



投稿画面本文フィールド(TinyMCE)のスタイルをカスタマイズする方法

最終更新日：2011年4月9日 shortlink：http://modx.jp/?id=529

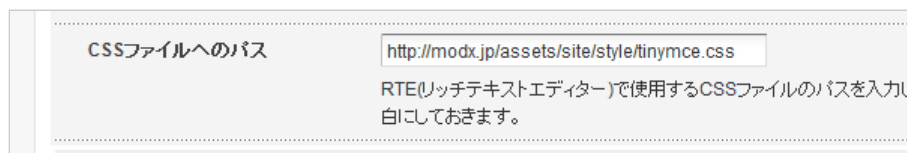
MODXの投稿画面の本文フィールドはTinyMCEが組み込まれており、見た目を反映したワープロ感覚の記事編集ができます。このフィールドのデザインをサイトのイメージに合わせることができます。見出しや段落のフォント色、領域の横幅、背景色など。実際のサイトと同じイメージで編集できるため分かりやすいです。



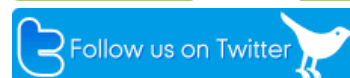
当サイトの投稿画面は上記のようになっています。見出しのスタイルや背景イメージ、本文領域の横幅などが実際のサイトの表示と揃っています。背景イメージは、スクリーンショットを加工して一枚の背景画像にしています。

設定方法

グローバル設定の「インターフェースとその他の機能」タブを開きます。



「CSSファイルへのパス」の項目で、投稿画面の本文フィールド(textarea部分)に適用するスタイルシートのパスを設定します。基本的な設定はこれだけです。投稿画面を開いて確認してみてください。TinyMCEのスタイルシート指定は頑固にキャッシュを読みに行くことがあるので、ブラウザのキャッシュをいったん削除する必要があるかもしれません。



PR

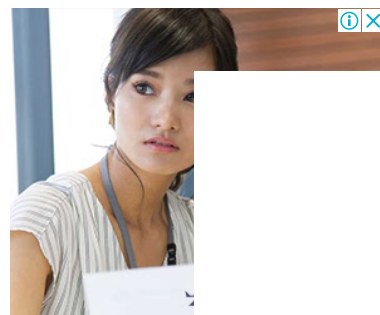
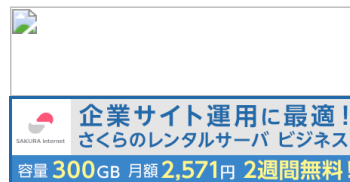
Webもメールもデータベースも
レンタルサーバーで

KAGOYA
Internet Routing

MODX導入
その他、Webアプリケーション
開発等お任せ下さい。
RSP All for Solution Provider

Most Promising Open Source CMS
[Winner]
CMS AWARDS '07
MODx Most Promising Open Source CMS

スポンサー募集中



応用設定

管理画面スタイルシートとのバッティングを避けるためには、固有のセクタを設定する必要があります。

パラメータ	値
body_id	"main"
valid_elements	"*[*]"

TinyMCEプラグインの設定タブを開き、上記のように設定します。body_idというのはTinyMCE本体が持つオプションで、本文領域textareaをbodyと見立ててIDを与えることができます。これにより、**textarea#xxx** となるべきところを **body#xxx** としてスタイルシートからアクセスでき、管理画面スタイルシートとのバッティングを避けることができます。上記の指定では **body#main** となります。

classを指定するbody_classというオプションもあるので、必要に応じてid指定かclass指定かを選ぶことができます。

背景画像
background-image: url(mce_bg.png);

横幅
width: 670px;

