

WordPress による人文情報学科
Web サイトの作成について

0948011 今倉純貴

目 次

1	はじめに	1
1	1 動機	1
2	2 WordPress とは	2
3	3 対象者について	3
4	4 類似物について	4
2	制作物について	4
1	1 人文情報学科オリジナル Web サイトの設計	4
3	制作過程について	5
1	1 WordPress の導入	5
2	2 外観の作成	7
3	3 Web サイトの記事の作成	11
4	Wordpress のバージョンアップとサーバー移行	13
1	1 バージョンアップ	13
2	2 サーバーの移行	16
5	アンケートについて	19
1	1 アンケートの内容	19
2	2 アンケート結果について	19
6	終わりに	21
1	1 自己評価	21

1 はじめに

(1) 動機

WordPress で大谷大学人文情報学科のオリジナル Web サイトを作成しようと考えたのは、今現在あるオリジナル Web サイトがある特定の人物しか編集・更新することができないことから誰でも手軽に編集・更新することができるようになればもっと頻繁に編集・更新ができ、この Web サイトが活性化するのではないかと思い立ったからである。

福田ゼミのテーマは「人の役に立つものをつくる」というものが軸になっている。入学した当初の私は、Web 関連について全く無知であり、興味があると言う程度であった。しかし、人文情報学科の演習や講義を受けていき、「4年間この学科で学ぶことを存分に活かして、自分の力で何かを作りたい」と思うようになった。そして、2回生のゼミを決める時福田先生の話聞き、このゼミなら自分のやりたいことができると感じ、このゼミを選んだ。

福田ゼミに入り、最初にやったことは人文情報学科のオリジナル Web サイトの編集・更新であった。ゼミ生全員が分担して編集・更新である。思った以上に時間がかかってしまった。もっと簡単に編集・更新できないか考えた。

3回生になり、卒論のテーマを考えている時に福田先生から「WordPress 使って人文情報オリジナル Web サイトを作成する」ことを勧められた。私はあのことを思い出した。このソフトで Web サイトを制作すれば誰でも簡単に運営・管理ができると考えた。当初私は WordPress の存在を知らず、とりあえず WordPress を使い、Web サイトを制作し始めたが、このソフトはブログ制作の時に用いられるソフトであると購入した本に書かれていたので、本当に理想の Web サイトを作ることが

できるのかと半信半疑であった。しかし、購入した本や WordPress に関する Web サイトを参考し、制作していくにつれ WordPress というソフトについて理解し始め、努力次第で理想の人文情報オリジナル Web サイトができると確信した。

これを作ることにより自分が目標にしていた「4年間で学んだことを活かし自分の力で何かを作る」ということを達成することができ、尚且つ人の役に立つものが作れると考え、この Web サイトを制作することにした。

(2) WordPress とは

WordPress は 2003 年に「b2/cafelog」というソフトの後継ソフトとして誕生した。オープンソースのブログ・CMS のプラットフォームであり、セマンティック Web、コードやデザインの美しさ、ユーザビリティなどを意識して開発されており、無料でダウンロードして使うことができ、世界の多くの人に利用されている。主にブログ制作に使用されるが今回 Web サイトの制作に使用するため編集がブログをしているのと同様な感覚で行える。

HTML や CSS、PHP などの言語で構成されている。主な機能として複数およびネストされたカテゴリーを記事に割り当てる機能やウィジェットを PHP、HTML を編集せずに配置する機能などがある。Web サイトの内容の変更も PHP、HTML を編集しなくても変更ができることも注目すべきところである。

しかし、短所としてエラー発生率が高く、問題解決に時間がかかり、未だ解決できていない問題が多々ある。WordPress のフォーラムや伝言板サイトなどで WordPress 利用者同士意見の交換を行い、問題解決

に努めている。

世界中で利用されているソフトでも完璧なものを作るということは困難であることが分かった。

今回のオリジナル Web サイト制作に使われる WordPress のファイルは、膨大なファイルがある中で Web サイトのすべての基本となる index.php やさまざまな Web の細かなデザイン設定を行う style.css、その他に header.php、footer.php、function.php が主に編集するテンプレートファイルであり、Web サイトの機能や外観などを変更するときに編集が必要となってくる。Web サイトの記事など内部の編集は WordPress 内にある投稿ページや固定ページの作成で行う。今回 Web サイトの制作が目的となるためコメント投稿などブログに必要な機能は省略した。

