

# ブログ風 Web ページ作成ツールの製作に ついて

嶋本萌

## 目 次

1	序論	1
1	今回製作したツール	1
2	ブログの利点	1
3	ブログの問題点	4
2	製作	5
1	HTML の補助	5
2	メニューシステム	6
3	メニューの削除に関して	8
4	サイトの表示	9
5	メニューの表示	10
6	ページの表示	11
7	3種類のサイト表示	12
8	サイト配色変更システム	14
3	結果	15
1	モニター結果	15
2	メニューシステムの課題と改善策	16
3	サイトの配色変更システムの課題	18
4	今回の製作で学んだ事	19
1	当初のツール製作の概念	19
2	矛盾	19
3	製作後のツール製作の概念	20
4	製作者に必要な3つの視点	21

## 1 序論

### (1) 今回製作したツール

今回製作したツールは、授業で製作したブログから派生した、メニューシステムに重点を置いた Web ページ作成ツールである。一般的なユーザーをターゲットとしており、情報整理に特化したツールの製作を目指した。

具体的には、ブログのカテゴリー機能をより発展させ、下位5つまでの階層構造を持つメニューシステムを取り入れたツールである。

メニューを階層構造にすることにより、情報（ページ）が整理され、それが閲覧者への情報伝達向上に繋がるのではないかと考え、今回の製作に至った。

使用したのは主に PHP と MySQL で、サイト情報や作成したメニュー、ページなどの情報を全てデータベースに保存しておき、サイト表示の際、データベースから情報を引き出し表示される形式をとっている。

そのため、一般的な Web サイト、ホームページの作成のように1つずつ HTML のファイルは作成されない。こうすることの利点に関しては、下記の「ブログの利点」の後半にて述べる。

### (2) ブログの利点

ツールの製作にあたり、参考にしたいブログの利点として、まず、HTML の知識がなくとも簡易にページが作成が可能であるという事があげられる。

利用者はテキストエリアに文字を入力し、決定ボタンを押すだけでも最低限のページが作成できるのである。また、大抵のブログにはテキス

トエリアの上部に入力を補助するツール（文字色変更、画像挿入など）が設置されている。そのため、HTML の知識がない者も、HTML の知識がある者も、容易に多様なページを作成できるのである。

もう一つの利点として、ページを繋ぐ必要がない事があげられる。これが 1 ページ 1 ページ HTML ファイルを作成しない利点である。

通常、Web ページとはリンク等によって Web ページ同士繋がっている。これは他者に閲覧してもらうという情報発信が目的であり、孤立しては意味がないものとなるからである。したがって、新しくページを作成した際には、そのページに跳ぶリンクを貼るという、ページを繋ぐ作業が必要になる。

しかしブログの場合は、利用者が意図的に操作しないかぎり、作成したばかりのページが孤立することはない。自動的に繋がりのあるページが作成されるのである。ページを作成する度に必要な作業が不要という点は、大きな利点だと考えられる。

これは 1 ページ 1 ページを HTML のファイルを持つ Web ページでは難しく、データベースを活用してこそ実現が容易な Web ページの特徴である。

実用的な面が目立つ上記 2 点以外に、デザインの変更が容易に可能である事もブログの利点として取り上げたい。理由は以下の通りである。

まず、注目したのはデザインがもたらす効力である。Web ページを作成するということは情報を発信するという事であり、より効果的に閲覧者に伝える事が Web ページ作成者の目的である。視覚的要素を持つデザインは、閲覧者に与える影響が大きく、これにより情報伝達力が左右されると言っても過言ではない。

例えば、レイアウト。Web ページはそのページに表示すべき内容（コ

コンテンツ) と、目的のページに辿りつくためのナビゲーション(メニュー表示) の両方が必要である。ナビゲーションの機能がなければ、閲覧者が目的のページに辿り着くのが困難になってしまう。そして、この二つをどのように配置するかによって、閲覧者がどのように辿れば目的のページに行けるか理解しやすいかそうでないかが決まる。

