画面の記事から**ニュース記事**をいくつか**作っておきましょう**。このときに記事にカテゴリの設定をしておくと、日付の隣にカテゴリ名を表示することができます。

作ったニュース記事を一覧にしてメインページに載せていきます。ニュース記事を表示させる場所は右図の枠で囲んだエリアです！

ウェブページと同様に、記事にも**タイトル、URLを表示させるMTタグ**があります。



<mt:Entries>

ブログ記事の一覧のためのブロックタグです。まずはこのブロックタグで囲みましょう。

<\$mt:EntryTitle\$>

ブログ記事のタイトルを表示します。

<\$mt:EntryPermalink\$>

ブログ記事を含むアーカイブの絶対URLを表示します。

<\$mt:EntryDate\$>

ブログ記事の公開日を表示します。モディファイアの指定で色々なオプションがつけられます。

<\$mt:EntryBody\$>

ブログ記事の本文を表示します。

<\$mt:EntryExcerpt\$>

ブログ記事の概要に入力した内容を表示します。指定がない場合はブログ記事の本文を、最初から 40 文字 (デフォルト値) 表示します。(後述のモディファイアで指定します。)

こちらもウェブページと同様に、URLを表示させるMTタグと組み合わせて、テンプレートに入れ込んでみましょう。

```

109     <p class="newsArea__newsRelease">
110     <a href="#"><!-- 一覧はこちら -->
111     </a>
112     </p>
113     <dl class="newsArea__dateList">
114     <dt>
115     <p class="newsArea__day"><!-- 2017.02.11 -->
116     <a href="news/archive.html">
117     <p class="newsArea__category"><!-- カテゴリ -->
118     </p>
119     </dt>
120     <dt>
121     <p class="newsArea__title"><!-- 記事タイトル -->
122     </p>
123     <dd>
124     <a href="#"><!-- テキストテキスト -->
125     <p class="newsArea__link">
126     <a href="#"><!-- 続きを読む -->
127     </a>
128     </p>
129     </dd>
130     </dl>
131     </div>
132     <div class="top-bannerArea">
133     <div>

```

}

```

144     <p class="newsArea__day"><!-- 2017.02.11 -->
145     <a href="news/"><p class="newsArea__category"><!-- カテゴリ -->
146     </p>
147     </dt>
148     <dt>
149     <p class="newsArea__title"><!-- 記事タイトル -->
150     </p>
151     <dd>
152     <a href="#"><!-- テキストテキスト -->
153     <p class="newsArea__link">
154     <a href="#"><!-- 続きを読む -->
155     </a>
156     </p>
157     </dd>
158     </dl>
159     </div>
160     <div class="top-bannerArea">
161     <div>

```



```

<mt:Entries>
  <dl class="newsArea__dateList">
    <dt>
      <p class="newsArea__day"><$mt:EntryDate$></p>
      <a href="news/"><p class="newsArea__category">カテゴリ</p></a>
    </dt>
    <dt>
      <p class="newsArea__title"><$mt:EntryTitle$></p>
    </dt>
    <dd>
      <$mt:EntryExcerpt$>
      <p class="newsArea__link">
        <a href="<$mt:EntryPermalink$>"> 続きを読む</a>
      </p>
    </dd>
  </dl>
</mt:Entries>

```

記事一覧が表示されました！！

一覧はこちらにはニュース一覧を表示させる、news.htmlへのリンクを貼ってあげましょう。前に使った、ブログのURLを表示させる<\$mt:BlogRelativeURL\$>を使ってリンクを完成させます。



```

105     <div class="top-newsArea__header">
106       <h2 class="top-newsArea__title">
107         <i class="fa fa-feed"></i>ニュース
108       </h2>
109       <p class="top-newsArea__newsRelease">
110         <a href="#">一覧はこちら</a>
111       </p>
112     </div>
113     <mt:Entries>
114       <dl class="newsArea__dateList">
115         <dt>

```



```

<a href="<$mt:BlogRelativeURL$>news.html">一覧はこちら</a>

```

❖ カテゴリ名を表示させる

記事の表示が完了したら、カテゴリ名も正しく表示していきましょう。現在「カテゴリ」と表示されている場所に、実際に記事が属しているカテゴリの名前を表示させます。さらに、クリックするとカテゴリ別アーカイブに飛ぶようにリンクを設置してみましょう。カテゴリ関連のMTタグはこちらです。

<mt:EntryCategories>

ブログ記事に指定したカテゴリの一覧のためのブロックタグです。

<\$mt:CategoryLabel\$>

カテゴリ名を表示します。

<\$mt:CategoryArchiveLink\$>

カテゴリアーカイブの URL を表示します。

先ほど書き換えた、<mt:Entries>～</mt:Entries> 内にあるカテゴリに関する箇所を、以下のように変更していきます。

```

<mt:Entries>
  <dl class="newsArea__dateList">
    <dt>
      <p class="newsArea__day">
        <$mt:EntryDate$>
      </p>
      <mt:EntryCategories>
        <a href="<$mt:CategoryArchiveLink$>" class="category"><p
class="newsArea__category"><$mt:CategoryLabel$></p></a>
      </mt:EntryCategories>
    </dt>
    <dt>
      <p class="newsArea__title"><$mt:EntryTitle$></p>
    </dt>
  </dl>

```

```

<dd>
  <$mt:EntryExcerpt$>
  <p class="newsArea__link">
    <a href="<$mt:EntryPermalink$>">続きを読む</a>
  </p>
</dd>
</dl>
</mt:Entries>

```

❖ モディファイアを使う

さて、マニュアルや既存のテーマを見ていると、MTタグの中に`lastn="1"`や`sort_by="title"`などが付いているものがありますね。これらを**モディファイア**といい、設定することで**MTタグに細かい指定ができる**ようになります。

例えば先ほど指定した、ブログ記事の一覧を表示する`<mt:Entries>`には、**記事を指定した数だけ表示する**モディファイアや、**並べる順番のルールを指定する**モディファイアが用意されています。またカスタムフィールドの値を使って表示したり、振り分けたりもできます。組み合わせればテーマの幅が広がります！ぜひマスターしていきましょう！

今回は、**“エントリー記事の一覧”を、決まった記事数だけ表示させてみよう**と思います。モディファイアは、`limit="N"`を追加しましょう。

`sort_by="foo", sort_order="ascend | descend"`を追加すると、**ブログ記事を並び替える対象と並び順を指定できる**ようになります。`sort_by="foo"`の初期値は公開日順、`sort_order="ascend | descend"`の初期値は降順（新しいものが上）、になっているため、今回は追記しなくても最新記事を指定した数だけ表示することができます。

limit="N"

limit モディファイアは、指定された条件でフィルタリングした結果を指定数を最大件数として表示します。指定可能な値は1-100のみです。

sort_by="foo"

ブログ記事を並び替える対象を指定します。初期値は公開日順です。

sort_order="ascend | descend"

並べる順序を指定します。初期値は `descend` です。

メインページに最新記事を3件だけ表示させてみます。

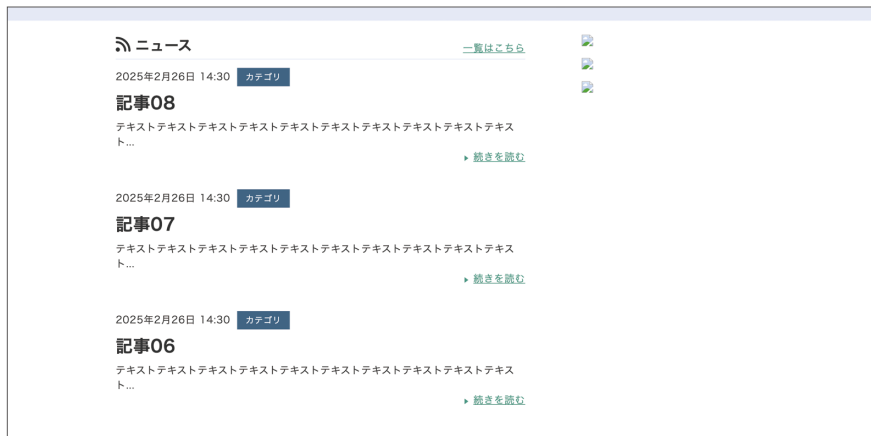
```

109 <p class="top-newsArea__newsRelease">
110 <a href="#">一覧はこちら</a>
111 </p>
112 </div>
113 <mt:Entries>
114 <dl class="newsArea__dateList">
115 <dt>
116 <p class="newsArea__day"><$mt:EntryDate$></p>