(3) 対象者について

本制作物は、人文情報学科に在学するすべての学生はもちろん、今後入学してくる学生も対象に制作している。そして、この私が制作した新しい人文情報学科オリジナル Web サイトをこれから運営していく人文情報学科の先生方全員を対象としている。

今まである決まった人しか運営できなかったが今回は、先生方がそれぞれ自分のページを編集し、スムーズに Web サイトの運営できるように制作した。記事の投稿や画像の挿入を Word を使用しているような感覚で編集が出来るようになっている。

(4) 類似物について

現在使われている人文情報学科オリジナル Web サイトは、上記の動機の部分で述べたように HTML などの言語で構成されるソースを直接編集しなくてはならない。編集に時間がかかり、さまざまな言語の知識が必要となってくる。人文情報学科のすべての先生方が容易に編集できない。

また、外観が少々地味であったため在學生や新入生がこのオリジナル Web サイトを意識していないように感じる。もっと外観を明るくし、フォントを大きく見やすくするなどの工夫を加えることでもっと学生がこのオリジナル Web サイトを意識的に観覧するようになれば、学科の情報を学生が認識するようになり、学生の学習意欲が高まるのではないかと考えている。また、大谷大学に入学しようとする学生や全く大谷大学に興味がない学生がこのオリジナル Web サイトを見て、大谷大学人文情報学科に興味を持ち、入学を希望してくれる学生が一人でも増えるように願って制作している。

2 制作物について

(1) 人文情報学科オリジナル Web サイトの設計

今回人文情報学科オリジナル Web サイトの制作にあたり上記にも述べたようにより見やすく、誰でも簡単に編集・更新が行えるといった操作性重視でこの Web サイトを制作しようとする。

まず Web サイトのデザインについては少し派手で尚且つ長時間観覧できる Web サイトを制作していきたい。そのためには、コンテナの色を薄く、サイドバーなどの色は赤系統の色で配色を行いたいと考える。大谷大学のテーマカラーは「えんじ」である。しかし、「えんじ」を使用

すると全体が暗くなってしまう可能性があるのであえてそれに似た色を使用していく。

また、先生専用のページにおいてそれぞれの先生が担当している授業の記載は削除する。先生個人の時間割を記載したところであり意味がないと考える。シラバスのように詳しく授業内容や授業目的が書かれているのであれば意味はある。それがないのであれば1・2・3回生で推奨する授業を参考にすれば十分である。実際に私も観覧することはなかった。無駄なものは削除し、よりシンプルに作り変えることもこの Web サイト制作の目標である。

今回のオリジナル Web サイトは従来のものとは違うデザインで制作していくつもりである。今あるオリジナル Web サイトは大谷大学の Web サイトに似せてデザインされており、統一感があっていいのだが「人文情報学科」の存在があまり目立たないのではないかと考えたからである。

この Web サイトを制作することによって人文情報学科の存在感を十分に出していけたらと考えている。

3 制作過程について

(1) WordPress の導入

今までオリジナル Web サイトはこのようなオープンソースのフリーソフトを使い作成されておらず、多くの知識が必要とされていた。今現在あるオリジナル Web サイトに使われている言語は html・CSS・jQuery などが使用され、気軽に編集するということは困難であった。そこでオリジナル Web サイトの内容をそのままに WordPress を用いて作り代えた。

まず WordPress をサイトからダウンロードし、いくつかの設定をした。この時重要視されるべきところは「パーミッションの設定」である。インストール実行前にまず「wp」、その wp フォルダの中にある「wp-content」をユーザーだけが書き込みができるようにしなくてはならない。ターミナルでパーミッションの設定をする。ターミナル内で `htdocs/imakura/` のフォルダに移動し、そこで「`chmod 755 wp`」と記述しパーミッションを設定する。パーミッションの「755」とは、「通常のディレクトリ、コマンド、CGI スクリプトなど。誰でも読み込みと実行はできるが、書き込みは自分だけ」というパーミッションである。同様に「wp-content」もパーミッションを設定する。

インストールが完了したら「wp」を再び書き込みできない最初の状態に戻す。最初に「wp」を書き込み可能にしたのはインストール中に「wp-config」という設定ファイルが作成されるからである。パーミッションの設定は「ファイル詳細」からでもできる。

ユーザーは今現在私一人だが管理画面で複数のユーザーが登録可能であり、自分だけのアカウントを所有することができる。今現在設定しているユーザーは、「junki.imakura」パスワード「imakura5210」そして、「otaniuniv」パスワード「local_mac」の2人である。

