チベット学術研究サイトの構築について

藤巻 昂

目　次

1　**はじめに**　1

1　何を作るのか 1

2　制作に関する構想 2

3　事前に習得すべき知識 3

2　Web **サイトの作成**　5

1　配色に関する考察 5

2　メニューバー作成 6

3　テンプレートの作成 6

4　CSS だけでドロップダウン型のメニューの作成 8

5　翻訳ページの作成 10

6　ログイン・ログアウト機能の作成 11

7　文献一覧の作成 14

8　その他のページについて 14

9　SEO 対策 15

3　**まとめ**　16

1　今後の課題 16

2　自己評価 18

――チベット学術研究サイトの構築について――

1　はじめに

(1) 何を作るのか

私の所属する福田ゼミでは「人に役立つものを作る」というテーマがある。私は自分が４年間の大学生活で学んできたことの集大成を、このテーマに沿って作ろうと考えた。

具体的にどのようなことを目標にしたかというと、既存の Web サイトを改良・再構築し、Web サイトを閲覧する人には見やすいデザインや環境を、Web サイトを管理運営する人には編集しやすい構造・システムにすることである。そして Web サイトの再構築をするにあたって構築の基となるサイトを、福田先生が管理運営しているチベット学術研究サイト「ツォンカパ中観思想の基礎的研究」(※資料集図 1 参照) を基に作成することにした。

そして、Web サイトの改善・構築をするにあたって「ツォンカパ中観思想の基礎的研究」の問題点は、デザイン的に文字が見えにくいことと SEO 対策がされておらず検索エンジンの優先順位が低く事前に「ツォンカパ中観思想の基礎的研究」を知っていないとなかなかたどり着けないことであった。この二つの問題に関しては先に Web デザインの方から進めていくことにした。当サイトを作成するにあたって、私が重要視しているのは、サイトの管理者ページなどの構造は不特定多数の利用者を対象にしたものではなく、特定の個人が使うことを前提としていることが上げられる。今回は福田洋一先生が管理運営することを前提として構築していくので、逐一福田先生と連絡を取りつつ後から福田先生が使う時に不満が無いようなサイトを作ることが目的である。

(2) 制作に関する構想

当サイトの使用目的はチベット語で記述されている文献と日本語訳を同じページに並べて記述しテキストファイルとして保存編集や、福田先生の今までの研究データのリスト・リンク作成などである。使いやすい Web サイトを作るにあたって私は、プログラムのような目に見えにくい内側の工夫を先に作るより、視覚的にわかりやすい、文字の見やすさや研究サイトらしい配色などの外枠から製作することにした。理由としては、今まで Web デザインについての学習する機会があまり無く、早く知識を得て不安要素を消したかったということもあるが、先に目に分かりやすく形にしやすいところから作ることによって次の案に繋がりやすいからというのが私の考えである。今回私が作る Web サイトはチベットに関する学術研究サイトである、Web デザインを製作する基準として、学術サイトなので派手過ぎず落ち着いた色合いにする、Web サイトのデザインは統一感を重視する、文字が読みやすい配色を心がける、とした。もう一つチベットという国を象徴する臙脂色やオレンジ色を使う研究サイトの側面よりチベット色を強調したサイトを作るという案もあったので実際に配色の感じを見るためにフリー素材の Web サイトテンプレートを使い比較、検証することにした。

次にメニューバーについては、普段は親項目の中に収納されていて展開するときは、見たい項目をクリックすることで子項目が展開される形式である開閉式ツリー型メニューにすることにした。本 Web サイトは学術研究サイトなので文献や研究内容などの項目は必然的に後から追加する項目の量も多くなるので開閉式ツリー型メニューにすれば長いメニューの収納が可能になり展開するまですっきりとした見た目になる。そして開閉式ツリー型メニューの子項目は下に向かって展開するの

で Web サイト上部に配置するメニューバーではなくサイドメニューを作ることにした (後にサイドバーが不要になったので開閉式ツリー型メニューではなくドロップダウン型メニューにすることにした)。

