

echo for Movable Type 利用マニュアル

改定: 2015年12月24日



CMSのウェビングスタジオ
<http://webbingstudio.com/>

目次

はじめに

このテーマの特徴	4
動作環境	6
ライセンスについて	7

導入の流れ

パッケージを確認する	8
テーマをインストールする	10
コンフィグファイルを修正する	12
ブログの全般設定をする	13
再構築をする	15
MtAppjQueryの設定をする	16
ページ本文を流し込む	18
記事を投稿する、ウィジェットを確認する	20
公開前のチェック	21

コンテンツ作成に関する機能

記事とページの違いについて	22
記事・ページを作成する	23
トップページの編集	24
コールトゥアクションの編集	25
バナーの編集	26
記事一覧の体裁を変更する	27

外観に関する機能

メイン画像を利用する	29
ページのレイアウトについて	30
ウィジェットをカスタマイズする	31
記事ループモジュールについて	33

CSS・JSに関する機能

CSSの構造について	35
SaSSの構造について	36
JavaScriptの構造について	38
gulpの利用について	39

その他の機能

ページ分割にPageButeを利用する	40
カテゴリー・フォルダの階層情報を利用する	42
依存ファイルのディレクトリを変更する	43

補足

コーディングルール	45
TIPS	47

はじめに

このたびは、プレミアムテーマ「echo」をお買い上げいただき、ありがとうございます。

本書は「echo」Movable Type+Bootstrap版の導入手順、利用方法、TIPS等を解説しています。echoが、お客様のワークフローの向上や、社内・チーム内でのよりよい意思統一につながれば幸いです。

このテーマの特徴

CSSフレームワークの採用

CSSフレームワーク「Bootstrap 4 Alpha」、アイコンフォント「Font Awesome」をコアに採用し、Bootstrapとechoのそれぞれにおいて、構造と装飾のCSSを分離しています。

現場に不要な機能の排除

「echo」は、現場の制作ではほとんど利用されない「コメント」「トラックバック」に対応していません。一部のテンプレートはダイナミック・パブリッシングでの表示にも対応していますが、こちらでも導入ケースが少なく、かつスタティックと同様の表示にならない不具合が見られるため、最低限の対応となっています。

利用頻度の高い機能を厳選

「特定の記事をピックアップ」「記事一覧にアイキャッチ画像を表示」「SEO関連の情報を操作」など、特に要望が多い機能（カスタムフィールド・条件・変数処理）を基本搭載しています。

詳細な資料を同梱

「echo」には本書のほか、「変数表」「テンプレート一覧」「カスタムフィールド一覧」をスプレッドシート・PDFにて同梱しています。制作の参考になるほか、Excel・Googleドライブでの改変ができます。

▶ [次ページへ続く](#)

論理的なマークアップ

「echo」では、誰でも読める、理解できるコーディングを心がけました。
HTML・CSS・Sassなどの表示部分だけでなく、テンプレートの独自タグ・コメント・変数処理に至るまで考慮しています。

制作をスムーズにする変数

「echo」のテンプレートでは、多くの変数を使用しています。
ウェブサイト・META・OGP情報をはじめ、実務制作でひんばんに求められる「現在・親・先祖」の三段階のカテゴリ・フォルダ情報を、計算の必要なく取得できます。

プログラミングと表示の分離

「echo」では、テンプレートの冒頭で、ページ内で使用する情報を変数にまとめて定義したうえで、表示の処理を行っています。表示部分に複雑な条件分岐や変数定義がほとんどないため、BEM・Foundation・Material Designなど、bootstrap以外のコーディングにも変更しやすくなっています。

デバッグモード

定義済みの変数を実際のページでチェックする「デバッグモード」を採用しています。制作補助にご利用ください。

動作環境

「echo」は、Movable Type6.1以上で動作を確認しています。

バージョン6の新変数を使用していないため、バージョン5でも利用できる可能性がありますが、動作の完全保証はいたしません。また、MTOS・MovableType.netでは必須機能が不足しているため、利用できません。

ページ分割について

ブログトップページ・カテゴリーアーカイブのページ分割に、プラグイン「PageBute」を利用することが可能です。

詳しくは詳細は本書内「[その他の機能 > ページ分割にPageButeを利用する](#)」を参照ください。

<http://www.skyarc.co.jp/engineerblog/entry/2642.html>

管理画面の整形・改変について

プラグイン「MTAppjQuery」を利用すると、管理画面の項目が正しい順番に整列し、入力がしやすくなります。

詳しくは詳細は本書内「[導入手順 > MTAppjQueryの設定をする](#)」を参照ください。

<http://bit-part.net/products/mtappjquery/>

PageButeは株式会社スカイアーク様、MTAppjQueryはbit part 合同会社様の制作物となります。当方での永続的な保証はできませんのでご了承ください。

ライセンスについて

WordPress版以外の「echo」は、MITライセンスにて提供しています。
ご購入後は商用・非商用問わずご利用いただけます。

https://osdn.jp/projects/opensource/wiki/licenses%2FMIT_license

不特定多数への再販を行う場合は必ず、下記のテーマ名、およびリンク元をテーマ内に明示してください。業務向けの有償テーマという性格上、節度を持ってご利用いただければ幸いです。

echo for Movable Type

<http://echo-mt.cms-skill.net/>

WordPress版に関する特記事項

WordPress版のみ「100%GPLライセンス」となります。

https://osdn.jp/projects/opensource/wiki/licenses%252FGNU_General_Public_License_version_3.0

MITライセンスとは異なり、GPLライセンス下では、改変後においても同一のライセンスを継承する義務があります。

このため、改変後のテーマにも、オープンソースの宣言が求められることをご了承ください。

依存ファイルについて

依存ファイル（画像・CSS・JS・SVG）についても、echo本体に限らず自由にご利用いただけます。

画像素材は、「Stocksnap」収録の素材と、ウェビングスタジオが撮影した写真となります。SVGアイコン（使用していないものも含む）は、ウェビングスタジオが各サービスのロゴ規定に基づき自作したものです。

いずれもCC0ライセンスとなります。

<https://stocksnap.io/>

<http://creativecommons.org/publicdomain/zero/1.0/deed.ja>

なお、テーマ本体については、著作権の完全放棄はしていません。

導入の流れ

パッケージを確認する

購入いただいた圧縮ファイルには、以下のフォルダ・ファイルが含まれています。

パッケージ全体

dist/themes/echo_bootstrap/

テーマ本体です。gulpを利用するところに生成されます。

_js/

圧縮前のJavaScriptファイル一式です。

_scss/

コンパイル前のSassファイル一式です。

config.rb

Compassの設定ファイルです。gulp・Sassの基本機能についての解説は割愛します。

gulpfile.js

gulpの設定ファイルです。

package.json

gulpを動作させるために必要なパッケージが記載されています。

manual_echo_mt_bootstrap(日付).pdf

本書です。

vars_echo_mt_bootstrap(日付)

echoの変数表です。PDF・XML形式があります。

templates_echo_mt_bootstrap(日付)

echoのテンプレート一覧です。PDF・XML形式があります。

fields_echo_mt_bootstrap(日付)

echoのカスタムフィールド一覧です。PDF・XML形式があります。

▶ [次ページへ続く](#)

テーマ本体

_pages/

初期状態のテーマに含まれている、最低限必要なウェブページの内容です。

_upload/

初期状態のテーマを表示するために必要なメイン画像・OGPです。

blog_static/

画像・CSS等の、テーマの必須ファイルです。テーマをインポートすると、ブログのルートフォルダにファイルが複製されます。事前に差し替えておくことで、工数を節減できます。

theme.yaml

テーマの構造を定義しているファイルです。編集することでテーマを案件名、作者を自分自身にしたり、カスタムフィールドやカテゴリー構造を変更することができます。本書では解説を割愛します。

thumbではじまる画像ファイル

管理画面で表示される、テーマのサムネイルです。制作サイトのスクリーンショットに差し替えてください。

テーマをインストールする

テーマ本体をアップロードする

テーマ内の「dist/themes/」内にある「echo_bootstrap」ディレクトリを、Movable Typeの以下の階層にアップロードしてください。

Movable Type本体を設置したディレクトリ/themes/

SVG等の読み込みを許可する

Movable Typeでは、テーマファイルからSVG、Woff等のファイル形式を読み込むことができないため、環境設定を変更します。

Movable Typeの下記ファイルを、テキストエディタ等で開いてください。

Movable Type本体を設置したディレクトリ/mt-config.cgi

mt-config.cgiの最終行に、以下のコードを追記してください。

```
ThemeStaticFileExtensions jpg jpeg gif png js css ico flv swf html otf ttf eot svg woff woff2 svg
```

