

CSS オンライン教材の作成について

山中健至

目 次

1	テーマ	1
1	テーマ	1
2	テーマ選択の理由	1
3	教材の対象者と目的	2
4	過去の似た教材について	2
2	計画	4
1	CSS の学習	4
2	教材の方針	4
3	教材作成の注意点	5
4	XHTML 使用の理由	5
3	実際の制作	6
1	教材の構造	6
2	教材のレイアウト	7
3	教材で使用した画像について	9
4	章とセクションの内容について	12
5	教材のリンクについて	19
4	評価	19
1	アンケート実施について	19
2	アンケート結果とそれに基づく自分の考え	20
3	自己評価と今後の課題	22

1 テーマ

(1) テーマ

私の所属するゼミでは「人の役に立つものを作る」という大きなテーマを掲げている。そして、人の役に立つものとはどういったものかという考察に基づいて私は、CSS⁽¹⁾に注目した。そして、その CSS を学習するための CSS オンライン教材を制作物とすることに決め、それに伴って「CSS オンライン教材の作成について」を卒業論文のテーマとした。

(2) テーマ選択の理由

私が CSS に注目した理由としては Web ページの観覧がきっかけである。私は普段、様々な Web ページを観覧しているが、そういった Web ページ観覧をしている最中にページのデザインが気に留まった。そしてこのページのデザインを施しているのが CSS であり、それに伴って CSS を学習しようと思った。HTML⁽²⁾が文書の「構造」を指定するための言語であるのに対し、CSS は Web ページの「視覚表現」をコントロールするための言語である。つまり、CSS を使用することによって Web ページにデザインを自在に付加させることができるのである。

私たちは普段、何気なく Web ページを観覧している。では、私たちにとって Web ページの価値とは何かと考えたとき、その価値の大半が内容、つまりコンテンツにあると言える。しかし、いかに充実したコンテンツであっても、単にそれを HTML で構造化して表示してあるだけの Web ページを見て、私たちは満足するであろうか。私たちが、Web ページを観覧するにおいてコンテンツと同様、必然的に求めているものはデザインであると言える。デザインの付加、つまり視覚表現の違いによって、Web ページ観覧における私たちの価値観も変わってくるので

はないだろうか。様々な Web ページにおいて、このデザインを有効に付加することが、ページ全体の価値を高める上で重要だと言える。

例えば、自分のホームページや Web サイトを作成したい時にも、この CSS の技術が必要になる。また、Web ページを観覧する際にも、デザインの構図を理解することで、ページ観覧における価値観も変わってくるのではないか。そして、CSS を初歩から学びたいと思っている人が、CSS を学習できるように、CSS オンライン教材を作成することに決めた。

(3) 教材の対象者と目的

この教材は CSS の初心者を対象としている。CSS をまったく知らない人、CSS を少しだけなら知っているという人の更なる CSS 技術の向上を目的としている。本教材では CSS を使うにおいて最もよく使用する必要最低限の技術を学習できるものとなっている。

(4) 過去の似た教材について

大谷大学福田ゼミ人文情報学科の 2005 年度の卒業論文「Web 標準デザインの初心者用オンライン教材の作成」、2007 年度の卒業論文「CSS を用いた Web デザイン・応用教材の作成について」において、同じように CSS の教材が作成されている。ここでは、過去に作成されたこれらの教材の問題点に加え、本教材を作成するにあたっての意図について述べる。

まず、2005 年度の教材においては HTML と CSS 両方を学習するための教材になっており、大半に HTML の内容が取り上げられているため、CSS については少ししか触れられていない。そして 2007 年度の教

材において、2005 年度の教材の CSS に関する内容の少なさが指摘されており、2007 年度の教材の内容は、いわゆる 2005 年度の教材の応用編になっている。そして、2007 年度の教材では、2005 年度の教材では取り上げられていない、更に高い技術を要する内容が取り上げられている。しかし、この 2 つの教材においていくつか気になる点があった。

