

MTDDC 2024

MT構築の現場から

～問題解決事例～

2024年11月23日



WIM WORLD IS MINE, INC.
株式会社ワールドイズマイン

© World Is Mine, Inc.

目次

1. 自己紹介
2. ワールドイズマインのご紹介
3. ディレクター目線で見たと、プロジェクトを成功に導くポイント
4. 構築時のアイデア実例:FLCF (FreeLayoutCustomField) で
ブロックエディタライクな管理画面を作る

1. 自己紹介

1.自己紹介



加藤輝樹

1975年生まれ

愛知県東海高校出身

1.自己紹介



2007年に起業

株式会社ワールドイズマイン

代表取締役

1.自己紹介



非エンジニア
コードを書かない
ディレクターです。

1.自己紹介



**MTとの関わりは
2009年8月から
16年目**

MT3系の頃ですね

1.自己紹介



元シックス・アパート

WEBディレクター

コーポレートサイト
製品サイト
の構築、運用をやっていました。

1.自己紹介



MTを利用した
サイト構築は
100件以上

1.自己紹介



特技は

1.自己紹介



炎上案件の火消し

燃え上がっているプロジェクトにアサインされて、消火活動をしています！

1.自己紹介



趣味は

1.自己紹介



釣り

6年ぶりにハワイで
釣りやってきました！

自社メディア「WIMEDIA」で
ブログ執筆中

<https://media.wim.jp/>

1.自己紹介



ゴルフ

も好きです

なかなか100が切れない

1.自己紹介



将来の 目標

1.自己紹介



ハワイ移住
ハワイ大好きです

2.ワールドイズマインの紹介

2. ワールドイズマインの紹介

WIM
WORLD IS MINE, INC.

社名の由来

2.ワールドイズマインの紹介

初音ミク

2.ワールドイズマインの紹介

ではありません。(好きだけど)

2.ワールドイズマインの紹介

こっちはです。

抗うな。

受け入れる。

すべては繋がっている。

ワールドイズマインが できること



Movable Type

(1) Movable Type 構築

WIMのメイン業務、受託開発。エンドユーザーからの直接受託、代理店経由の受託、制作会社経由の受託と様々な形態での受託開発に対応。

作業領域は現状分析、企画、設計、マーケティング、コンサルティングから、デザイン、コーディング、CMS実装、プラグインの開発、保守までプロジェクトごとに必要な部分だけのお手伝いをいたします。

2.ワールドイズマインの紹介

Movable Type 構築実績

官庁、金融、IT、メーカー etc 業種を問わず対応しています！

Movable Type 以外のCMSやスクラッチでのシステム開発にも対応しております。

2. ワールドイズマインの紹介

シックス・アパート社のウェブサイトの構築

<https://www.sixapart.jp/>

<https://www.movabletype.jp/>

<https://movabletype.net/>

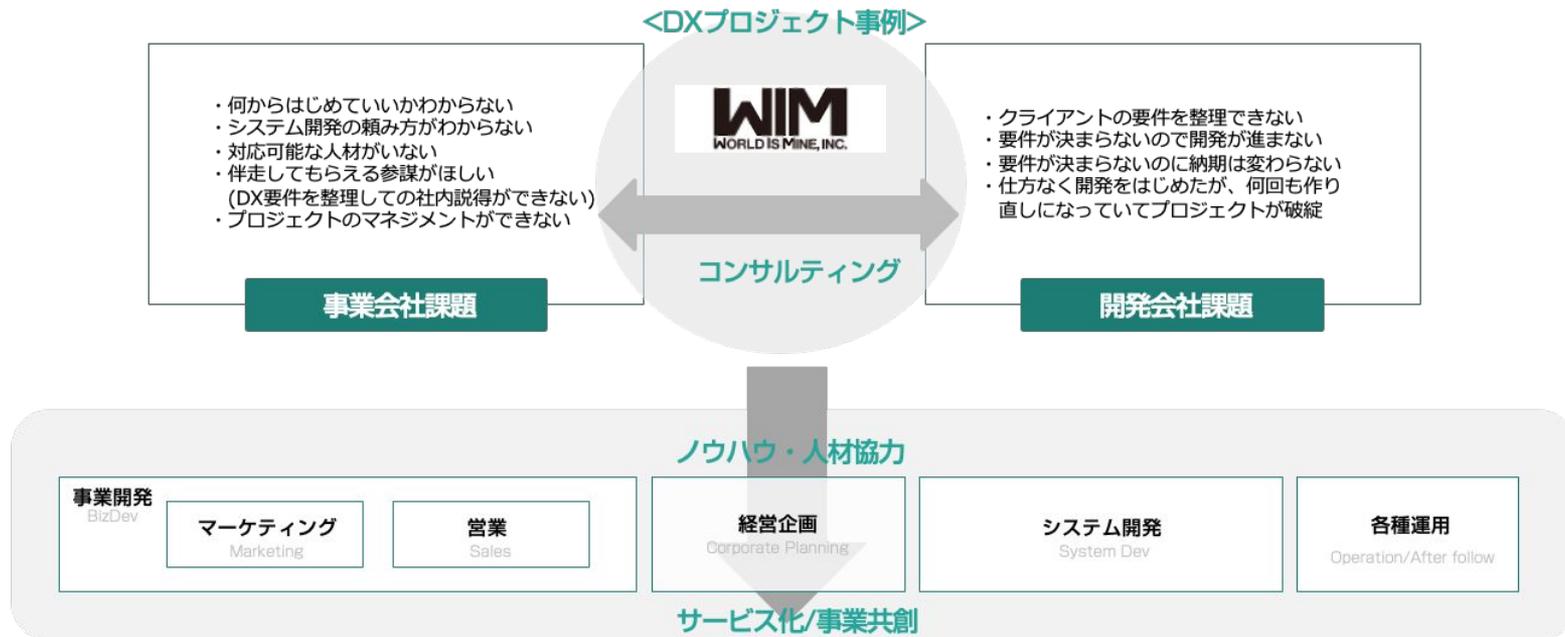
<https://www.lekumo.jp/>

コーポレートサイト、製品サイトの構築、運用を行なっておりました。また、Movable Type のテクニカルサポートについてもお手伝いさせていただいておりました。

印刷物のデザインもやっています！

2.ワールドイズマインの紹介

(2)DX コンサルティング・マーケティング



27

2. ワールドイズマインの紹介

DX事例

社会福祉法人さま

介護スタッフが紙で行っていた「利用者(入居者)の生活記録」のシステム化におけるプロジェクトマネジメントを担当。

本契約が同社会福祉法人のDX化の契機となり、現在は、

- ・人事管理/給与設計システム
- ・ショートステイ予約管理システム
- ・経営層向けダッシュボード(データ統合/リアルタイム出力)

のプロジェクトマネジメントから開発までを担当。

2.ワールドイズマインの紹介

WEB 担当者／構築会社向け Movable Type 導入支援／運用サポートサービス

mtrs

MT リモートサービス

WIM WORLD IS MINE, INC.
株式会社ワールドイズマイン

2.ワールドイズマインの紹介

(3)MTRS(MT リモートサービス)を運営

**お抱えの構築業者を持たないユーザーの、
MTに関するお困りごとを有償でサポートしています。**

<https://www.sixapart.jp/movabletype/solutions/mtrs.html>

2.ワールドイズマインの紹介

(3)MTRS(MT リモートサービス)のメニュー

- ・ログインパスワードの復旧
- ・MTのインストール
- ・MTのアップグレード
- ・サーバ移行
- ・サーバ構築
- ・AWS版への移行
- ・クラウド版への移行
- ・mt.netへの移行
- ・SSL化、SSL証明書の発行、設置、CSRの作成
- ・設定変更、メンテナンス
- ・WordPressからMTへの移行(その他CMSからも)

mtrs プラグインシヨツプ

2. ワールドイズマインの紹介

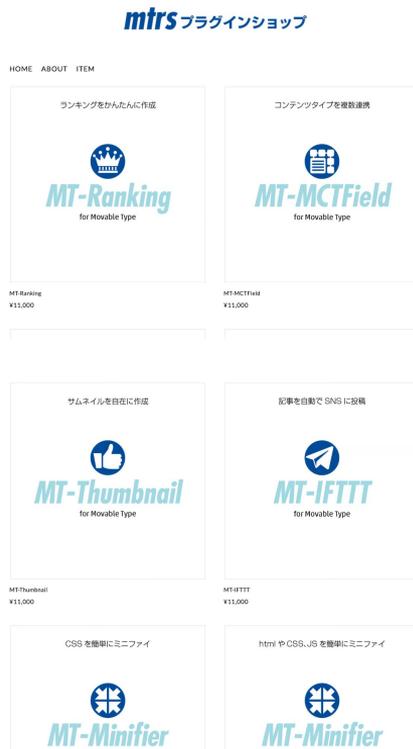
Movable Type プラグインの開発

MTのプラグイン開発を行なっております。一般向けに販売しているものとクライアントに合わせたオーダーメイドがあります。

<https://www.wim.jp/plugins/>

2024年2月 藤本 亓氏の開発したプラグインを継承

藤本氏の開発されたプラグインを引き継ぎ、MT8への対応など継続開発を進めています。



2.ワールドイズマインの紹介

MT8への対応状況

対応済みプラグイン

- Free Layout Custom Field
- Free Layout Content Field
- MailForm
- ExtendArchives

今後対応予定のプラグイン

- CustomFieldListing : 管理画面の一覧画面にカスタムフィールドの値を表示
- SearchEntries : 複数の条件でブログ記事を検索する
- SearchContents : 複数の条件でコンテンツデータを検索する
- NoRebuild : ファイル名に「*no_rebuild*」を含むページを、再構築対象外とする
- ArrayHashUtil : 配列/ハッシュ関連のテンプレートタグを追加する
- BlogLoadContextEx : 複数ブログ処理のモディファイアを拡張する
- EntryCategoriesOrdered : 記事のカテゴリを並び順通りに出力する
etc...