そしてプログラミングは、主に Web デザイン・メニューバーなどの基礎部分には HTML・CSS を使い、データのやり取りやログイン機能等には PHP を使うことにした。今回のチベット学術研究サイトは福田先生が管理するのでテストバージョンができ次第確認をしてもらう予定である。また、Web サイトのページは閲覧者用の資料やメモの編集ができないページと管理者用のログインしなければ入れない各項目を追加、削除、編集ができるページに分けるつもりである。

(3) 事前に習得すべき知識

必要な知識としてはプログラム面に関しては、HTML・CSS の知識の習得と PHP を使うことにした。PHP とは動的に Web ページを生成する Web サーバの拡張機能の一つであり、また、そこで使われるスクリプト言語の事を指す。レイアウトの「雛形」となる HTML ファイル内に、処理内容を記述したスクリプトを埋め込み、処理結果に応じて動的に文書を生成し、送信することができる。PHP は動的に生成されるページの作成に向いているので、本 Web サイトでは、研究内容や翻訳文などをサイト上で保存できる機能をつけようと考えている。具体的には研究内容をサイトのテキストエリアに書き込み保存をするとテキストに書かれた内容が保存されたテキストファイルを指定したフォルダーに生成し、保存した内容が書かれたページも同時に自動生成する機能をつけようと思っているので PHP を選んだ。そのほかの理由としてはソースの書き方が利用者が多い、CSS や Javascript に非常に似ていて CSS

や Javascript を利用している人にとっては編集が簡単だということもある。

また、今回は開閉式ツリー型メニューにしたサブメニューを制作するつもりなので JavaScript の知識も必要となる。JavaScript とは、Web 上でインタラクティブな表現をする為に開発されたオブジェクト指向のスクリプト言語である。HTML 内にプログラムを埋め込むことで、Web ページに様々な機能を付加できる為、HTML や CSS では表現できないユーザーの動きに応じたものを作ることができる。例としては、マウスの動きにあわせてデザインが変化する複雑な WEB ページなどの Web ページに動的な要素を加えることが可能になる。従来 Web ページは、静止画や印刷物のような静的な表現しか作ることが出来なかったが、JavaScript の登場により幅広い表現（動的表現）が可能となった。また Java と名前は似ているが Sun 社の Java 言語に似た記法を用いることが名称の由来となっただけで異なるものである。

もちろんのことだが管理者画面に一般閲覧者がいけないようにするセキュリティーなどの Web 上で最低限必要な管理者権限についての学習も必要だ。これに関しては PHP を使い管理していくつもりだ。そのほかに Adobe Photoshop や Corel Paint Shop Pro などを使った Web デザインについての知識の習得も必要なことである。Web デザインについての学習の方法は福田先生に相談したところ大学にある Web デザインについての教材を借りる場所を教えて頂いたので大学の教材とインターネットでフリー素材を使って学んでいく予定だ。

2　Web サイトの作成

(1) 配色に関する考察

はじめに Web デザインのテンプレートの作成にとりかかることにした。まず実際に Web サイトを管理運営する福田先生の要望を聞いたところ、先生が希望するホームページのサンプルを送って頂いた。(資料集図 2 参照) この他に、ホームページテンプレートの無料配布サイトのフリー素材や他のチベット学術サイトを制作の参考にすることにした。まず私は福田先生から頂いたサンプルページの配色をチベットの象徴的な色である臙脂色やオレンジ色に差し替えてテンプレートを作成した。結果として、全体的に派手な配色になり、チベットに関するサイトという面では強く印象付けられるのだが、当 Web サイトは学術研究サイトであり、注目されるべきはデザインではなく内容であるので、適していないという結果になった。この結果から配色は背景色はデフォルトの白または淡色系などの文章を読むことの妨げにならないような配色が適しているということと、メニューバーやヘッダー、タイトルなどの目立つようにする部分は、焦げ茶色などの彩度を落とした落ち着いた色を使用することにした。サンプルページを制作して福田先生と相談した結果、背景は淡い乳白色を使用し、メニューバー等は焦げ茶色やクリーム色等のシックな色合いを使うことにした。また配色の決定に伴いチベットに関するサイトという印象付けは画像で行うことにした。画像については著作権などの問題を考慮して先生から使用しても問題のないチベットに関係する画像を幾つか頂いたので背景やメインページに活用することにした。