まず 2005 年度の教材は先ほども述べたように CSS については一部しか触れられていない。そして内容の説明も至って簡易に述べられている。これでは CSS を使用するに於ける基礎レベルとまでもいえないと言える。しかし、2007 年度の教材では 2005 年度の教材がリンクとして貼られており、2005 年度の教材を使用して、まず基礎を学んでから取り組むように指摘してある。しかし、2005 年度の教材の内容だけでは基礎としては少なすぎるため、不十分な知識で 2007 年度の教材を始めしてしまうことになり、CSS の基礎の部分が抜けてしまう。そこで、まず基礎として必要な最低限の知識を先に取り上げるべきだと考えた。

2007 年度の教材の内容については技術レベルが上がっているのにも関わらず説明が簡易に感じられた。複雑なレイアウトを施す技術ほど手順を踏んで説明していかなければ概念を理解することはできない。教材の対象者が CSS の初心者となることおさらのことである。従って、内容はしっかりと理解すべきことから手順を踏んで説明するべきだと考えた。

2005 年度と 2007 年度の教材におけるこれらの気になる点を踏まえた上で、本教材では、この 2 つの教材の要素を取り入れた CSS オンライン教材を作成するのが目的である。

2 計画

(1) CSS の学習

まず CSS の教材を作成するにあたって自分が CSS の知識をつけなければ話にならない。教材作成の第一段階として CSS の学習を始めた。学習方法としては、まず過去の 2 つの CSS の教材を参考にして CSS を基礎から学習した。更に、とほほの WWW 入門⁽³⁾というサイトに CSS の要素が多数載せられているので、それを参考にして必要な要素を補う形で学習した。そして、一通りの知識が身に付いた後は、実際に自分自身でページを作成して CSS でレイアウトを施しつつ試行錯誤した。大谷大学人文情報学科のホームページの HTML と CSS のソースを参考にすることもあった。そうして学習を続けていき CSS において必要性の高い要素を模索していった。

(2) 教材の方針

CSS の学習を終えたところで、次の段階として教材の方針について考えた。CSS の学習にも使用したとほほの WWW 入門というサイトには多数の CSS の要素が載せられている。これは過去の教材のリンクにも貼られているものであり、これによって CSS の学習をすることは可能である。しかし、このサイトは 1 つ 1 つの要素の内容の説明が文章のみで綴ってある辞書のようなものであり、要素の意味や記述方法は理解できても実際に表示させたときのレイアウトが載せられていないので、扱い方が理解しづらいと思われる。

そこで、教材とは何かと考えた時に、やはり初心者でも理解しやすい内容ということ念頭におく必要がある。私自身が CSS を学習し始めた時の気持ちになって、どうすれば理解し易いかを考察した結果、

HTML と CSS のソース、ブラウザ表示時の画像をそれぞれ貼った上で説明を付加するという進め方を考えた。文章のみで説明するよりも教材使用者が実際にファイルを作成しながら試行錯誤して進めていくというのが一番好ましい方法だと判断したためである。

(3) 教材作成の注意点

ここでは教材作成にあたって注意すべき点の考察について述べる。まず始めに浮かんだものがページレイアウトであった。CSS を使用することによって様々なページレイアウトが可能になるが、本教材では CSS 初心者が対象になるので、ページ全体がシンプルで見やすい方が学習意欲を向上できるのではないかと考えた。複雑なレイアウトで様々なリンクをページ上に配置するとかえって任意の場所が探しづらくなるためである。次に 1 つのページの内容を適度な長さにする必要があると考えた。1 つのページをあまり長くしすぎると画面スクロールが長くなってしまい全体のバランスが悪くなってしまうためである。更に、本教材では説明に少し長めの文を用いている箇所がいくつかあるので、そういった中でいかに読みやすく文を綴るかにも要点をおいた。

(4) XHTML 使用の理由

本教材では、ページを作成する構造言語として XHTML⁽⁴⁾ を使用して進めている。この理由としては、HTML、XHTML、XML⁽⁵⁾ の 3 つの構造言語の内、現在では XHTML を使用するのが一般的となっているためである。XHTML が一般的となっている理由としては次のようなことが挙げられる。

XHTML では使用できる要素や属性が旧来の HTML とほぼ共通す

るため、習得するのが容易であることが挙げられる。次に、XHTML は HTML に比べて厳しい書式を備えているため、処理に与える負荷が軽減されるということが挙げられる。XHTML では終了タグの省略が認められていない。つまり、
などの空要素のタグには
というように最後に半角スペースとスラッシュを付ける必要がある。このように、より厳密な書式で構成されることによって、データ処理が高速化され、ブラウザでの安定動作が促進されるのである。