WordPress のデザインは「テーマ」として管理される。上記にも記載されている `index.php`、`style.css` などである。これらは「wp-content」>「themes」>「sample」というファイルの中に作成し、これらも書き込み可能なパーミッションの設定が必要である。これらを経て下準備は終了となる。

(2) 外観の作成

外観の編集は、管理画面（資料 1 参照）の「外観 > テーマ編集（資料 2 参照）」から行うことができる。この段階では html や CSS などを書く必要がある。しかし、WordPress はテンプレートタグをこの段階で挿入することにより、管理画面で投稿や設定を行ったすべてのものを出力し、今後この Web サイトを運営の際、ソースを書き換えるなどの面倒な作業を省き、Web サイトの運営を効率良く行うことができる。

<タグの例>

```
<?php if ( have_posts()):while(have_posts()):  
the_post(); ?>~<?php endwhile; endif; ? >
```

このタグは「ループ」と呼ぶ。これは記入することによりページに表示する記事と記事に関するデータを取得して、記事ごとにループ内に記述した処理を実行するタグである。

```
<?phpthe_title(); ? >
```

記事のタイトルを出力

```
<?phpthe_content(); ?>
```

記事の本文を出力

```
<?phpthe_permalink();?>
```

記事の個別ページの URL を出力

この 3 つを最初に記述したタグの中に入れることにより記事のタイトル、本文が投稿日が新しい順に表示され、タイトルと個別ページがリンクする。

外観のデザインは従来のものより明るくシンプルに制作した。明るくシンプルにすることにより観覧者の目に留まりやすくなり、外観のデザインよりも Web サイトの内容に注目するのではと考えたからである。

大谷大学の Web サイトとデザインが全く違うため違和感を感じてしまう可能性が大いにある。しかし、この Web サイトは人文情報学科として独立した Web サイトであり、上記でも述べたようにこの Web サイトで人文情報学科の存在感をアピールできると考える。

以下外観の作成にあたり注目すべき点を述べていく。

(i) テーマ

テーマにあるテンプレートは最初 index.php と style.css が主なテンプレートであったが index.php の中にあるヘッダーとフッター、それぞれ独立したテンプレートを作成した。今後それぞれを書き換えるときに書き換えやすいようにするためである。index.php に記述されたヘッダーとフッターについてのすべての記述（テンプレートタグなど）を新しいそれぞれのテンプレートにコピーし、index.php にはヘッダー記述があった場所には `<?phpget_header(); ?>` フッターの記述があった場所には `<?phpge_footer(); ?>` を記述することでそれぞれのテンプレートを読み込むことができるようになる。

(ii) ナビゲーションバーの追加

Web サイトの構成は従来のものとあまり変わらないが、今回「ナビゲーションバー」を追加した。ナビゲーションバーを追加することで大谷大学の Web サイトに飛ぶことができるようになり、よく観覧されるであろうページにもすぐに飛ぶことができるようになった。

サイドバーにある項目とかぶるものがあり、ナビゲーションバーにそこまでの重要性はない。しかし、操作性の観点からみてサイドバーの中から項目を探しスクロールさせる手間が省け、効率がいいのではないかと考え追加することにした。

この時使用されるテンプレートの中に function.php がある。func-

tion.php は index.php などのページを作成するテンプレートとは違い、WordPress に関する設定を記述するテンプレートである。これに記述を行い、header.php にも記述の書き加えが必要となってくる。すべてが完了したら「外観＞メニュー（資料3参照）」でナビゲーションバーの機能を有効にすると使用が可能になる。ナビゲーションバーの項目は同画面で自由に項目を入れ替えすることができる。

(iii) ヘッダーの画像設定の簡易化

ヘッダーの画像設定は、header.php でタグ<?phpbloginfo('template_url'); ?>を書き加え、その後ろに画像の入っているフォルダを指定し出力する必要があった。しかし、function.php を使い、その手間を省くことができる。function.php に記述を行い、header.php に記述の書き換えが完了したら「外観＞ヘッダー（資料4参照）」でファイルをアップロードし、ヘッダーの設定が自由にできるようになる。この時画像はあらかじめ「wp-content」＞「themes」＞「sample」のフォルダの中に用意しなくてはならない。画像の大きさは縦 850 横 200 にヘッダーのサイズが設定されているためその大きさに近い画像を用意するのが好ましい。