また、『プロとして恥ずかしくない WEB デザインの大原則』という書籍にて『Web サイトは情報を伝える事が最大の目的だ。それだけに視覚効果の分野で大きな役割を果たす配色は Web デザインにおける重要な要素となる』という配色の重要性について石田恭嗣さんの考えが述べられている。確かに、私達は日常において色から受ける印象が強い事を知っている。例えば、赤は危険、黄色は注意、青は安全と定められた信号機は、色によって人間が受ける印象を利用したものである。

Web ページにどれだけの情報を積み込むかは制作者の自由で、制限はない。しかし、その Web ページの情報全てを閲覧者が見るとは限らない。そのため、ほんの一瞬でどれほどの情報を閲覧者に伝えられるか、引き込めるかが重要になる。文章と違い色は一瞬にして閲覧者に届く。よって、サイトの目的にあう色の印象を閲覧者に与えるというのは、理にかなっていると私は考える。

また、閲覧者への影響以外に、作成者の意向を顧慮する必要がある。Web ページは情報発信をすると同時に、作成者が Web を通して不特定多数の人に、共感、同調など繋がりを求めるものでもあり、作成者が自分の意向に沿ったデザインを求める傾向は強い。

現在サービスとして提供されているブログは、デザイン変更のシステムが非常に充実している。予め用意してあるテンプレートの他にも、個人で編集できる機能や、それらを利用同士で共有できるシステムなども

ある。その結果、膨大な量のテンプレートが存在し、利用者はそれらを吟味し使用している。

サービス側がここまで重点を置いているデザインシステムだが、このシステムは、情報伝達の向上という目的よりも、利用者の好みに合うテンプレートの提供という目的で作られている。

この事から、自分の好みに合うデザインのサイトを利用できるか否かにを、重要にとらえている利用者が多いということが伺える。

ツールは利用する人がいて意味のあるものである。そのため情報整理に特化したシステムだけでは、利用価値のあるツールとは言えないのではないかと考えた。

しかし、今回の製作にあたり、時間、技術、経験などの要因から、情報整理システムとデザインに関するシステムの2つを同時に極める事は難しい。

そのため今回は、ページの整理に重点を置いてのツールを製作となり、デザインのシステムに関してはサイトの配色を変更するものにとどめる事にした。

### **(3) ブログの問題点**

多くの利点があるブログだが、通常のブログを一般的な Web サイトとして扱うには問題がある。それはページを整理する機能が十分ではないという事である。

ブログは日記形式であるため、投票された記事は十分な分類が施されないまま、ただ蓄積されるという状態で Web サイトを形成している。

これに対し、通常 Web サイトは一つのカテゴリーの中に、更に細かく分けられたカテゴリーが存在しているケースが多い。

ブログにはカテゴリーという記事を分別ができる機能があるが、非常に限られた範囲でのみであり、更に細かなカテゴリーへと分別する機能はない。上記のようにカテゴリーの中にカテゴリーがある Web サイトの作成には十分な対応はしていないのである。そのため、ブログを用いて一般的なサイトを作成するには、利用者が各自努力する必要がある。

故にページ整理に関して十分とはいえず、一般的な Web サイト作成ツールとするにはこれらを改善する必要があると考えた。

## 2 製作

### (1) HTML の補助

HTML の知識がなくとも作成できる工夫として、以下の仕組みを取り入れた。

まず、テキストエリアでのエンターキーによる改行を反映させるために、本文のデータを変数\$honbun とし、nl2br(\$honbun) とした。そして、その他の入力補助として、本文を入力するテキストエリアの上部に、JavaScript を利用し、テキストエリアで反転選択している箇所の最初と最後の部分に< b ></b >や< u ></u >といったの太文字や斜線などのタグをクリックで挿入できるシステムを取り付けた。

しかし、これは上記のように開始タグと閉じタグが同じである単純な形式のタグのみに対応しており、< font color="#999999" ></font >や< a href="http//..." ></a >というような前後の開始タグと閉じタグが異なる、かつ URL や色コードのように特定されていない形式には対応できない。< font ></font >< a ></a >だけの挿入なら可能だが、これでは HTML の知識がない者には扱えず、「HTML の知識がなくとも作成できる工夫」とは言えない。