なお、本教材は CSS の技術を習得する目的で作成されているものであるため、これらのような XHTML についての説明は省いている。

3 実際の制作

(1) 教材の構造

本教材は大きく分けてトップページと第一章から第五章までの5つの章で構成されている。章を55つに分けた理由としては、本教材で扱う CSS の技術内容を考慮し、段階ごとにグループ分けした所、5つが妥当と判断したためである。各章の内容はそれぞれ番号順で複数のセクションに分かれている。これは、1つの章に1章分の内容を全て記述してしまうと1ページが長くなりすぎてしまい、非常に使いづらく効率が悪いと判断したためである。なお、各章ごとのセクションの数については各章で扱う内容によって異なるので統一はしていない。

各セクションの内容、つまり本文にあたる部分はすべて1ページに収められている。この理由としては、本文の内容を更に複数のページに分けてしまうとレイアウトが複雑になってしまい、かえって使いづらいと判断したためである。本文の内容は、教材使用者が読みやすく、全体的に綺麗にまとまるようそれぞれ短めになっている。

つまり、本教材は「章」、「セクション」、「本文」の3つの要素に分かれており、章からセクションへ、セクションから本文へとリンクしている構造である。初心者には物事を学習させるにはきちんとした手順が必要であると私は考える。更に、複雑で難しい表現は学習意欲を低下させてしまう可能性が高い。章、セクション、本文の3つの要素で全体を繋げる構造とすることが、本教材を初心者でも理解しやすいシンプルかつ有効な内容にまとめるのに適していると判断した。

(2) 教材のレイアウト

本教材を快適に使用してもらうために重要になってくるのがページレイアウトである。CSSで施せるページレイアウトは様々であるが、私は、「サイドスペース」、「メインスペース」の二段組に「ヘッダー」を付加した至ってシンプルなレイアウトを選んだ（図1参照）。

このようなレイアウトにした理由としては、先ほどの章、セクション、本文の3つの要素を収めるのに最適な段組であると同時に、シンプルなレイアウトにより初心者にも見やすいと判断したためである。ページ中央上部に章、ページ左側にセクション、ページ中心部に本文を配置している。つまり、章とセクションがメニューとしてそれぞれへとリンクしており、本文が実際に説明を記述していく部分にあたる。トップページを除く全てのページにこの段組が適用されている。

まず、ヘッダーの部分に章がリスト表示でレイアウトされている。章はCSSで横向きにし、メニューバーとしての装飾が施されている。サイドスペースにはセクションがリスト表示で縦向きにレイアウトされている。

ここで、この章とセクションの2つの位置関係について説明する。

ページ全体のバランスと見やすさを考慮すると、章とセクションの位置は左サイドとトップの二箇所に分ける必要があると判断した。ここで、章とセクションとで、どちらを左サイドに配置し、どちらをトップに配置するかを考察した。章のリスト数は、トップページを含んだ6つと一定なのに加え、章のタイトルもトップ、第一章から第五章と統一されているのに対し、セクションのリスト数はそれぞれ異なり、セクションタイトルの長さもそれぞれ異なる。つまり、多数のリスト数を含んだセクションが横向き表示されるとセクション全体の横幅が長くなってしまいレイアウトのバランスを崩してしまう恐れがあるという理由から、章をトップ、セクションを左サイドに配置することにした。セクションを左サイドに配置することでリストが縦表示されるのでリスト数が多い場合でもレイアウトに支障をきたすことはなくなった。

メインスペースでは、セクションで選ばれた CSS の要素の内容を説明していくのである。全体のレイアウトに使用されている色に関しては、画面全体に薄い青、サイドスペースとメインスペースに白よりも少し濃い色と、全体的に薄い色を使用している。この理由としては、サイドスペースとメインスペースにはセクションと本文が入るので、教材使用者に文字がしっかり読めるようにするためである。白ではなく、白よりも少し濃い色を選んだのは本文に使用されているサンプル画像と背景との区別を付けるためである。白にしてしまうと本文において画像と背景が同化し、区別が付かなくなってしまう。逆に、白以外の色を使用してしまうと今度は文字が読みづらくなってしまう上、全体背景色との調和も悪いので全体的なバランスを考慮し、白よりも少し濃い色を使用した。