この設定は、テーマのインポート・書き出しの際のみ必要となります。環境設定を変更できないなど、サーバーに制限がある場合は、テーマのインポートを行ったあとで、テーマの「blog_static」ディレクトリ内のファイルを、公開ディレクトリにアップロードし直してください。

重複ファイルを退避する

公開予定のサイト内に、以下のディレクトリが既に存在する場合、テーマのインポート時ファイルを上書きしてしまう可能性があります。

css imgaes js fonts share MTAppjQuery

該当ファイルを前もって退避するか、本書内「[その他の機能 > 依存ファイルのディレクトリを変更する](#)」の手順に従って、echoの依存ファイルを移動してください。

▶ [次ページへ続く](#)

テーマを有効化する

テーマ一式をアップロードすると、管理画面の「**デザイン > テーマ**」から「echo Bootstrap」テーマを選択できるようになります。

ウェブサイトを新規作成するか、既存のウェブサイトに反映して適用してください。カスタムフィールド・ページ・フォルダ・カテゴリー・依存ファイルが自動的に読み込まれます。



ウェブサイトが存在するブログのコンテンツを表示するブログポータルを作成します。

▶ **プロフェッショナル ウェブサイト 1.11**
作者: [Six Apart, Ltd.](#)
バナー画像、水平型のナビゲーションなど、ホームページ用途に適したデザインです。あらかじめ用意されたペ

▶ **echo Bootstrap 1.0.0**
作者: [WebbingStudio](#)
「echo」は、プロの制作者向けに、効率よくウェブサイトを構築できるよう設計したフレームワークテーマです。ベースに、構造と装飾を分離した、再利用性の高い設計を行っています。

▶ **Eiger 1.01**
作者: [Six Apart, Ltd.](#)
Eiger は企業サイトのようなブログを作成できる、レスポンシブ Web デザインを採用したテーマです。Media Q Type の機能を使ってロゴやトップページの画像、ナビゲーションの内容を簡単にカスタマイズできます。

▶ **Pico 1.3**
作者: [Six Apart, Ltd.](#)
Picoはマイクロブログを作成するのに適した、テキストや写真といったコンテンツを引き立てるシンプルなデザインは、メインコンテンツの下に配置されます。

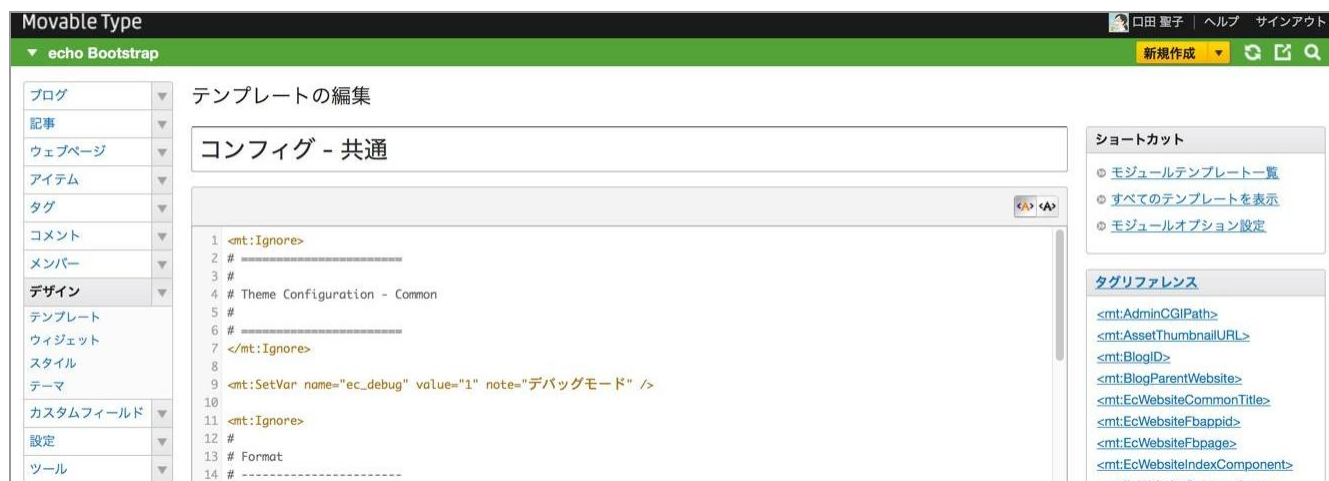
▶ **Rainier 1.21**
作者: [Six Apart, Ltd.](#)

既存のウェブサイトに反映した場合、変数・カスタムフィールドに不具合が出る可能性があります（変数名に接頭辞を付けることで、変数名の衝突自体は回避しています）。新規作成での導入をお勧めします。

コンフィグファイルを修正する

管理画面の「**デザイン** > **テンプレート**」からテンプレート一覧へ移動し、テンプレートモジュールの「**コンフィグ-共通**」の編集画面を開いてください。

制作するサイトの状況に応じ、以下の箇所を任意に修正してください。



`<mt:SetVar name="ec_blog_contents_label" value="ブログ" . . .`

時系列コンテンツの名称が「新着情報」「お知らせ」など、「ブログ」ではないときは、こちらのvalueモディファイアを変更してください。

`<mt:SetVar name="ec_breadcrumb_home_label" value="ホーム" . . .`

パンくずリストの起点の名称が「Home」「(Movable Type外のサイト名)」など、「ホーム」ではないときは、こちらのvalueモディファイアを変更してください。

`<mt:SetVar name="ec_entry_date" note="記事の年月日フォーマット" value="%Y-%m-%d" . . .`

`<mt:SetVar name="ec_entry_time" note="記事の時分秒フォーマット" value="%H:%m:%S" . . .`

時系列コンテンツに表示される、公開日の表記フォーマットです。制作予定のサイトのデザインに合わせて変更してください。Movable Typeの日付フォーマットについては、下記を参照ください。

<http://www.movabletype.jp/documentation/appendices/date-formats.html>

ブログの全般設定をする

ウェブサイト（ブログに設置した場合はブログ）の管理画面の「[設定](#) > [全般](#)」へ移動してください。

Movable Type本来の設定の他に、echoのウェブサイト設定が追加されています。任意の設定値を入力してください。

Movable Type

ウェブページ

▼ echo

ブログ

▼

記事

▼

ウェブページ

▼

アイテム

▼

タグ

▼

コメント

▼

メンバー

▼

デザイン

▼

カスタムフィールド

▼

設定

▼

全般

コミュニティ

投稿

コミュニケーション

登録/認証

Webサービス

ツール

▼

全般設定

ウェブサイト設定

名前 *

echo

説明

「echo」は、プロの制作者向けに、効率よくウェブサイトを構築できるよう設計したフレームワークテーマです。「Bootstrap 4」とアイコンフォント「Font Awesome」をベースに、構造と装飾を分離した、再利用性の高い設計を行っています。

FacebookページのURL

https://www.facebook.com/cmsskill/

FacebookページのURLがあれば入力してください。

メイン画像5枚目



画像を選択 画像を削除する

ホームのメイン画像をアップロードしてください。

Twitter ID

cmsskill

TwitterのアカウントIDがあれば入力してください。

ホームのMETAの概要文

MTEAppjQueryを有効にするまでは、入力欄が順不同となっています。気になる場合は、以降の「再構築をする」「MTEAppjQueryの設定をする」を先に行っても構いません。

▶ [次ページへ続く](#)

重要な設定

■ アップロード先

バージョン6.2から、画像のアップロード先の固定が可能となりました。必ず最初に定義し、一旦保存してください。

ウェビングスタジオの推薦は、「%s/upload/%y/%m/」です。

■ メイン画像1～5

テーマのトップページデザインをそのまま利用する場合は、必須となります。

画像素材を別途用意するか、テーマパッケージ内「_upload」フォルダに含まれている、5点のダミー画像をアップロードしてください。

■ og:image

SNSでシェアしたときに、記事・ページ専用の画像がないときに参照する画像です。任意のOGP画像をアップロードしてください。

画像を設定しなかった場合は、og:imageが定義されない状態となります。

■ 1ページの件数

検索結果や、PageButeプラグインでのページ分割の「1ページの記事件数」は、**管理画面の設定内容を反映しません。**

変更する場合はテーマ内「**コンフィグ-共通**」の以下のコードを探し、変数のvalueを編集してください。

```
<mt:SetVar name="ec_search_count" value="6" note="一覧の1ページあたりの件数: リスト関連では2倍になる" />
```

