

本書「Movable Type デザインカスタマイズブック MT4.2対応 ～クリエイターのための Value Design～」213,214ページに記載の内容に誤りがありました。

謹んでお詫びさせていただくとともに、下記のように訂正情報を追加いたします。

Movable Type デザインカスタマイズブック MT4.2 対応 ～クリエイターのための Value Design～

Chapter4 プラグイン・JavaScript による機能拡張 08 Ajax 検索を用いたサイト内検索機能を導入する

テンプレートE

210 ページ～215 ページ

当該の項は、Ajax の仕組みを利用して、MT の標準機能とは異なる「サイト内検索」を可能にする、というものです。

本書 210 ページから 213 ページ上部までの間で、検索用の XML のサイトマップを独自に作っておくという趣旨で、XML データの文字コードを UTF8 で生成されるようにする「Ogawa::Code MT-I18N プラグイン」を利用する方法を解説しています。

213 ページ中部より、上記の UTF-8 化された XML データを、JavaScript で検索し結果を表示する仕組みを利用する方法を解説しています。この段落は、Ajax と Json というテクノロジーを用いるという趣旨で、「世界中の 1% の人々へ blog_ajax_json_search」で公開されている JavaScript の改造方法を紹介していますが、本著に掲載したコードに誤りがありました。改めて、お詫びさせていただくとともに、以後は 213,214 ページに掲載しているコードの訂正となります。

213 ページ中部、blog_ajax_json_search.txt の拡張子を js に変更し、エディターで開いて、コードを編集していきます。

MEMO

お問い合わせの中で、コードをワードプロセッサモードで開かれて、表示行の違いに戸惑ってしまう方がいらっしゃるように感じました。JavaScript はかならず、エディターモードで開いて、コード 1 行に対して表示される数字をもとに、編集行を確認してください。

まずは、blog_ajax_json_search.js の 147 行目を修正します。

<修正前>

```
html += '<li><a href="' + result_data[i].link + '#search_word=' + search_keyword + '">' +  
highlight( result_data[i].title, keywords ) + '</a><br />' +  
highlight( trunc( result_data[i].body, keywords[0] ), keywords ) + '</li>';
```

<修正後>

```
html += '<li><a href="' + result_data[i].link + '">' + highlight( result_data[i].title, keywords )  
+ '</a><br />' + highlight( trunc( result_data[i].body, keywords[0] ), keywords ) + '</li>';
```

続いて、149 行目を修正します。

<修正前>

```
html += '</ol></div></div><div class="entry-body-bottom"></div><p class="posted"  
style="text-align:right; font-size: smaller;"><span class="post-footers"></span><span  
class="separator"></span><span class="bo_so_copyright"><a href="#" title="Go Back page"  
onclick="location.reload(true); return false;" onkeypress="location.reload(true); return  
false;">[Go Back Page]</a><a href="/" title="Go Back Index" >[Go Back  
Index]</a></span></p></div><div class="entry-bottom"></div><div  
class="date-footer"></div>;
```

<修正後>

```
html += '</ol></div></div><div class="entry-body-bottom"></div><p class="posted"  
style="text-align:right; font-size: smaller;"><span class="post-footers"></span><span  
class="separator"></span><span class="bo_so_copyright"><a href="#" title="Go Back page"  
onclick="location.reload(true); return false;" onkeypress="location.reload(true); return  
false;">[Go Back Page]</a><a href="ブログのトップページの絶対パス" title="Go Back Index"  
>[Go Back Index]</a></span></p></div><div class="entry-bottom"></div><div  
class="date-footer"></div>;
```

続いて、155 行目を修正します。

<修正前>

```
html += '</ol></div></div><div class="entry-body-bottom"></div><p class="posted" style="text-align:right; font-size: smaller;"><span class="post-footers"></span><span class="separator"></span><span class="bo_so_copyright"><a href="#" title="Go Back page" onclick="location.reload(true); return false;" onkeypress="location.reload(true); return false;">[Go Back Page]</a><a href="/" title="Go Back Index" >[Go Back Index]</a></span></p></div><div class="entry-bottom"></div><div class="date-footer"></div>';
```

<修正後>

```
html += '</ol></div></div><div class="entry-body-bottom"></div><p class="posted" style="text-align:right; font-size: smaller;"><span class="post-footers"></span><span class="separator"></span><span class="bo_so_copyright"><a href="#" title="Go Back page" onclick="location.reload(true); return false;" onkeypress="location.reload(true); return false;">[Go Back Page]</a><a href="ブログのトップページの絶対パス" title="Go Back Index" >[Go Back Index]</a></span></p></div><div class="entry-bottom"></div><div class="date-footer"></div>';
```

最後に、187 行目を修正します。

<修正前>

```
return text.replace( new RegExp( '(' + keyword + ')', "ig"), '<span style="background-color: #FFCC33;">' + "$1" + '</span>' );
```

<修正後>

```
return text.replace( new RegExp( '(' + keyword + ')', "ig"), '<span class="highlight">' + "$1" + '</span>' );
```

訂正は、以上です。

次の段落では、XML データと Json データのやりとりを行うために、prototype.js を利用することを示しています。これについては、本書の 200-205 ページの「Chapter4 08 What's New エリアにタブインターフェイスを設置する」という項で、非同期通信の Ajax 用ライブラリファイル「Prototype JavaScript framework prototype.js」を解説していますので、ここでおさらいしてください。

扱いとしては、「blog_ajax_json_search.js」と「prototype.js」を、FTP クライアントでサーバーの出力ディレクトリにある「js」にアップロードしてください。

215 ページでは、機能を整え、最後にフォームを形成するテンプレートのカスタマイズ方法を解説しています。

出力されるページ「Ajax 検索」のテンプレートは、ウィジェット化してあります。本書 215 ページのコードで、赤字でマークされている部分を確認してください。blog_ajax_json_search.js は、ここで読み込まれています。

prototype.js は、テンプレートモジュール「ヘッダー」から読み込まれています。本書 202 ページのコードで、赤字でマークされている部分も確認してください。

ここまで整えば、検索結果を表示させるエリアに、空タグとなるクラス名を貼り付ければいいだけです。テンプレート E では、インデックステンプレート「メインページ」に<div class="result"></div>というタグを挿入させ、blog_ajax_json_search.js の65行目にある result を読み込んで表示させています。

検索という機能は、Web サイトのアクセシビリティを支える要ですから、使い勝手にはこだわりたいところです。簡潔な Ajax 検索を設置する Web サイトは 1 つの優れた解となるでしょう。