なお、本教材では、HTML に CSS を反映する方法として、外部に

CSS ファイルを作成して反映させるという方法を取っている。この方法以外にも HTML に CSS を反映させる方法としては、HTML ファイルの内部に CSS を記述するためのタグを入力し、その中に CSS の指定を記述することによって反映するという方法が存在する。内部 CSS のメリットとしては、1つのページで HTML と CSS の2つの記述ができるため、CSS ファイル作成の手間をなくし、複数のページを作成する際のファイル数を抑えることができる。しかし、この方法は HTML の内容量が大幅に増加し、ページ内が非常に混雑してしまう。そうになると、画面が見づらくなってしまい、記述の際にミスが生じやすくなってしまいうのではないかと考えた。HTML と CSS のファイルをそれぞれ作成することで、ファイルの内容量も適度に収まり、内容をしっかりと把握することが可能であると判断した。従って、CSS ファイルを外部に作成し、反映させる方法を取った。

(3) 教材で使用した画像について

教材を作成するにおいの方針で挙げたように、本教材はサンプル画像を用いて進められている。本教材では、グラフを使用して HTML ファイル、CSS ファイル、ブラウザ表示画面を画像ファイル化し、本文に使用している。ここではサンプルを載せる方法についての考察を述べる。

本文中に HTML と CSS の記述方法とブラウザ表示のサンプルを載せる方法として最初は、本文の HTML にサンプルのソースを記述して表示させる方法を考えていた。しかし、この方法には様々な問題があった。まず、本文に説明したい内容のサンプルを全て載せなければならないので、本文の HTML が非常に複雑になってしまうのである。HTML

ソースのサンプルを記述する際、普通にタグと文章を記述してしまうと、タグがブラウザ表示に反映されてしまうのでソースのサンプルとしての役割を果たすことができない。つまり<p>のように記述できないのである。<>を HTML で表示させるには< と> を使用しなければならない。< は<、> は>の役割を果たす。つまり、HTML で<p>とブラウザ表示させるには文中に<p>と記述する必要がある。本教材はサンプルを中心として進めていくため、多くのサンプルを必要とする。つまり、HTML ソースをサンプルとして記述する際、タグを表示させるのに頻繁に< と> を使用しなければならず、記述にかかる手間を考慮すると非常に効率が悪いと判断した。

次に挙げた問題は、HTML で文章として表示させたソースのサンプルはコピーとペーストができてしまうという点である。本教材の趣旨は、使用者が実際に自分でファイルを作成、記述、表示させ、その過程で理解を深めるものであり、サンプルは、より理解しやすくするための手助けである。しかし、コピーとペーストができてしまうと、教材使用者が自分自身での記述を怠り、学習意欲を低下させてしまう恐れがあるということが懸念された。

これらのような理由から、本文の HTML にサンプルを記述して載せる方法は本教材には適していないと判断し、グラブを使用してファイルを画像化し、本文に貼るという方法を取った。なお、本教材の第四章では一部例外がある。ここではメニューバーの作成方法を説明しており、ブラウザ表示のサンプルを画像化せずに HTML に記述する方法で載せている。理由としては、メニューバーはカーソルをメニューに合わせることで文字色や背景色が変わるようになっているので画像化して貼ってしまうと全体の装飾が確認できるだけで、カーソルを合わせることによ

る変化を確認できないからである。

さて、この方法のメリットとしては先ほども挙げたように、コピーとペーストができないこと、HTML と CSS のファイルをそのまま画像化しているので HTML と CSS の区別が付けやすく、初心者にも理解しやすいなどが挙げられる。デメリットとしては、それぞれの HTML と CSS のサンプルのファイルを全て画像化し、本文に貼らなければならないので、画像ファイルが多くなってしまったことに加え、貼り付けに手間が掛かったという点である。

なお、本教材で使用しているサンプル画像のサイズは統一されていない。この理由としては、HTML のサンプル、CSS のサンプル、ブラウザ表示のサンプルのそれぞれ適した画像サイズが異なるためである。HTML のサンプルは、常にひな型を付加したものになっている。これは HTML ファイルだという識別がしやすいようにするためである。HTML のサンプルは、このひな型も含め、文章を記述するので、全体がしっかり表示されるようにサンプル画像はやや大きめのサイズになっている。