(2) メニューバー作成

配色が決定したので Web ページテンプレートの無料配布サイトのフリー素材を参考にして Adobe Photoshop CS5 でボタンを作成した。当初は目次ごとに個別にに作成していたのだが新しい項目を追加又は削除するときに不便だと感じたので、メニューバーの画像は何も書かれていないものを繋げて表示し、CSS を使い
タグで追加された文字の左右に仕切りをつけることによってできる擬似的なメニューボタンを使うことにした。この形式に変えたことによって簡単に追加・削除・変更が出来るようになった。問題点は項目が増えすぎると文字がメニューバーの画像の範囲から飛び出してしまうことである。この問題点について福田先生と話し合いをしたところ学術研究サイトは子項目が増えることはよくあるが親項目が増えることはそうそう無いとのことだったので、問題無しと判断しこの形式でメニューバーの作成をすることに決定した。

(3) テンプレートの作成

背景は福田先生から頂いたチベット語の文献の画像を Adobe Photoshop CS5 で編集し淡い乳白色にしたものを使用した。また当初予定していた開閉式ツリー型メニューはサブメニューとして組み込む予定であったが、サブメニューがサイトの幅を圧迫するという問題があったので福田先生と相談した結果廃止することとなった。それに伴いサイト上部のメニューバーで親項目をすべて表示することになったのでよく横長のメニューバーに使われるドロップダウン型のメニューバーにすることにした。これはメニューバーの項目の上にカーソルを合わせたときのみ子項目が展開されるという形式である。今回は様々な要素を入れる

とソースが複雑になると考え CSS だけでドロップダウン型のメニューバーを作成し、統一感を重視したのだが、逆にソースが長くなってしまったことが反省点である。しかし項目を追加編集すること自体は簡単に出来るので問題なしと判断した。

また、ヘッダー (メニューも含まれる)・フッターのようなほとんどのページで表示させる要素は、一つのファイルとしてまとめ、そのファイルの内容が必要な個所に PHP の require 文を使い呼び出す形にした。require 文とは指定されたファイルを読み込み、実行する関数である。また似た関数として include という関数が存在する。両関数の違いとして、ファイルを見つけられない場合に include に関するスクリプトの処理だけを中断し warning を発行する。一方 require の場合は、同じ場合に fatal error を発行しそのページのスクリプト処理全体が中断されるという点が異なる。 fatal error の場合エラーが出た時点ですべての処理が中断されるので本来ならば include のほうがいいのだが、include の場合データが読み込まれなくても他の箇所が動いてしまうのでエラーに気付かないことがあると考慮し require を使うことにした。この require 文はソースの簡略化だけでなく頻繁に編集を行う箇所を分けることによってどのファイルに何に関するソースが記述されているかが分かりやすくなるので、編集をする人にとってはとても有用である。require 文を使う利点として長くなりがちなプログラミングのソースを複数のファイルに分割することによって大本のソースを簡略化し見やすく改善できる点があげられる。

(4) CSS だけでドロップダウン型のメニューの作成

HTML の部分は (資料集図 3 参照)
を使ういたってシンプルな作りである。ドロップダウン表示させたい項目を<ul>の中に入れ子上に
を使い書き込むだけでドロップダウン表示がされるようにしている。そしてメニュー全体の ul の設定については、ドロップダウン型メニュー全体の横幅を変えたいときは「dropmenu」の width の値を変更する (資料集図 4 参照)。このドロップダウン型のメニューの「dropmenu」の中にある width の値を変更するだけで他の箇所はいじらなくても大丈夫なようになっている。他の横幅を変更するとドロップダウン表示させる項目がずれる可能性があるので変更する必要は無い。高さについては、後で出てくる「dropmenu li a」(資料集図 5 参照) の上下の padding と font-size の値の合計で決まるようになっている。「dropmenu li」では横幅の値を%で指定するようになっている。何故かというと、こうすることによって全体の大きさを変えるときに先ほどの「dropmenu」の幅を変えるだけで他の横幅の値も自動的に適切な大きさに変更されるので、いちいち全部の値を変更する手間が省けるからである。そしてサブメニューの表示位置の基準となるため、position: relative; を指定している。position: relative を指定すると top、bottom、left、right の各プロパティの値の分だけそれぞれ通常の位置の上、下、左、右からずらして各要素を配置することが可能になる。また top、bottom、left、right の各プロパティにはマイナス値も使えるようになるので position: relative; を指定した箇所を自由に配置できるようになる。この position: relative; によりドロップダウンの箇所の表示位置の基準を定めているので、当サイトの CSS だけで作るドロップダウン型メニューにはとても重要な要素である。