そのため、文字色の変更やリンクの挿入に関しては、別ページにそれぞれのタグ作成ツールを製作し、テキストエリアの上部に、別ウィンドウでそのページを開くリンクを貼る事で、この問題を解決する事にした。(資料8 参照)。

2つのタグ作成ツールは同じ仕様になっている。文字色変更のツールを例に説明すると、まず、色を付ける文字をフォームに入力してもらい変数\$moji とする。そして何色つけるのかを選択してもらい変数\$color とする。これは、直接色コードを入力してもらう事も可能である。取得した\$moji と\$color のデータをテキストエリアに `< span style="color:$color">$moji </span >` と表示し、利用者にはテキストエリアに表示されたタグをコピー&ペーストして使ってもらう仕組みになっている。

以上が私が用意した、HTML の知識がなくとも Web ページが作成できる補助システムである。

## (2) メニューシステム

次に、メニューシステムの開発だが、これはフォルダの概念から製作のヒントを得た。

フォルダは Macintosh 及び Windows のファイル管理に用いられている概念で、UNIX ではディレクトリに相当する。1つのフォルダの中にさらに下位のフォルダを持ち、階層構造をなして、木(ツリー)構造ともいわれているものである。(資料9 参照)。

無限に下位の層を作成できる上記に対し、このツールには5階層までという制限を設けた。(資料10 参照)。

具体的には、メニューを作成する際に5桁の保存番号というもの定



め、左から順に $\$menu\_a \sim \$menu\_e$  という具合に変数名を割り振り、保存番号 10000 のメニューであれば $\$menu\_a=1$   $\$menu\_b=0$   $\$menu\_c=0$   $\$menu\_d=0$   $\$menu\_e=0$  とした。

保存番号を 5 桁と制限した理由は、メニューがどの階層であるのか保存番号の下桁の 0 の数で判断しているからである。例えば、保存番号が $\$menu\_b \sim \$menu\_e$  が全て 0 のならば資料 10 の 1 の階層となる。

保存番号 10000 のメニューに子メニューを作成すると、保存番号 11000 のメニューが作成され、さらにその下位にも同様の作成される仕組みになっている。

当初、保存番号は利用者に入力してもらう形式でメニュー作成ページを製作していた。(資料 11 参照)。

しかし、保存番号の概念を利用者に理解してもらう必要があるという問題があった。5 桁の保存番号は私個人が考えた概念であり、一般的に普及しているものではない。利用者には難しく感じられる可能性が高く、また、利用者に指定してもらう方法を採用すると、保存番号の重複、親メニューを持たない子メニューの作成(保存番号 20000 のメニューが存在しないのに 21000 のメニューが存在する)など、多数の問題が出てくるため、保存番号は利用者が定めるものではなく、メニュー作成時に自動的に決定し作成されるシステムに変更した。(資料 12 参照)。

自動的に保存番号を定める仕組みは、以下のようにして製作した。

例えば、TOP のページに表示される 1 の階層のメニューを作成するとする。まずデータベースに登録されているメニューから、保存番号 $\$menu\_b$  から $\$menu\_e$  が全て 0 であるメニューをカウントし、その数に 1 を足した数を $\$menu\_a$  とし、 $\$menu\_b$  から $\$menu\_e$  を 0 として登録する。(資料 13 参照)。

その下位に作成する子メニューも、親メニューの番号を受け継ぎ、同様の仕組みで作成されるようになっていて。例えば、保存番号 40000 のメニューの下に子メニューを作成する場合、保存番号 40000 のメニュー下にまだ子メニューが作成されていなければ \$menu.a=4 の値を受け継ぎ、 \$menu.b にはカウント数 + 1 で「1」の値が定められ、保存番号 41000 のメニューが作成される。(資料 14 参照)。