その他ご希望ありましたら、ご相談ください！

2. ワールドイズマインの紹介



MT-Minifier

for Movable Type

MT-Minifier はテンプレートの再構築時に、出力ファイルのミニファイ処理を自動で行います。

対象となるのは html と CSS 、JS ファイルです。

テンプレートはミニファイ前の状態を保持します。

Movable Type 6系、7系、8系で動作します。

2.ワールドイズマインの紹介



MT-Eraser

for Movable Type

MT-Eraserはインデックステンプレートの非公開時に出力ファイルを自動で削除します。

記事やウェブページと同じ感覚でサーバから出力ファイルを削除します。

Movable Type 6系、7系、8系で動作します。

2. ワールドイズマインの紹介



MT-Thumbnail

for Movable Type

MT-Thumbnail は自由な画角でサムネイルを作成します。

指定された幅と高さで画像は生成され、オーバーフロー領域は自動的にトリミングされます。

Movable Type 6系、7系、8系で動作します。

2.ワールドイズマインの紹介



MT-MCTField

for Movable Type

MT-MCTField (MT-MultipleContentTypeField) はコンテンツタイプを拡張します。

1つのコンテンツタイプフィールドに複数のコンテンツタイプを読み込むことができます。

Movable Type 7系、8系で動作します。

2.ワールドイズマインの紹介



MT-Ranking

for Movable Type

MT-Ranking は管理画面から簡単にランキングを作成できます。

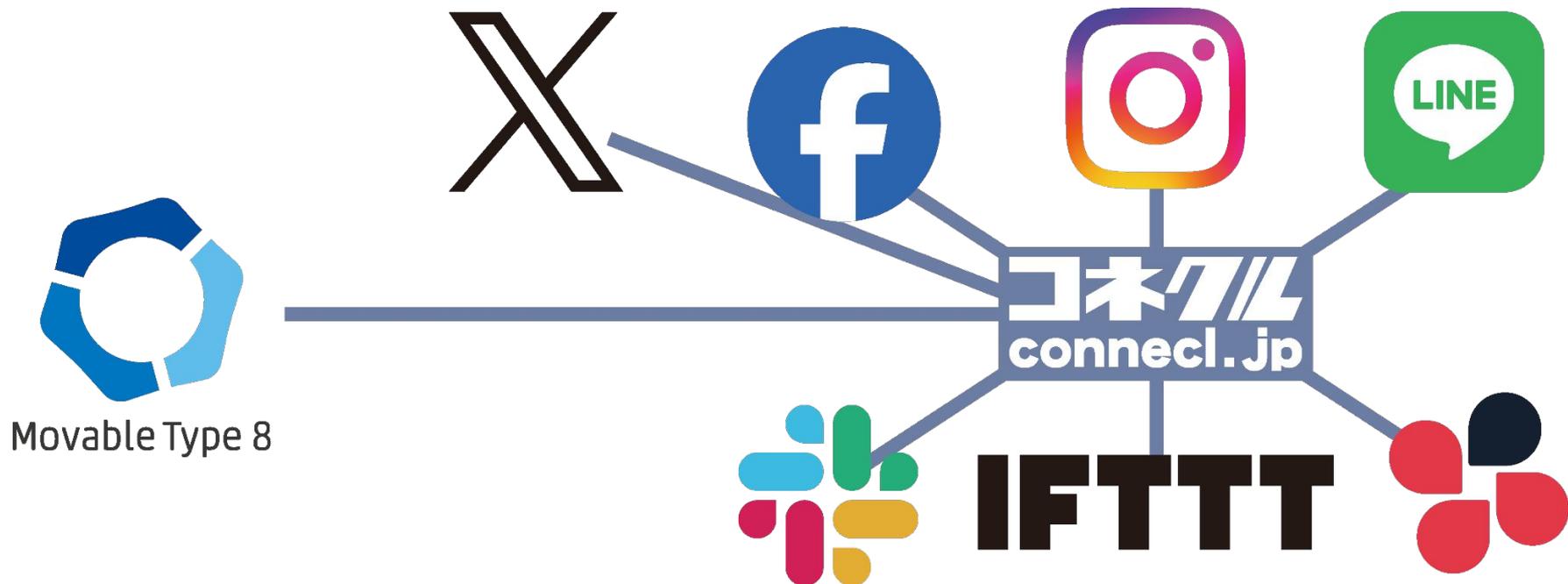
MTタグを利用して表示項目をカスタマイズすることが可能です。

Movable Type 7系、8系で動作します。

2.ワールドイズマインの紹介



2. ワールドイズマインの紹介



2.ワールドイズマインの紹介

コネクルをハブとして、MTとSNSや他のサービスを繋ぎます。

Xへの自動ポスト

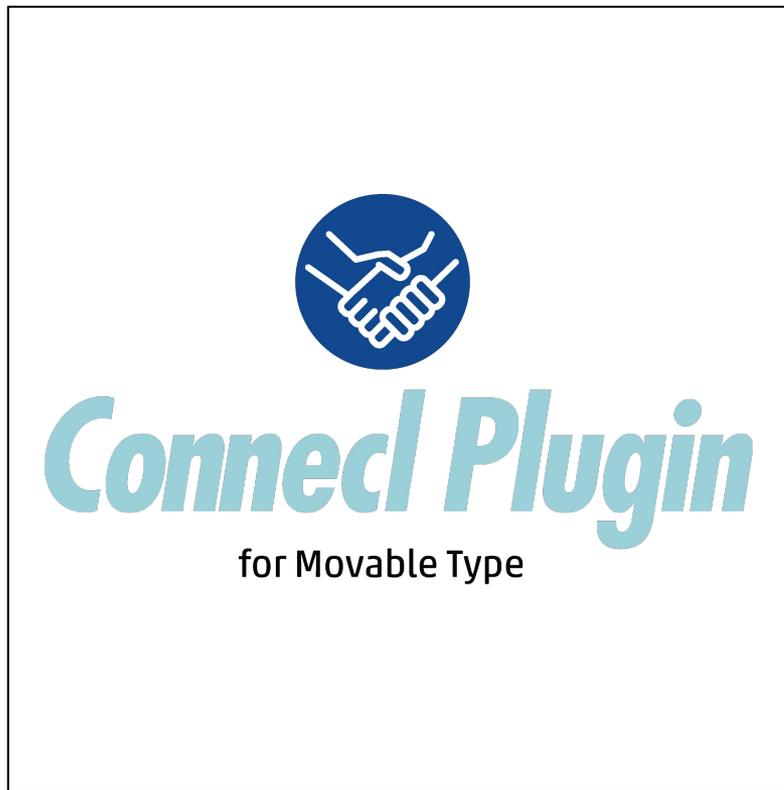
FBへの自動ポスト

etc...

現在ベータ版提供中！

Xとの連携を無料でお試しいただけます！

2.ワールドイズマインの紹介



7.MT専用プラグイン Conecel Plugin for MT の提供

プラグインの提供により、MTの管理画面上から直接コネクルのサービスUIへシームレスにアクセスが可能になります。

2. ワールドイズマインの紹介

The image shows a CMS editor interface with a sidebar on the left containing a 'tiny' logo and several empty text input fields. The main content area displays a settings panel for the 'Connect' plugin. The settings are organized into sections: '変更メモ' (Change memo) with a text input and '公開' (Publish) / 'プレビュー' (Preview) buttons; 'カテゴリ' (Category) with a search input and '追加' (Add) button; 'コメント' (Comments) with '受信設定' (Reception settings) and a checked 'コメントを許可' (Allow comments) option; '記事アセット' (Article assets) with an '追加' (Add) button and the text 'アセットはありません' (No assets); and 'Connect' (highlighted with a red box) with 'Publish to Connect' options (including 'Post to X'), a '説明' (Description) text input, and a checkbox for 'Use entry excerpt instead of description.'.