次にメインメニューのリンク部分については、line-height: 15px; を指定しておくとことで、全体の高さが上下の padding と font-size の合計になるようになっている。これは最初に指定した dropmenu の高さと同じ値である。次にマウスを乗せたときの色の設定に関しては、（当サイトでは色の代わりに画像を使っている。）この設定はメインメニュー、サブメニュー共通で設定することができる。また「dropmenu < li:hover < a」の border-radius の項目でメインメニューの上にマウスを乗せた時に角が丸くなるように設定している。この時「<」を挟むことでサブメニューには適用されないようにしている (資料集図 6 参照)。

次にドロップダウン表示されるサブメニューの設定についての解説は、まずサブメニュー全体の ul の設定については「dropmenu li」を基準として position: absolute; を使い絶対値で表示する位置を指定している (資料集図 7 参照)。今回の場合「top: 10037;」でメインメニュー真下に表示するように設定している。またサブメニューに関しては border-radius で下側の角を丸くなるようにしている。次にサブメニューの幅についての設定は、今回サブメニューの幅は width の値を 10037; にしてメインメニューと同じにしているのだが (資料集図 8 参照)、見やすさやボタンの押しやすさに不満を感じた場合には、サブメニューの幅をメインメニューの 2 倍にすることも可能である (width の値を 20037; にする)。しかし 2 倍にした場合一番右のサブメニューが右側にはみ出ることになってしまうので、それを防ぐために一番右のサブメニューのみメインメニュー 1 つ分左にずらすなどの微調整が必要になる。

次にサブメニュー内の li については、transition でオンマウス時の変化に動きを付けている。また初期状態では「overflow: hidden;」「height:

0;」でサブメニューを表示されないようにしている (資料集図 9 参照)。メインメニューにマウスを乗せた際のサブメニューの動きについての設定はここ (資料集図 10 参照) で行っている。高さは「dropmenu li ul li a」の上下の padding の各ピクセルと font-size12px の合計の値になっている。展開前は平面に見えるが、実は画像が重なった状態になっている。最後に先ほど (図 6 参照) で行ったことと同じように一番下のサブメニューの下の角を丸くし形を整えた (資料集図 11 参照)。このドロップダウン型メニューについては、「Webpark CSS だけで作る動きのあるドロップダウンメニュー」というサイトを参考にして作成した。

苦労した点として、私はメニューのデザインを画像で構成していたのだが、本来私が参考にしたサイトではメニュー全体のデザインは画像ではなく CSS のカラーコードで作られていたので、全体大きさの配分やそれに伴う配置の調節が一番苦労した。

(5) 翻訳ページの作成

このページの用途としてはチベット語の原文と翻訳文を同じウィンドウで並行して編集が行える (資料集図 12 参照)。ファイル名、題名を記述して保存すると翻訳フォルダーにテキストファイルとして保存できる。また保存することによって翻訳一覧ページに保存したページの内容が記録され自動的に項目として追加されるのでいつでも編集・削除が行うことができる、各操作は PHP を使いプログラミングを行っている。各操作の具体的な説明として、保存・編集はそれぞれのテキストエリアに書き込まれたデータを関数に代入し、代入したデータ fwrite 関数を使い、指定したフォルダーにテキストファイルとして作成・上書きを行う (資料集図 13 参照)。またそれぞれのテキストエリアに書き込ま