### (3) メニューの削除に関して

メニューを削除する仕組みだが、メニューを削除した場合、そのメニューを親とする子メニューも同時に削除される仕組みをとっている。だがここで問題がある。

例えば保存番号 20000 (\$menu.a=2 \$menu.b=0 \$menu.c=0 \$menu.d=0 \$menu.e=0) と下位にある子メニューを削除した場合、保存番号 30000 以降の番号を持つメニュー全ての \$menu.a の値を 1 ずつ引かなければ、次にメニューが作成される際に支障を来すということである。

具体的にどのような支障を来すのかというと、上記で述べたように 1 の階層にメニューを作成する際、 \$menu.a 以外つまり \$menu.b ~ \$menu.e が全て 0 の保存番号を持つメニューをカウントし、その数に 1 を足して保存番号を定めてメニューを作成しているので、保存番号 20000 のメニューが削除された場合、カウントされるメニューは、保存番号 10000、30000 となりカウント数は 2、そしてカウント数に 1 を足した数を \$menu.a とし \$menu.b ~ \$menu.e を 0 とした保存番号 30000 を持つ新規メニューが作成される。だが、保存番号 30000 のメニューはすでに存在しており、保存番号の重複というエラーを招く。(資料 15

参照)。

よって、メニューを削除する場合、行わなければならない処理として、削除の対象となるメニューとそのメニューを親とする子メニューの削除。そして、削除対象のメニューと以降の保存番号の変更及び更新が必要である。

具体的には、削除処理を行うと同時に、削除対象のメニューの保存番号を読み取り、以降変更が必要な保存番号を持つメニューが1ずつ引かれてデータベースに更新されるという仕組みをとっている。

#### (4) サイトの表示

サイト表示を行うにあたり、ここでエリアの名称を定義しておく。(資料 16 参照)。

サイトは4つのエリアに分かれており、予めそれぞれに表示する内容を定めている。

サイトの上下の位置にあるのがヘッダーとフッターで、このエリアにはサイト名を表示する。メニューが表示される左のエリアをメニューエリアとし、ページ情報が表示される右側をメインエリアとする。

ヘッダーとフッターエリアはサイト名のみ表示なので、変化があるのはメニューエリアとメインエリアの2つとなる。

まず、ヘッダエリアにサイトの基本情報をデータベースから呼び出しサイト名を表示する。

サイト名は利用者が定めるものであるため変数であり、ウィンドウのタイトル箇所も `< title ><?php echo $title;?></title >` と記述する事により反映が可能である。

次に、表示すべきメニューの項目をデータベースから引き出し、メ

ニューエリアに表示する。そして、次にメインエリアに表示すべきページの情報をデータベースから引き出し表示し、最後にフッタエリアに最初同様にサイト名の情報を読み込み表示する。

### (5) メニューの表示

メニュー項目は上記で定めたようにメニューエリアに表示される。

このシステムでは、サイトにメニューを表示させるにあたり、親メニューや子メニューというように、上下のメニューも表示させる仕組みをとっている。(資料 17 参照)。

このツールで作成される Web ページは 5 階層にも及ぶ階層構造になっているため、ブログと違い、保存できる場所が多様に存在する。そのため同階層のメニューだけでなく、上下の階層のメニューも表示しなければ、閲覧者には、現在どの階層にメニューにいるのか判断しにくく感じられる構造になると考え、この方法をとった。

実際には、ツリー形式を用い、資料 10 の階層構造に近いものを表示するという工夫を施した。ただし、5 階層全ては表示せず、上下の階層のみの表示とする。理由は限られたメニューエリアに 5 階層ものメニューを表示するには、スペースに問題があるからである。

この形式は、開いているメニューの保存番号を 11000 とすれば、まず、その親メニューである保存番号 10000 を表示し、次に対象となる保存番号 11000 のメニュー、そしてその下に子メニューである 11100 や 11200 を表示し、次に同階層とである 12000 や 13000 を表示させるようにプログラムを組んだ。メニュー名を表示する際、階層であることがわかるように空白や記号を用いてツリー形式に近い状態にした。(資料 17 の例参照)。