CSS のサンプルは、HTML のようなひな型が存在しないので比較的に内容量の少ないものになっており、初期段階では一行の入力で済んでしまうものが多い。しかし、段階が上がるにつれて内容量も多くなっていくのでサンプル画像は小さなサイズから大きなサイズまで様々である。ブラウザ表示のサンプルは、はっきり見やすいように基本的に大きめのサイズになっている。

これらのことからサンプル画像のサイズはそれぞれのサンプルの内容によって異なることが分かる。従ってサンプル画像のサイズを統一する場合、大きいサイズに統一すると内容量の少ないサンプル画像に余分な

スペースができてしまい無駄になる。小さいサイズに統一すると内容量の多いサンプル画像が圧縮されてしまい見づらくなってしまう。これらをバランス良く載せるにはそれぞれのサンプルの内容量に適した画像サイズにするのが適切だと判断した。

(4) 章とセクションの内容について

(i) はじめに

ここでは本教材の章とセクションで取り上げている内容について述べる。教材の構造で述べたように本教材の内容は、第一章から第五章で構成されており、章が進むにつれて技術レベルを上げている。

(ii) CSS の基礎

第一章では、CSS の基礎について取り上げている。内容は CSS において最もよく使用するプロパティである。そして、この第一章の内容が 2005 年度の教材の内容を補ったものになっている。第一章では、CSS の概念や書式、頻繁に使用するプロパティを理解し、CSS に慣れてもらうのが目的である。

(iii) 要素のグループ化

第二章では、要素のグループ化を取り上げている。要素のグループ化で扱う内容は、CSS のプロパティではなく、HTML で指定する要素である。しかし、要素のグループ化は CSS でのレイアウトにおいて必要不可欠となるため、本教材で取り上げた。

(iv) 段組レイアウト

第三章では、段組レイアウトを取り上げている。段組レイアウトとは、ページの内容を複数の列に分割して配置するレイアウト方法である(図 2 参照)。段組レイアウトは初心者にとって、CSS の技術で最も理

解するのに苦労する内容だと言える。2007 年度の教材に段組レイアウトについての項目があるが、段階を踏まずに説明されており、初心者の立場に立って考えると、理解するには少し難しいように思える。ここでは、初心者が理解しやすいという目的に基づいた段組レイアウトの説明と作成手順について述べる。

段組レイアウトを使用していないページは、ブラウザ表示させたときの画面全体がページ外枠となっており、画面全体の大きさを変えると当然ページの外枠の大きさも変わる。例えば、CSS を使用せずに HTML のみで一定の大きさのブラウザ画面を基にページを作成したとしよう。実際ブラウザ表示したときに、ページ作成の基にしていたブラウザ画面の大きさで任意の表示はできても、基にしていたブラウザ画面の大きさよりも画面を大きくしていくと、本来は縦に並んでいた文字の行が、横にずれてしまうといった現象が起こるのである。つまり、段組レイアウトを使用しなければ、画面全体の大きさによって表示のされ方が変わってしまい、安定したレイアウトができないのである。

本教材では、段組レイアウト作成の過程を複数の段階に分け、それぞれの段階をセクションとしている。段組レイアウトの構造を初心者が理解しやすいようにするには細かく段階を分ける必要があると判断したためである。まず 1 つ目のセクションとして、外枠の作成を取り上げていく。この外枠の作成は、段組レイアウトの構造を理解する上でとても重要である。段組レイアウトにおいて外枠の作成は、簡単に言ってしまうと全体画面の位置を固定する作業である。

まず HTML の記述で div を使用して外枠を作成するためのグループを作る。そして、作成したグループに CSS で width と height を使用して幅と高さを指定する。外枠の作成はこれだけの作業である。本教材

では、作成した外枠の大きさが分かるように色を付けている。この外枠は、これからこの外枠の中に段組を組み立てていくにおいての全体画面にあたり、外枠の中の要素をどのように配置しても、外枠の外側に要素が出てしまうことはない。つまり、外枠作成によって安定したレイアウトを保つことができるのである。外枠を作成するにあたっては、枠の範囲が分かるように枠内の背景色を変えておくといいだろう。