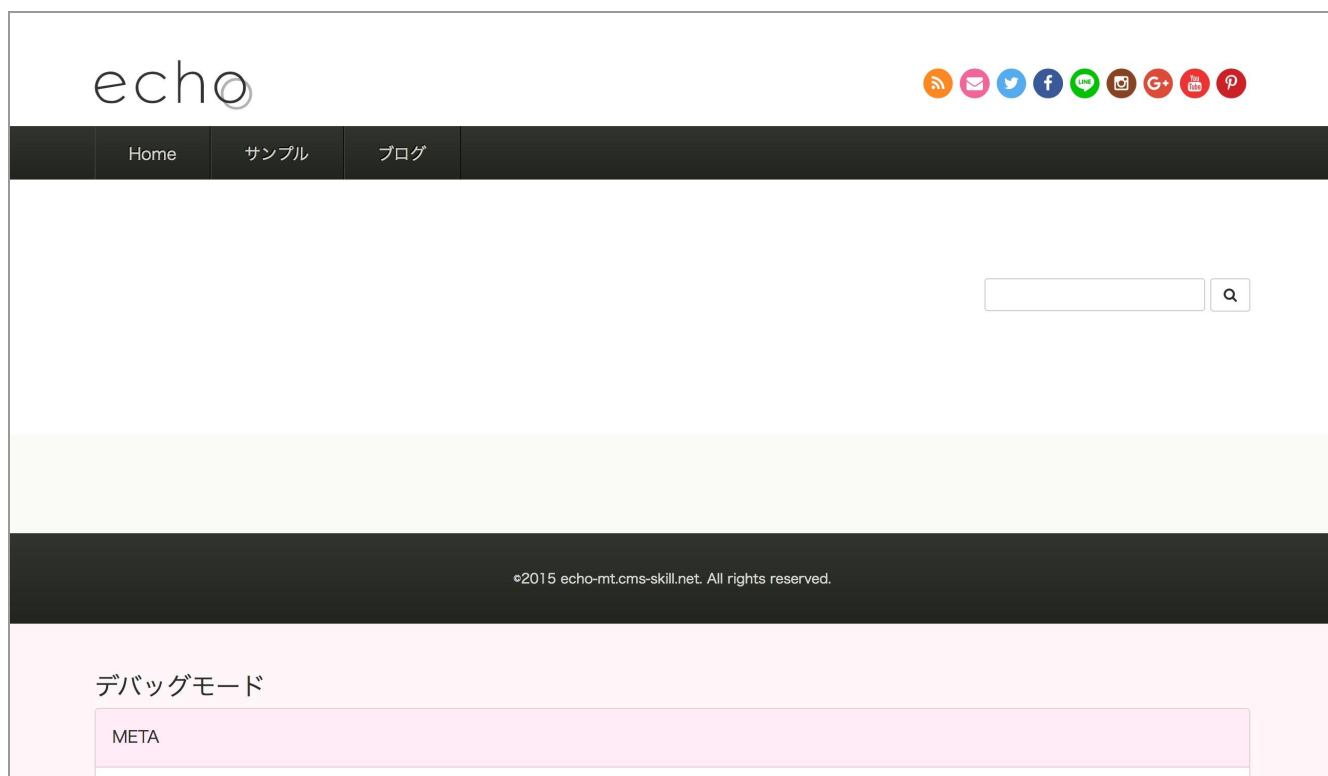
ページ分割の詳細は、本書内「**その他の機能 > ページ分割にPageButeを利用する**」を参照ください。

再構築をする

ここまで設定すると、ブログを再構築することができます。

通常通りの手順で、ウェブサイトもしくはブログの再構築を行ってください。

既存コンテンツがある場合は、重複ファイルがないか慎重に確認してください。



トップページを表示し、ページが崩れていないかを確認してください。この時点では本文には何も表示されません。

ページを下にスクロールしていくと「デバッグモード」が表示されます。

ウェブサイト・ブログ・カテゴリー・ページ・記事の設定内容がMETAタグ等に正しく反映されているか、チェックすることができます。

MTAppjQueryの設定をする


メイン画像1枚目



[画像を選択](#) [画像を削除する](#)

ホームのメイン画像をアップロードしてください。


メイン画像2枚目



[画像を選択](#) [画像を削除する](#)

ホームのメイン画像をアップロードしてください。


メイン画像3枚目



[画像を選択](#) [画像を削除する](#)

ホームのメイン画像をアップロードしてください。


メイン画像4枚目



[画像を選択](#) [画像を削除する](#)

ホームのメイン画像をアップロードしてください。

メイン画像5枚目



このテーマは、プラグイン「MTAppjQuery」対応のスク립トファイルを生成します。

MTAppjQueryを有効にすると、管理画面が以下の通り最適化されます。

- メインメニューの「コメント」が非表示になる
- 記事・ページの項目の並び替えを禁止する
- 表示オプションが非表示となる（項目を変更することができなくなります）
- ブログの「全般設定」の項目が正しい並び順となり、メイン画像が左右二段組となる
- ウェブサイト・ブログ・記事・ページカスタムフィールドの入力欄が大きくなり、ヒントが強調される

MTAppjQueryには上記以外にも、管理画面をカスタマイズする、さまざまなjQueryライブラリが含まれています。詳細は公式ドキュメントを参照ください。

<https://document.bit-part.net/mtappjquery/usage/>

▶ 次ページへ続く

user.js・user.cssを設定する

設置済みのMovable Typeに、「MTAppjQuery」をインストールしてください。

続いて、ウェブサイト（ブログに設置した場合はブログ）の管理画面の「[ツール > プラグイン](#)」へ移動し、「MTAppjQuery」の詳細設定を開いてください。

※プラグイン名をクリックして展開後、「詳細」タブを選択します

プラグイン

 **CSVDatalmExporter 2.03**

 **Facebook Commenters 2.3**

 **MTAppjQuery 1.8.1**

[詳細](#) [リソース](#) [設定](#)

全般設定

有効

user.jsの設定

有効

URL

http://example.com/MTAppjQuery/user.js

user.cssの設定

有効

URL

http://example.com/MTAppjQuery/user.css

jquery_ready.jsの設定

有効

[URLを変更する](#)

自由テキストエリア

管理画面テンプレートのそれぞれの位置に挿入するソースを設定します。管理画面用のMTタグが使えます。それぞれの値がどの位置に挿入されるかは、ソースコードの中の<!-- 変数名 (MTAppjQuery) -->等で確認できます

<head></head>タグ内

「user.jsの設定」「user.cssの設定」欄の「URLを変更する」文字列をクリックして、以下の通りURLを入力して、プラグイン設定の最下部にある「変更を保存」ボタンを押してください。