ただし、最初のページである TOP は例外とし、上記とは別の形式でサイト表示を行っている。TOP の表示に関しては以降の「3種類のサイト表示」にて述べる。

## (6) ページの表示

メインエリアに表示されるページの情報、日付順に1件ずつ表示される仕組みで、前後にページがあるならば、ページの下に「前へ」「次へ」と表示され、そこから前後のページへ移動ができるようにした。利用者がページの順番を入れ替えたい場合は、管理画面の『ページ情報』の編集にて日付を書き換えれば可能である。

「前へ」「次へ」の表示はデータベースで前後のページが存在するか否かを判断した上で表示されなければ、空白のページがメインエリアに表示されてしまう。ページの移動の仕組みに関して、資料 18 を元に解説すると以下ようになる。

現在「ページ 3」を表示しているとした場合、「ページ 3」のメニューは1の通り「メニュー 1」である。同じ「メニュー 1」のページを「ページ 3」を除いて割り出すと2の通り「ページ 1、4、6」が該当する。

「前へ」のリンク表示の有無を判断するにはまず、「メニュー 1」で「ページ 3」の日付までのページをカウントする。この場合は「ページ 1」と「ページ 3」の2つであり「ページ 3」は「メニュー 1」で2番目である事がわかる。これを $\$m\_c=2$ とする。 $\$m\_c=1$ であったならば、1番目のページであるため、前にあるページは存在しない事になるので、その場合は「前へ」のリンク表示をしない。

資料の例の場合 $\$m\_c=2$ であるので、前のページが存在する事になる。前のページを特定するには、前のページが「メニュー 1」で何番

目かを割り出す必要がある。そこで $\$m\_c$  から 1 を引き、前のページが「メニュー 1」で 1 番目のページだということがわかる。

同じように「次へ」のリンク表示の判断だが、まず「ページ 3」を含む「メニュー 1」のメニューであるページを全てカウントする。例ではカウント数は 4 となり、これを $\$m\_c\_gokei$  とする。上記でカウントした $\$m\_c$  が、 $\$m\_c\_gokei$  と同じならば、次のページは存在しない事になる。

資料の例の場合、 $\$m\_c=2$ 、 $\$m\_c\_gokei=4$  と異なるので、次のページが存在する事になる。次のページを特定するには「前へ」と同様に、次のページが何番目かを割り出す必要がある。 $\$m\_c$  に 1 を足して、次のページが 3 番目のページだという事がわかる。

後は、「メニュー 1」のページを何番目か指定して、情報をデータベースから引き出し、取得したページの id を、ページ id からサイトを表示させるページへと送信すれば良い仕組みになっている。

### (7) 3種類のサイト表示

このツールによって作成された Web ページは、3 種類の方法で表示される仕組みをとっている。(資料 19 参照)。

1 つは TOP のページである。これはサイトの入り口にあたるので、メニューエリアに最新記事が表示され、メインにはサイトの基本情報である、サイト名、管理人名、サイトの紹介文が表示される。このページは上記で説明したような、階層構造を表示させるメニュー方式はとっていない。理由は TOP であるため、メニュー項目以外に最新記事といった情報の記載があり、メニューを階層構造で表記しては、TOP のメニューエリアの情報量が多くなり、閲覧者を惑わしかねないと判断し

たためである。

また、大きく違うのは、このページは TOP であるため、送信されてくる id を元にサイト表示は行っていない事である。

2つ目と3つ目は上記の階層構造をメニューに表示させる方式をとっているという点、id を元に表示しているという点においては双方同じであるが、送信されてくる id の種類が異なるため別ページとした。取得元の id がメニューの id であるパターンと、ページの id であるパターンの2種類あるのである。

メニューの id で表示される場合は、メニュー項目を選択したことによりサイト表示に変化をもたらす場合である。ページの id で表示される場合は、最新記事や、前後のページに移動するなどページの情報を元に次の表示を行う場合である。(資料 20 参照)。

メニューの id を元にサイトを表示する場合。まず、メニューエリアに取得したメニューの id を元に親メニューや子メニューの情報をデータベースから引き出し表示する。そして、取得したメニューの id を持つページの中から、日付の古い1件をメインエリアに表示する。これが、メニューの id を元にサイトを表示する順序である。