また、大きめの画像を使用することもでき、画像選択時に大きさに合わせてカットされる。縮尺ではないので画質が変化することはない。

(iv) サイドバーメニューの全固定ページ化

サイドバーは function.php と index.php への記述によって「外観＞ウェジェット（資料5参照）」で設定可能となるが最初にサイドバー内に「カテゴリー」を選択することによってメニュー項目は「カテゴリー」で成り立っていた。

そのためすべての投稿した記事は何らかのカテゴリーに属さなければ

いけない。カテゴリーを Web サイトのメニューのように設定したのはいいのだが問題が起こってしまった。

例えば、「授業紹介」というカテゴリーがある。このカテゴリーは「親」となる。「親」の中に「1回生から受講できる授業」「2回生から受講できる授業」などの子のカテゴリーを作る。この時点でサイドバーに記載され、選択することができるようになるのは「授業紹介」「1回生から受講できる授業」「2回生から受講できる授業」などが選択可能になる。サイドバーで「授業紹介」を選択するとそれを親に持つ子カテゴリーの記事がすべて表示される。これは非常に見にくくなってしまう。そこを選択して見ないようにできなかった。そこですべてメニューを「固定ページ」にしてしまえばいいのではないかと思いついた。

固定ページはどのカテゴリーに属することなくサイドバーに表示することができる。尚且つカテゴリーのように親の固定ページと子の固定ページを作成でき、サイドバーにはカテゴリーのように表示される。ページなので記事が親に集中して掲載されないのである。しかし、サイドバーに固定ページを選択するとき「固定ページ」としてしかサイドバーに表示されないでカテゴリーのようにしようとすると一つひとつの項目にすべての固定ページが表示される。

例えば「進路」という固定ページを親に「人文情報学生からの声」と「QA」の固定ページを表示させたい。しかし、その下に「授業紹介」などの固定ページが表示されてしまう。これを防ぐには「ウェジェット」でサイドバーに選択されている固定ページの設定画面で除外ページを設定すれば解決する。これを設定するにはページ id を記入する。ページの id はそのページの url の一番最後に記載されている。

(3) Web サイトの記事の作成

(i) 記事の投稿

記事の投稿は管理画面の「投稿 (資料6 参照)」で行える。本文はヴィジュアルか html で書くことができる。ヴィジュアルでは、word とほぼ同じ感覚で記述することができる。html では html で記事の作成を行える。基本はヴィジュアルで編集を行えば誰でも簡単に行える。フォントの大きさもヴィジュアルで行えるが「見出し2」(html 内で言う < h2 > のタグ) に設定してしまうと上記のタグの例にある「ループ」内で < h2 > はマークアップして中見出しとして表示されるように設定しているため本文を記述するときフォントの大きさは「見出し3」以降が適していると考えられる。

ページ同士のリンクも可能であり、ツール内にある「鎖」のマークがリンクのツールである。リンクする部分を選択しそれぞれのページのページ id が記載された url を貼ることによってリンクが完了となる。画像の挿入も可能であり、「アップロード・挿入」のツールで行える。この時ヘッダーの画像と同様に「wp-content」>「themes」>「sample」のフォルダ内に画像を用意しておかなくてはならない。画像がアップロード完了すればその他設定の画面になる。画像の配置とサイズを設定する。この時キャプションを設定すれば枠とともにキャプション (注1) が表示される。記事によって構成される Web サイトの内容は「各先生のゼミページ」「先生の個人ページ」「インタビューページ」である。

(ii) 固定ページの作成

固定ページの作成は管理画面の「固定ページ (資料7 参照)」で行える。本来 Web サイトの時系列に依存しない項目を独立したページとして表示するために作成するのが固定ページである。そのため上記にも述

べたようにどのカテゴリーにも属することなく、サイドバーのメニューにも独立したものとして表示することが出来る。ページの作成は、記事の投稿作成を行うときと同じである。

固定ページで作成したのは「はじめに」「授業紹介」「ゼミ」「進路」である。これらは、サイドバーメニューで区切られて表示される。サイドバーに表示するときの設定は上記に説明したとおりである。ゼミの一覧を表示している固定ページは「table」が使用されているためにヴィジュアルでの記述ではなく、html への記述が必要となってくる。

(iii) RSS フィードへのリンクを追加

RSS フィードのリンクを追加することにより Web の記事の更新が効率よく知らせることが出来るので追加することにした。「ウェジェット」を利用してサイドバーに追加することができ、「メタ情報」という項目を追加する。RSS フィードだけでなく、サイトのログアウトを行ったり直接管理画面に移動することが出来るようになる。(資料8参照)