```



```
<mt:Entries limit="3">
```



最新記事が3件表示されました！

最新の3件以外にも、モディファイアの指定で任意の記事を出力することができます。タグリスト (<https://movabletype.net/tags/>) を参考にカスタマイズしてみましょう！同様に、日付を表示するタグもモディファイアで好きな表示方法にできます。(日付に関するテンプレートタグのモディファイアリファレンス: <https://movabletype.net/tags/date-formats.html>)

ブログ記事の概要に入力した内容を表示する `<$mt:EntryExcerpt$>` も、モディファイア `words="20"` で出力する文字数を指定できます。この

とき、記事の概要フィールドに入力がなければ、本文が出力されます。

また、カテゴリの一覧を表示する

`<mt:EntryCategories> ~ </mt:EntryCategories>` は、モディファイアの指定がない場合、記事が属している全てのカテゴリが表示されます。

今回は、メインに設定したカテゴリのみを表示させたいので、モディファイア `type="primary"` を付けましょう。上の最新ニュース一覧エリアはこのようなりました。

```

<mt:Entries limit="3">
  <dl class="newsArea__dateList">
    <dt>
      <p class="newsArea__day">
        <$mt:EntryDate format="%Y.%m.%d"$>
      </p>
      <mt:EntryCategories type="primary">
        <a href="<$mt:CategoryArchiveLink$>" class="category">
          <p class="newsArea__category"><$mt:CategoryLabel encode_html="1"$></p>
        </a>
      </mt:EntryCategories>
    </dt>
    <dt>
      <p class="newsArea__title"><$mt:EntryTitle$></p>
    </dt>
    <dd><$mt:EntryExcerpt words="20"$>
      <p class="newsArea__link">
        <a href="<$mt:EntryPermalink$>">続きを読む</a>
      </p>
    </dd>
  </dl>
</mt:Entries>

```

❖ ウェブページの一覧を表示させる

同様に、ウェブページの一覧を表示させましょう！以下の画像の箇所になります。

ウェブページの情報を表示させるには<mt:Pages>～</mt:Pages>で囲み、タイトルを表示させるのは<\$mt:PageTitle\$>…ここまで使ったMTタグで作れそうですね！

```

71 <div class="top-pagesArea">
72   <div class="mt-container">
73     <ul class="top-pagesArea__thumbnailList">
74       <li>
75         <a href="#">
76           <div class="top-pages__item">
77             <h2 class="item__title">ページ1</h2>

```

```

    }

```

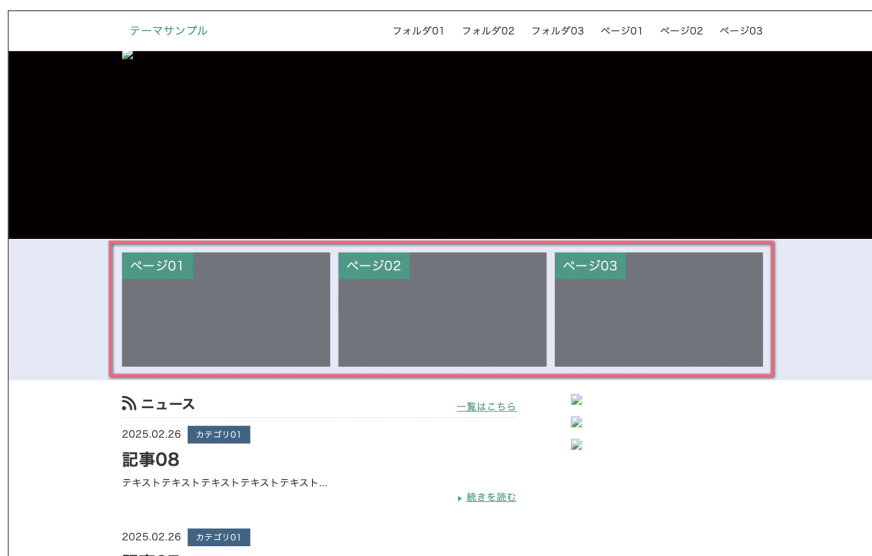
```

93       <h2 class="item__title">ページ3</h2>
94       
95     </div>
96   </a>
97 </li>
98 </ul>
99 </div>
100 </div>

```



```
<ul class="top-pagesArea__thumbnailList">
  <mt:Pages limit="3">
    <li>
      <a href="<$mt:PagePermalink$">">
        <div class="top-pages__item">
          <h2 class="item__title"><$mt:PageTitle$></h2>
        </div>
      </a>
    </li>
  </mt:Pages>
</ul>
```



ウェブページ3件と、ページタイトルが一覧表示できました！<mt:Entries>同様、<mt:Pages>もモディファイア<limit="N">で、**3件のみ表示させています**。

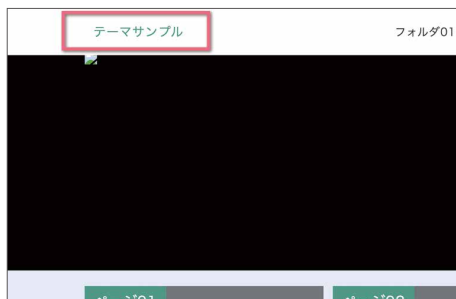
ウェブページ一覧のサムネイル画像は、**カスタムフィールド**を設定することで、ウェブページ編集画面から設定できるようになります。カスタムフィールドの使い方は、後ほど説明します。

ファイルマネージャを使おう

ファイルマネージャにあるファイルを表示させる

ここまでで、メインページのMTタグを使う箇所は完了しました。最後に、**ファイルマネージャにアップロードした画像を表示させましょう**。メインページでは、ロゴ画像とバナーを表示するエリアにあたります。

まずは通常どおりを使って画像を表示させます。画像のアドレスは絶対パスで指定したいので、`http://サイトのアドレス/images/logo.png`となります。サイトのアドレスの部分は、MTタグで指定できますね。使ってみましょう！



ロゴ画像を表示するエリア

```

```

先ほど設定した<h1 class="mt-h1">～</h1>内を書き換えます。<\$mt:BlogRelativeURL\$>のタグの出力には末尾に"/"が含まれているので重複に注意しましょう。

```
21 <div class="mt-container">
22 <a href="<$mt:BlogURL$>">
23   <h1 class="mt-h1"><$mt:BlogName$></h1>
24 </a>
25 <ul class="navi-list">
```



```
<h1 class="mt-h1">
  " />
</h1>
```

同様に、メイン画像エリアや、バナーエリアも作っていきます。



メイン画像エリア

```

67
68 <section class="mainImage">
69   <div class="mt-container">
70     
71   </div>
72 </section>
73 </main>

```



```

<section class="mainImage">
  <div class="mt-container">
    
  </div>
</section>

```



バナーエリア

```

122
123 <div class="top-bannerArea">
124   <ul>
125     <li>
126       <a href="#"></a>
127     </li>
128     <li>
129       <a href="#"></a>
130     </li>
131     <li>
132       <a href="#"></a>
133     </li>
134   </ul>
135 </div>
136 </div>

```



```

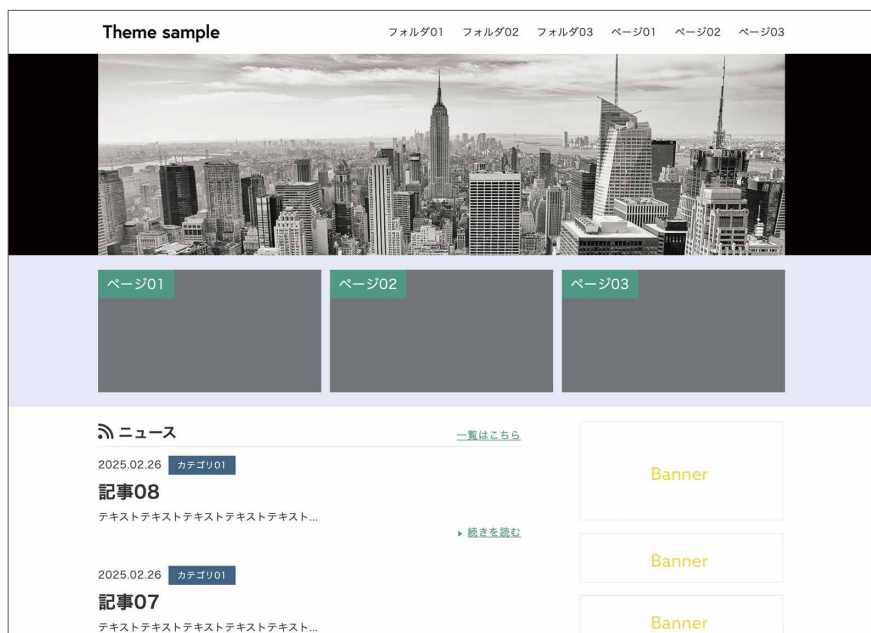
<div class="top-bannerArea">
  <ul>
    <li>
      <a href="リンク先アドレス">
        
      </a>
    </li>
    <li>

```

```

<a href="リンク先アドレス">
  
</a>
</li>
<li>
  <a href="リンク先アドレス">
    
  </a>
</li>
</ul>
</div>

```



画像が表示されました！ファイルマネージャから画像を使用する場合は、このように設定します。oggイメージやfaviconも、同様にして表示させましょう。

これでMTタグを使ったメインページが完成しました。この調子で、他ページもMT化を進めていきます。テーマの作り方、つかめてきましたか？

モジュールの活用

モジュールを作って効率的なサイトに!