さて、次の段階として2つ目のセクションにメインとサイドの2つのスペースの作成を取り上げている。始めに作成した外枠の中に実際に段組を組み立てていく作業である。ここでは外枠の中にメインとサイドの2つのスペースを作成するので二段組レイアウトということになる。まず、外枠の作成に使用した all のグループの間に、新たに2つのグループを作成する。これは、これから作成する2つのスペースを外枠の中に収めるためである。グループを作成できれば、CSS でそれぞれのグループに幅と高さを指定していく。

ここで注意しなければならないのはメインとサイドの幅と高さの設定である。外枠の width と height がそれぞれ 500px に指定されていた場合、メインとサイドの width と height のそれぞれの合計が 500px を超えてしまうとレイアウトにずれが生じてしまう。メインとサイドの幅、高さは外枠の大きさに応じて指定しなければならないのである。それぞれのグループの指定ができれば float を使用してグループの左右振り分けを指定する。ブラウザ表示させたときに外枠の中にメインとサイドのグループがしっかりと収まれば問題ないだろう。

メインとサイドの二段組ができた段階で段組レイアウトになっていると言えるが、更にページの利便性を上げるためにヘッダーとフッターの作成を3つ目のセクションとして取り上げている。ヘッダーとフッター

を作成するにあたって、これまでと同じようにヘッダーとフッターにあたるグループをそれぞれ作成する。なお、ヘッダーとフッターのグループは外枠のグループである all の外側に作成する。この理由としては、ヘッダーとフッターをのグループを外枠の中に入れてしまうと、メインとサイドの大きさを指定する際にヘッダーとフッターの大きさも考慮しなければならず、指定が複雑になってしまうからである。ヘッダーとフッターをそれぞれ外枠の外側に作成すれば、ブラウザ表示したときに外枠の上下に配置されるので、それらを別々に指定すればよいのである。幅を外枠の幅と同じにし、高さを小さめにすると綺麗に整ったレイアウトになるだろう。

さて、外枠、メインとサイド、ヘッダーとフッター、と段階を踏んで段組レイアウトの作成について説明してきたが、これらの応用として三段組レイアウトの作成を最後のセクションに取り上げている。この三段組レイアウトの作成は、二段組レイアウトに比べ少し複雑になるが、基本的な概念は同じであり、二段組レイアウトの作成の内容をしっかりと理解していれば難しいものではないだろう。三段組レイアウトでは、新たなサイドを 1 つ追加したメイン、サイド、サイド 2 の三段で構成される。

まず、サイド 2 にあたるグループを新たに作成する。二段組レイアウトと違うのは、all のグループの中に別々に存在していたメインとサイドのグループを 1 つのグループにまとめるということである。つまり、メインとサイドを統一したグループ、サイド 2 のグループの 2 つのグループで二段組レイアウトを作成するという概念である。なお、CSS での指定は少し複雑になる。CSS ではメインとサイドを統一したグループとサイド 2 のグループの指定をする。

ここで注意すべきなのは、サイド 2 のスペースを新たに追加するので、サイド 2 が収まるスペースを空ける必要がある。つまり、メインとサイドの幅を、サイド 2 で取りたい幅に応じて小さくしなければならない。そして、メインとサイドを統一したグループの幅をメインとサイドの幅の合計にし、残りの幅がサイド 2 を収めるスペースにあたる。最後に float で左右の振り分けをすれば三段組レイアウトになるのである。これらの技術を応用すれば、メイン、サイド、サイド 2 を統一し、新たにサイド 3 を追加することで四段組レイアウトにすることも可能である。

このように、段組レイアウトとはグループ化した要素に幅と高さを指定して 1 つの枠を作り、その枠を様々な配置させるというものである。そして、段組レイアウトは応用次第で実に様々なレイアウトを実現することができ、ページレイアウトには必要不可欠な技術である。