Connect Plugin for Movable Type

記事やコンテンツデータごとに
編集画面でコネクトへ連携するか
選択可能

※画面は開発中のものとなります。

2. ワールドイズマインの紹介



Connect API
記事 / コンテンツデータの更新



Twitter API
記事 / コンテンツデータの通知



MT-Ranking の場合

記事の編集

タイトル: 「あったかいな」MTRSが発表される (2/20) Movable Type プラグイン「MT-Ranking」の紹介

URL: パーマリンク: https://media.wm.jp/2023/10/17/mi-ranking.html

本文:

```
strong em: blockquote ul of li {
  -img style="margin-bottom: 20px; float: left; border: 1px solid #ccc; padding: 5px; width: 100px; height: 100px; margin-right: 10px;">
  <div> Movable Type Advert Calendar 2023 </div>
  <pre><a href="https://calendar.org/advert-calendar-2023/" target="_blank">Movable Type Advert Calendar 2023</a> の2/4日目の記事です。</pre>
  <div> 自己紹介</div>
  <pre><a href="https://www.wm.jp">株式会社ワールドイズマイン</a>で Movable Type の開発をメインの業務としているチームです。<a href="https://www.facebook.com/worldismine/">https://www.facebook.com/worldismine/</a>です。</pre>
  <pre><a href="https://www.wm.jp/advert-calendar-2023.html">MTRS (MT-Ranking) をはじめ、MT-Ranking に関する、サブドメインの広告サービスの紹介<a href="https://www.wm.jp/plugins/" target="_blank">MT-Ranking</a>の紹介<a href="https://www.wm.jp/plugins/mi-ranking/" target="_blank">MT-Ranking</a>の紹介<a href="https://www.wm.jp/plugins/mi-ranking/" target="_blank">MT-Ranking</a>の紹介<a href="https://www.wm.jp/plugins/mi-ranking/" target="_blank">MT-Ranking</a>の紹介</pre>
```

WIMEDIA (@WIMEDIA)

株式会社ワールドイズマインが運営しています。本社はwebデザイン、webシステム開発をしています。 趣味や仕事の話もゆる〜くお伝えします。 スモールマウスの作りのお話を中心に発信します。 約70名がいて仕事しています。 必死の努力で頑張っているつもりです。 #webデザイン #代客制作 #web制作 #web制作 #web制作

2023年7月23日

2018年7月19日 Twitter を利用しています

504 フォロワー 147 フォロワー

ポスト

WIMEDIA (@WIMEDIA) · 4月26日
MT-Ranking トクダ 仁志 (MTRS) | 2023年 W 中の選ばれるお祝いサービス (MTRS) #MTRS #MTRS

WIMEDIA (@WIMEDIA) · 4月26日
MT-Ranking 1.4 リリースノート - NEWS | 株式会社ワールドイズマインにて開発された記事またはコンテンツの更新を通知するプラグインです。 #web

3.ディレクター目線を見た プロジェクトを 成功へ導くポイント

3.ディレクター目線を見た、プロジェクトを成功に導くポイント

Movable Typeを利用したウェブサイト構築で、
これまでにあった要望で実装が大変だったものや、
その実装例についてご紹介したいと思います。

よくある要望

- ・ブロックエディタで自由なフィールドを作りたい
- ・記事をシリーズ化してトビラページを作りたい
- ・ランキングを表示したい
- ・サイト内検索をリッチにつくりたい
- ・パンくずや構造化データの自動化をしたい

3.ディレクター目線で見た、プロジェクトを成功に導くポイント

今回はブロックエディタの構築事例についてご紹介します

3.ディレクター目線を見た、プロジェクトを成功に導くポイント

が、
その前に！

3.ディレクター目線を見た、プロジェクトを成功に導くポイント

**クライアントの要望通りに実装できないケースや
要望通りに実装できるけれど、ちょっと無理あるよね？
といった経験はありませんか？**

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

**構築作業に入る前のディレクションが正しく行われれば、
無理めな要件を削ぎ落としたり、正しい要件へ変更を行うこと
ができるので、
プロジェクトは成功に近づきます！**

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

**正しいディレクションで、
プロジェクトを成功に導きましょう！**

**というわけで、実装例のご紹介の前に
ディレクターのお仕事と
そのポイント、
サイト構築の流れをご紹介します！**

サイト構築の流れ

- (1)現状分析
- (2)要件定義・設計
- (3)データ移行計画(リニューアルの場合)
- (4)テンプレートの構築(デザイン作成も含む)
- (5)記事などデータの移行
- (6)動作テスト
- (7)調整・修正
- (8)差分のマージ(リニューアルの場合)

デザイナーや開発者が手を動かす前の(1)～(3)のタスクが重要！

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

(1)現状分析

移行元のサイト(CMS)の調査

- ・どんなCMSでつくられていて、どんな管理画面になっているか
 - どのCMSでもできることは似ているが、細かいところは当然違う
 - 運用マニュアルを入手して、サイト制作者のコンセプトや構築意図を探る
- ・どんなディレクトリ構造で作られているか
 - パンくずの実装やアーカイブマッピングに影響がある
 - ディレクトリマップを支給されても自分でも作る！
- ・ユーザーが満足している点、不満に感じている点
 - ヒアリングの過程で(構築者としての)マストな要件とそうでない要件を精査する

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

(1)現状分析

移行元のサイト(CMS)の調査でのポイント

ポイント1:

記事やウェブページ、コンテンツデータの登録画面(ユーザーに見えやすい部分)は移行元のCMSでできたことがMTできないと許容されないケースがあることを頭に入れておく。

ポイント2:

インデックステンプレートで構築する部分はユーザーには意識されないところなので力技でもなんとかできることが多い。

ポイント3:

大規模サイトや公官庁系のサイトの場合、移行元サイトの構築者からの協力を得られない場合がある。

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

ポイントを押さえた上で、意識したいことは、

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

移行元のCMSについては、権限を全てもらえなかったり、未知のCMSであったり、実際の動作が100%把握できないことがある。

プロジェクトの前提として、要件定義は100%FIXではなく、

構築の過程で変更の可能性もあること、
できることとできないことがあることを事前に明示し、

クライアントと**柔軟な関係**を構築することが大切！

その結果、作業者にとっても無理がなく、クライアントも

満足できる要件にまとめられます！

3.ディレクター目線を見た、プロジェクトを成功に導くポイント

まとめ：

技術的なポイントは押さえつつ、

**現状分析段階からのクライアントとの意思の疎通、
すり合わせが最も大事。**

**クライアントと言うべきことを言い合える関係になることで
作業者にストレスフリーな要件にまとめましょう！**

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

結論：

クライアントと仲良くなれ！

作業者にやさしくあれ！

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

お陰さまで特技が火消しになりました！

サイト構築の流れ

- (1)現状分析
- (2)要件定義・設計**
- (3)データ移行計画(リニューアルの場合)
- (4)テンプレートの構築(デザイン作成も含む)
- (5)記事などデータの移行
- (6)動作テスト
- (7)調整・修正
- (8)差分のマージ(リニューアルの場合)

デザイナーや開発者が手を動かす前の(1)～(3)のタスクが重要！

(2)要件定義・設計

MT上でのフィールドの設計や動作についての定義

- ・サイトの構造

→どの粒度でサイトをわけるか

- ・フィールドの構造

→移行元のフィールド構造を維持することが妥当なのか検討

- ・連携する動作の定義

→自動で出力する部分と手動で登録する部分の精査

(2)要件定義・設計

MT上でのフィールドの設計や動作についての定義でのポイント

ポイント1:

どこまで移行元の機能を踏襲するか。要求仕様にあっても妥当でないと思われる機能は要求に対する温度感を把握し、時間をかけてディスカッションし調整する。

ポイント2:

コンテンツタイプで作るべきなのか、記事で作るべきなのか。インクルードする内容やフィールドの数、既存のデータ形式、必要なアーカイブなどから総合的に判断する。

ポイント3:

潜在化していない要件があることを想定し、極力シンプルで自由度の高い設計を心がける。
プラグインはなるべく導入しない。

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

ポイントを押さえた上で、意識したいことは、

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

クライアントがCMSを移行するには理由があります。

- ・使いにくい
- ・保守が切れた
- ・費用が高い

など

それぞれのケースによって移行先のCMSに対する温度感は変わります。
RFPには書かれていない要望も含めて汲み取ることが大切。

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

どのような場合でも、我々が目指すゴールは

- ・ユーザーにとってより使いやすい
- ・メンテナンスがしやすい
- ・MT的に妥当な実装になっている

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

クライアントはMTの専門家ではなく、利用ユーザーなので、
今の使い勝手との比較や、今できることを移行後もしたいという要望があります。

その要望は必ずしも構築者にとってはベストな選択ではないことがあります。

なるべく要望に沿いつつも、MTとして妥当な選択を提案し、納得してもらうことが、その後の開発をスムーズにし、結果的にクライアントにとっても良いものができます。

3.ディレクター目線で見ると、プロジェクトを成功に導くポイント

例えば、記事にカスタムフィールドが数十個あるような場合。

再構築の負荷の問題は常にあります。

記事を作成するユーザーのスキルレベルは？

定型文やサンプルhtmlを用意して記事を作るという選択は許容できないか？

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

例えば、記事を自由に並べ替えたい。

移行元ではフィールドに数値を入力して制御している場合があります。

公開日で制御すれば、1つフィールドを減らせます。

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

**要件の整理後、外せない要件をまとめて、
設計を行い、実装に進みます。**

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

まとめ:

要求仕様作成者はMTの専門家ではない(ことも多い)ので、要求を100%実現することが正義ではないかも？

取捨選択、別の方法もあることを提示することを恐れない。

要望を実現するだけでは、結果的に使いにくいものになってしまふことがよくあります。恐れず話しましょう。

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

結論:

ディスカッションを怖がらない

何がクライアントにとって一番なのかを想像する

サイト構築の流れ

- (1)現状分析
- (2)要件定義・設計
- (3)データ移行計画(リニューアルの場合)**
- (4)テンプレートの構築(デザイン作成も含む)
- (5)記事などデータの移行
- (6)動作テスト
- (7)調整・修正
- (8)差分のマージ(リニューアルの場合)

デザイナーや開発者が手を動かす前の(1)～(3)のタスクが重要！

(3) データ移行計画

どうやってデータを引っ越すか

- ・データベースを改変して取り込むケース

→PowerCMSからの移行、バナー機能などの独自機能をリプレイスしてデータを取り込む

- ・手動で取り込むケース

→MySQLが利用されていないケースやテーブル構造が著しく違う場合

→1000ページくらいまでのサイトであれば手動でやったほうが結果早いことが多い

- ・上記のハイブリット型

→DBから取り込めないところだけ手動で移行する

- ・エクスポートデータを利用するケース

→WPとPowerCMS以外ではあまりみたことない

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

具体的な事例

(1)複雑なブロックエディタ風な実装が行われている

ハートコア→フリーレイアウトカスタムフィールドで実装し、データは手動で移行

(2)MTにはない機能が使われている

PowerCMS→拡張されたカスタムフィールド→MTAppjQueryで同機能を実装し、MT6のDBを作成。→PowerCMSのテーブル構造と突合し、データを流し込み→MT6で実装してからMTをアップグレードする

(3)エクスポートデータが不完全

WordPress→プラグインが導入されていたり、カスタム投稿タイプが利用されている場合、完全なデータがエクスポートできないことがある。

移行できるものだけを移行し、他は手動で対応。

(4)移行元のデータ構造と移行先のデータ構造が著しく異なる

データは移行したいが、今後作られるデータは新しいフォーマットで作っていきたい。→移行用と新規作成用のフィールドを分けて設計

3.ディレクター目線で見ると、プロジェクトを成功に導くポイント

まとめ:

**移行元のデータによって、適切な移行方法は異なる。
エンジニア、コーダー、ディレクターとスキルセットの違うメン
バーでの検討が有効。**

**移行に時間がかかることはイメージしづらいので、バッファをも
つ。**

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

結論：

最終的には手動で移行すればなんとかなる！

サイト構築の流れ

- (1)現状分析
- (2)要件定義・設計
- (3)データ移行計画(リニューアルの場合)
- (4)テンプレートの構築(デザイン作成も含む)
- (5)記事などデータの移行
- (6)動作テスト
- (7)調整・修正
- (8)差分のマージ(リニューアルの場合)

(1)～(3)の段階での準備がうまくいけば、(4)から先の作業時に、現場から、これ無理なんですけど～～といわれずに済みます！

3.ディレクター目線で見たと、プロジェクトを成功に導くポイント

ここまでで、クライアントとも十分なコミュニケーションがとれているので、

全部作ったあとに、イメージと違うんですけど

といわれたい！

炎上もしたくない！

3.ディレクター目線で見た、プロジェクトを成功に導くポイント

ここからは、構築についての事例をご紹介します。

4.構築時のアイデア実例 : FreeLayoutCustomFieldで ブロックエディタライクな 管理画面を作る

Free Layout Custom Fieldとは？

FreeLayoutCustomFieldプラグインは、HTMLとCSSでレイアウトできるカスタムフィールドを追加するプラグインです。

1つのカスタムフィールドの中に複数の入力欄(フィールド)を入れることができ、それらをHTMLとCSSでレイアウトすることができます。

フィールドの種類を追加

同じフィールドの組み合わせを複数回追加

異なるフィールドの組み合わせを複数回追加(ユニット)

今回はユニット機能を利用します

複雑なブロックエディタの実装事例

ヘッドレスCMSの流行などサイト構築において、求められるスキルセットは日々高度になっています。しかしながら、多くの制作会社はperl、JSなどを巧みに使いこなすエンジニアをチームに持たないことも多いと思います。

待望のMTブロックエディタがMT8から同梱されましたので、ご利用されている方も多いと思います。

自由度が高いので使いこなせれば他の選択肢は不要ですが、カスタマイズのハードルは少し高いと感じられている方もいらっしゃるのではないのでしょうか？

今回はもう一つの選択肢として、FreeLayoutCustomFieldを利用したブロックエディタの実装事例をご紹介します。

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

(1)MTブロックエディタ

フロントエンジニアがチームにアサインされている場合にはカスタムスクリプトを駆使して作成が可能

- ・自由度が高い
- ・カスタムスクリプトを使いこなすにはJSの理解が必要。
- ・シックス・アパート社製

(2)Free Layout Custom Field

jQueryがわかる、MTはわかるといったスキルセットのコーダーでも複雑なブロックエディタを作成可能

- ・追加フィールドが用意されているので、ゼロから作らなくて良い
リッチテキスト
複数選択チェックボックス
行数可変の表など
- ・HTMLとCSSの知識があれば複雑なフィールドも作成できる

Free Layout Custom Fieldでブロックエディタを作る

(1)見出し(H2)

(10)画像、見出し、本文のセット(画像右)

(19)ボタン

(2)見出し(H3)

(11)Youtube

(20)プロフィール

(3)文章(p)

(12)会話

(21)商品紹介

(4)画像

(13)テーブル(会話)

(22)商品紹介(SP時カード型)

(5)画像2列

(14)テーブル(自由入力)

(23)店舗紹介

(6)画像3列

(15)テーブル(リスト)

(24)罫線

(7)リスト

(16)注釈

(25)カード

(8)番号付きリスト

(17)注釈(赤)

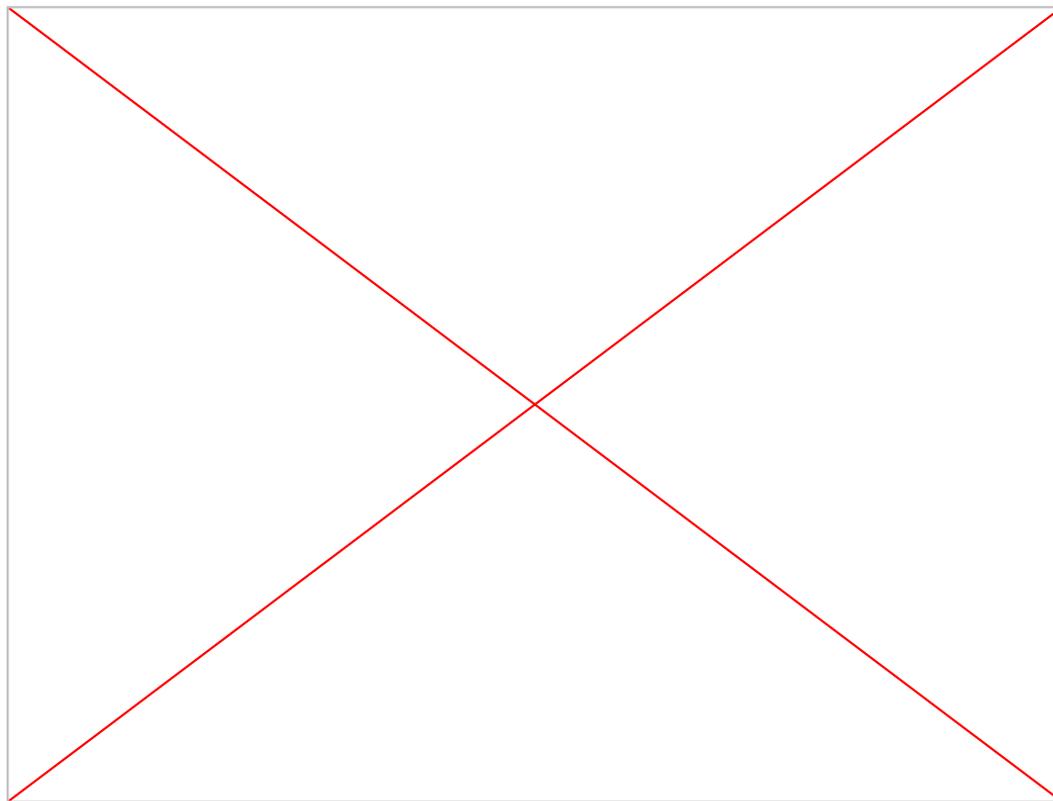
(26)目次

(9)画像、見出し、本文のセット(画像左)

(18)ボックス(グレー)

(27)フリーレイアウト枠

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る



FreeLayoutCustomFieldの仕組み

(1)フィールド(ユニット)の定義

(2)HTMLの定義

(3)CSSの定義

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

ユニット(フィールド)の定義

種類 [必須]
フリーレイアウト

必須?
 フィールドに値は必須ですか?