しかし、RSS フィードを追加することによって問題が起こった。エラーが発生してしまうのである。プログラミングにおいてエラーの発生は日常茶飯事であり、エラーとの戦いである。それでもこのエラーは WordPress を使用している多くの人たちが悩まされるエラーであることが分かった。WordPress のフォーラムや様々な Web サイトで問題解決のための書き込みが多数あった。その書き込みを参考に問題解決に努めた。やり方はさまざまであったが WordPress の RSS 関連ファイルに書き加えるというものを取り入れた。「ob_end_clean();」をソースに書き加えることが必要となってくる。「ob_end_clean();」は出力バッファを削除し、出力バッファリング (注2) をオフにするものである。追加するファイルは「wp-includes > feed-atom.php,

feed-rss.php, feed-rss2.php」の3つのファイルの先頭に書き加える。
これは WordPress をアップデートするたびに追加がある。

4 Wordpress のバージョンアップとサーバー移行

(1) バージョンアップ

今現在の WordPress のバージョンは 3.2.1 を使用している。バージョンアップデートを行うことにより今までなかった機能などが追加される。

また、今現在 WordPress はローカルのサーバーで作業が行なわれている。今後 Web サイトを運営していくためには大谷大学のサーバーに移行しなくてはならない。そのため新しいサーバーに新バージョンの WordPress をインストールしておく必要がある。今現在使用している WordPress のバージョンは古いためバージョンアップは必ずしなくてはならない。

しかし、簡単にはバージョンアップは出来ない。アップグレードを行うときに必ず各ファイルなどデータのバックアップはもちろんデータベースのバックアップもとらなくてはならない。データがバージョンアップした時に消えてしまう可能性があるためである。

WordPress のプラグインで「BackwpUp」というプラグインがあるがエラーが発生してしまい、解決するのに時間がかかる。そのため WordPress を使う人たちに敬遠されているためこのまじい方法ではないとされている。しかし、私は「BackwpUp」はデータファイルとデータベースのバックアップを同時に行えるためエラーが発生しないなら使用の方が効率が上がると考える。

実際にこのプラグイン自体はインストール済みであと少しで使用でき

るところまできている。しかし、このプラグインをインストールしプラグインを有効化すると

```
「BackWPup:-Temp folder '/Applications/xampp/xamppfiles/htdocs/imakura/wp/wp-content/plugins/backwpup/tmp/' does not exists!-Temp folder '/Applications/xampp/xamppfiles/htdocs/imakura/wp/wp-content/plugins/backwpup/tmp/' is not writeable!」
```

と管理画面に表示がされる。バックアップをすることは可能なのだがそのバックアップをとったデータをインポートすることができなかった。原因としてあげられるのが先ほど述べたプラグイン自体に起きているエラーが原因だと考える。WordPress のフォーラムにエラー解決のために書き込みをしたが、別のやり方でバックアップすることを進められた。

他にも USB などの外部メモリにバックアップをとる方法もあるのだがこれはアップグレードの時に手作業でやらなくてはいけないので手間と時間がかかる。一般の方法としてデータのバックアップの方法はいくつかあるが FTP クライアントを利用してデスクトップにコピーする方法がある。この方法で私はやっていきたいと考えた。ここで使うのは「Cyberduck」という FTP クライアントである。アプリケーションを起動しクイック接続で「local_mac@localhost」に接続し、アクションでフォルダ「wp」を「wp-1」として htdocs/imakura にアップロードする。

またデータベースのバックアップはターミナルから行うか「phpMyAdmin」で行う方法が一番適切だと考える。しかし、「phpMyAdmin」は sql でパスワードの設定が義務づけられており、私たちが使用

する sql はパスワードが設定されていないため使用することが出来ない。まだデータベースのバックアップの方法について決定的な方法はわかっていない。

バックアップの段階で問題が多数発生したが問題を時間内に解決できなかったため先にバージョンアップの段階の説明に入っていきたい。

まず新しいバージョンの WordPress をダウンロードし、バージョンアップをするためデータベース上の古いファイルを削除しなくてはならない。しかし、下記にあげるファイルは削除せずに残さなければいけない。

- wp-config.php ファイル

- wp-content ディレクトリ

この中の wp-content/cache ディレクトリ

wp-content/plugins/widgets ディレクトリは削除。

- wp-images ディレクトリ (あれば)

