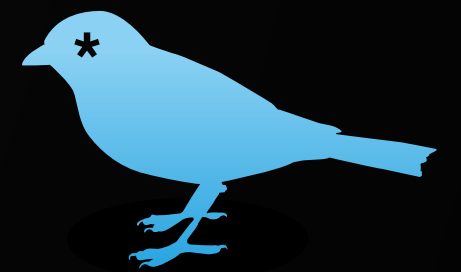
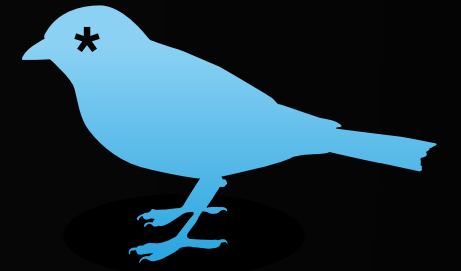


静的 HTML を CMS 用データにコンバート

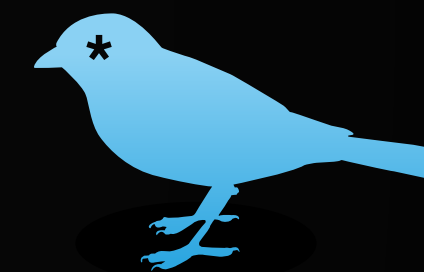


きっかけ



某大学サイトのリニューアル案件

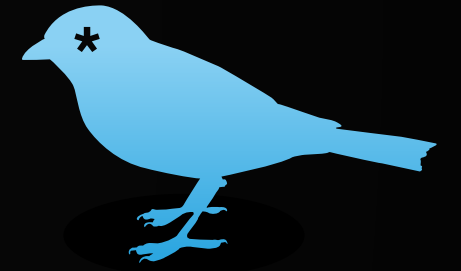
- 大学担当者でも編集できるようにしたい → CMS か？
- HTML のファイル数が約 2000 → 手作業でコピペしたくない



基本ルール

- ・コンテンツ量が多く、複数のブログに分割する必要があるので CMS は MovableType に
- ・既存ツールで出来るところに対応しない
今回のリニューアルではデザイン、ディレクトリー構造も変更するため、個々の HTML の編集やディレクトリーの移動などは DW に対応する
- ・Perl で作る（手持ちの言語の中ではテキスト処理が一番楽なので）

作成したツール

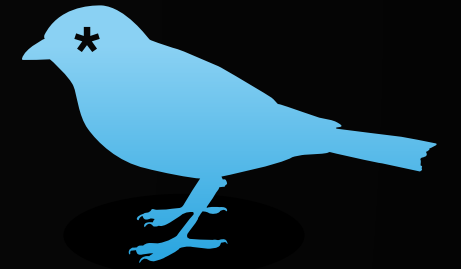


1. 静的 HTML から編集用 HTMLML に変換するツール

2. 編集用 HTML から MT 用バックアップデータに変換するツール

※インポート / エクスポート機能ではカテゴリーの階層が再現できない

動作環境



- Perl 5.8 以上 (ローカル側)

Encode::InCharset::shiftjis (IBM 拡張の Shift-JIS に対応するために利用)

Unicode::Japanese (IBM 拡張の Shift-JIS も上手い事変換してくれる)

HTML::Entities (記号類をエンティティ化してくれる)

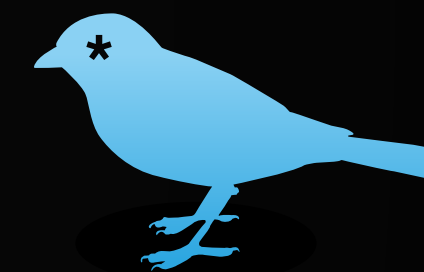
HTML::TreeBuilder (壊れた HTML も無理矢理に XML ツリーみたいにアクセスできる)

HTML::TreeBuilder::XPath (HTML で XPath を使える！)

※通常は以下のモジュール類はインストールされていないので CPAN なり、 PPM なりでインストールする

CPAN でのインストールには VisualStudio (win) や Xcode(mac) などのコンパイラ出来る環境が必要です。

- MT4.0 以上 (サーバ側)



手順 1：静的 HTML から編集用 HTMLML に変換

- 変換するコンテンツ一式を htdocs フォルダにコピー
- change2edit.pl をダブルクリックして、しばし待ちます
(mac の場合は mi エディタで「実行」)
- htdocs 内の HTML が編集用 HTML に変換されます