フィールドの定義

レイアウト用HTML

```
fields:  
  . . . (各フィールドの定義) . . .  
options:  
  . . . (multiple等の定義) . . .  
units:  
  names:  
    - name: ユニット名  
      label: ラベル  
    - name: ユニット名  
      label: ラベル  
  . . .
```

フィールドの定義では、各ユニットで使うフィールドを定義します。例えば、「見出し」「文章」「画像」の3つのフィールドを組み合わせたい場合、「見出し」「文章」「画像」の3つのフィールドを定義します。

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

レイアウト用HTMLの定義

種類 [必須]
フリーレイアウト

必須?
 フィールドに値は必須ですか?

フィールドの定義

レイアウト用HTML

```
<!-- {{unit:ユニット1}} -->  
ユニット1のレイアウトを行うHTML  
<!-- {{unit:ユニット2}} -->  
ユニット2のレイアウトを行うHTML  
...  
<!-- {{unit:ユニットn}} -->  
ユニット2のレイアウトを行うHTML
```

ユニットを使う場合、ユニットごとにフィールドのレイアウトが異なります。したがって、「レイアウト用HTML」の欄に、個々のユニットのHTMLを書きます。

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

レイアウト用CSSの定義

レイアウト用HTML

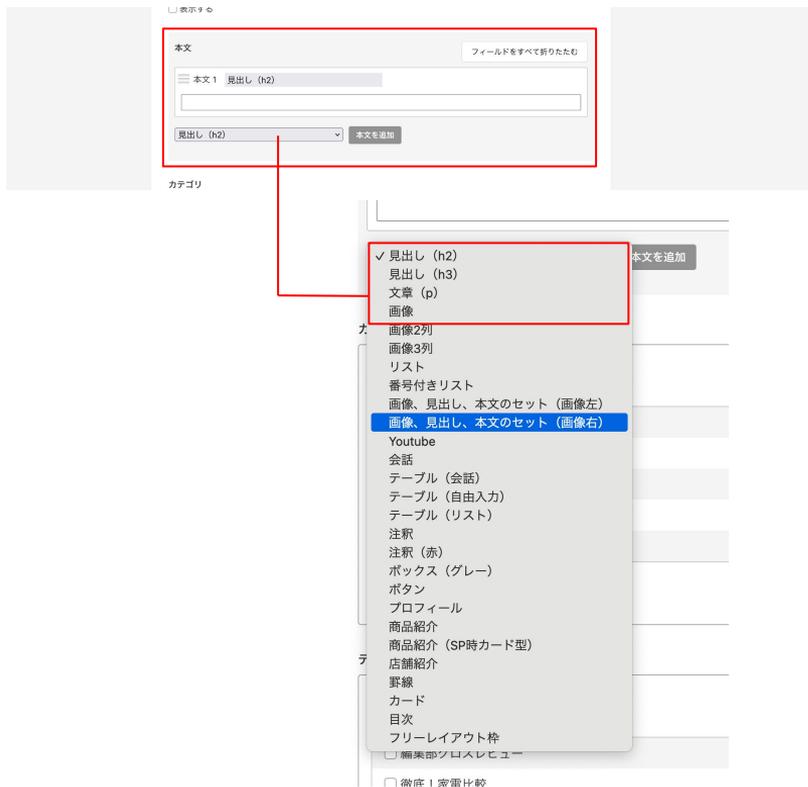
レイアウト用CSS

カスタムJavaScript(初期化用)

```
<style type="text/css">
{{field_id}} .クラス名 {
    入力要素に対するCSS
}
{{field_id}} .ラベルのクラス名 {
    入力要素のラベルに対するCSS
}
</style>
```

フィールドのデザインは、CSSで細かく調整することができます。このCSSは、コンテンツフィールドの設定ページで、「レイアウト用CSS」の欄に入力します。レイアウト用CSSの欄に入力した内容は、管理画面（のHTMLのヘッダ一部分）にそのまま組み込まれます。

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る



(1)見出し(h2):
登録すると<h2>でマークアップされます。

(2)見出し(h3):
登録すると<h3>でマークアップされます。

(3)文章(p):
登録すると<p>でマークアップされます。

(4)画像:
画像とキャプションを登録できます。



(5)画像2列:
横並び2列の画像とキャプションを登録できます。



Ultra Open Earbuds 概要

【H3】中見出しが入ります中見出しが入ります_22pt

ケーブルの取り回しを気にすることなく使用できるのが魅力の「ワイヤレスヘッドホン」。Bluetoothによるワイヤレス接続で快適に音楽が楽しめます。また、断線を心配する必要がないのもメリットです。そこで今回は、おすすめのワイヤレスヘッドホンを個別にご紹介。選び方や人気メーカーについても解説しているので、購入を検討している方はぜひチェックしてみてください。ケーブルの取り回しを気にすることなく使用できるのが魅力の「ワイヤレスヘッドホン」。Bluetoothによるワイヤレス接続で快適に音楽が楽しめます。また、断線を心配する必要がないのもメリットです。そこで今回は、おすすめのワイヤレスヘッドホンを個別にご紹介。選び方や人気メーカーについても解説しているので、購入を検討している方はぜひチェックしてみてください。



キャプションで説明します。キャプションで説明します。



キャプションで説明します。キャプションで説明します。



キャプションで説明します。キャプションで説明します。

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

ユニットの定義(具体例)

フィールドの定義

fields:

headline1:
type: text
label: 見出し

headline2:
type: text
label: 中見出し

sentence:
type: textarea
label: 文章

image_center:
type: image
label: 画像
image_center_caption:
type: text
label: キャプション

units:

names:

- name: headline1
label: 見出し (h2)

- name: headline2
label: 見出し (h3)

- name: sentence
label: 文章 (p)

- name: image_center
label: 画像

レイアウト用HTML

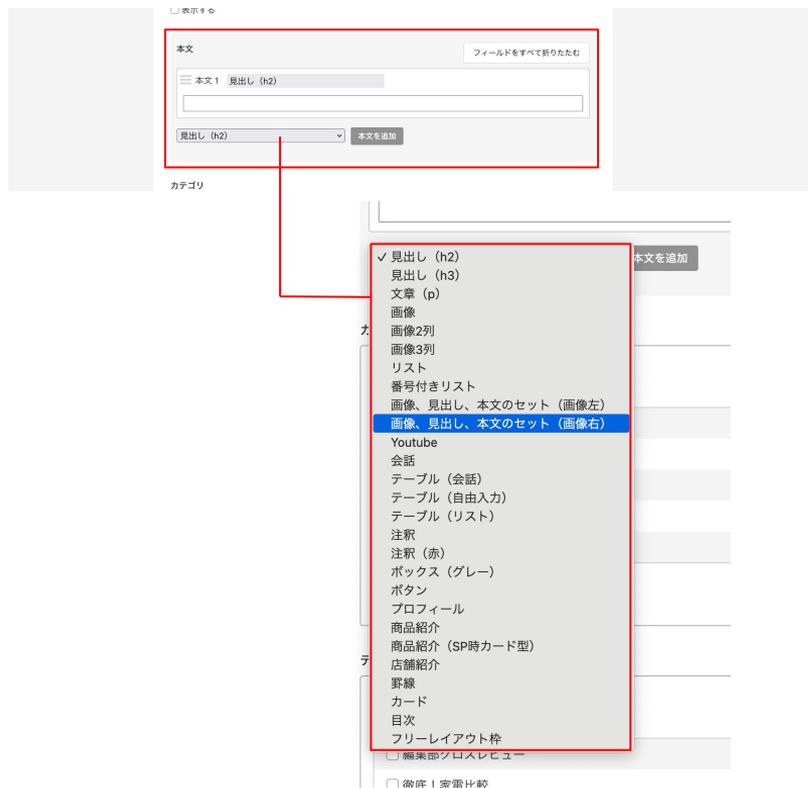
```
<!-- {{unit:headline1}} -->  
<div class="flcf-field">  
  <div class="flcf-field-body">{{headline1}}</div>  
</div>
```

```
<!-- {{unit:headline2}} -->  
<div class="flcf-field">  
  <div class="flcf-field-body">{{headline2}}</div>  
</div>
```

```
<!-- {{unit:sentence}} -->  
<div class="flcf-field">  
  <div class="flcf-field-body">{{sentence}}</div>  
</div>
```

```
<!-- {{unit:image_center}} -->  
<div class="flcf-field">  
  <div class="flcf-field-box-body-title">  
    <span class="flcf-field-box-label">画像</span>  
    {{image_center}}  
  </div>  
  <div class="flcf-field-box-body-title">  
    <span class="flcf-field-box-label">キャプション</span>  
    {{image_center_caption}}  
  </div>  
</div>
```