れたデータが入った関数は翻訳一覧ページから各翻訳ページの情報を取得して呼び出すためにも利用している。削除機能は厳密にはデータを消しているのではなく fwrite 関数を使いテキストファイルの名前を書き換え各ページでデータの取得をできないようにしている。削除しない理由としては、福田先生と話し合った結果翻訳ページのデータはテキストファイルなのでデータの容量も少なく完全に消去する利点が少ないので、データを完全に削除するより過去のデータを参照出来る方が使い勝手がいいという結論に至ったからである。

またこの翻訳ページは閲覧者用ページと管理者用ページでは仕様が違い、閲覧者用ページでは文章を見ることは可能だが、編集・保存・削除ができなくなっている。またマウスでの範囲選択からのコピーも disabled を使うことによってできなくしてある。しかし検証の結果 disabled はウェブブラウザによって実装されていないものがあり、私が普段利用している Safari では正常に機能せず書き込みは制限できるが範囲選択からのコピーは出来てしまう状態であった。改善のために他の方法も試してみたのだが結果はあまり変わらなかったので一番ソースが短く分かりやすい disabled のままで行くことにした。(後に IE で翻訳ページを表示した場合原文と翻訳文の欄が 1 行ずれるという問題が発生したが、解決するには至らなかった。)

(6) ログイン・ログアウト機能の作成

ログイン・ログアウトの機能はまず PHP の define 関数で入力されたパスワードのチェックを行い正誤を判断し、場合分けでパスワードが誤っていた場合はログイン画面から先に進めず、パスワードが正しかった場合は isset 関数を使い管理者 ID として使用する変数の取得と、正常

に変数が取得されていることを確認してから<form action>で変数を送信して次のページに移動するようにしている (資料集図 14 参照)。次のページでは前のページで<form action>で送信された変数を cookie で管理者 ID として取得してから管理者用ページに移動する仕組みになっている。cookie とは、Web サイトの提供者が、Web ブラウザを通じて訪問者のコンピュータに一時的にデータを書き込んで保存させるしくみである。cookie にはユーザに関する情報や最後にサイトを訪れた日時、そのサイトの訪問回数などを記録しておくことができる。cookie はユーザの識別に使われ、認証システムや、WWW によるサービスをユーザごとにカスタマイズするパーソナライズシステムの要素技術として利用される。当ページでは正しいパスワードの ID 情報が cookie に組み込まれていないと管理者ページに入れないように認識システムとして利用している。

また、このページは cookie の取得のためのページであり長くとどまる必要はないので<META http-equiv='refresh' content='5; url=kanri_index.php'>を使い 5 秒後に管理者用メインページに自動的に移動するようになっている。

なぜ cookie でパスワードを ID として取得してからなのかというと各管理者用ページのソースの最上部には正しいパスワードを入力してＩＤを取得していないとページの表示がされないように require_once で制限がされているからである。require_once とは基本的には require と働きは同じだが、ファイルがすでに読み込まれているかどうかをチェックするという点が異なる。当サイトではこの機能をパスワード ID を取得しているかをチェックし ID を取得していない場合は表示しないというセキュリティーとして使用している。また require_once を使うとき

の注意点として必ずソースの最上部に配置しないといけないようになっている。そのため空行が入っているだけでもエラーが起こるので注意が必要である。

ログアウトの機能も cookie を利用したもので、管理者用ページのメニューの時のみに表示されるログアウトボタンを押すことによって今まで取得していたパスワード ID を削除してから閲覧者用のメインページに移動するようになっている。なぜパスワード ID を削除してからなのかというと、そうしないと次に管理者用のページに移動する時にパスワードの入力をしなくても管理者用ページに行けてしまうのでセキュリティー面で問題があるからである。ID 削除の方法としては PHP で取得した ID に isset 関数で ID の部分にに空白を上書きすることで ID を消している。当初はログアウトして移動する閲覧者用メインページでこの処理を行うつもりだったのだが、ほかの関数と干渉しあってしまいうまく働かないという問題が発生してしまった。なのでログアウトボタンを押してから閲覧者用メインページに移動するまでにパスワード ID を削除する専用のページを挟むことで問題を解決した。しかし、ログアウト時にあまり意味のないページにとばされるのは手間がかかるので、header 関数を使い ID の削除専用ページに移動した瞬間に次の閲覧者用のメインページに移動するようにした。実際に何度かテストを行ったがログアウトボタンを押して閲覧者用メインページに直接移動する時とほとんど同じ速度でページ移動するが可能だったので。