引き続き、ページに必要なものをMT化していきます。
メインページには**フッター**があるのでそれを追加します。
今回はシンプルに、ブログの名前を入れたものを作ります。
ここまでで覚えたMTタグが使えますね。



```

143 </div>
144 </main>
145 <footer class="footerArea">
146   <div class="mt-container">
147     <p class="footerArea_logo">Theme sample</p>
148   </div>
149 </footer>
150 </body>
151 </html>

```



```

<footer class="footerArea">
  <div class="mt-container">
    <p class="footerArea_logo">
      <$mt:BlogName$>
    </p>
  </div>
</footer>

```

ストテキストテキストテキスト...

[▶ 続きを読む](#)

テーマサンプル

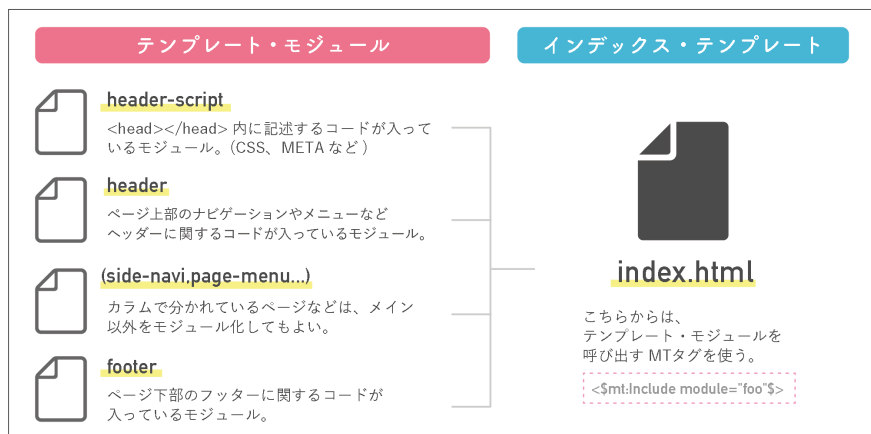
表示されました。

❖ テンプレート・モジュールを使う

さて、今回作ったフッターもですが、先に作ったヘッダー、ナビゲーションなど…どこのページにも共通する部分が出てきました。これをほかのテンプレートにも何度も書くのは面倒ですし、修正が必要な際には全てを書き換えなければなりません。

こんなときは、**テンプレート・モジュール**の出番です。知識編でも少しご紹介しましたね。

複数のページの共通事項はモジュール・テンプレートに記述し、**インデックス・テンプレート**や、**アーカイブ・テンプレート**から呼び出して使います。



呼び出すためのMTタグを1行書けばフッターを呼び出せるので、インデックス・テンプレート側はすっきりしますね。さらに、フッターを修正する場合にもテンプレート・モジュールの中を変えれば、全ページのフッターが書き換わるのでとっても楽です♪**使いまわせるパーツは、モジュール化しておきましょう。**

❖ テンプレート・モジュールを作ってみよう！

では、実際にヘッダーをモジュール化してみましょう！
デザイン>テンプレートに**テンプレート・モジュール**の項目があり、ここから作成します。

今回はヘッダーのモジュールなので、名前をheaderとつけました。モジュールはたくさん作ることができるので、**わかりやすい名前をつけてあげてくださいね。**

削除

テンプレート・モジュール

+ 新規

☐ 名前

削除

システムテンプレート

他テンプレート同様に、編集画面から記述ができます。そのモジュールが担う部分を書いて、保存しましょう。今回はヘッダー内の内容をモジュール化するので、該当の部分をインデックス・テンプレートからコピーして持ってきます。



❖ テンプレート・モジュールを呼び出そう!

続いて、インデックス・テンプレート側を書き換えます。モジュールを呼び出す際に使うMTタグはこちらです。

```
<$mt:Include module="モジュールの名前"$>
```

"モジュールの名前"部分には、作ったモジュールの名前を入れます。先ほどヘッダーの記述を入れたモジュール、headerを作ったので、ここは<\$mt:Include module="header"\$>ですね。インデックス・テンプレート側に書いて、呼び出します。

<\$mt:Include\$>

モジュールを読み込むときに使うタグです。

```
18 </head>
19 <body>
20 <$mt:Include module="header"$>
21
22 <section class="mainImage">
23   <div class="mt-container">
```



先ほどと変わらずに表示されました! 同様にして、フッターのモジュールも作っておきましょう。

- ウェブページ、ニュースページ編

ウェブページテンプレートを作る!

◆準備

メインページと同様に、ウェブページもMT化を進めていきましょう！ウェブページのテンプレートは、デザイン>テンプレート>アーカイブ・テンプレート>ウェブページから開くことができます。例によって、確認用にウェブページをいくつか作っておいてくださいな。

必要なヘッダー、フッターのモジュールはindex.html
のインデックス・テンプレートからコピーしておきま
しょう。

また、CSSやファイルのリンクなど、index.htmlと共通して必要なものも、前回と同じようにパスの指定を<code><mt:BlogRelativeURL\$></code>~から始まるように変えておきましょう。

❖ ページタイトル、本文を表示させる

まずはウェブページに最低限必要なタイトルと本文を表示させましょう。

<\$mt:PageTitle\$>

ウェブページのタイトルを表示します。

<\$mt:PageBody\$>

ウェブページの本文を表示します。

<\$mt:PageMore\$>

ウェブページの続きに入力した内容を表示します。

```
<section class="UL-mainImage">
<div class="UL-mainImage">
<h2 class="UL-mainImage_title">
<1 class="fa fa-file-text-o"></1>ページ
</h2>
</div>
</section>
<main class="pages-wrap">
<div class="mt-container">
<h3 class="pages_title">
タイトルタイトルタイトルタイトルタイトル
</h3>
<div class="pages-textArea">
<p>
<strong>テキストテキストテキスト</strong>
<br>テキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
</p>
<p>
テキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
</p>

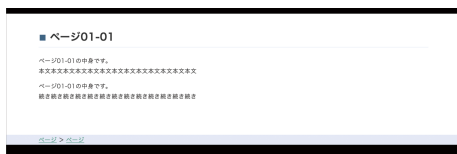
</div>
</div>
</main>
```

```

<section class="UL-mainImage">
  <div class="mt-container">
    <h2 class="UL-mainImage__title">
      <i class="fa fa-file-text-o"></i><$mt:FolderLabel$>
    </h2>
  </div>
</section>
<main class="pages-wrap">
  <div class="mt-container">
    <h3 class="pages__title"><$mt:PageTitle$></h3>
    <div class="pages-textArea">
      <$mt:PageBody$>
      <$mt:PageMore$>
    </div>
  </div>
</main>

```

ナビからウェブページに飛んで、確認してみましょう。
すると…？ウェブページのタイトルと本文が表示されました。



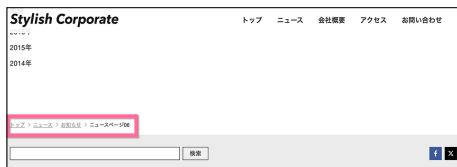
❖ パンくずリストを表示させる

「メインページを作る・その2」でナビゲーションを作成したときに説明したように、ウェブページはフォルダに入れることができます。さらにフォルダの中にフォルダを入れ、その中にウェブページを作することもできます。細かくフォルダ分けをすると管理はしやすいですが、サイトの構造は複雑になってしまいます。