ページの id を元にサイトを表示する場合。メニュー id のパターンと同じように、まず、メニューエリアにメニュー一覧を表示しなければならないが、取得している id はページの id であるため、まず、ページの id を元に、メニュー id を引き出さなければならない。メニューの id を取得した後は、上記と同様の処理が行われる。

このように、1つのページに2種類の id が読み込まれるため、データの上書き、また混同を避けるために、製作時には、メニューの id とページの id の2つを取得時に別の変数に代入し、id を保存するよう仕

組みを取る必要があった。

## (8) サイト配色変更システム

上記の「ブログの利点」で述べた通り、利用者がデザインに少しでも手を加えられるようにと製作したシステムが、サイトの配色変更システムである。

背景色や、文字サイズなどをデータベースに登録しておき、サイトを表示する際、CSS として書き出し、サイトの配色等を決める仕組みをとっている。テンプレートがデータベースに保存されているため、以前使った配色を再度利用する事が容易であるという利点がある。

サイトの配色を変えたい利用者は、これを使ってテンプレートを新規に作成してもらい、特にこだわらない利用者には、用意してあるサンプルを利用してもらう仕組みになっている。

まず、利用者に管理画面の『サイトのデザイン変更』のページで、テンプレート名及び、背景色や文字色といった色コードや文字サイズなどを入力してもらい、それらの情報をデータベースに登録してもらう。

配色対象となるヘッダー、フッター、メニュー、メインのエリアは <div> により予め定めている。Web ページを作成をした事がある者には、ヘッダーやフッターという概念は比較的知られているが、HTML 初心者などにはわからない場合がある。そこで入力項目をエリアごとに色分けし、対象となるエリアがどの部分かわかるように、入力項目の隣に部分図を表示させた。(資料 6 参照)。

利用者によって作成されたテンプレートは、管理画面の『サイト基本情報』のページで『サイトデザインの変更』から変更が可能である。また、サイトを作成してからテンプレートの作成が可能になるので、サイ



ト作成時は、サンプルとして sql ファイルに予め用意してある 2 つのテンプレートからデザインを選んでもらうようになっている。

テンプレートを選択する際、実際どのようなテンプレートなのか、予め利用者が把握できる方が利便性が高いと考え、登録されているテンプレートの配色具合を視覚的に確認できるようにした。

具体的には、テーブルタグを使って、サイトの縮小バージョンをつくり、読み込んだテンプレート一覧を視覚的に表現したのである。(資料 7 参照)。

テンプレートを削除するにあたり、削除できるものとできないものとを識別する必要があった。

削除できないものと定めたのは使用中のテンプレートと、sql ファイルに予め用意してあるサンプルとして用意したテンプレートである。識別した理由は、サイトを表示するにあたり問題が生じるためである。使用中のテンプレートのデータがない場合、選択できるテンプレートがない場合、ともに CSS でデザインが組まれているサイトであるため、CSS 部分のデータがなくなることにより、資料 7-a のようにサイトが表示されてしまうのである。

### 3 結果

#### (1) モニター結果

今回、このツールのモニターには人文情報学科三回生の 7 名に協力して頂いた。

メニューやページの作成自体は容易との評価を頂いたが、作成したメニューやページがどこにあるのか把握し難いという指摘があった。

原因は、メニューのシステム自体は階層構造になっているが、利用者

がそれを管理ページにて視覚的に確認できないという事であった。子メニューがどこにあるのかを確認するには、親メニューを開かなければいけないという構造になっていたのである。子メニューがどこにあるのか把握が難しかったため、その中に作成したページの場所も把握が難しくなっていたと考えられる。

サイトの配色変更システムに関しては、入力する各エリアが色分けされていて、わかりやすかったという意見があった。

また、サイトの配色を変えるだけでなく、サイトデザイン変更システムとして、より機能の充実を求める意見が多く。サイトを作成する上で、デザインにこだわる利用者が多い事がわかった。