ディレクトリーの移動やデザインやタイトルなど HTML 編集、その他リンクチェック等は DW を使って対応。

ファイルパスもルート相対パスに変換しておく (DW 拡張で対応 :

<http://mypocket.blogspot.com/2007/07/dreamweaver.html>)

ページタイトルはカテゴリー名としても使用する場合がありますので、正しいページ名にします

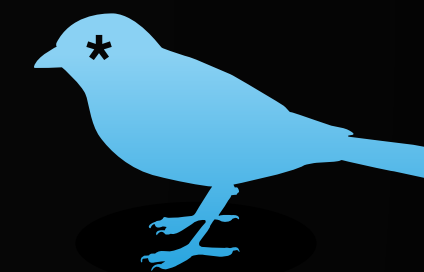
(手順 2 でディレクトリー構造をカテゴリーに変換します)。



手順 2：編集用 HTML から MT バックアップデータに変換

- ・ 公開時に使用する MT 上にてブログを新規作成し、バックアップデータを作成
- ・ 作成したバックアップデータのファイル名を restore.xml として、change2restore.pl と同一階層にコピー。
- ・ 変換するコンテンツ一式を htdocs フォルダにコピー
- ・ change2restore.pl をダブルクリックして、しばし待ちます (mac の場合は mi エディタで「実行」)
- ・ restore.xml にデータが追記されるので、MT 上にて復元

※復元する際は FTP など import フォルダにバックアップデータを格納してから復元するようにしてください。
ブラウザ上でファイルを指定する方法ではファイルサイズが大きすぎて、処理できない場合があります。



技術的な補足説明

- 静的 HTML から編集用 HTMLML に変換

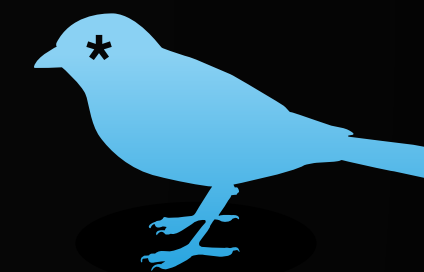
ページタイトルとコンテンツエリアを切り出して、特定のタグでマークアップ。

「ページタイトル」、「コンテンツエリア」の指定は原則、XPath で行う

- 編集用 HTML から MT バックアップデータに変換

MT のバックアップデータにはアカウント情報なども含まれているため、空ブログのバックアップデータに追記する形でデータを作成。

ディレクトリーはカテゴリーとして、親子関係を維持しつつ変換。カテゴリー名は各ディレクトリーのトップページっぽい HTML の見出しを取得。



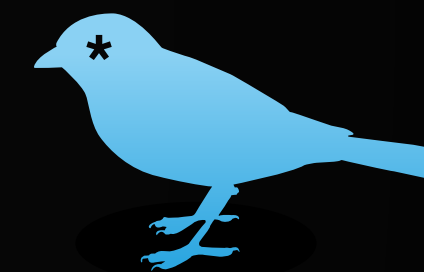
MT バックアップデータのフォーマット 1

○基本フォーマット

```
<movabletype>  
  <author />  
  <blog />  
  <template />  
  <entry />  
  <category />  
  <placement />  
</movabletype>
```

○<blog> の属性 : ブログ

id : ブログに紐付くテンプレート、エントリーなどで使用



MT バックアップデータのフォーマット 2

○<entry> の属性：個々のエントリー

id : 連番 (で可。エントリーとの紐付けの時に必要)

atom_id : tag の後ろの文字列は無視しても可。最後の数字は id と合わせる

blog_id : 共通 (<blog> タグの ID を使用)

title : ページタイトル

<text> : コンテンツ (HTML エンティティ化すること)

○<category> の属性：個々のカテゴリー

id : 連番 (で可。エントリーとの紐付けの時に必要)

blog_id : 共通 (<blog> タグの ID を使用)

basename : カテゴリー KEY (フォルダ名と一致させる)

label : カテゴリー名 (表示用)

parent : 親カテゴリーの ID (トップカテゴリーの場合は 0)

○<placement> の属性：エントリーとカテゴリー紐付け

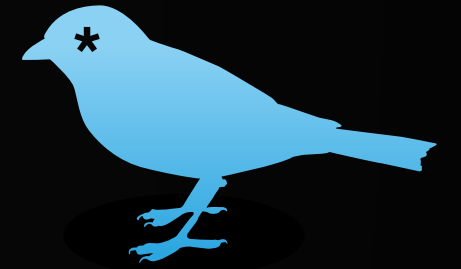
id : 連番 (他では使わないはずなので、単純に連番)

blog_id : 共通 (<blog> タグの ID を使用)

category_id : 紐付けたいカテゴリーの ID

entry_id : 紐付けたいエントリーの ID

DW Tips



あんまりファイル数が多いサイトを編集すると DW がファイルを壊します。4 桁後半辺りを超えると顕著。特にリンクエラーの結果画面からリンク先を編集すると高確率で壊れます。