ここに表示されているのがパンくずリスト

そんなときは、**フォルダから現在位置を表示するパンくずリスト**を作りましょう！パンくずリストは、現在見ているページを格納しているフォルダを**上から順番に並べて表示**しています。「**上位のフォルダを表示する**」MTタグと、前回使った「**フォルダ名を表示する**」MTタグを組み合わせると正しい順で表示できます。



「Stylish Corporate」でも、パンくずリストが設置されています

<mt:ParentFolders>

最上位のフォルダから現在のフォルダまでを繰り返して表示するブロックタグです。

```

36     </div>
37 </main>
38 <aside class="bread-list">
39     <div class="mt-container">
40         <ul>
41             <li><a href="#">ページ</a></li>
42             <li><a href="#">ページ</a></li>
43         </ul>
44     </div>
45 </aside>
46 <%=Include module="footer"%>
47 </body>
48 </html>

```

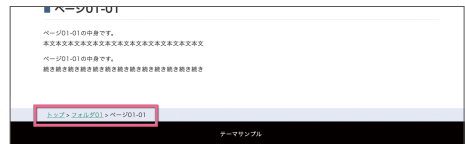


```

<ul>
<li><a href="<%=BlogRelativeURL%>">トップ</a></li>
<%=ParentFolders>
<li>
<a href="<%=PagePermalink%>"><%=FolderLabel encode_html="1"%></a>
</li>
</%=ParentFolders>
<li><span><%=PageTitle encode_html="1"%></span></li>
</ul>

```

パンくずリストが表示されました!とってもシンプルですがウェブページのテンプレートが完成しました。



ニュースページを作る・その1 - 記事ページ編

❖ ニュースページを作る!

記事ページとアーカイブ・テンプレートを使って、**ニュースページ**を作りましょう。必要なのは記事そのものになる記事と、ニュースページを一覧させるのに使う、記事ページ一覧の**アーカイブ・テンプレート**。それとnews.htmlになる**インデックス・テンプレート**です。どちらもいままでの要領で進めていけば大丈夫!さっそく作っていきましょう。

インデックス・テンプレート	
+ 新規	
☐ 名前	出力ファイル
☐ 最新記事のフィード	atom.xml
☐ ニュース TOP 	news.html
☐ メインページ	index.html
☐ スタイルシート	styles.css

アーカイブ・テンプレート	
+ 新規	
☐ 名前	
☐ 記事	
☐ ウェブページ	
☐ カテゴリ別ニュース記事リスト	
☐ 月別ニュース記事リスト	

❖ 記事ページを作る

まずは、ニュース記事そのものに使うインデックス・テンプレートの方を作っていきます。デザイン>テンプレート>アーカイブ・テンプレートにある、記事を編集し、必要なモジュールを置き換えておきましょう。

今回、ニュースページは2カラムのデザインにしています。メインカラムには記事内容を表示させます。必要なのは、記事の日付、カテゴリ、記事タイトル、本文、本文追記分です。サブカラムには、各記事リストをアーカイブにしたものを置きましょう。

メインカラム、記事を表示する部分を作っていきます。必要になるMTタグは下記の通りです。以前に記事の一覧を作った際に使ったMTタグも出てきますね。思い出して、作ってみましょう。



<\$mt:EntryDate\$>

ブログ記事の公開日を表示します。モディファイアの指定で色々オプションがつけられます。

<mt:EntryCategories\$>

ブログ記事に指定したカテゴリの一覧のためのブロックタグです。

<\$mt:CategoryLabel\$>

カテゴリ名を表示します。

<\$mt:CategoryArchiveLink\$>

カテゴリアーカイブの URL を表示します。

<\$mt:EntryBody\$>

ブログ記事の本文を表示します。

<\$mt:EntryMore\$>

ブログ記事の続きに入力した内容を表示します。

これを、記事ページになるように組み立てていきます。ウェブページを作った時と同じように、考えてみましょう。

```

29 <div class="news-columnWrap">
30   <div class="news-newsArea">
31     <div class="news__titleArea">
32       <p class="news__day">2017.02.11</p>
33       <p class="news__category">カテゴリ</p>
34       <h3 class="news__title">
35         タイトルタイトルタイトルタイトルタイトル
36       </h3>
37     </div>
38     <div class="news-textArea">
39       <p>
40         テキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
41       </p>
42       <p>
43         テキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
44       </p>
45     </div>
46     <!-- SNSエリア -->
47     <!-- /SNSエリア -->
48   </div>
49 </div class="news-sideArea">

```



```

<div class="news-newsArea">
  <div class="news__titleArea">
    <p class="news__day">
      <$mt:EntryDate format="%Y.%m.%d"%$>
    </p>
    <mt:EntryCategories type="primary">
      <a href="<$mt:CategoryArchiveLink$>" class="category">
        <p class="news__category">
          <$mt:CategoryLabel$>
        </p>
      </a>
    </mt:EntryCategories>
    <h3 class="news__title">
      <$mt:EntryTitle$>
    </h3>
  </div>
</div>
<div class="news-textArea">
  <$mt:EntryBody$>
  <$mt:EntryMore$>
</div>
<!-- SNSエリア -->
<!-- /SNSエリア -->
</div>

```

記事が表示されました！これでニュース本体のアーカイブ・テンプレートが完成しました。続いて、ニュースのトップページ news.html のインデックス・テンプレートを編集します。



ニュースページを作る・その2 - news.html編

❖ news.htmlにニュース一覧ページを作る

続いて、news.htmlのインデックス・テンプレートを作ります。メインページでリンク先を指定した、**一覧はこちら**のリンクから、ページを確認することができます。

まだページが表示されていませんが、記事ページと同じく、メインとサブの2コラム構成です。こちらは、メインコラムに**最新の記事一覧**が並ぶようにします。同じものをメインページで作りましたね！

```
<mt:Entries limit="3">
  <dl class="newsArea__dateList">
    <dt>
      <h4 class="newsArea__title"><$mt:EntryTitle$></h4>
    </dt>
    <dd class="newsArea__text">
      <div class="newsArea__body">
        <$mt:EntryExcerpt words="20"$>
          <p class="newsArea__link">
            <a href="<$mt:EntryPermalink$>">続きを読む</a>
          </p>
        </div>
      </dd>
    </dl>
  </mt:Entries>
```

一覧を作るのは2回目なので、こちらは少しレベルアップしてみましょう！記事を投稿した**日付**と、選択された**カテゴリ**をMTタグを使って取得し、表示させます。**記事のサムネイル画像**も表示させましょう。（後ほど、投稿記事に対応できるようにカスタマイズする

ので、ここではimages/noimage.pngを表示させてみます。）

コードは、次のように書き換えました。

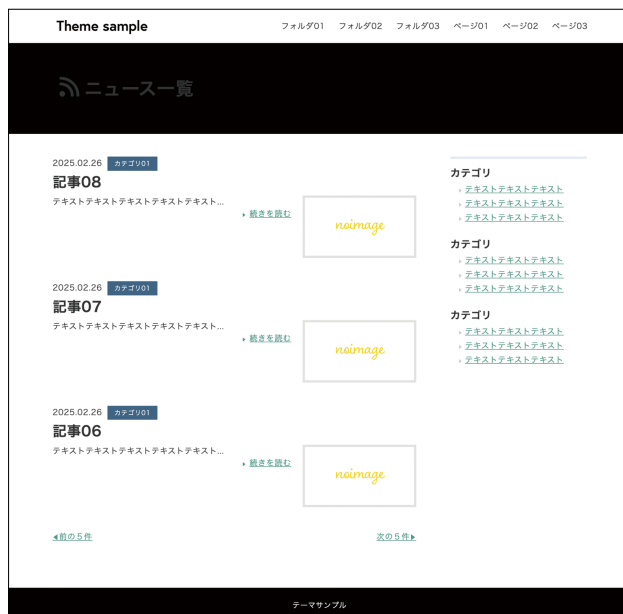
```
31 <div class="news-columnWrap">
32   <div class="news-newsArea">
33     <dl class="newsArea__dateList">
34       <dt>
35         <p class="newsArea__day">2017.02.11</p>
36         <a href="news/"><p class="newsArea__category">カテゴリ</p></a>
37         <h4 class="newsArea__title">
          }
38       </h4>
39     </dt>
40     <dd class="newsArea__text">
41       <div class="newsArea__img">
42         
43       </div>
44     </dd>
45   </dl>
46   <ul class="newsArea__nextArea">
```



```

<mt:Entries limit="3">
  <dl class="newsArea__dateList">
    <dt>
      <p class="newsArea__day">
        <$mt:EntryDate format="%Y.%m.%d"$>
      </p>
      <p class="newsArea__category">
        <$mt:CategoryLabel$>
      </p>
      <h4 class="newsArea__title">
        <$mt:EntryTitle$>
      </h4>
    </dt>
    <dd class="newsArea__text">
      <div class="newsArea__body">
        <$mt:EntryExcerpt words="20"$>
      </div>
      <div class="newsArea__img">
        
      </div>
    </dd>
  </dl>
</mt:Entries>