- wp-includes/languages/ ディレクトリ

(言語ファイルをあえてここに置いている場合)

- .htaccess ファイル (パーマリンク (注3) 設定をしている場合)

- robots.txt ファイル

(ブログがルートにあって、このファイルを作成している場合)

この時削除は FTP クライアントで行い、新しいファイルも FTP クライアントでアップロードする。そして、WordPress の管理画面にいきアップグレードの可否を確認する。必要とあれば

「This link will lead you to run the WordPress upgrade script by accessing wp-admin/upgrade.php. Follow the instructions presented on your screen. Note: Make sure the

database user name registered to WordPress has permission to create, modify, and delete database tables before you do this step. If you installed WordPress in the standard way, and nothing has changed since then, you are fine.」 とリンクが表示される。

パーマリンクと .htaccess を更新も行う必要があり「管理パネル > 設定 > パーマリンク設定画面」にて、パーマリンク構造を更新する。必要であれば .htaccess ファイルにカスタムルールを追加する。これが完了すれば最新版のプラグインとテーマをインストールする。そして wp-config.php ファイルにセキュリティ・キーの定義を追加する。最新版の WordPress では、ユーザの Cookie に格納される情報をより強固な暗号化で守るため、AUTH_KEY および SECURE_AUTH_KEY、LOGGED_IN_KEY、NONCE_KEY という 4 つのセキュリティ・キーを使用する。

サーバから wp-config.php ファイルをダウンロードしてテキストエディタで開き、これらの定義が入っているか確認する。行が見つからないか、値が初期値のままだったら、wp-config.php の編集の説明に従って行を追加／修正し、更新した wp-config.php ファイルを再びサーバへアップロードする。以上がバージョンアップの手順である。

(2) サーバーの移行

上記に説明したように今後この Web サイトを運営していくには、大谷大学のサーバへ移行しなくてはならない。WordPress のバージョンアップが完了したらまず新しいサーバに WordPress をインストールしなくてはならない。そして、サーバが変わるとデータベースの接

続環境 (ホスト名やデータベース名) なども変わる場合がある。ダウンロードしたプログラムの中のファイル「wp-config.php」をテキストエディタで開き、データベースの設定情報を移行後のサーバーの情報に書き換える。

また、以下のような場合は、移行前のサーバーからエクスポート・保存したデータベースのファイルの書き換えが必要となってくる。

- ・移行前/後のサーバーで利用するドメインが変わる場合
- ・移行前のドメインを移行後のサーバーでも使用する予定だが、ネームサーバー (注4) の切り替えまでは一時的にアクセス可能な仮のドメインや IP アドレスでアクセスする場合
- ・移行前/後のサーバーでルートディレクトリ (注5) からドキュメントルート (注6) までのパスが異なる場合

ドメイン (URL) が異なる場合は、データベースのファイルをテキストエディタで開き、ドメインに関する記述をすべて移行後のサーバーで使用するドメインや IP アドレスなどに書き換える。また、パスが異なる場合は移行後のサーバーの環境に合わせたパス情報に書き換える。

サーバー移行後の作業として移行前にバックアップをとったデータを FTP を使いデータをアップロードさせる。この時データはそのままの状態アップロードさせる。そして新しいサーバーにデータベースを作成し、そこに移行前のデータベースの情報を書き加える。この時多くの人は上記で記述した「phpAdmin」を使用するのだが「phpAdmin」を使用するには SQL のパスワードの設定が義務づけられており、大学の SQL はパスワードが設定されていないため使用ができない。今他の方法を模索中である。

データベースの書き換えが完了した過程で移行先のサーバーでちゃん

と WordPress が起動するか確認する。その次にネームサーバーの変更が必要になる。ドメインのネームサーバーを移行後に提示されたネームサーバーに変更する。WordPress のドメイン設定も変更する。ネームサーバーの変更前まで、移行後のサーバーに一時的にアクセス可能な仮のドメインや IP アドレスでアクセスしている場合、ネームサーバーの変更後、正しいドメインでアクセスするためにデータベースの記述変更をする。

SQL を移行後のサーバーで使用できるようにし、必要に応じて以下の SQL で作業を行う。

- ・ サイトのドメイン (URL) の変更

```
UPDATE wp_options SET option_value = REPLACE (option_value,
'変更前の URL', '変更後の URL') WHERE option_name = 'home'
OR option_name = 'siteurl';
```

- ・ 記事のパーマリンク (guid) の変更

```
UPDATE wp_posts SET guid = REPLACE (guid, '変更前の URL',
'変更後の URL');
```