(v) メニューバー作成

第四章では、メニューバーの作成を取り上げている。メニューバーとは HTML タグの ul と li を使用したリストの構成を CSS で様々な向きに配置し、様々なデザインに変更したものである（図 3 参照）。過去の教材では、メニューバーの作成については取り上げられていない。しかし、メニューバーはページレイアウトにおいて頻繁に用いられる技術であり、ページの価値を高める上で重要な要素だと言える。ここでは、メニューバー作成の手順について述べる。本教材では横向き表示で、カーソルを合わせると色が変わるという一般的によく用いられるメニューバーを取り上げている。

メニューバーを作成する手順としては、まず ul と li の要素をグループ化する。そして、このグループ化した要素に CSS で指定していく訳

だが、ここで注意しなければならないのはセレクトアの指定方法である。グループ化した要素には ul と li の両方の要素が含まれているので、単にグループ名をセレクトアに指定してしまうと ul と li の両方にプロパティで指定した内容が反映されてしまい、両方に任意の指定ができないのである。両方をそれぞれ個々に指定するにはセレクトアにグループ名を指定する際、グループ名の後ろに ul と li を記述する必要がある。ul 要素を指定する場合は ul、li 要素を指定する場合は li とそれぞれ指定したい要素をグループ名の後ろに記述することによって同じグループ名でも ul と li のそれぞれを個々に指定することが可能である。

グループ化できれば実際に CSS で指定していく訳だが、まず ul と li の概念を理解する必要がある。ul はリスト全体の要素であり、li はリスト 1 つ 1 つの要素である。メニューバー作成においてはこれまでに本教材で扱ってきたプロパティに加え、新たに list-style-type、text-decoration、display、hover を使用する。ul はリスト全体の要素なので CSS で指定する際はメニューバーをページのどこに表示させるかの位置を指定するだけでよい。li はリスト 1 つ 1 つの要素なので、この li の指定がメニューバーをどう表示させるかの要と云っていい。li の指定にはまず、list-style-type を使用してリストの横に表示される記号を消す。これは、リストの横の記号を色々な形に変えるプロパティであるが、メニューバー作成においては必要ないので消しておく。次に float を使用して左右に寄せると同時に横一列に並べる。float を指定した段階でリストは横向きに表示される。リストを横向き表示にした後は、border を使用して枠線を付け、padding、margin を使用してリストの間隔を指定していく。

ここまではメニューバー全体の表示の指定になる訳だが、Web ペー

ジ等でのメニューバーの用途はリンク先のリストの表示であり、メニューバー表示されている ul、li の要素の内容はリンクを貼るケースがほとんどである。つまり、リンク部分の指定をしなければならない。リンク部分の指定をするには、セレクトア指定でグループ名、li に続いて a を記述する。a の指定には text-decoration、display を使用する。text-decoration はテキストの装飾を施すプロパティであり、リンク部分の下線を消すことができる。display は画面での表示方法を指定するプロパティであり、本教材ではブロック要素にしている。ここまでがリンク部分の指定である。

最後はメニューバーにカーソルを合わせたときの装飾の指定をする。カーソルを合わせたときの指定には、セレクトア指定の a の後に hover を記述する。hover はカーソルを合わせている状態のリンクを指定するプロパティであり、背景色やフォント色を指定することができる。ここまで指定ができればメニューバーは完成である。

(vi) まとめ問題

第五章では、これまで本教材で学習してきた内容を活かしてページのレイアウトを作成するまとめ問題になっている。これは、サンプルとなるレイアウト画像を貼っておき、そのサンプルと同じレイアウトを作成してもらうという内容である。なお、レイアウトの解答となる HTML、CSS のソースは別に保存してあるが (図 4 参照)、本教材には載せていない。この理由としては、解答のソースを載せてしまうと使用者がそれに頼ってしまい、これまで学習してきたことを自分で応用してレイアウトを作成するという目的を損ねる恐れがあるためである。また、厳密にはまとめ問題に解答という概念はない。あくまでサンプルはレイアウトの例であって、実際に使用者がサンプルを参考にし、自分自