## (2) メニューシステムの課題と改善策

作成したメニューやページの把握し難い事に関しては、管理画面でも、ツリー形式などを用い、子メニューも含めて1つのページで表示できるようにして、メニューの階層構造を利用者に視覚的にわかりやすくする必要がある。また、メニューの一覧から、そのメニューに作成されているページ数などの情報も確認できれば、より利用者にわかりやすくなると考えられる。

上記のように改善するとして、5桁の保存番号全てを取り入れた変数 \$menu\_no を新たに設け、order by で保存番号 \$menu\_no の値が小さいメニューから順に表示するという案がある。ただし、このまま表示したのではツリー形式にならないため、表示する前に読み込むメニューがどの階層であるかを識別し、下位のメニューには空白など前に挿入して階層構造を視覚的に表す必要がある。

なぜ新たに \$menu\_no という変数を設ける必要があるのかというと、

現段階でメニューの保存番号は、各桁ごとに\$menu\_a~\$menu\_e というように単体で変数名を持ち、データベースに登録されているので、例えば保存番号 10000 のメニューの場合、実際にデータベースに 10000 の値は登録されておらず、\$menu\_a=1 \$menu\_b=0 \$menu\_c=0 \$menu\_d=0 \$menu\_e=0 と、全て単体で値が登録されている。そのため、単体での保存番号を利用して、子メニューを含む全てのメニューを保存番号順に一覧表示させようとなると、条件式が複雑になり製作が困難になる。よって、新たに\$menu\_no という変数を設ける案を提示する。

\$menu\_no の値を定めるには、単体で定められた各桁の値を用いる。 $\$menu\_a \times 10000 + \$menu\_b \times 1000 + \$menu\_c \times 100 + \$menu\_d \times 10 + \$menu\_e \times 1$  が\$menu\_no である。メニュー作成時、及びメニューを削除し更新される時に、\$menu\_no の値を割り出し登録する必要がある。

その他、モニター後に気がついた改善すべき点について述べる。

まず、ページが作成されていないメニューは、サイト表示にてメニューエリアに表示されないようにすべきであったという事があげられる。メニューをクリックしてもページが作成されていないので、空白のページが表示されてしまうのである。

さらに、ページが作成されていないメニューには、子メニューの作成ができないように制限をしておくべきだったと感じた。

理由は、目的のページにいく前にページが作成されていないメニューを通る事になり、それはつまり空白のページを通ることになるからである。よって子メニューを作成する条件として、そのメニューが1つ以上ページを作成していることを付け加える必要があったと考えられる。

### (3) サイトの配色変更システムの課題

配色変更だけでなく、サイトデザイン変更システムとして、より機能を求める意見が多数あったが、メニューシステムに重点をおいた Web ページ作成ツールであることを考えれば、今後の課題とする必要はない。

しかし、「ブログの利点」で述べた通り、私は「Web ページ作成においてデザインは重要性の高いものであり、ツールとは利用価値があつて意味のあるものである」と考えているため、サイトデザイン変更システムの向上も今後の課題としてとりあげたい。

今回製作したテンプレートの作成システムでは、ヘッダー、フッター、メニュー、メインといった各エリアの制限が比較的多く、利用者が自由に変更できる箇所が少ない。モニター後に頂いた意見の中には、上記の幅なども指定できればよかつたともある。確かに、メニューのエリアの幅は予め設定されているため、作成したメニューの名前の長さによっては、バランスの問題もでてくるだろう。

また、デザインの重要性を考えるならば、画像の利用も今後の課題になる。ツール製作の際に画像のアップロードも顧慮に入れて製作に当たっていたが、あまりにもアップロードに時間がかかる事、また、アップロードした画像を効率よく利用できる状態にできなかった事から、今回のこのツールには取り入れていない。