デザインに関してはログインフォームのデザインに特に力を入れて取り組んだ。ログインフォームのデザインはすべて CSS で作成したのだが、最初のころはうまくデザインがまとまらずログインフォーム作成は難航した。その時に私が活用したのが CSS Button Generator という

Web サイトである。この Web サイトの CSS ジェネレーターは細かいデザインまで簡単に操作ができ、なおかつ実際にどのようにデザインが変わるのかを見ることができるので一々 CSS のソースを書き換える手間が省けてとても重宝した。

(7) 文献一覧の作成

文献一覧は言語によって分類分けされていて、日本語、サンスクリット語、チベット語、欧文の４種類がある。各項目にはメニューバーから移動ができるようになっている。また各文献ページを選択した時はページに移動するのではなく、適度な大きさの新規ウィンドウを開く形式になっている。これは Javascript の window.open を使用して行っている。これにより複数の作業を並行してできるので使い勝手は向上していると思われる。

文献一覧の作成にあたって福田先生と使いやすいと思う構造について相談したところ、題名・著者名などの項目に記述欄を分けて記述していく形式ではなく広いテキストエリアでいつでも自由に編集ができる環境がいいということになったので、管理者用画面では広いテキストエリアに翻訳ページにも使われている PHP を使った保存方法といった至ってシンプルな作りにした。管理者ページで編集・保存された文章は閲覧者用ページではテキストエリアに書かれた文字としてではなく、通常のソースに書き込まれた文章のように表示される。

(8) その他のページについて

リンク集のページについては一応作成現時点では大谷大学の Web サイトへのリンクと、大谷大学人文情報学科オリジナルサイトの福田先生

のページへのリンクしか存在しない。なので、随時追加できるようにサイト上でもリンクが作成できるようにしている。次に福田先生の今まで行ってきた研究内容のページについてなのだが、当サイトを作成している途中に福田先生が今までの研究データを書き込んでいたデータベースの更新が行われてその影響で研究内容のデータが無くなってしまうというアクシデントが発生したため、研究内容のページについてもリンク集と同じようにサイト上でプログラミングが行えるようにしている。

(9) SEO 対策

まず SEO とは「google」や「Yahoo」などのサーチエンジンの検索結果のページの表示順の上位に自らの Web サイトが表示されるように工夫することである。なぜ SEO 対策をするのかというと、検索結果のページの表示順の上位に自らの Web サイトが表示されるようにすることによって人の目にとまりやすくなるので、自身のサイトを多くの人に見てもらう確率が上がるからである。

具体的な SEO 対策の方法としては、自分の Web サイトを表示順の上位にしたいサーチエンジンの優先順位が高いキーワードの適切な選択や、メタタグの description や keywords を使う、自分の Web サイトを表示順の上位にしたいサーチエンジンに登録してからより多くのサイトにリンクしてもらうなどの手段がある。今回はサーチエンジンに登録などは出来ないのでメタタグを使用する対策をすることにした。まず description については、サーチエンジンで検索をしたときにサイト名の下に書かれるサイト内容の紹介文を表示するタグである。「goo」や「Infoseek」等では検索結果に対する優先順位が高いので、キーワードを盛り込みつつ的確にサイト紹介をすることが重要で

ある。次に keywords については、そのページの重要なキーワードを書き出し、検索ワードに引っかかりやすくするタグである。使い方としては description に書ききれなかった単語を補うように使う。キーワードは最大 10 ワード以内におさめ、ワードの選択はページのコンテンツと合っているものを書くようにする。またメタタグはメインページのソースだけではなくすべてのページに異なる内容で書き込む方が有効的とのことだったので閲覧者用のページにはすべて書き込んだ。