```



❖ ページネーションの機能をつける

さらに、**ページネーション**の機能を持たせてみましょう。このページには**最新3件の記事を表示させ、それ以降は次のページに送る**ようにします。それ以降の記事は、「次の3件」のようなリンクをつけ、次のページに表示させるようにしましょう。

<mt:Entries paginate_by="N">

ページ送り等を行いたい場合、モディファイアの値に件数を指定することで、指定した数分の件数が表示されます。limit="N"との併用はできないので書き換えましょう。

```
31 <div class="news-columnWrap">
32   <div class="news-newsArea">
33     <mt:Entries limit="3">
34       <dl class="newsArea_datelist">
35         <dt>
```



<mt:Entries paginate_by="3">

こちらのタグでも3件の記事が表示されます。ですが、これだけではナビゲーションがないので、次の3件のページにいきませんね。次のタグも追加して、その中に**「前に戻る」「次に進む」**を作っていきます。

<mt:EntriesFooter>

ブロックタグの中で、最後のブログ記事のときだけ実行するブロックタグです。

<mt:PaginationHasPrevious>

<mt:EntriesHeader>と<mt:EntriesFooter>の中で利用でき、前のページがある場合に内容を出します。

<\$mt:PaginationPreviousURL\$>

前のページの URL を出力します。存在しない場合、何も出力しません。

<mt:PaginationHasNext>

<mt:EntriesHeader>と<mt:EntriesFooter>の中で利用でき、次のページがある場合に内容を出します。

<\$mt:PaginationNextURL\$>

次のページの URL を出力します。存在しない場合、何も出力しません。

<mt:Entries paginate_by="3"> ~ </mt:Entries>の間の最後の部分に<mt:EntriesFooter>を用意しましょう。ここは繰り返しの処理にならないので、1 回だけ表示されます。その中に、内容の表示/非表示を判断する、<mt:PaginationHasNext>、<mt:PaginationHasPrevious>を作っていきます。その中にURLを表示させる<\$mt:PaginationPreviousURL\$>、<\$mt:PaginationNextURL\$>を、アンカータグの中に入れ、完成です！

```

<mt:Entries paginate_by="3">
  <dl class="newsArea__dateList">
    <dt>
      <p class="newsArea__day">
        <$mt:EntryDate format="%Y.%m.%d"$>
      </p>
      <p class="newsArea__category">
        <$mt:CategoryLabel$>
      </p>
      <h4 class="newsArea__title">
        <$mt:EntryTitle$>
      </h4>
    </dt>
    <dd class="newsArea__text">
      <div class="newsArea__body">
        <$mt:EntryExcerpt words="20"$>
        <p class="newsArea__link">
          <a href="<$mt:EntryPermalink$>">続きを読む</a>
        </p>
      </div>
      <div class="newsArea__img">
        
      </div>
    </dd>
  </dl>
</mt:EntriesFooter>
<ul class="newsArea__nextArea">
  <mt:PaginationHasPrevious>
    <li class="newsArea__nextArea--prev">
      <a href="<$mt:PaginationPreviousURL$>"><i>◀</i>前の3件</a>
    </li>
  </mt:PaginationHasPrevious>
  <mt:PaginationHasNext>
    <li class="newsArea__nextArea--next">
      <a href="<$mt:PaginationNextURL$>">次の3件<i>▶</i></a>
    </li>
  </mt:PaginationHasNext>
</ul>
</mt:EntriesFooter>
</mt:Entries>

```

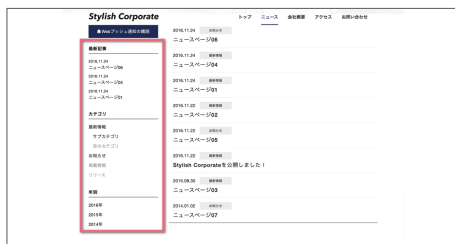
記事をたくさん作って確認してみましょう…ページネーションができました！

2つの機能を追加して、メインページの一覧よりパワーアップしましたね。ニュース記事一覧は、アーカイブ・テンプレートで何度か使うことがありそうなので、ここもニュース記事一覧のパーツとして、モジュール化しておきましょう。

ニュースページを作る・その3 - アーカイブ編

ここからは、記事リストのアーカイブを作っていきます。記事はカテゴリ別や日付別、月別などで**アーカイブリストを作ることができます**。これをニュースページのサブカラムに設置してみましょう！

右の画像はテーマ「Stylish Corporate」のニュース一覧ページです。左カラムにアーカイブリストがありますね。



❖ カテゴリ、月別のアーカイブリストを作る

作りたいアーカイブの**アーカイブ・テンプレート**を作っておきます。最初のテンプレートの書き換え時に用意したのは**カテゴリ別記事リスト**と、**月別記事リスト**でした。この2つを編集していきましょう。

ニュース一覧ページを作るときに使うMTタグは同じです。記事は**アーカイブタイプに合わせて、カテゴリ別になる**ように集めてきてくれます。news.htmlの

インデックス・テンプレートから、必要なMTタグをコピーしてきましょう。一覧の部分はモジュールしておく、アーカイブ・テンプレートの種類が増えた時にも使えます。それに加えて、記事リストの内容がわかるようにしたいので、`<$mt:ArchiveTitle$>`を使って**テンプレート名を表示させ、一覧の上にページの見出しをつけておきます**。

`<mt:ArchiveTitle>`

ブログ記事一覧アーカイブのタイトルを表示します。表示されるタイトルは、アーカイブの種類に依存します。

```
30 <div class="mt-container">
31   <div class="news-columnWrap">
32     <div class="news-newsArea">
33       <h3 class="newsArea__categoryName">カテゴリ</h3>
34       <dl class="newsArea__dateList">
35         <dt>
```

```
<h3 class="newsArea__categoryName"><$mt:ArchiveTitle$></h3>
```

それぞれにテンプレート名が表示されて、ちょうど良い見出しになりました。これでカテゴリ別記事リスト、月別記事リストのページができました。



❖ アーカイブを一覧で表示する

ついに、アーカイブリストをページに表示します。インデックス・テンプレート、news.htmlのサブカラムを編集していきます。カテゴリ別記事リストをアーカイブ表示させるのに使えるMTタグには、次の種類があります。

<mt:ArchiveList>

アーカイブの一覧を表示するためのブロックタグです。モディファイアarchive_type="archive_type_foo"の中を変えることで他アーカイブを選択できます。今回はカテゴリ別を表示させたいので、archive_type="Category"を使います。

<\$mt:ArchiveLink\$>

ブログ記事一覧アーカイブの絶対 URL を表示します。ブログ記事一覧アーカイブへのリンクを作るときに使います。mt:ArchiveListブロックの中か、アーカイブに関連したテンプレートで使用できます。