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る



(6) 画像3列:
横並び3列の画像とキャプションを登録できます。



(7) リスト:
登録するとでマークアップされます。



- 箇条書き内容が箇条書き内容が箇条書き内容が入ります。
- 箇条書き内容が箇条書き内容が箇条書き内容が入ります。箇条書き内容が入ります。
- 箇条書き箇条書き箇条書き内容が箇条書き内容が箇条書き内容が入ります。
- 箇条書き内容が箇条書き内容が箇条書き内容が入ります箇条書き内容が箇条書き内容が箇条書き内容が入ります

(8) 番号付きリスト:
登録するとでマークアップされます。



1. 箇条書き内容が箇条書き内容が箇条書き内容が入ります。
2. 箇条書き内容が箇条書き内容が箇条書き内容が入ります。箇条書き内容が入ります。
3. 箇条書き箇条書き箇条書き内容が箇条書き内容が箇条書き内容が入ります。
4. 箇条書き内容が箇条書き内容が箇条書き内容が入ります箇条書き内容が箇条書き内容が箇条書き内容が入ります

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

ユニットの定義(具体例)

フィールドの定義

```
list:  
  type: table  
  label: リスト  
  initRows: 2  
  updown: 1  
  columns:  
    - name: listtext  
      type: textarea  
      label: テキスト
```

```
orderedlist:  
  type: table  
  label: 番号付きリスト  
  initRows: 2  
  updown: 1
```

フィールドの定義

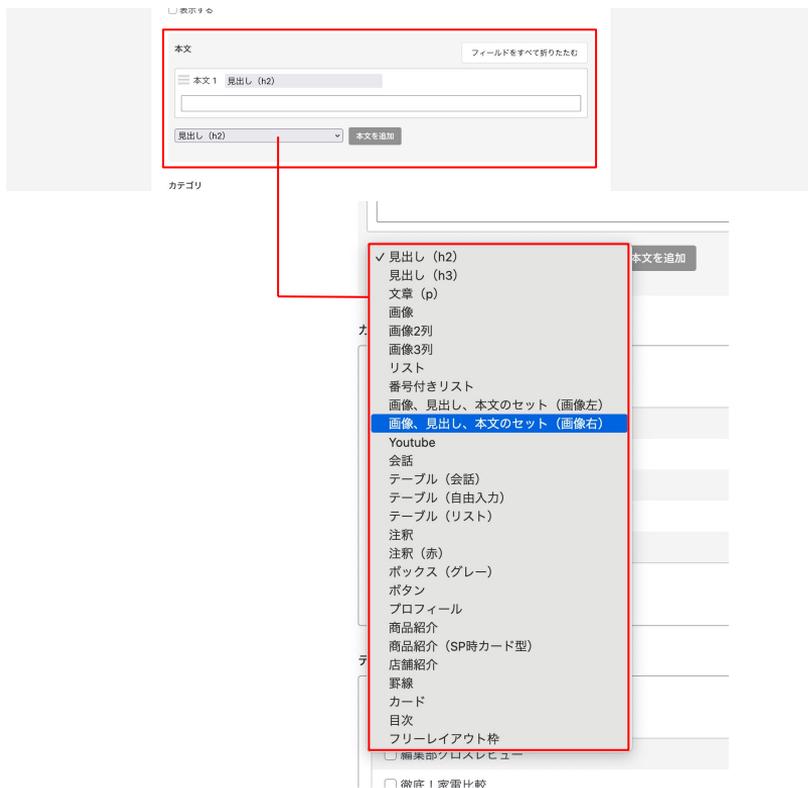
```
units:  
  names:  
    - name: headline1  
      label: 見出し (h2)  
    - name: headline2  
      label: 見出し (h3)  
    - name: sentence  
      label: 文章 (p)  
    - name: image_center  
      label: 画像  
    - name: image2column  
      label: 画像2列  
    - name: image3column  
      label: 画像3列  
    - name: list  
      label: リスト  
    - name: orderedlist  
      label: 番号付きリスト  
    - name: set_leftimage
```

レイアウト用HTML

```
<!-- {{unit:list}} -->  
<div class="flcf-field">  
  <div class="flcf-field-body">{{list}}</div>  
</div>
```

```
<!-- {{unit:orderedlist}} -->  
<div class="flcf-field">  
  <div class="flcf-field-body">{{orderedlist}}</div>  
</div>
```

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る



(9) 画像、見出し、本文のセット (画像左):
画像、見出し、本文のセットを登録できます。



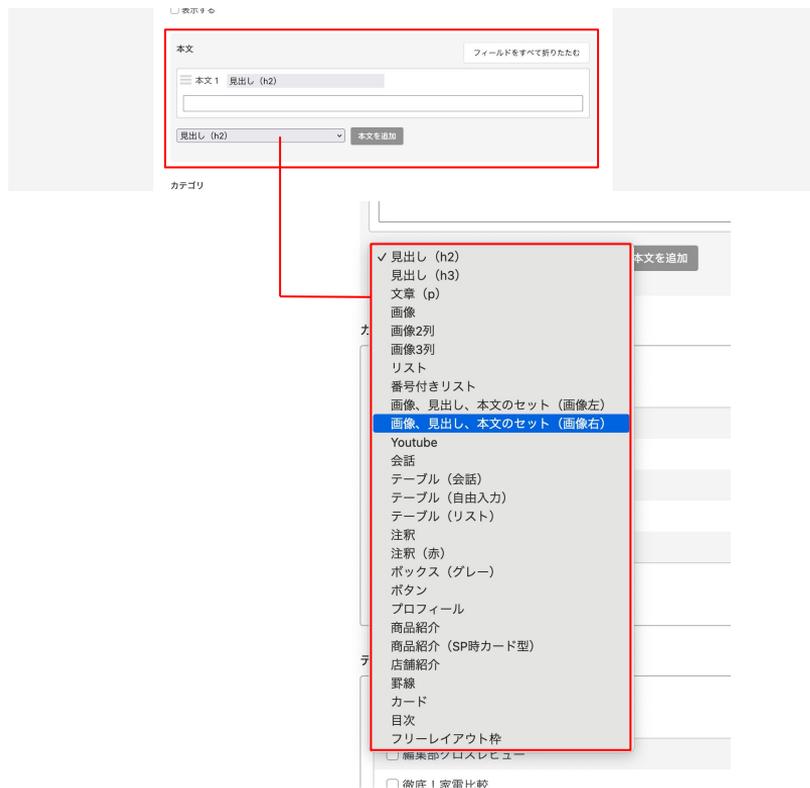
(10) 画像、見出し、本文のセット (画像右):
画像、見出し、本文のセットを登録できます。



(11) Youtube:
Youtubeの動画とキャプションを登録できます。



4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る



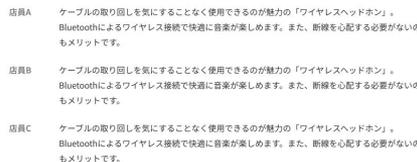
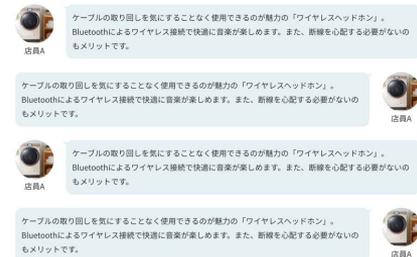
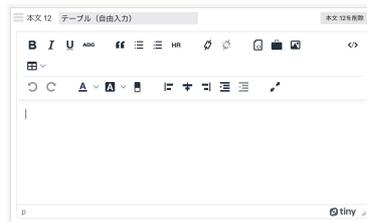
(12)会話:
画像、見出し、会話文のセットを登録できます。