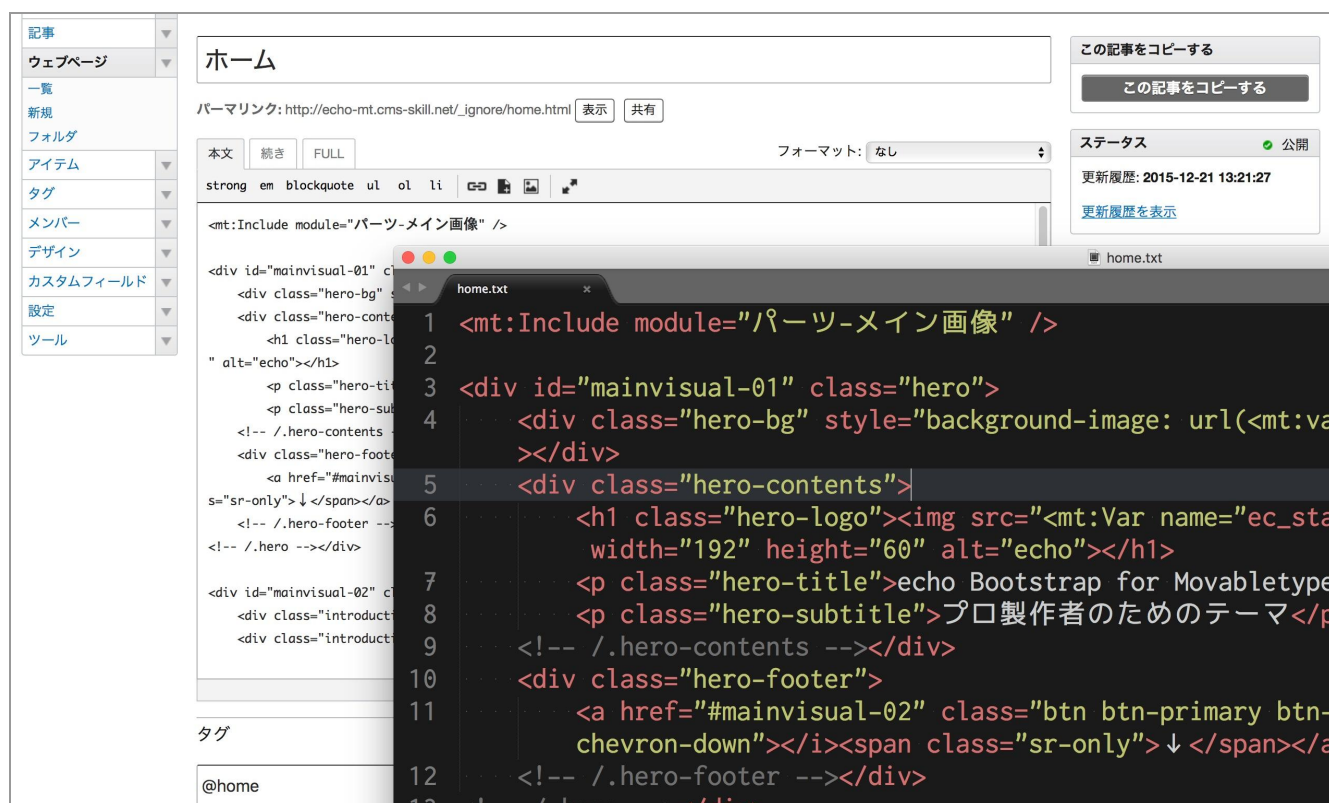
user.js

（公開サイトのトップページのURL） MTAppjQuery/user.js

user.css

（公開サイトのトップページのURL） MTAppjQuery/user.css

ページ本文を流し込む

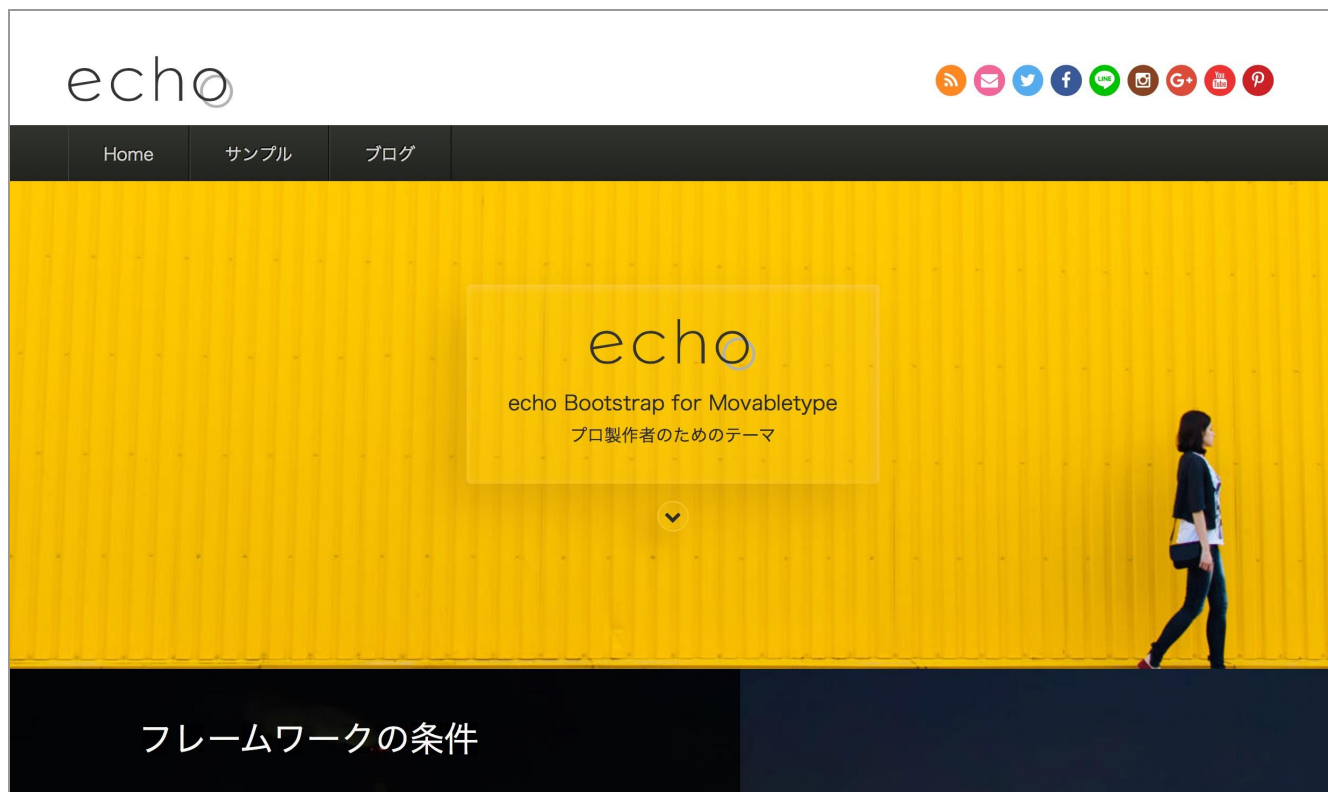


このテーマは「MTML（Movable Typeの独自タグ）を記述していると、インストール時にエンコードされる」という理由から、初期ページの本文をインストールしません。

ページの本文は、テーマ内にテキスト形式で含めています。初期状態を確認したい場合は、下記の通り本文をコピー・ペーストしてください。本文のフォーマットは「なし」としてください。

ページタイトル	ページ本文のファイル
ホーム	_pages/home.txt
コールトゥアクション	_pages/cta.txt
バナー	_pages/banner.txt
リッチテキストのサンプル	_pages/sample.txt
Bootstrap 4のサンプル	_pages/bootstrap.txt
追加コンポーネントのサンプル	_pages/echo.txt

▶ 次ページへ続く



トップページの本文を流し込むと、全般設定でアップロードしたメイン画像も表示されるようになります。

「リッチテキストのサンプル」「Bootstrap 4のサンプル」のページは、bootstrap等のCSSカスタマイズの際の表示確認にご利用ください。

ロリポップ！レンタルサーバー・ヘテムルなどの「WAF」機能を採用しているサーバーでは、「head要素」にコードを追加すると403エラーとなることがあります。
その場合はWAFを無効にするか、ページ本文に追加してください。

テスト記事を投稿する

Movable Type echo Bootstrap ウェビングスタジオ ヘルプ サインアウト

▼ echo Bootstrap 新規作成

記事の編集

ピックアップ・アイキャッチを指定した記事

パーマリンク: <http://echo-mt.cms-skill.net/blog/post/10/index.html> 表示 共有

本文 続き FULL フォーマット: リッチテキスト

この記事は「ピックアップ」にチェックを入れ、「アイキャッチ画像」を登録しています。

ピックアップされた記事は、公開日に関係なく、専用ウィジェットでサイト全体に呼び出すことができます。

アイキャッチ画像をアップロードすると、一覧ページにサムネイルが表示されます。

- 一覧ページでは、サムネイルの指定がない場合はアイキャッチ画像を表示し、アイキャッチ画像も指定されていない場合はテーマ内の「ノイメージ (images/noimg.jpg)」を表示します。
- ノイメージは4:3となっているため、アイキャッチ画像を他の比率で統一する場合は差し替えてください。
- SNSでシェアした時にも利用されるため、幅1200ピクセル以上の画像が望ましいです。
- アイキャッチ画像の本文への自動表示は行いません。必要の場合は「記事ループ-本文」テンプレートを修正してください。