<\$mt:CategoryArchiveLink\$>

カテゴリアーカイブのURLを表示します。

<mt:ArchiveListHeader>

<mt:ArchiveList>タグの中で、最初にだけ実行する条件タグです。

<mt:ArchiveListFooter>

<mt:Archives>タグの中で、最後にだけ実行する条件タグです

<\$mt:ArchiveCount\$>

ブログ記事一覧アーカイブのブログ記事数を表示します。

<\$mt:CategoryCount\$>

カテゴリに含まれるブログ記事の件数を表示します。

まず、カテゴリ別記事リストを呼び出し、その中からカテゴリ名と、そのカテゴリに含まれる記事の件数をそれぞれ取得し、表示させていきます。

```
34 <div class="news-sidebar">
35   <h3 class="sidebar_title">カテゴリ</h3>
36   <ul class="sidebar_list">
37     <li><a href="#">テキストテキストテキスト</a></li>
38     <li><a href="#">テキストテキストテキスト</a></li>
39     <li><a href="#">テキストテキストテキスト</a></li>
40   </ul>
41
42   <h3 class="sidebar_title">カテゴリ</h3>
43   <ul class="sidebar_list">
44     <li><a href="#">テキストテキストテキスト</a></li>
45     <li><a href="#">テキストテキストテキスト</a></li>
46     <li><a href="#">テキストテキストテキスト</a></li>
47   </ul>
48
49   <h3 class="sidebar_title">カテゴリ</h3>
50   <ul class="sidebar_list">
51     <li><a href="#">テキストテキストテキスト</a></li>
52     <li><a href="#">テキストテキストテキスト</a></li>
53     <li><a href="#">テキストテキストテキスト</a></li>
54   </ul>
55 </div>
```



```

<mt:ArchiveList archive_type="Category">
  <mt:ArchiveListHeader>
    <h3 class="sideArea__title">カテゴリ</h3>
    <ul class="sideArea__list">
      </mt:ArchiveListHeader>
      <li>
        <a href="<$mt:CategoryArchiveLink$">">
          <mt:ArchiveTitle> (<$mt:CategoryCount$>)
        </a>
      </li>
      <mt:ArchiveListFooter>
    </ul>
  </mt:ArchiveListFooter>
</mt:ArchiveList>

```

サブカラムのナビゲーションに**カテゴリ別記事リスト**と、それぞれの**カテゴリに含まれる記事の数のアーカイブ**が入りました。続いて、月別記事リストも表示させていただきます。今作成したカテゴリ別記事リストのコードの下に続けて、次のコードを記述します。

```

<mt:ArchiveList archive_type="Monthly">
  <mt:ArchiveListHeader>
    <h3 class="sideArea__title">月別</h3>
    <ul class="sideArea__list">
      </mt:ArchiveListHeader>
      <li>
        <a href="<$mt:ArchiveLink$">">
          <mt:ArchiveTitle> (<$mt:ArchiveCount$>)
        </a>
      </li>
      <mt:ArchiveListFooter>
    </ul>
  </mt:ArchiveListFooter>
</mt:ArchiveList>

```

2つのアーカイブが表示できました！ほかの記事別リストがある場合もarchive_type="archive_type_foo"の部分を変更して表示させていきましょう。



❖ ニュースページの仕上げ

ここで作ったアーカイブをモジュールにしたものを、**カテゴリ別ニュース記事リスト**、**月別ニュース記事リスト**、**記事のテンプレート**、**news.htmlのテンプレート**の共通する箇所書き換えておきましょう。

❖ ニュースページへのリンクをヘッダーナビに表示する

最後に、ヘッダーのナビに、ニュース一覧ページを表示させるリンクを追加しましょう。news.htmlのページへのリンクURLは、<\$mt:BlogRelativeURL\$>news.htmlと記述することができます。このURLへのリンクを、ナビの一番左に表示できるよう、ヘッダーのモジュール・テンプレートに、次のコードを記述しましょう。

```
<li>
  <a href="<$mt:BlogRelativeURL$>news.html">ニュース</a>
</li>
```



```
<ul class="navi-list">
  <li>
    <a href=" <$mt:BlogRelativeURL$>news.html">ニュース</a>
  </li>
  <mt:TopLevelFolders>
  <li>
    <a href="#">
    <$mt:FolderLabel$>
    </a>
    <ul class="navi-list__sub">
      <mt:Pages>
      <li>
        <a href="<$mt:PagePermalink$>">
          <$mt:PageTitle$>
        </a>
      </li>
    </ul>
  </li>
  ...
```

ナビの一番左に、ニュースページへのリンクが表示されます。これで**ニュースページは完成**です！



カスタムフィールドを使おう

カスタムフィールドを設定する

❖ カスタムフィールドで何ができるの？

「もっと管理画面を使いやすくカスタマイズしたい!」
というときには、**カスタムフィールドを使ってみましょう**。カスタムフィールドは、記事、ウェブページ、
ブログ、アイテム、カテゴリ、フォルダに、「テキスト」
「URL」「ドロップボタン」など**様々なタイプのフィールドを追加できる機能**です。

例えば、作成した記事にサムネイル画像を設定できる

カスタムフィールドを作成し、設定された画像の記事
のタイトル、内容と一緒に表示することができます。

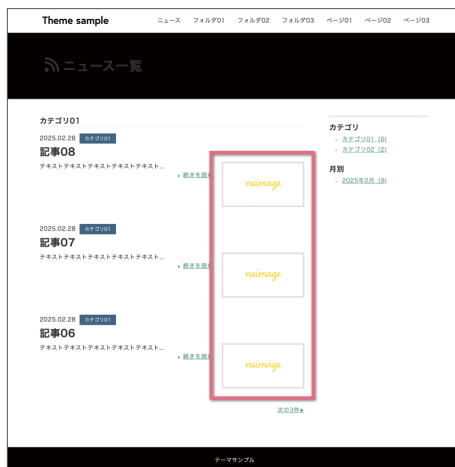
このように画像を表示する使い方以外にも、「**チェック
ボックスにチェックがある時は表示する**」、「**フィールド
に入力されている値が同じであれば表示する**」など、
幅広くカスタマイズができます。

画像を表示させるカスタムフィールドを作る

では、実際に作ってみましょう! まずはすぐに使えそ
うな、**画像表示のカスタムフィールド**を作ります。記
事の一覧にサムネイル画像を表示させてみましょう!

カスタムフィールドから記事の関連画像を表示させる
手順は次のとおりです。では、早速作っていきましょ
う!

1. カスタムフィールド > 新規から新しくカスタムフィールドを作る
2. 記事テンプレートにカスタムフィールド用のMTタグを追加する
3. 記事編集画面からカスタムフィールドを確認し、画像を投稿する



❖ 1.新しくカスタムフィールドを作る

カスタムフィールド>新規から新しいカスタムフィールドを作ります。



共有: ウェブサイトを作成している場合に表示される項目です。ウェブサイト内にあるブログすべてでこのカスタムフィールドを使う場合にチェックします。ブログを作成している場合には表示されません。

システムオブジェクト: このフィールドをどこに使うか?を設定します。記事／ウェブページ／ブログ／アイテム／カテゴリ／フォルダから選びましょう。今回は「記事」ですね。

名前: このフィールドの名前です。フィールドのタイトルになり、編集画面の入力部分に表示されます。

説明: 編集画面の入力部分に表示される説明です。使う人がわかりやすいように入れてあげましょう。

タイプ: テキスト、チェックボックス、画像、URLなど…フィールドのタイプを選択します。今回は「画像」を選択します。

必須: このフィールドへの入力を必須にする場合はチェックを入れます。

既定値: デフォルトの値を設定します。文字や正/誤の場合に入れておくとも良いでしょう。画像など既定値が存在しないタイプの場合、この項目は表示されません。

識別子: テンプレートからこのフィールドを呼び出す際に使う文字列です。半角英数で、わかりやすく付けましょう!

記事のサムネイル画像をアップロードする、という動きなのでこのように設定しました。



❖ 2.記事テンプレートにカスタムフィールド用のMTタグを追加する

このフィールドを表示できるようにテンプレート側を変更していきます。45～46ページで作ったニュース記事一覧のテンプレート・モジュールを開いて、画像部分を書き換えましょう。先ほど付けた識別子を使います。

<mt:CustomFieldAsset>

アイテム（ファイル、画像、ビデオ、オーディオ）のカスタムフィールドでの値を表示するブロックタグです。identifierモディファイアでカスタムフィールドの識別子を指定して使用します。ブロックの内部では<mt:AssetURL>などのアイテム用のタグが利用できます。

identifier="識別子"

指定した識別子を持つカスタムフィールドの値を利用できるようにします。識別子は先ほどのnews_imageにします。

<\$mt:AssetThumbnailURL\$>

アイテムの種類が画像 (image) のファイル、またはアイテムに登録されたPDFファイルのサムネイルのURLを表示します。PDFファイルが複数ページ構成になっている場合に作成されるサムネイルは、PDFファイルの1ページ目が使用されます。

```
50         <a href="<$mt:EntryPermalink$>">続きを読む</a>
51     </p>
52 </div>
53     <div class="newsArea__img">
54         
55     </div>
56 </dd>
57 </dl>
58 <mt:EntriesFooter>
```