(13)テーブル(会話):
テーブル型の見出し、文章のセットを登録できます。



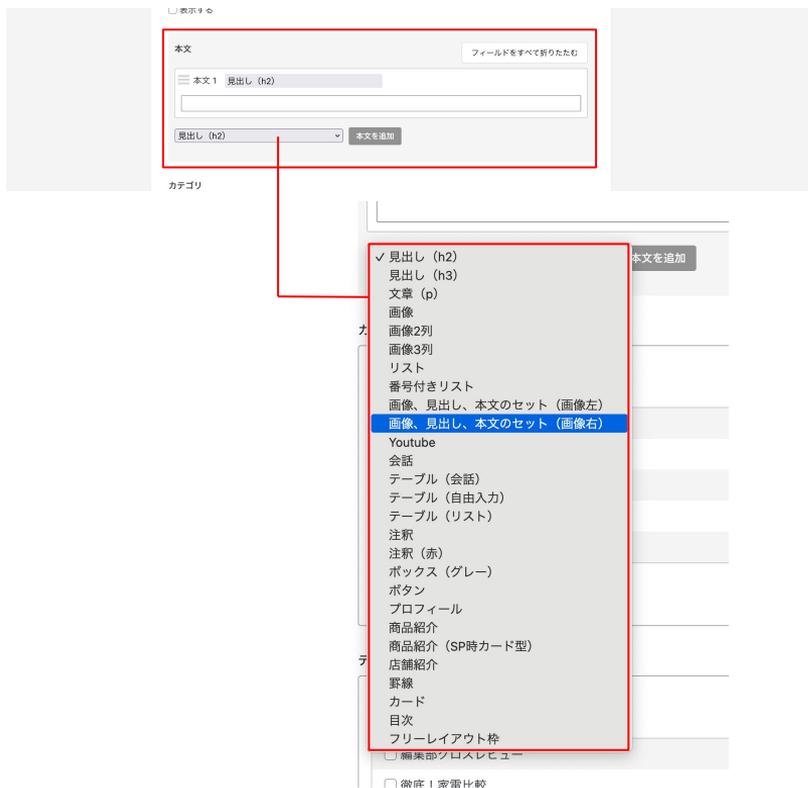
(14)テーブル(自由入力):
WYSIWYGで自由にテーブルを作成できます。



タイトル	タイトル	タイトル
説明説明説明	説明説明説明	説明説明説明

※このフィールドでテーブルを作成すると横長の表の時、スマホビューで自動でスクロールがつかます。

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る



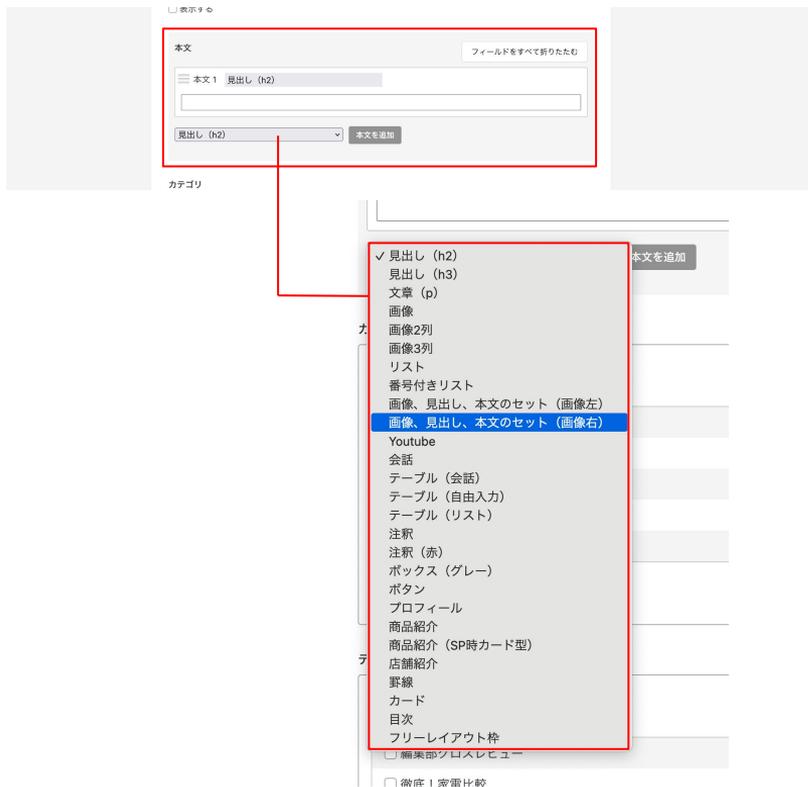
(18)ボックス(グレー):
見出し、テキストのセットを登録できます。



(19)ボタン:
テキスト、リンク先を登録し、ボタンの色を選択します。
フィールド上で「+」ボタンを押下すると追加でき、横に並びます。
3つ以上のボタンが横に並ぶことは想定されていません。



4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る



(20)プロフィール:(スクロールします)
画像、肩書、名前、紹介文、SNSリンクの
セットを登録できます。通常はコンテンツタイプ「著者」
を利用しますが、文中に著者を挿入したい場合に利用します。

本文 19 プロフィール 本文 19を削除

	画像	肩書き	名前	紹介文
1	選択			

✕

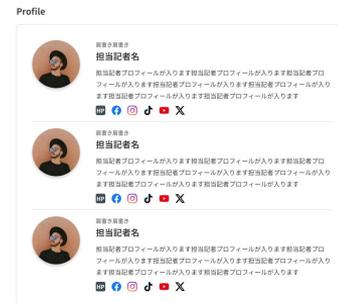
本文 19 プロフィール 本文 19を削除

	HP	facebook	instagram	tictok	youtube

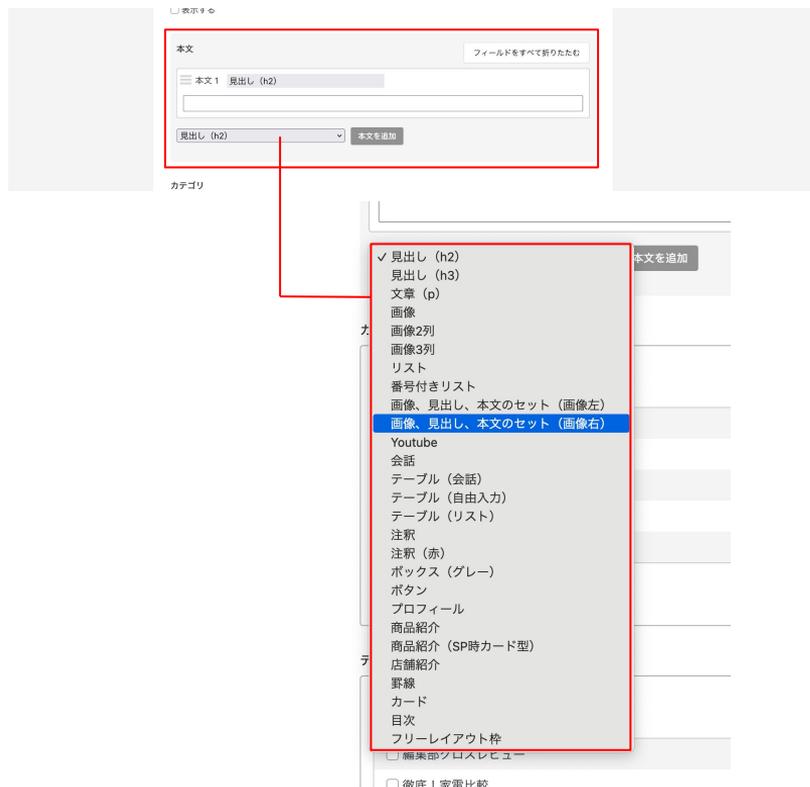
本文 19 プロフィール 本文 19を削除

	instagram	tictok	youtube	X

📄 📄 📄 📄



4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

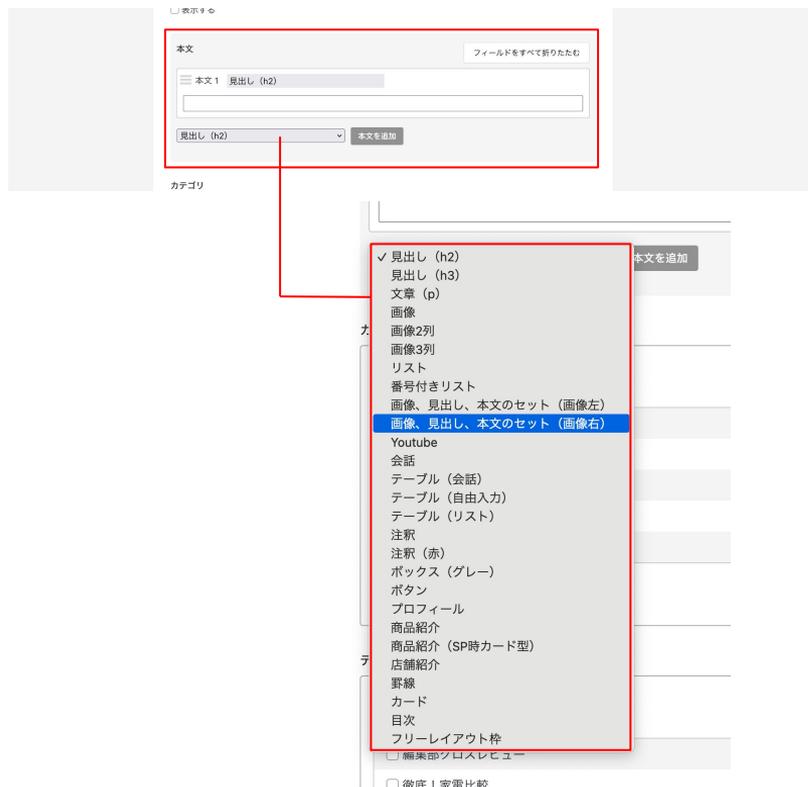


(21)商品紹介:(スクロールします)

画像、商品名、紹介文、リンク先のセットを登録できます。



4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

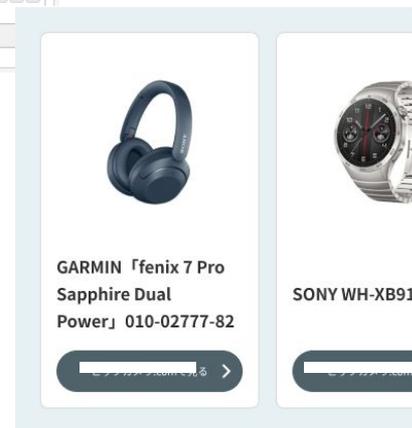


(22)商品紹介:(スクロールします)