この記事をコピーする

この記事をコピーする

ステータス 公開

更新履歴: 2015-12-21 17:17:47

更新履歴を表示

公開

ステータス 公開

公開日 2015-12-21 @ 16:35:18

公開終了日 指定されていません 編集

出力ファイル名 post 編集

変更メモ

更新 プレビュー 削除

最後に、テスト記事を投稿します。公開デモサイトの「ブログ」内に、テスト記事のサンプルがありますので参照ください。

<http://echo-mt.cms-skill.net/blog/>

記事を投稿すると、ウィジェットとして設定されていた、各種新着記事リストが表示されるようになります。それらが正しく表示されているか、並べ換えが正しくできるかをご確認ください。

初期状態では、すべてのテンプレートが「スタティック・パブリッシング」となっています。関連記事・ローカルナビを確認する場合は都度サイト全体を再構築するか、サーバーサイドインクルードを利用してください。

各テンプレートの、ダイナミック・パブリッシングの利用の可・不可については、別添のテンプレート一覧を参照ください。

公開前のチェック

SNSボタン

SNSボタンのアカウント、リンク先のテンプレート編集画面を開き、不要なボタンを表示していないか確認してください。

ヘッダ右側のSNSリンクは、テーマ内「**ヘッダ**」、記事本文のSNSボタンは、「**記事ループ-本文**」のテンプレートを参照ください。

アクセス解析・OGP

管理画面の「**設定 > 全般**」へ移動し、head要素内に表示するアクセス解析のコード、Twitterアカウント、FacebookのアプリケーションIDが間違っていないか確認してください。

OGP画像は設定しなくても構いません。設定しなかった場合はog:imageを出力しません。

ファビコン・apple-touch-icon

ファビコンとapple-touch-iconは、Movable Type側から変更できません。**初期状態ではechoのロゴとなっている**ため、FTPソフト等で必ずすべて差し替えてください。

デバッグモードを無効にする

管理画面の「**デザイン > テンプレート**」からテンプレート一覧へ移動し、テンプレートモジュールの「**コンフィグ-共通**」の編集画面を開いてください。

テンプレート前半に、デバッグモードを設定している以下のコードがあります。

valueを「0」にして再構築すると、ページ末尾にチェック用の変数リストが表示されなくなります。

```
<mt:SetVar name="ec_debug" value="1" note="デバッグモード" />
```

echoの初期設定は、以上となります。

以降のページでは、echoが持つ各機能を解説いたします。

コンテンツ作成に関する機能

記事とページの違いについて

「echo」は「記事／カテゴリー」と「ページ／フォルダ」に対応しています。これらの違いと推薦する利用法について解説します。

記事

ブログなどの、時系列コンテンツへの利用を前提としています。
ダイナミック・パブリッシングを利用でき、「タグ」に対応しています。また、一覧のリンクをクリックしたときに、個別ページではなく任意ページへ移動させることができます。

カテゴリー

ウィジェットで、階層ナビゲーションを自動生成することができます。また、一覧ページ専用の概要・アイキャッチ画像を登録することができます。

ページ

恒久的なコンテンツへの利用を前提としており、一覧ページを生成しません。
ページごとにレイアウトを大幅に変えることができ、ランディングページも作成可能です。
ダイナミック・パブリッシングを利用すると、親・先祖のフォルダ情報を取得できないため、ステイックでの利用をお勧めします。

フォルダ

フォルダはあくまで、ページのディレクトリを指定するためにのみ使用しています。
Movable Typeの「SubFoders」タグは、カテゴリー用の「SubCategories」タグと比較し、複雑な階層ナビゲーションを作成しにくいという仕様があります。このため、階層構造のナビゲーションを作成したい場合は、すべて記事で作成することをお勧めします。

記事・ページを作成する



記事・ページの作成については、Movable Typeの一般的な投稿手順と同様です。

リッチテキストエディタ用のCSSには対応していません。作成したサイトのデザインに合わせて別途ご用意ください。

(scssファイルをコピーして流用すると、作成がスムーズです)

ページの本文にはMTMLを使用することができ、ウィジェット・モジュールテンプレートを呼び出すことが可能です。この機能を制限したい場合は、テーマ内「**ページ**」のテンプレートを編集してください。

ローカルナビゲーションでのページの並び順は「公開日昇順」となります。

トップページの編集



タグに「@home」が指定されたページの本文は、ウェブサイトのホーム前半に表示されます。

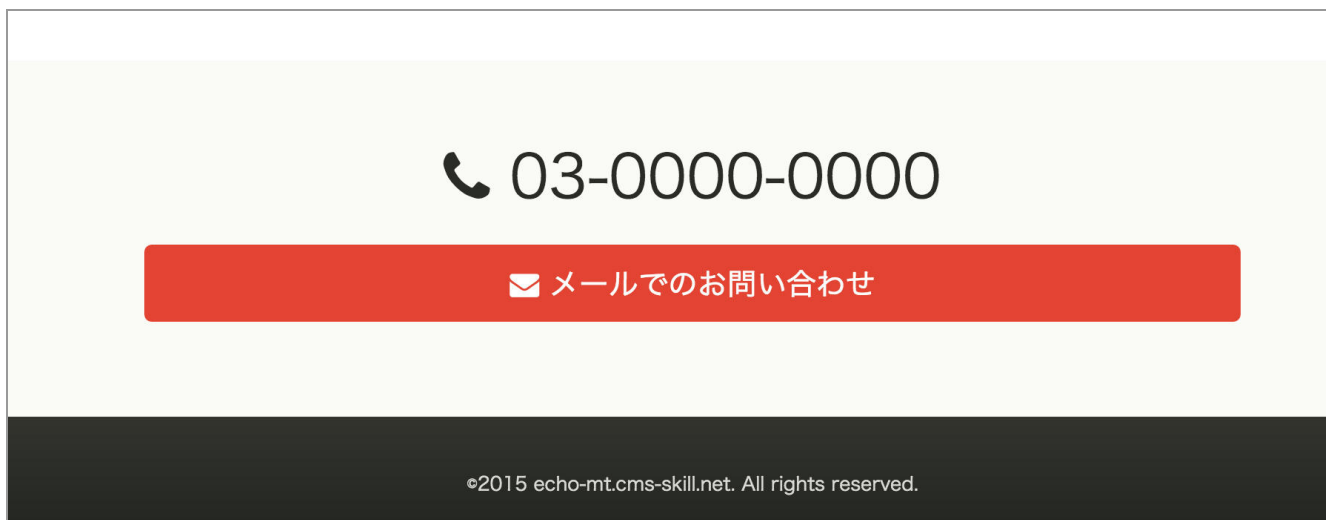
後半はホーム専用のウィジェットと、コールトゥアクションとなります。

本文にはMTMLを使用することができ、ウィジェット・モジュールテンプレートを呼び出すことも可能です。作成予定のページに併せて編集してください。

「@home」を指定したページが複数ある場合は、公開日順にすべて表示されます。

Movable Typeの仕様上、通常のページが別途生成されます。非公開のフォルダに収納するようにするか、別途プラグインなどを利用してファイル生成を制限してください。

コールトゥアクションの編集



タグに「@cta」が指定されたページの本文は、「ウィジェット-コールトゥアクション」を呼び出した箇所に表示されます。

コールトゥアクションを使用しない場合は、ウィジェットを外すだけで非表示にできます。

異なる用途に使用したい場合は、ページ名・付与するタグ名を変更し、テーマ内「**ウィジェット-コールトゥアクション**」の表示条件を修正してください。

「@cta」を指定したページが複数ある場合は、公開日順にすべて表示されます。

Movable Typeの仕様上、通常のページが別途生成されます。非公開のフォルダに収納するようにするか、別途プラグインなどを利用してファイル生成を制限してください。

バナーの編集



タグに「@banner」が指定されたページの本文は、「ウィジェット-バナー」を呼び出した箇所に表示されます。

コールトゥアクションを使用しない場合は、ウィジェットを外すだけで非表示にできます。

PowerCMSのバナー機能等に変更する場合は、テーマ内「[ウィジェット-バナー](#)」等を参照ください。

「@banner」を指定したページが複数ある場合は、公開日順にすべて表示されます。

Movable Typeの仕様上、通常のページが別途生成されます。非公開のフォルダに収納するようにするか、別途プラグインなどを利用してファイル生成を制限してください。



ヘッドライン (headline)