今後、画像のアップロードを取り入れるならば、アップロードに時間がかかってしまう原因の究明と、アップロードした画像を利用者が容易に扱える工夫を施す事が必要である。

## 4 今回の製作で学んだ事

### (1) 当初のツール製作の概念

私は今回、このツールの製作に当たって、2つのワードを気かけながら製作に当たった。

それは「簡易」と「自由」である。

「簡易」は言葉の通り、利用者にとって利用しやすいという事、また、面倒だと思われる手間が省かれる事である。そして「自由」とは利用者がこのツールで自分が思い描く Web ページを作成できる事、自分の道具として自由自在に扱う事ができるかという事である。

私はこの2つの両立が最高の Web ページ作成ツールだと考え、製作にあたった。

### (2) 矛盾

しかし実際に製作して、一種の矛盾を感じた。利用者にとって簡易なツールを製作するには、プログラム上で利用者の行動を制限しなければならないのである。

例えば、今回重点を置いたメニュー作成システムだが、5桁の保存番号という制限を設けた。これは利用者が自由に定めるものでなく自動的に定められるものとした。理由は、保存番号を利用者が定める仕様では、利用が難しくなるからという理由の他に、プログラムを組む上での問題があった。

そもそもプログラムは、ある前提があって組まれるものである。その前提が多種多様であればあるほど、製作者は様々なパターンに想定し、対応できるプログラムを組まなければならない。しかし、どんなに多種多様に対応できるプログラムを組んでも、結局、決まった枠組みがある

のは、変える事のできないプログラムの本質である。

簡易であるためにプログラムを組む事は、制限を前提を設けるという事である。しかし、制限を設けるというのは、当初、私が目指した「簡易」と「自由」の両立に背く事になるのではないかという考えに至る。

製作後、私は、プログラムでツールを製作するにあたり「簡易」と「自由」は、両立できるものではないのだと、理解した。

### (3) 製作後のツール製作の概念

製作後に気がついたのは、ツール制作の際に考えるべきなのは「利用者が自由に使える」ではないことである。

もともと「自由」とは曖昧な表現であり、利用者が自由に使えるという事は、利用者が0から作り得ることをも意味する。それは簡易ではなく、利用者が最初から最後まで責任を持って作成するという事である。

確かに0から作成できる事は自由以外のなにものでもないが、しかし利用者が求めている自由とは異なる。利用者が求めるのは、自分が作成するにあたり、何かしら障害がないこと。つまり「不自由を感じない」ことなのである。

故に念頭に置くべきであったのは「利用者が自由に使える」ではなく、「利用者に不自由を感じさせない」「不満を感じさせない」といった事である。

「自由」と「不自由を感じない」は同義語ではない。不自由を感じないということは、何かしら制限があってもそれが利用者の障害にならないという事である。

問題は、どのようにすれば利用者に障害を与えない制限を設ける事が

できるかである。これは、試行錯誤を繰り返し、改善していくよりないだろう。

#### (4) 製作者に必要な3つの視点

また、ツールの製作には大きく3つの視点から考え、製作していかなければならない事を学んだ。

1つは、プログラムを組むというコンピュータに近い視点。2つは、製作したツールを扱う利用者の視点、管理画面や作成画面などである。3つめは利用者がツールを使って作成したものを、閲覧もしくは利用する第三者の視点、つまりサイト表示で見られる視点である。

この3つのいずれかが不足しても、利用しやすいツールとは言えない。

今回の製作においては、モニター結果から2つめと3つめの視点が十分でなかったと考えられる。

2つめ3つめは、システムの仕組みだけでなく、視覚的な部分、デザイン力が試されると感じた。どこにどのように配置して表示するかといった事が、想像していたよりも重要であったのである。

なぜなら、どれほど考えシステムを組み立てても、利用者や、閲覧者が見にくい使いづらいと判断したのならば、意味のないツールになってしまうからである。

以上が、今回取り組んだ製作で学んだ事である。今後、ツール製作を行うにあたり、今回学んだ事を活かし、製作にあたりたい。

——ブログ風 Web ページ作成ツールの製作について——

## 文献表

小島まさご 2007 『実践マスター PHP+MySQL』株式会社ソーテック社

2007 『プロとして恥ずかしくない WEB デザインの大原則 改訂版』エムディエヌ・ムック