当サイトでは上記のような指定になっています。コンテンツを入力できるのは横幅670pxの部分のみです。

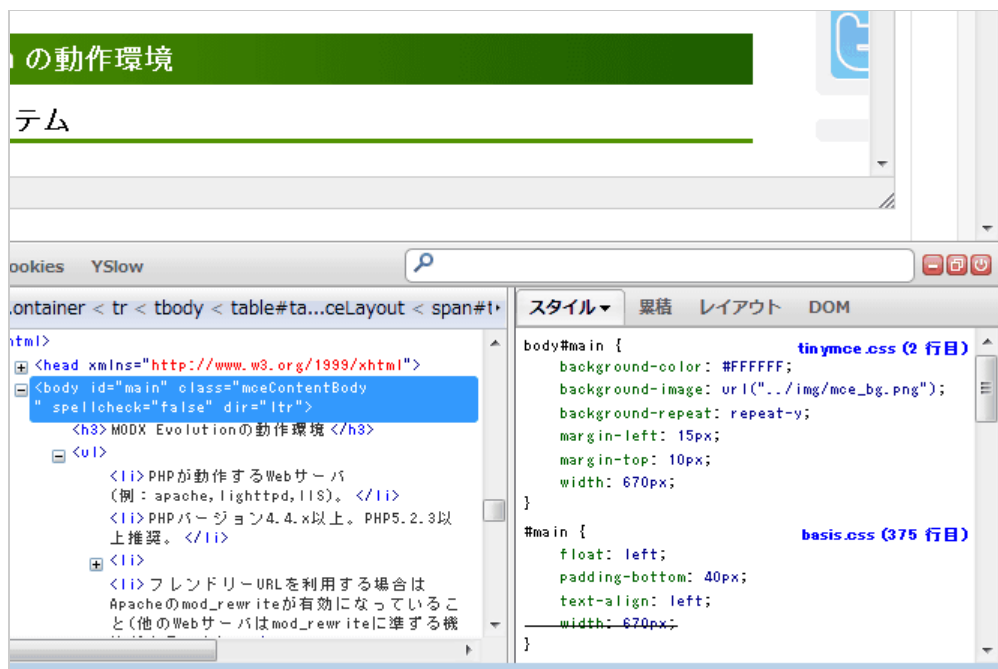


横幅を意識しやすいように、サイトイメージの背景画像を作って配置しています。スクリーンショットを適当に切り抜いて、薄い色合いに加工。サイドバーが左側にあるデザインのサイトでも、右側にあるものとして背景画像を作るとよいでしょう。

問題は、p要素とかh2要素とかは分かるとしても、横幅や背景画像を設定するセレクトは何？ということなのですが、textareaではありません。さっきも少し書きましたが、body要素になります。下記の説明をお読みください。

カスタマイズのコツ

TinyMCEのスタイルカスタマイズは、htmlのソースを見て確認できるわけではないので、直感的に難しく感じる人もいます。こういう場合に便利なツールがあります。



Firebugを用いると構造を把握しやすいです。値を一時的に書き換えて違いを確認することもできます。

上記の画像で見られるとおり、TinyMCEのデフォルト状態ではmceContentBodyというclassが与えられており(セレクトがbodyになっていることに注目)、少し以前までのMODXではこれが用いられてました。この指定を外すためにはmceContentBodyというclass名を覚えておく必要があり、使い勝手が悪いため現在のMODXでは使っていません。現在はh2・h3・pなど要素単体でセレクトを指定しているので、簡単に上書きできます。

投稿画面専用のスタイルシートを作るのではなく、サイトに用いているスタイルシートをそのまま設定することも可能です。そのためには、スタイルシートの構造がコンテンツ領域とテンプレート領域の区別を意識した作りになっている必要があります。たとえばコンテンツ

領域が `div#main` の中に収まる形でスタイルシートが整理されていれば、投稿画面は `body#main` であると見立ててデザインをコントロールできます。

```
body#main {.....}
#main p {.....}
#main h1 {.....}
#main h2 {.....}
#main h3 {.....}
```

このようなスタイルシート構成になっていれば、投稿画面と共用できます。

CMS運用を意識したサイト構成としては、サイトで用いるスタイルシートを `template.css` と `content.css` の二つに分けると便利かもしれません。管理画面では `content.css` をサイトと共用します。

今回はMODX自体というよりTinyMCEの使いこなしに関する話題でした。他のCMSでも応用できるかもしれません。TinyMCEは今回紹介した `body_id` 以外にも 多くのオプション を備えており、工夫次第でカスタマイズの幅を広げることができます。さまざまなパターンの 実装サンプル がありますので参考にしてみてください。



MODX Japan

ニュース
ダウンロード
MODX日本公式フォーラム

MODXについて

Revolution
Evolution
ドキュメント (Evo)

導入ガイド

FAQ(よくある質問)
MODXデモサイト
MODXが稼働するレンタルサーバー

MODX Japanについて

このサイトについて
活動の目的と内容
運営への協力について
貢献・支援・寄付
協賛企業



Copyright © 2010 - 2019 MODX JAPAN All rights reserved. [stat : 1.07 MB 0.0270 s 4クエリ 15ファイル]
Powered by MODX. Sponsord by KAGOYA Japan & ASP@Akihabara. Design by STUDIO DOTLINE.