[ウェブサイト](#) < [ブログ](#) < [カテゴリー](#)

例2：ウェブサイトが「headline」、子ブログBが「body」の場合は、ウェブサイトの検索以外のすべての一覧が「headline」、子ブログBの検索以外のすべての一覧が「body」となる

▶ 次ページへ続く

コンポーネントの詳細

初期パッケージで選択可能な、6種類のコンポーネントの一覧です。

コンポーネント内で呼び出している「記事の繰り返し部分」は、すべて名称に「記事ループ」が付いているモジュールテンプレートとして作成しています。案件に応じて追加・編集も可能です。

詳細は「[外観に関する機能 > 記事ループモジュールについて](#)」を参照ください。

カード (card)

上にアイキャッチ画像、下にタイトル・日付が表示されます。Bootstrapの「card」コンポーネントが表示されます。

ヘッドライン (headline)

左に日付、右にタイトルが表示されます。モバイル画面では折りたたまれ、リストグループと同じ状態になります。

メディア (media)

左にアイキャッチ画像、右にタイトル・日付・概要120文字が表示されます。Bootstrapの「media」コンポーネントが表示されます。

リストグループ (listgroup)

日付・タイトルがリストで表示されます。Bootstrapの「list-group」コンポーネントが表示されます。

本文 (body)

旧来のブログの形式で、アイキャッチ画像以外のすべての情報を表示します。このモジュールのみ「本文」と「続き」を区別し、一覧ページでは本文のみが表示されます。また、SNSのシェアボタンも表示されます。

装飾なし (none)

デバッグ、および新しい記事ループを追加するときのひな形用のモジュールです。すべての記事情報を装飾なしで表示します。

上記以外の特殊なコンポーネントも、別売にて順次リリース予定です。

外観に関する機能

メイン画像を利用する

「echo」では、管理画面の「[設定 > 全般](#)」で、メイン画像を5点までアップロードすることができます。

メイン画像の情報は変数に代入され、MTMLが使用できる以下の箇所で参照できます。メインビジュアル・スライドショーだけでなく、重要なページのメイン画像として利用することも可能です。

- テンプレート
- ページ本文・続き
- ウィジェット

変数を呼び出す際は、必ず「パーツ-メイン画像」モジュールを呼び出す必要があります。以下のサンプルコードをテンプレート等に貼り付けてご確認ください。

```
<mt:Include module="パーツ-メイン画像" />
```

メイン画像1

URL: <mt:var name="ec_main_1_url" />

alt: <mt:var name="ec_main_1_alt" _default="(なし)" />

幅: <mt:var name="ec_main_1_width" /> 高さ: <mt:var name="ec_main_1_height" />

メイン画像2

URL: <mt:var name="ec_main_2_url" />

alt: <mt:var name="ec_main_2_alt" _default="(なし)" />

幅: <mt:var name="ec_main_2_width" /> 高さ: <mt:var name="ec_main_2_height" />

メイン画像3

URL: <mt:var name="ec_main_3_url" />

alt: <mt:var name="ec_main_3_alt" _default="(なし)" />

幅: <mt:var name="ec_main_3_width" /> 高さ: <mt:var name="ec_main_3_height" />

メイン画像4

URL: <mt:var name="ec_main_4_url" />

alt: <mt:var name="ec_main_4_alt" _default="(なし)" />

幅: <mt:var name="ec_main_4_width" /> 高さ: <mt:var name="ec_main_4_height" />

メイン画像5

URL: <mt:var name="ec_main_5_url" />

alt: <mt:var name="ec_main_5_alt" _default="(なし)" />

幅: <mt:var name="ec_main_5_width" /> 高さ: <mt:var name="ec_main_5_height" />

ページのレイアウトについて



The screenshot shows a web form for configuring a page layout. It includes a link '画像を選択' (Select image), a text box for uploading a page icon, a dropdown menu for 'ページレイアウト *' (Page layout) with options 'col2' (selected), 'col1', and 'none', and a text box for 'title要素' (Title element).

ページでは、三種類の「レイアウト」を選択することができます。

デフォルト値は「col2」となっています。

col2

通常の2カラムレイアウトでページを作成します。

col1

サイドメニューが本文の下となり、ややスリムな1カラムレイアウトとなります。

none

サイドメニューやコンテンツの上下余白を表示しなくなります。ランディングページなどにご利用ください。

ウィジェットをカスタマイズする



「echo」は、テーマ内の各所に「ウィジェットエリア」を設けており、コンテンツを容易に差し替えることができます。

テーマ内のウィジェットエリアは、以下の9箇所です。

- コンテンツ-下段
- コンテンツ-本文の前-ブログ
- コンテンツ-本文の前-ページ
- コンテンツ-本文の後-ブログ
- コンテンツ-本文の後-ページ
- サブメニュー-ブログ
- サブメニュー-ページ
- ホーム-下段
- ホーム-中段1
- ホーム-中段2

表示した場所でマークアップを切り替える

「サブメニュー-ブログ」「サブメニュー-ページ」「ホーム-中段2」でウィジェットを呼び出しているときは、変数「`__is_sub__`」の値が1になっています。

これを利用すると、メインコンテンツに使用したときと、サブメニューで使用したときで、マークアップを変更することができます。

詳しくは、テーマ内「[ウィジェット-ブログ新着-ヘッドライン](#)」等のテンプレートを参照ください。

▶ [次ページへ続く](#)

記事一覧に公開日を出さないようにする



新着ウィジェットに「カード」「メディア」「リストグループ」を選択した場合、公開日が表示されます。

この日付は、簡単な修正で非表示にすることができます。

「ヘッドライン」は日付を非表示にすることはできません。「リストグループ」を利用してください。

管理画面の「[デザイン > ウィジェット](#)」へ移動し、各ウィジェットの編集画面を開いて、以下のコードを探してください。

```
<mt:include module="記事ループ- (各コンポーネントの名前)" />
```

コードに以下の通り、「date="0"」を追加して再構築すると、そのウィジェットでは日付が表示されなくなります。

```
<mt:include module="記事ループ- (各コンポーネントの名前)" date="0" />
```

「記事の繰り返し部分」に関するモジュールは、他にも様々なオプション変数を持っています。詳細は次ページ「[記事ループモジュールについて](#)」を参照ください。

記事ループモジュールについて

「echo」では、さまざまな一覧表示に対応できるよう「記事の繰り返し部分」をモジュール化しています。

記事一覧の「コンポーネント」や、ウィジェットの内部で呼び出しています。

名称に「記事ループ」が付いているモジュールがこれにあたり、初期パッケージでは以下の6種類を用意しています。それぞれの体裁については、本書内「[コンテンツ作成に関する機能 > 記事一覧の体裁を変更する](#)」を参照ください。

- 記事ループ-カード
- 記事ループ-ヘッドライン
- 記事ループ-メディア
- 記事ループ-リストグループ
- 記事ループ-本文
- 記事ループ-装飾なし

「装飾なし」以外のモジュールには「オプション変数」があり、以下のようにモディファイアとして規定の値を付与することで、表示結果を変えることができます。

```
<mt:include module="記事ループ-メディア" tag="p" thumbnail="0" date="0" excerpt="0" />
```

オプション変数

変数の詳細は、別添資料「変数表」を参照ください。

■ 記事ループ-カード

tag	タイトルを包括する要素名
date	0で日付を出さない

■ 記事ループ-ヘッドライン

tag	タイトルを包括する要素名
-----	--------------

▶ 次ページへ続く

■ 記事ループ-メディア

tag	タイトルを包括する要素名
thumbnail	0でアイキャッチ画像を出さない
date	0で日付を出さない
excerpt	0で概要を出さない

■ 記事ループ-リストグループ

date	0で日付を出さない
flush	0で全体にボーダーを引く（ウィジェット内で表示しないときに利用する）

■ 記事ループ-本文

tag	タイトルを包括する要素名
date	0で日付を出さない
sns_top	1で本文の上にSNSボタンを出す
sns_bottom	1で本文の下にSNSボタンを出す