- ・ 記事内の URL の変更

```
UPDATE wp_posts SET post_content = REPLACE (post_content,
'変更前の URL', '変更後の URL');
```

- ・ 画像のパスの変更

```
UPDATE wp_posts SET post_content = REPLACE (post_content,
'src="変更前の URL', 'src="変更後の URL'); UPDATE wp_posts
SET guid = REPLACE (guid, '変更前の URL', '変更後の URL')
WHERE post_type = 'attachment';
```

- ・ 各投稿のメタ情報の変更

```
UPDATE wp_postmeta SET meta_value = REPLACE (meta_value,  
'変更前の URL','変更後の URL');
```

これで WordPress の起動確認でサーバーの移行は完了となる。

5 アンケートについて

(1) アンケートの内容

この WordPress で人文情報学科オリジナル Web サイト作成について 3 回生の福田ゼミ生を対象にアンケートを実施した。

1. Web サイト全体の配色はどうか？
2. 記事は見やすいか？
3. よかった点
4. 改善すべき点
5. 編集をしてみてどうだったか？
6. この Web サイトを運営できそうか？

(2) アンケート結果について

まず Web サイトの外観についてアンケートを行った結果ほぼ全員が「見やすい」と回答した。しかし、少数意見にサイドバーの色が少しきついのではないかという意見があった。私はサイドバーの色について試行錯誤を繰り返して「ff3300」の色が適していると結論が出た。私はこの意見に反対するわけではないがサイドバーの色が見にくいのではなく、Web サイト全体の色に問題があると考えた。そこで全体の色を黄系統から白系統の色に変更した。そうすることによりサイドバーの色が抑えられ見やすくなったように感じた。

また、フレームが少し狭いのではないかという意見もあった。私もそ

のとおりだと思った。若干記事が中央に集まりすぎていると感じた。そこでコンテナの幅を 800 から 1000 に変更し Web 全体の配置を広げた。そうすることにより窮屈感を感じなくなり非常に見やすくなった。

今現在のオリジナル Web サイトと比べ、明るく作ったが学生は明るい方が目に留まりやすいのだと考えていた。しかし、今回のアンケートからわかったのだが明るすぎるのにも問題があり、簡単であると思っていた配色の選択も難しいものだと感じた。私がサイドバーで設定している配色は「ff3300」で赤系統の中でも明るめの色を設定したのでそこを考えて作成するべきであったと感じている。Web サイトのフレームが窮屈だと観覧する人に対して違和感を与えてしまうことが分かった。そして、フレームを広げすぎても見にくくなってしまうことも分かった。「見やすい Web サイト」を作成するためにはこのような配色やフレームの幅など細かなところを改善していかななくては目指すものは作れないと感じた。(資料 9 参照・資料 10 は比較対象)

編集に関してのアンケートは全員が「使いやすく、編集しやすい」と回答した。編集に関してはこのままでいこうと考える。しかし、要望として「配色などの全体のレイアウトをボタンで簡単にできたらいい」という意見があった。編集する人にとってより簡単な方法で編集できる方が Web サイトを運営しやすいのは言うまでもない。

私はこの意見から Web サイトのバックカラーを「管理画面」で行えるように作り変えた。function.php からバックカラーの編集ができるように書き換えることによってそれは可能となった。こうすることによってバックにカラーだけでなく画像バックに設定を簡単にできるようになった。設定の仕方はヘッダーを管理画面で変更するときとほぼ同じで画像をバックにするにはまずフォルダに画像データを用意し、アッ

プロードする。画像の配置・表示の仕方などを選択し変更することができる。

今回のアンケートを行っていい意見がほとんどではあったが、その中にもちゃんと改善点を指摘してくれる意見が多くあった。この意見を素直に受け止め、すべての人の要望に応えることで Web サイトは自分の理想の Web サイトになっていくと感じた。それと同時に理想だけでは思い通りのものは作成できず、知識と技術が必ず必要となってくると感じた。

アンケートを行ったすべての学生が最後の質問の「この Web サイトを運営できそうですか」という問いに「はい」と答えてくれることを望んでいたのだがそうはいかなかった。3人程「いいえ」と答えてしまい、不本意な結果になってしまった。一人が見やすく使いやすいと答えるのではなく、すべての人がそう思うように作成しなければこれは成功とは言えない。なぜ運営できないと答えたのか私なりに考えた結果私と同じように知識に乏しく、簡単と言っても最初は私もそうであったようになかなか思ったようにできないと考える。WordPress がどのようなものでどんな風に使えばいいのかを知らなければならない。運営するためには WordPress について多少の知識が必要である。