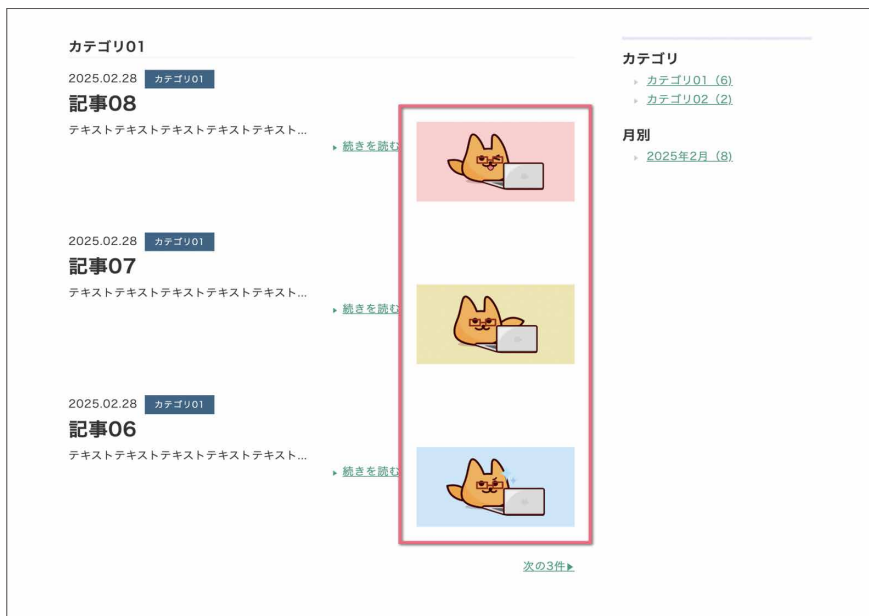


```
<div class="newsArea__img">
  <$mt:AssetThumbnailU
RL$></mt:CustomFieldAsset>">
</div>
```

❖ 3.記事編集画面からカスタムフィールドを確認し、画像を投稿する

続いて、記事の編集画面を見てみましょう。先ほど作ったカスタムフィールドが表示されていますね！

画像を選択から、画像をアップロードしてみましょう。すると…？



画像が表示されました！カスタムフィールドで画像を表示させる方法はいこれで完了です。
ですが、実際に利用する際にはもう少しカスタマイズが必要になります。

```

    <$mt:AssetThumbnailURL$>
  </mt:CustomFieldAsset>
">
```

この記述だと、カスタムフィールドに画像が入っていれば画像が表示されますが、**入っていない場合は空になっ**てしまい、ちょっと不自然です。カスタムフィールドに画像があればそれを表示させ、画像がなかったら記事に入っている画像を表示する という動きが理想的ですね。

そのためには `<mt:If>` という MT タグを使用し、**条件によった振り分け**をします。`<mt:If>` の使い方、カスタムフィールドとの組み合わせの活用は、Web 版「トフでもできる！？テーマ開発」内、カスタムフィールドを設定する！ (<https://developer.movabletype.net/themeguide/practice/p-01/customfield-setting.html>) で詳しく解説していますので、そちらをご参照ください。

MTタグを使いこなそう

configモジュールとは？

❖ configモジュールって？

一部のベーステーマには、デザイン>テンプレートモジュール・テンプレートの中に、configモジュールがあります。この中では、`mt:SetVarBlock`を使って、**変数が設定**されています。変数にテンプレート内で何度も出てくるカラーコードやURLをまとめておき、テンプレートから**変数名で呼び出し、管理している**のです。

例えばconfig内で`pointColor`の変数に強調色を設定をしておき、CSSから強調文字にしたい部分を変数名で呼び出して強調色を表示できるようにしておきます。こうすると、色を変更したい場合にconfig内の変数を書き換えるだけで、同じ変数が使われている箇所の色を一括で変えることができます。作った変数はCSS、HTML両方のテンプレート内で使用できます。

```

1 <mt:Ignore>
2 各種設定はこのモジュールで行います。
3 </mt:Ignore>
4
5 <mt:Ignore>色指定</mt:Ignore>
6 <mt:Ignore>背景色</mt:Ignore>
7 <mt:SetVarBlock name="bgColor">#ffffff</mt:SetVarBlock>
8
9 <mt:Ignore>背景画像 (url)の中にアドレスを入れます。必要のない場合はnoneを入力してください。 </mt:Ignore>
10 <mt:SetVarBlock name="bgimg">url<mt:if:isset:blogURL encode="html">1</mt:if:isset:blogURL</mt:SetVarBlock>
11
12 <mt:Ignore>文字色</mt:Ignore>
13 <mt:SetVarBlock name="textColor">#000000</mt:SetVarBlock>
14
15 <mt:Ignore>白抜き文字色</mt:Ignore>
16 <mt:SetVarBlock name="whiteTextColor">#fff</mt:SetVarBlock>
17
18 <mt:Ignore>本文リンク色</mt:Ignore>
19 <mt:SetVarBlock name="linkColor">#FF7F00</mt:SetVarBlock>
20
21 <mt:Ignore>見出し色</mt:Ignore>
22 <mt:SetVarBlock name="headerColor">#000000</mt:SetVarBlock>
23
24 <mt:Ignore>メイン色 (ヘッダー・フッターなど) </mt:Ignore>
25 <mt:SetVarBlock name="mainColor">#FF0000</mt:SetVarBlock>
26
27 <mt:Ignore>メインより少し明るい色 (ストライプ柄にする場合に指定してください。しない場合はメインと同じ色を指定)
28 <mt:SetVarBlock name="mainColor-Light">#FF0000</mt:SetVarBlock>
29

```

例：テーマ「Character」のconfigモジュール

テーマ「Character」では文字、サイトのメインカラー、サブカラーなどの色が、configモジュールで指定されているのがわかりますね。色の他にも、記事の表示件数やURLなど、テーマ内で何度も出てくる数値を変数に設定することもできます。設定次第で**簡易的な編集もできる**ようになるので、テンプレートを編集した経験のないサイト運営者に渡す際にも活用できそうです。

❖ configモジュールを設定してみよう

上で使われている変数を設定するMTタグはこちらです。

<mt:Ignore>

指定した範囲について、処理を行わないブロックタグです。コメントなどに使用します。

<mt:SetVarBlock>

ブロックタグで囲まれた内容を、変数として値を設定します。

name="name_foo"

値を代入する変数の名前を指定します。

続いて、変数呼び出すHTML・CSS側に記述するタグです。使用するHTML、CSSの最初に<\$mt:Include module="モジュール名"\$>で、**configモジュールを読み込ませる**のがポイントです。

<\$mt:Include module="モジュール名"\$>

テンプレート・モジュールや、外部ファイルを読み込むときに使うタグです。モディファイアmoduleやfileなどにより、読み込む対象を指定します。

<\$mt:Var name="var_foo"\$>

mt:SetVar, mt:SetVarBlock, mt:SetVarTemplateタグで定義した変数から値を呼び出します。

configモジュールを使ってみよう

❖ 1.configモジュールを作成し、設定する

デザイン>テンプレート テンプレート・モジュールの「**+新規**」からモジュールを新規作成し、右のコードを入力します。名前は、わかりやすくconfigとつけます。

```
<mt:Ignore>ポイントカラー</mt:Ignore>
<mt:SetVarBlock
name="pointColor">#FF0000</
mt:SetVarBlock>
```

❖ 2.読み込むテンプレート側に、configモジュールの読み込みと呼び出し設定をする

```
<$mt:Include module="config"$>
<p>テスト<span style="color:<$mt:Var
name="pointColor"$>;">ここを色変え</span>
テストテストテスト</p>
```

テスト **ここを色変え** テストテストテスト

configモジュールを使って、文字色の変更ができました！CSSにも同じように記述ができるので、よく使うカラーコードはまとめておくと、あとあとの管理に役に立つでしょう。

MT化する -404 ページ編

404ページを作ろう！

ここまでで、大体のページと作り方を習得しました。ですが…デザイン>テンプレート内に、もう1つ**システムテンプレート**がありますね。システムテンプレートには存在しないページをリクエストした場合に表示される、「**404ページ**」が用意されています。これを設定しないと、なんとも味気ないエラーページが出てしまいます。

Theme Sample

[トップ](#)
[ニュース](#)
[ページ](#)
[ページ](#)
[ページ](#)