同じリンク先をまとめて修正できるので便利ですが、全く関係ないファイルにリンクしてたりします (HTML にリンクしてなきゃいけないのに、画像にリンクしてたり)。

サイト全体に影響する編集は 1 ステップ毎にサイトキャッシュを作り直した方がいいです。

若干、貧弱ですが正規表現による置換も便利です。後方参照 (\$1、\$2...) も使えます。DW のヘルプには日付の書式変更の例が載ってます。

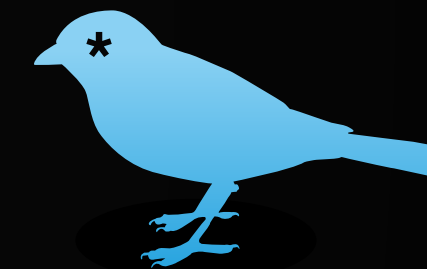
例：

id が「○×Area」となっているものを「○×Col」に置換 (○× は半角英数字)

検索 : id="(\\w+)Area"

置換 : id="\$1Col"

XPath



XML ファイル内の特定の箇所を指定するもの。HTML も XML と同じマークアップ言語ですので、タグの対応が取れていれば使えます。CSS のセレクタの親戚みたいなものです。

例：

ルートノードから全指定：html タグ直下の body タグ直下にある 4 個目の div タグ（なぜか 1 から始まる）

```
/html/body/div[4]
```

属性による指定：class が hoge な div タグ全部

```
//div[@class="hoge"]
```

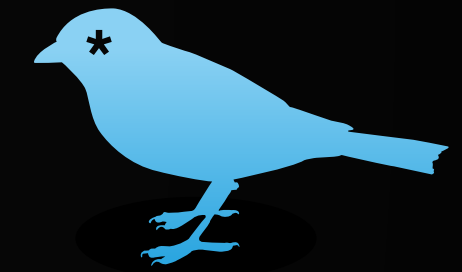
参考：

JavaScript-XPath をリリースしました！さあ、あなたも XPath を使おう！

<http://d.hatena.ne.jp/amachang/20071112/1194856493>

JQuery でも似たようなことが出来ますが、上記ページのライブラリを使用すると JavaScript で本格的に XPath が使えるので、ゴニョゴニョしたい人はぜひ。HTML に id や class を定義することなく、HTML 内の任意の要素にアクセス出来るし、RSS 読み込んでゴニョゴニョしたり、サーバサイドの言語なら適当にスクレイピングしてまとめサイト自動作成できたりと大活躍です。

その他



「HTML::ExtractContent」

コンテンツらしい部分を勝手に判別してくれる Perl モジュールですが、リンク集ページがコンテンツのないページ扱いされてしまうなど今回の案件では使いづらいので不採用。

多分、ブログサイトはきれいに取り出せそう。リンクの密集度や句読点などで判別するらしいです。

ヒドイ HTML

標準のテンプレートになっていない HTML（独自のナビゲーションがあったり）や Word や Excel、PPT で作成した HTML も多々存在したため、XPath だけでは取得できないものもありました。この場合、body タグの中身を全部持ってきてます。

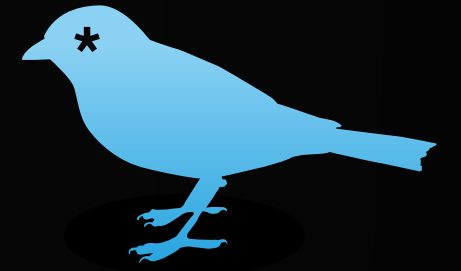
ただし、オフィス系で作成したコンテンツは IE に特化した HTML になっている場合が多いため、使い物にならない事が多い。ルビとかがヒドイ。

コンテンツの整理

この手の大規模サイトのリニューアルのキモはコンテンツの整理です。

5年前のオープンキャンパスの告知ページが今でも残っていたり、「テスト」とだけ書かれた HTML があったり、ヒドいアリサマでした。

現状



今のところ、某大学サイト→MT 専用ツールです。
まだ、実運用もしてません。

見出しとコンテンツエリアを抽出する指定を GUI とかで変更できるようにすれば、幅広く使えると思います。後は複数の CMS に対応できるようにすればもっと幅広く使えると思います。