6 終わりに

(1) 自己評価

今回人文情報学科オリジナル Web サイトの制作をして Web サイト自体のはあまりつまづくことなく制作することが出来た。Web サイト全体のデザインや管理画面で行う操作性などは十分満足いく制作ができた。

しかし、出来なかったことも多い。ページ同士のリンクについてページからページへ移動するとき前のオリジナル Web サイトはページの上にとどのページから飛んできたのかが分かるように記載されていた。例えばホームからゼミ・教員に移動したとき「HOME >ゼミ・教員」とページの上に記載され、そこにもリンクがされており、自由にページを行き来することが出来るようになっている。

今回それを取り入れようとした。テーマの編集ではなく、記事の投稿のところで行う。記事投稿の本文に前の Web サイトと同じように今まで飛んできた各ページのリンクを記述しようとしたのだがそうしてしまうとタイトルの下にそれが記載されてしまう。テンプレートタグにより記事の本文よりも先にタイトルが記載されるようにしているためである。タイトルの下に記載されては見栄えがあまりよろしくない。そこでタイトルを出力するテンプレートタグの上にもう一つ記事の本文を出力するテンプレートタグを記述し、どうにかしてそのリンクのところだけ出力させるようにさせればうまくいくのだが上手くいかなかった。これが外観においての一番の課題である。

また、各所のエラーの対応も終わっていない。管理画面にエラーが表示されているのだがこれは RSS に問題がある。上記にも RSS エラーの対処法について述べているのだがまた別のエラーが発生している。しかし、このエラーはこの Web サイトを運営するにあたりあまり問題にはならなかったため、今まで放置してきたが本当にこの Web サイトを公にし運営していくとなれば解決しなくてはならない問題である。

その問題とは、管理画面の WordPress ブログ、WordPress フォーラムに「WP HTTP Error: Failed to connect to 192.168.199.184: Host is down」というエラーが発生している。このエラーにある番号はペー

ジの id であると思われるのでページに何らかのエラーがあると思われる。そして管理画面以外でもプラグインを使用したり、新しいプラグインを検索するとき「API のリクエストに際して、予期しない HTTP エラーが発生しました。」というエラーが発生してしまう。これを解決しようと他の Web サイトを参考に各ファイルを編集してみたところ「不明なエラーが発生しました。」と問題が悪化してしまったため元に戻し放置状態になってしまっている。

また、外観だけではなく、WordPress のバックアップやバージョンアップ、そしてサーバーを移行も大きな課題である。上記にも説明してある通り、データのバックアップには成功したのだが、データベースのバックアップ、バージョンアップ、サーバーの移行に関してはやり方を説明しているだけで成功はしていない。今後この Web サイトを今ある人文情報学科オリジナル Web サイトの代わりに使用するのあればこの問題はこの Web サイトの制作において最大の課題となってくる。

上記でも述べたようにまだたくさんのがあるため時間の許す限り問題解決に努め、今ある人文情報学科オリジナル Web サイトより良い Web サイトにしていきたい。

注

(1) キャプションとは

主に図版や写真について説明のために付け加えられた文字情報のこと

(2) バッファリングとは

複数の機器やソフトウェアの間でデータをやり取りするときに、処理速度や転送速度の差を補うためにデータを専用に設けられた記憶領域一時的に保存しておくこと

(3) パーマリンクとは

Web ページの中の個別のコンテンツに対して設けられた URL のこと

(5) ネームサーバーとは

インターネット上でのコンピュータの名前にあたるドメイン名を、住所にあたる IP アドレスと呼ばれる 4 つの数字の列に変換するコンピュータのこと

(5) ルートディレクトリとは

ツリー型ディレクトリ構造の最上層にあるディレクトリのこと

(6) ドキュメントルートとは

コンピュータ (システム) 上の実際のルートディレクトリ?とは異なり web サーバ上に公開するためのルートディレクトリのこと

文献表

エビスコム

2010 「WordPress レッスンブック」『ソシム株式会社』発行

とほほのスタイルシート

<http://www.tohoho-web.com/css/>

——WordPress による人文情報学科 Web サイトの作成について——

firstserver

<http://wp.fsv.jp/>

WordPress フォーラム

<http://ja.forums.wordpress.org/>