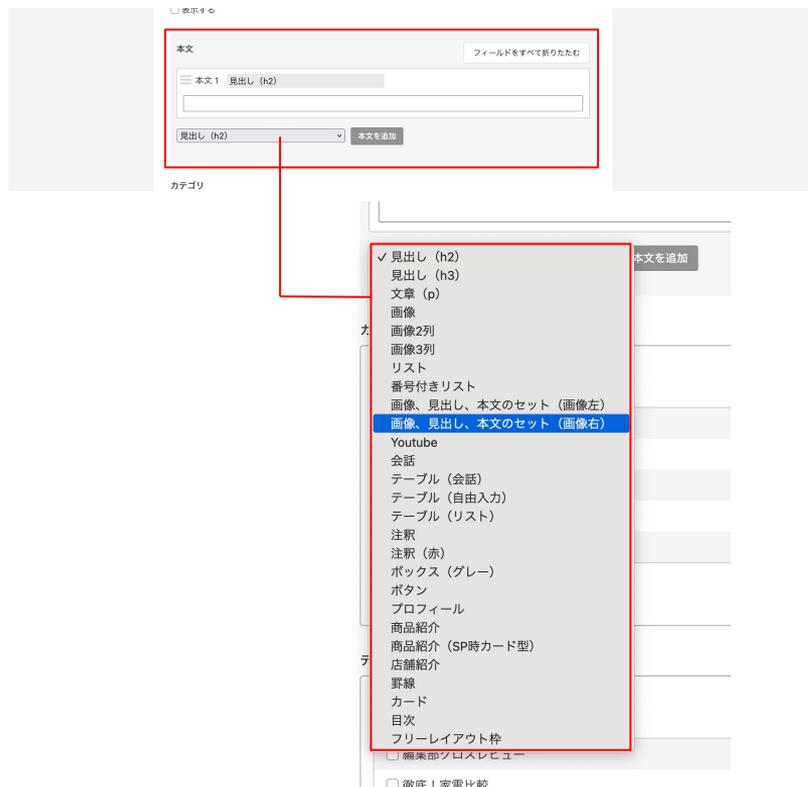
画像、商品名、紹介文、リンク先のセットを登録できます。

画像	商品名	紹介文	リンク先
1 選択	<input type="text"/>	<input type="text"/>	<input type="text"/>

商品名	紹介文	リンク先	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>



4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

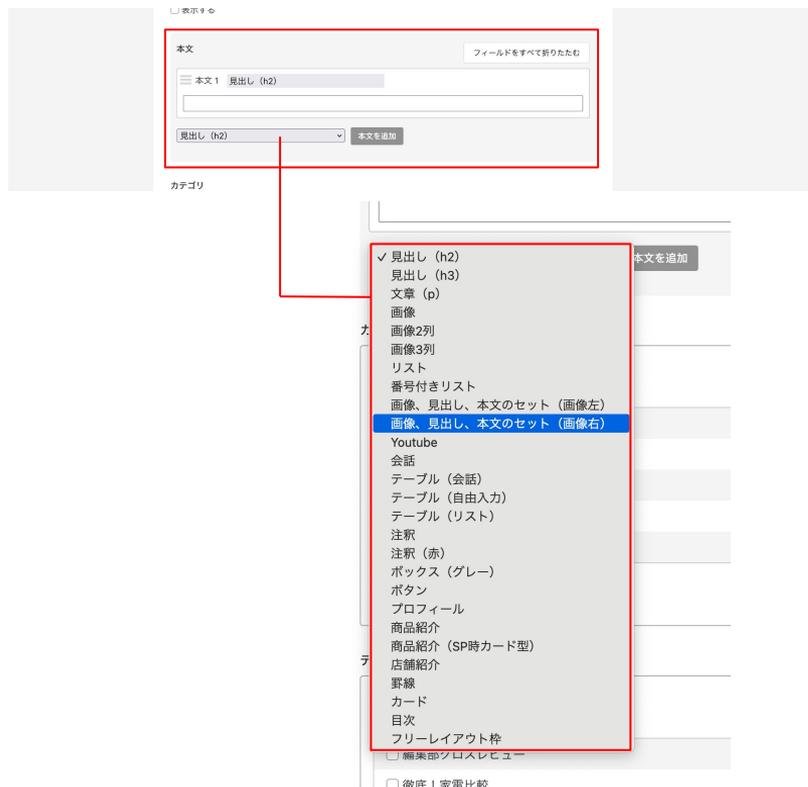


(23) 店舗紹介: (スクロールします)

画像、ショップ名、紹介文、所在地、電話番号、営業時間、アクセス、詳細リンクのセットを登録できます。通常はコンテンツタイプ「店舗」を利用しますが、文中に表示が必要な場合に利用します。



4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る



(24)罫線:
罫線を追加します。



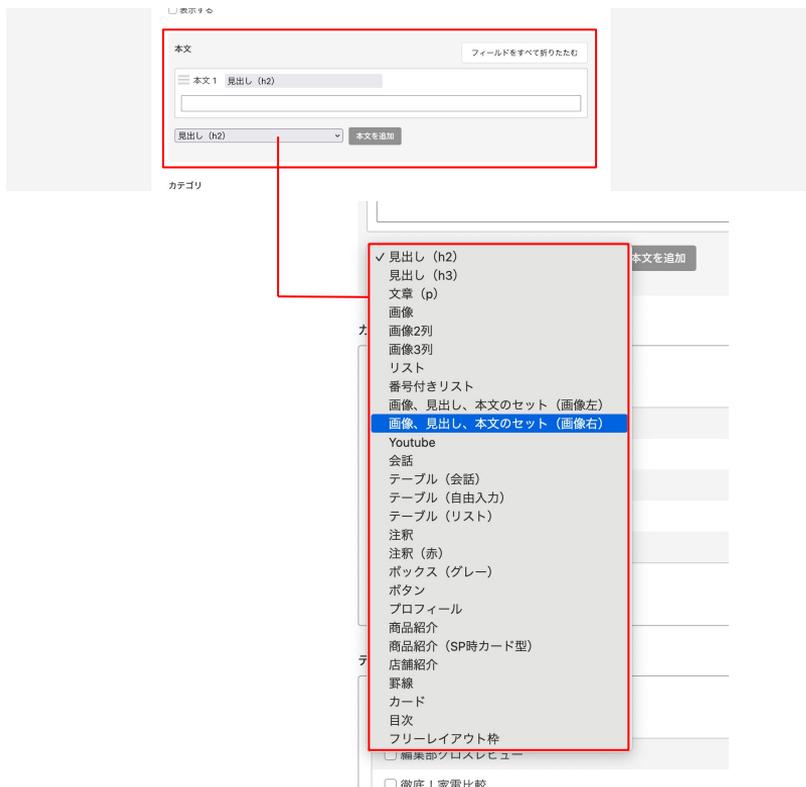
(25)カード:
画像、リンク先、タイトル、説明文を登録できます。



ブログカードブログタイトルが入ります

ブログのディスクリプション説明が入ります。ブログのディスクリプション説明が入ります。ブログのディスクリプション説明が入ります。

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る



(26) 目次:

目次を追加します。

目次の内容は本文中のh2見出しの内容が自動で入力され、リンクされます。



目次

1. 簡条書き内容が簡条書き内容が簡条書き内容が入ります。
2. 簡条書き内容が簡条書き内容が簡条書き内容が入ります。簡条書き内容が入ります。
3. 簡条書き簡条書き簡条書き内容が簡条書き内容が簡条書き内容が入ります。
4. 簡条書き内容が簡条書き内容が簡条書き内容が入ります簡条書き内容が簡条書き内容が入ります簡条書き内容が簡条書き内容が入ります。

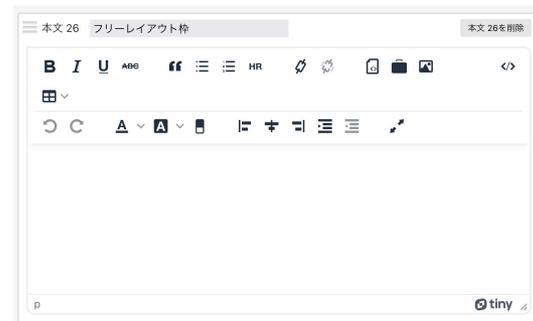
(27) フリーレイアウト枠:

自由にhtmlを記述できます。

WYSIWYGを利用することも可能です。

用意されたコンポーネント以外のものを

作成する場合に利用します。



**フィールドの定義が終わったら、
テンプレートに読み込むだけ！**

組み合わせは自由！

思い通りのフィールドの組み合わせを自由に作れます

コンテンツタイプでの利用は

FreeLayoutContentField

をご利用ください！

4.構築時のアイデア実例:FLCFでブロックエディタライクな管理画面を作る

以上

ご清聴ありがとうございました。