しかし、サーチエンジンは登録されている Web ページをキーワードに応じて表示するが、その際の表示順位はそれぞれのサーチエンジンによって異なる法則に則って決定している。さらにサーチエンジンの表示順位を決めるの法則は年々高度化が進む上、頻繁に変更が行われその度に激しく順位が変動するので、SEO 対策の方法に厳密な正解は存在しないといわれている。なので本来ならば地道に自分の Web サイトを表示順の上位にしたいサーチエンジンの利用者が求める情報をリサーチし、コンテンツを充実させて認知を広げることが確実な方法だといえる。

3　まとめ

(1) 今後の課題

Web サイト全体としての課題は、私は今まで先生が日常的に使用する Safari を基準に編集作業をしていたのだが、Web サイト構築の終盤で Firefox などのブラウザでは仕様が違うプログラミングの関数や構文がありテキストエリアがすこし横にずれていたり背景がうまく表示されなかったりなどの不具合が発生したので、他の閲覧者がこのサイトをどのような環境で当サイトを見るのかという点についての対策が自分には

不足していたと感じた。

また今回は管理者が福田先生が管理・運営するということで、各項目には項目ごとに入力欄があり誰でも簡単に編集作業ができるといった丁寧さより、一々対象のファイルが存在するフォルダーまで移動しなくても Web サイト上で編集作業が可能といった自由度を重視した Web 作りをしたが、これはあくまで福田先生がプログラミングについての知識があることが前提での構造であり、もし Web サイトの管理する人がプログラミングに関する知識を持ち合わせていなかったら、その管理者にとっては使いづらい Web サイトだと感じた。今後私が Web サイトの構築をすることがあるのならプログラミングに関する技術力が低い人でも使いやすいと感じるものを作りたいと思う。

デザイン面の課題については、同じゼミ生数人に実際に見てもらい評価してもらったのだが、いささかデザインが古い、もう少し派手さがあってもいいとの評価を頂いた。私は他の研究サイトやフリーの Web ページテンプレートを参考にして当サイトのデザインを作成したのだが、Web サイト全体のデザインの統一感の重要性や研究サイトに適した Web デザインを重視しすぎて、面白味のないデザインになってしまったのではないかと感じた。

プログラミング面については、編集のしやすさやソースの見やすさを考え、言語は HTML、CSS、PHP、に絞り統一性を重視したのだが、逆にソースが長く複雑になってしまう箇所が出来てしまったので、ソースの見やすさや編集のしやすさについては、まだ改良の余地があると感じた。翻訳

(2) 自己評価

今回のチベット学術サイトは閲覧者にとってはある程度のラインまでは見やすく操作しやすくなったと感じている。福田ゼミのテーマである「人の役に立つものを作る」は達成できているのではないだろうか。しかし今回のサイト構築は不特定多数の閲覧者が見やすく操作しやすいと感じるサイト構築と、Web サイト管理者 (当サイトの場合 福田洋一先生) が見やすく操作しやすいと感じるサイト構築の二つの視点を両立し兼ね備えたサイト構築を作らなければいけなかったので、両視点をバランスよく取り入れるということにはまだまだ課題が残っていると感じた。また Web サイトの管理者の視点で見れば、使いやすいとは言い難い箇所が存在すると言わざるおえない。何度も福田先生に確認してもらい問題があればその都度改善していたつもりであったが、まだまだ使いやすい Web サイトには課題が多いと感じた。

当サイトの作成をして、作るのは私だが実際にこのサイトを管理運営していくのは、福田洋一先生であるということを考えた時にどの程度まで私の案を形にしてもいいのかということについて深く考えさせられた。私は管理運営を行っていく人の要望実現していくだけでは管理者の利点ばかり重視され、閲覧者にとって見やすく操作しやすい Web サイトとしての側面が弱くなってしまうと感じた。なので制作者は管理者の視点を持つことはもちろんだが閲覧者としての視点も持ち合わせないとよりよい Web サイトは作れないという結論に至った。

文献表

Webpark 「CSS だけで作る動きのあるドロップダウンメニュー」

・http://weboook.blog22.fc2.com/blog-entry-359.html

杜甫々「とほほのスタイルシート」

・http://www.tohoho-web.com/css/

Steven Wanderski「CSS Button Generator」

・http://css3buttongenerator.com/

Nikukyu-Punch「Nikukyu-Punch」

・http://nikukyu-punch.com