CSS・JSに関する機能

CSSの構造について

テーマ内のスタイルシートファイルは、以下の4点で構成されています。それぞれファイル名に「.min」が付与された圧縮版があり、初期状態では圧縮版を参照しています。

CSSをカスタマイズする際は「bootstrap-theme.css」と「style.css」を変更してください。

bootstrap.css

Bootstrap 4 Alphaのコアファイルです。このファイルは編集しないようにしてください。

bootstrap-theme.css

コンポーネント、ボタン、文字サイズなど、BootstrapのCSS定義を上書きしているファイルです。

utility.css

ヘッダ・フッタ・メインビジュアルの基礎、ヘッドラインなど、このテーマのために追加したCSS定義です。

追加コンポーネントについては、初期ページ「追加コンポーネントのサンプル」を参照ください。

style.css

メインビジュアルのカラーリングなど、テーマの外観調整用のCSS定義ファイルです。

ビルドツールを使用せず、CSSファイルを直接編集したい場合は、テーマ内「**head要素**」テンプレートを修正し、未圧縮版を参照するようにしてください。

SaSSの構造について

Sassファイル一式は「_scss」ディレクトリ以下に格納されています。

mixinファイル一式は「Bootstrap 4 Alpha」の公開バージョンと同一となっています。 mixinは変更しないことを強くお勧めします。

bootstrap/_mixins.scss

`bootstrap/mixins/`以下の、Bootstrapのmixinを呼び出しているファイルです。

bootstrap/mixins/ 以下

Bootstrapのmixin一式です。

_variables.scss

カラーリング、見出しの基本文字サイズなど、echoの基本的な設定をここで定義することができます。

「Bootstrap 4 Beta」の公開バージョンとは内容が異なります。次ページ「**bootstrap-theme.cssを自作する**」を参照ください。

bootstrap-theme.scss • utility.scss • style.scss

CSSの出力元となるSassファイルです。前述のmixinと共通変数を参照しています。

▶ [次ページへ続く](#)

bootstrap-theme.cssを自作する

bootstrap-theme.css等のCSSファイルを自作したい場合は、以下の点にご注意ください。

■ 「bg-hover」変数を追加する

echoでは、「media・cardなどにカーソルを置いたときの背景色」の変数を追加しています。公開版のBootstrapの公開ファイルの「_variables.scss」に、以下の一行を追記してください。

```
$bg-hover: (カラーコードもしくは変数) !default;
```

■ Bootstrapの全ファイルを最新版にする

Bootstrap 4はベータ版で、逐一変更されています。最新の環境にしたい場合は、mixinも含めてすべて差し替えてください。

JavaScriptの構造について

テーマ内のJavaScriptファイルは、以下の3点で構成されています。このうち「all.js」のみがechoオリジナルのスク립トファイルとなり、ファイル名に「.min」が付与された圧縮版があります。

jquery.min.js

jQueryの、Bootstrap 4 Alphaで推薦されているバージョンです。

bootstrap.min.js

Bootstrap 4 Alphaのスク립トです。カルーセルやタブを利用する場合は必須となります。

all.js

echoの演出用のスク립トです。スムーズスクロール、cardの高さを揃えるなど、最低限のスク립トのみとなっています。

gulpの利用について

「echo」はタスクランナー「gulp」でのCSS・JS生成に対応しています。フォルダ構成も、ウェビングスタジオの実際のプロジェクトをそのまま再現したものとなっています。

<http://gulpjs.com/>

パッケージにはgulpfile.js・package.jsonが含まれていますが、関連プラグイン（node_modules）は含めていません。必ず制作前に、最新版パッケージをダウンロードするようにしてください。

gulpfile.jsの処理の内容

■ _scss/**/*.scss 以下に変更があった場合

該当SCSSをCompassに基づいてCSSにビルドし、`dist/themes/echo_bootstrap/blog_static/css/`に保存する。次いで全CSSファイルの圧縮版を更新する

■ _js/**/*.js 以下に変更があった場合

該当JSを`dist/themes/echo_bootstrap/blog_static/js`へ複製し、「.min」を付与した圧縮ファイルを同時に生成する

ウェビングスタジオはJavaScriptを専門としてないため、gulpfile.jsの内容もやや冗長となっています。制作環境に併せ、任意に修正してください。

その他の機能

ページ分割にPageButeを利用する



ブログトップページ・カテゴリーアーカイブ一覧に、ページ分割プラグイン「PageBute」を利用することが可能です。

初期状態のテンプレートでは、ページ分割を行いません。PageButeを利用したい場合は、以下の手順で導入してください。

PageButeをインストールする

PageButeは株式会社スカイアーク様のウェブサイトで配布されています。下記URLから最新版をダウンロードし、Movable Typeのプラグインディレクトリにアップロードしてください。

<http://www.skyarc.co.jp/engineerblog/entry/2642.html>

▶ 次ページへ続く

テンプレートを入れ替える

テーマの `/dist/themes/echo_bootstrap/_pagebute/` ディレクトリ内の、以下の3ファイルがあることを確認してください。

- `archive_paging.mtml`
- `blog_index_paging.mtml`
- `nav_pagination_paging.mtml`

これらのファイルの内容を、以下のとおり既存テンプレートと入れ替えてください。テンプレートファイルを入れ替えてインストールする方法でも構いません。

ファイル名	対応するテンプレート名
<code>archive_paging.mtml</code>	アーカイブ
<code>blog_index_paging.mtml</code>	ブログトップページ
<code>nav_pagination_paging.mtml</code>	ナビ-ページネーション

サイト全体を再構築すると、記事が下記件数を超過している場合は、ページ分割が行われるようになります。

ブログトップページ	「コンフィグ-共通」で設定した値（headline、listgroupは3倍）
アーカイブ	「コンフィグ-共通」で設定した値（headline、listgroupは3倍）
検索結果・タグ	「コンフィグ-共通」で設定した値の2倍

PageButeはファイルをすべて静的に出力するため、記事件数がひじょうに多いと再構築時間がかかるようになります。記事件数が増えることが予想される場合は、別の手段を検討ください。

カテゴリー・フォルダの階層情報を利用する

「echo」では、カテゴリー情報、フォルダ情報があるテンプレートでは、その記事もしくはページが属している階層を起点に「現在」「親」「先祖（三階層以上であれば最上層）」の情報を変数で取得することができます。

変数名はカテゴリー・フォルダともに「ec_taxonomy」が接頭辞となります。

例えば、カテゴリーに含まれている記事ページで以下のタグを記述すると、先祖アーカイブのURLが表示されます。

```
<mt:Var name="ec_taxonomy_ancestor_link">
```

「親カテゴリーを起点にナビゲーションを表示したい」「先祖フォルダの名前をコンテンツタイトルに常に表示したい」というウェブサイト制作でよくある要望に簡単に対応することができます。

取得できる情報は三階層までとなっていますが、よほどの大規模サイトでなければ三階層以上になることはないため、ほぼ変数のみで対応できます。

依存ファイルのディレクトリを変更する

「echo」は本格的なWeb制作に特化しているため、依存ファイルのフォルダ名は、一般的な名称となっています。

「既存のディレクトリ名が重複している」「コーディングルールが決まっている」等の理由で、依存ファイルを移動したい場合は、以下の手順で修正してください。

1. インストール時に依存ファイルを読み込まないようにする

テーマパッケージ直下の「theme.yaml」をテキストエディタで開き、以下の記述を探してください。

```
blog_static_files:
  component: ~
  data:
    - images
    - css
    - js
    - fonts
    - share
  importer: blog_static_files
```

「blog_static_files:」の行から「importer: blog_static_files」の行までを削除すると、インポート時に依存ファイルを読み込まなくなります。

2. 依存ファイルを別途アップロードする

テーマパッケージ内の「blog_static」に含まれるファイルを、サーバー内の任意の場所にアップロードしてください。

▶ [次ページへ続く](#)

3. 依存ファイルの参照元を変更する

管理画面の「デザイン > テンプレート」へ移動し、「**コンフィグ-共通**」テンプレートの編集画面から、以下の記述を探してください。

```
<mt:WebsiteRelativeURL setvar="ec_static_path" note="画像、CSS等の依存ファイルのパス: 異なる階層に設置する場合は変更する" />
```

ここで、依存ファイルの起点となるパスを定義しています。初期状態では「**mt:WebsiteRelativeURL = 親ウェブサイトのルートディレクトリ**」となっているため、任意の値に変更してください。

たとえば、「/path/to/static/」に変更したい場合は、以下のとおり書き換えてください。

```
<mt:SetVar name="ec_static_path" value="/path/to/static/" />
```