404 notfound
 お探しのページは見つかりませんでした。
 Theme sample

404ページも、**サイトに合わせたデザインのものに作り変えましょう**！今まで使ったタグやモジュールを使えば、カンタンなはずですよ♪ウェブページから、必要なヘッダーやフッターのモジュールをコピーしてきましょう。

```
<body>
<$mt:Include module="header"$>
<section class="UL-mainImage">
  <div class="mt-container">
    <h2 class="UL-mainImage__title">
      <i class="fa fa-frown-o"></i>
      404 notfound
    </h2>
  </div>
</section>
<main class="pages-wrap">
  <div class="mt-container">
    <div class="notfound-textArea">
      お探しのページは見つかりませんでした。
    </div>
  </div>
</main>
<$mt:Include module="footer"$>
</body>
```

このようになりました！ページ内でちゃんとリンクが貼られていればあまり見ることのないページですが、作成しているサイト・ブログのURLに、存在しないページ名を入れると確認することができます。



テーマを書き出す・読み込む

作成したテーマは書き出すことができ、書き出したテーマを読み込ませることで別のサイト・ブログに同じテーマを適用することができます。大枠ができたなら細かくカスタマイズする前に一度保存し、それをベースに新しいテーマを作っていくこともできます。ここでついに、**作ったテーマを書き出してみましょう！**

テーマを書き出す

デザイン>テーマ>現在のテーマから**書き出す**をクリックします。任意の出力ファイル名を入力し、書き出したいファイルとカスタムフィールドにチェックを入れます。

書き出すをクリックするとzipファイルのダウンロードがはじまります。保存して完了しましょう。



テーマを読み込む

デザイン>テーマ>現在のテーマから**読み込む**をクリックします。テーマが入ったzipファイルをアップロードします。無事にアップロードが完了し、テーマが変更されていれば成功です。



注意

読み込み前のテーマのカスタムフィールドが残ってしまうため、テーマの読み込みをする前に**以前のカスタムフィールドをすべて削除**しておきましょう。

MTタグでサイトの設定をしよう

ページ内部から作りましたが、言語設定やCanonical設定などの**基本情報もMTタグを使って指定することができます**。例えば、`<$mt:BlogDescription$>`はMT内のブログ>設定>全般の説明フィールドに入力された文言を表示するタグです。これをdescriptionに使えば、ブログの説明が表示されます。このように、基本情報の設定にもMTタグを使ってみましょう！主なタグはこちらです。

言語設定…`<$mt:BlogLanguage$>`

ブログに表示する日付の言語を表示します。

文字コード設定…`<$mt:PublishCharset$>`

システムで管理するブログの文字エンコードを表示します。UTF-8固定となります。

Description設定…`<$mt:BlogDescription$>`

ブログの説明を表示します。

Canonical設定…`<$mt:CanonicalLINK$>`

インデックス・テンプレート、およびアーカイブ・テンプレート内で現在のページのリンク付きURLを返します。

`<$mt:Link$>`

インデックス・テンプレートもしくは特定のブログ記事へのリンクを出力します。メディアファイア `template` や `entry_id` により、リンク先を指定します。

他にも、ページ作成で使った `<$mt:BlogURL$>`（ブログのURLを表示）や、`<$mt:BlogName$>`（ブログ名を表示）が使えるですね。meta情報をMTタグを使って書き換えてみましょう。

```

1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>サンプルテーマ</title>
8   <!-- Optional theme -->
9   <link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.css">
10  <link href="$mt:BlogRelativeURL$styles.css" rel="stylesheet">
11  <link
12    rel="stylesheet"
13    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css"
14    integrity="sha512-Ev84M4kqVGRN5gIGL/F/aIDqQb7xQ2vcd1wxjThSH8CSR7PBEakCrS1Ck+w+/U6swU2Im1vVX0S5V9ABhg=="
15    crossorigin="anonymous"
16    referrerpolicy="no-referrer">
17  >
18 </head>
19 <body>

```

テンプレート

名前

メインペ

出力ファイ

index.htm

メモ

最終更新:



```
<html lang="<$mt:BlogLanguage$">
<head>
  <meta charset="<$mt:PublishCharset$">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="<$mt:BlogDescription$">
  <meta name="keywords" content="oooooooo,ooo,ooo,ooo">
  <title><$mt:BlogName encode_html="1"$></title>
  <link rel="shortcut icon" href="<$mt:BlogURL encode_html="1"$>images/favicon.
ico">
  <link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/
meyer-reset/2.0/reset.css">
  <link href="<$mt:BlogRelativeURL$>styles.css" rel="stylesheet">
  <link
    rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.
css"
    integrity="sha512-Evv84Mr4kqVGRNSgIGL/F/aIDqQb7xQ2vcrdIwx fjThSH8CSR7PBEakCr51
Ck+w+/U6swU2Im1vVX0SVk9ABhg=="
    crossorigin="anonymous"
    referrerpolicy="no-referrer"
  >
  <link rel="start" href="<$mt:BlogURL encode_html="1"$>">
  <link rel="alternate" type="application/atom+xml" title="Recent Entries"
href="<$mt:Link template="feed_recent"$>">
    <$mt:CanonicalLink$>
</head>
```

他のページにも必要になりそうな部分は、適宜モジュール化しておくのがオススメです。metaキーワードなどはconfigにまとめて記述し、変数で呼び出しをしても良いでしょう。ogp設定する際にもURLをMTタグで指定するなど、対応していきましょう。

```
<meta name="keywords" content="<$mt:Var name="meta_keywords"$>">
```

Web版「トフでもできる！？テーマ開発講座」では、SNSボタンの設置方法や、configでのアカウント制御、表示／非表示の切り替えについて紹介しています。SNS対応にもぜひチャレンジしてみましょう！



さらにMTタグの学習をしよう！



ここまでで、基本となるテンプレートの作り方は習得できました！

さらに、用途に合わせたテーマの作り方、管理のしやすいテンプレートの仕組み、カスタムフィールドの有効活用などできることはまだまだたくさんあります！

MTタグが理解できれば**活用ブログ**、**マニュアル**を見ながら、カスタマイズができるはずです。また、デフォルトで用意されているテーマのテンプレートがどのような仕組みになっているのかも参考になるでしょう。

作成に必要な情報を得たいときに

・ 制作者向けポータル

MovableType.net のテンプレートをカスタマイズをする方向けに、マニュアルや関連記事をまとめています。

<https://developer.movabletype.net/>



「このタグ、なんだっけ？」というときに

・ タグリスト

MovableType.net で利用可能な独自タグ、MTタグを記述例とともに一覧で掲載しています。

<https://movabletype.net/tags/>

「本の内容をWebで確認したい」というときに

・ トフでもできる!? テーマ開発

本では省略した内容や、完成版のサイトを確認できます。ソースのコピーもできて便利です。

<https://developer.movabletype.net/themeguide/>

「実例や、追加機能を知りたい」というときに

・ MovableType.net 活用ブログ

機能の追加情報や、利用方法を画面キャプチャ付きで解説！運営・構築に役立つTipsも紹介しています。

<https://movabletype.net/blog/>



「この動作に使うMTタグは？」というときに

・ MTタグ逆引き辞典

実現したいことからMTタグを検索できる逆引き辞典です。

<https://movabletype.net/dict/>

技術的な問題が発生したときに

・ テクニカルサポート

MovableType.net をご利用中の方は、テクニカルサポートよりご質問いただけます。

<https://movabletype.net/support/account/support.html>

MovableType.net

トフでもできる!?

テーマ開発BOOK

2018年3月 1日 初版第1刷発行
2025年4月24日 初版第3刷発行

©2025 Six Apart Ltd.

企画・編集・作成：シックス・アパート株式会社

デザイン：原田 沙織・齊藤 有香・佐藤 こすえ（シックス・アパート株式会社）

デザイン協力：川角友太

印刷：株式会社グラフィック

発行元：シックス・アパート株式会社 <https://www.sixapart.jp/>

〒101-0051

東京都千代田区神田神保町3-17-15ヨシダFGビル5F

03-6261-4640（代表）

※製品に関するお問い合わせは、お電話ではお受けしておりません。

「トフでもできる!?テーマ開発BOOK」は

ウェブサイト「[トフでもできる!? テーマ開発](https://developer.movabletype.net/themesample202503/)」の内容を、書籍用に編集したものです。本書で省略されている内容や、**実践編**に登場するサンプルテーマ、学習用ファイルはウェブサイトにて確認できます。



Web版 「トフでもできる!?テーマ開発」

<https://developer.movabletype.net/themeguide/>



サンプルテーマ

<https://developer.movabletype.io/themesample202503/>



学習用ファイル(zipダウンロード)

<https://developer.movabletype.io/themeguide/assets/sampletheme-2025.zip>