4. 個別に変更する

ファイルごとにパスを変更したい場合は、「**head要素**」テンプレートを直接編集してください。

補足

コーディングルール

HTML

- インデントは、スペース4個とする（ただし、Movable Typeの設定で出力後はインデントがすべて削除される）
- html・head・bodyはインデントしない
- div・navは閉じタグの前にクラス名のコメントを付与する
- 閉じタグがない要素の末尾にはスラッシュを入れない

CSS

- インデントは、スペース2個とする（ただし、コンパイル後はSassの設定に従って整形される）
- クラス名は小文字、ハイフン・数字のみとする
- クラス名は「コンポーネント名-部位名-役割名-パターン」とする
- IDはアンカー対象、および一度しか使用しないことがわかっている要素以外では極力使用しない

MTML

- テンプレートの先頭には、参照しなければならない変数、および、そのテンプレートのみで利用している変数を明示する
- 接頭辞は「MT」ではなく「mt:」とする
- 否定条件はmt:Ifの「ne」モディファイアではなく、mt:Unlessの「eq」モディファイアで統一する
- タグに「\$」は使用しない（Sublime TextのHTMLシンタックスに対応するため）
- 閉じタグがないファンクションタグの末尾にはスラッシュを入れる
- 条件分岐にコメントが必要となる場合はnoteモディファイアを使用する
- mt:Elseなどでnoteモディファイアを利用できない、もしくは長文となる場合はmt:TemplateNoteで直前に書く

▶ [次ページへ続く](#)

変数、フィールドの命名

変数は役割によって命名規則が異なっている。これにより、名称で役割を判別可能にする。詳しくは別添の変数表を参照。

- グローバル変数: 接頭辞「ec_」＋分類＋名称 (lowercase)
- ローカル変数: 「_」＋名称＋「_」 (lowercase)
- モディファイア用変数: 名称 (lowercase)
- カスタムフィールド名: 管理画面での利用箇所＋名称 (capitalize)

モディファイア用変数はわかりやすさを考慮して接頭辞を付けていないが、厳密には何らかの接頭辞を付けた方が重複などの事故を回避できる。

ダイナミック・パブリッシングに対応できなくなるため、ハッシュは使用していないが、高速化・効率化を考慮する場合は積極的に利用するのが望ましい。

TIPS

mt:Entriesをウィジェットに使用する場合の注意

記事新着などのウィジェットを、カテゴリーアーカイブに使用すると記事が一件も表示されないことがある。これは、以下のとおり条件に「category」を明示していないと、表示しているページのカテゴリーを参照してしまうためである。

```
<mt:Entries sort_by="authored_on" sort_order="descend" limit="7">
```

以下のように「存在しないカテゴリー名」をNOT条件としてcategoryモディファイアを指定すると、どのページでもすべてのカテゴリーを表示することができる。

```
<mt:Entries category="NOT xxx" sort_by="authored_on" sort_order="descend" limit="7">
```

「公開日時降順」であれば「limit」ではなく「lastn」を利用した方が、このような事故が起きないので無難。

アーカイブタイプの判定

アーカイブタイプから「カテゴリー系であるか」を判断するときは、eqではなくlikeを使用すること。
eqだと、カテゴリー+月別などのアーカイブを判別できない。

```
<mt:ArchiveType setvar="ec_archive_type" note="アーカイブタイプ" />
<mt:If name="ec_archive_type" like="Category" note="カテゴリー関連のとき">
  . . . 処理 . . .
</mt:If>
```

▶ 次ページへ続く

モジュールにモディファイア変数を付与する場合の注意

以下のとおり、何も渡さなかった場合が初期値の「1 (true)」で、「0」を渡した場合を「false」とする場合の処理について。

```
<mt:Include module="モジュール名" hoge="0" />
```

モジュールの冒頭で「何も渡さなかった場合＝変数が存在しない場合」の処理を書くこととなるが、以下は正しく動作しない。0はプログラムの「false」であり、Unlessの対象となるためである。

```
<mt:Unless name="hoge">
  <mt:SetVar name="hoge" value="1" />
</mt:Unless>
```

これを回避するにはブラックリスト式とし、「0でない」ことを条件にUnlessを書く。

```
<mt:Unless name="hoge" eq="0">
  <mt:SetVar name="hoge" value="1" />
</mt:Unless>
```


補足

コーディングルール

HTML

- インデントは、スペース4個とする（ただし、Movable Typeの設定で出力後はインデントがすべて削除される）
- html・head・bodyはインデントしない
- div・navは閉じタグの前にクラス名のコメントを付与する
- 閉じタグがない要素の末尾にはスラッシュを入れない

CSS

- インデントは、スペース2個とする（ただし、コンパイル後はSassの設定に従って整形される）
- クラス名は小文字、ハイフン・数字のみとする
- クラス名は「コンポーネント名-部位名-役割名-パターン」とする
- IDはアンカー対象、および一度しか使用しないことがわかっている要素以外では極力使用しない

MTML

- テンプレートの先頭には、参照しなければならない変数、および、そのテンプレートのみで利用している変数を明示する
- 接頭辞は「MT」ではなく「mt:」とする
- 否定条件はmt:Ifの「ne」モディファイアではなく、mt:Unlessの「eq」モディファイアで統一する
- タグに「\$」は使用しない（Sublime TextのHTMLシンタックスに対応するため）
- 閉じタグがないファンクションタグの末尾にはスラッシュを入れる
- 条件分岐にコメントが必要となる場合はnoteモディファイアを使用する
- mt:Elseなどでnoteモディファイアを利用できない、もしくは長文となる場合はmt:TemplateNoteで直前に書く

▶ 次ページへ続く

変数、フィールドの命名

変数は役割によって命名規則が異なっている。これにより、名称で役割を判別可能にする。詳しくは別添の変数表を参照。

- グローバル変数: 接頭辞「ec_」 + 分類 + 名称 (lowercase)
- ローカル変数: 「_」 + 名称 + 「_」 (lowercase)
- モディファイア用変数: 名称 (lowercase)
- カスタムフィールド名: 管理画面での利用箇所 + 名称 (capitalize)

モディファイア用変数はわかりやすさを考慮して接頭辞を付けていないが、厳密には何らかの接頭辞を付けた方が重複などの事故を回避できる。

ダイナミック・パブリッシングに対応できなくなるため、ハッシュは使用していないが、高速化・効率化を考慮する場合は積極的に利用するのが望ましい。

TIPS

mt:Entriesをウィジェットに使用する場合の注意

記事新着などのウィジェットを、カテゴリーアーカイブに使用すると記事が一件も表示されないことがある。これは、以下のとおり条件に「category」を明示していないと、表示しているページのカテゴリーを参照してしまうためである。

```
<mt:Entries sort_by="authored_on" sort_order="descend" limit="7">
```

以下のように「存在しないカテゴリー名」をNOT条件としてcategoryモディファイアを指定すると、どのページでもすべてのカテゴリーを表示することができる。

```
<mt:Entries category="NOT xxx" sort_by="authored_on" sort_order="descend" limit="7">
```

「公開日時降順」であれば「limit」ではなく「lastn」を利用した方が、このような事故が起きないので無難。

アーカイブタイプの判定

アーカイブタイプから「カテゴリー系であるか」を判断するときは、eqではなくlikeを使用すること。
eqだと、カテゴリー+月別などのアーカイブを判別できない。

```
<mt:ArchiveType setvar="ec_archive_type" note="アーカイブタイプ" />
<mt:If name="ec_archive_type" like="Category" note="カテゴリー関連のとき">
  . . . 処理 . . .
</mt:If>
```

▶ 次ページへ続く

モジュールにモディファイア変数を付与する場合の注意

以下のとおり、何も渡さなかった場合が初期値の「1 (true)」で、「0」を渡した場合を「false」とする場合の処理について。

```
<mt:Include module="モジュール名" hoge="0" />
```

モジュールの冒頭で「何も渡さなかった場合＝変数が存在しない場合」の処理を書くこととなるが、以下は正しく動作しない。0はプログラムの「false」であり、Unlessの対象となるためである。

```
<mt:Unless name="hoge">
  <mt:SetVar name="hoge" value="1" />
</mt:Unless>
```

これを回避するにはブラックリスト式とし、「0でない」ことを条件にUnlessを書く。

```
<mt:Unless name="hoge" eq="0">
  <mt:SetVar name="hoge" value="1" />
</mt:Unless>
```