身の方法でレイアウトを作成するというのが望ましいことである。

(5) 教材のリンクについて

本教材では、教材を進めるにあたって役に立つサイトをリンクとして載せている。ここでは、リンクを載せる方法の考察について述べる。

リンクを載せる方法としてまず考えたのは、本教材の最後にリンク専用の章を作成し、そこにリンクをまとめるというものである。しかし、リンクの中には CSS での指定に使用するカラーコードをまとめたサイトなどの実用的なものも存在しているので、最後の章にリンクをまとめてしまうと、そのようなサイトを使用しづらいと考えた。そこで、すべてのページのサイドメニューに表示させているセクションの下にリンクを載せる方法考えた (図 5 参照)。この方法で載せることによって、どのページを見ている場合であってもリンクのページに移動することができ、使いやすいと判断したためである。

4 評価

(1) アンケート実施について

教材作成後に人文情報学科福田ゼミの三回生に実際に本教材を使用してもらい、アンケートを実施した。アンケートの内容としては、「教材の良かった点」と「教材の問題点や気付いた点」の 2 つの項目を挙げた。「教材の良かった点」を項目内容にした意図としては、本教材を作成する前に挙げた、初心者が見やすく、学習意欲を削がれないようなレイアウトを作成するという、サンプル画像を貼ることによって、より実用的で理解しやすい進め方にするということを教材使用者に実感してもらえるかを確認するためである。

「教材の問題点や気付いた点」を項目内容とした意図としては、実際に教材を使用してもらう中で、使用者の立場から教材の問題点や気付いた点を挙げてもらい、それを参考にして必要に応じて教材を修正することによって、更に完成度が高まるのではないかと考えたからである。

(2) アンケート結果とそれに基づく自分の考え

さて、ここでは実施したアンケートの結果とそれに基づいた自分の考えについて述べる。まず、教材の良かった点については次のような内容が挙げられた。

- ・ レイアウトが綺麗にまとまっており文章が少なく見やすい
- ・ サンプル画像を用いた進め方で理解しやすい
- ・ 全てのページにリンクがあるので便利

これらの良かった点の結果については非常に満足のいく結果であったと実感している。初心者でも見やすく、学習意欲を削がれないようなレイアウトの実現に基づいたシンプルな段組と色使いへの考慮、また、サンプル画像を用いることによる要素を理解しやすい進め方、教材を使用するにおけるサポートとして取り入れたリンクを実用的に載せる方法などが、実際に教材を使用してもらう中で注目してもらえたということによって、初心者がより快適に使いやすくするという目的が果たせたのではないかと思っている。

教材の問題点や気付いた点については次のような内容が挙げられた。

- ・ 現在表示しているページへのリンクは必要ない
- ・ 各ページの一番下に次の項目へのリンクがほしい
- ・ まとめ問題の解答がほしい

まず、現在表示しているページへのリンクは必要ないという意見についてだが、これは実際に開いているページのメニューの現在ページを表す項目にも同ページのリンクが貼られており、クリックしてもまた同じページに行くため、何も起こらないというものであった。この意見については的を得ており、レイアウト上で必要ないものと判断し、修正した。

次に、各ページの一番下に次の項目へのリンクがほしいという意見についてだが、本教材では、本文として扱っている各ページの内容は短めにしてあるため、次の項目へのリンクを付けても、教材を扱うにおいてのメリットは少ないと考えている。また、現在見ているページの内容を学習する最中で、これまでに学習してきた前のページを再確認するケースも多く、必ずしもページが順序良く開けられるとは限らない。これらの理由から次の項目へのリンクは必要ないと判断し、修正は加えなかった。

最後にまとめ問題の解答がほしいという意見についてだが、本論文でも述べたようにまとめ問題は、教材使用者がこれまでの知識を活かして、自分自身でレイアウトを施すという目的で作成しているため、解答は載せないものとしている。本教材で学習した内容を応用すれば必ずできるようになっているので、試行錯誤しながら載せられているサンプル画像に近いものを作成できれば良いと考えている。従って、まとめ問題の解答については修正を加えていない。

このようなアンケート実施については、実際に教材を使用してもらい、様々な意見を挙げてもらうことによって、教材作成における目標が果たされているかの確認、また本教材を更に使いやすく修正するための参考になり、非常に価値あるものだったと思っている。

(3) 自己評価と今後の課題

本教材では、CSS を使用するにおいて必要最低限の内容を取り上げてきたが、初心者が見やすく、理解しやすい内容にするという目的は実現できたと感じている。本教材で取り上げた内容を応用することで、実に様々なページレイアウトを施すことが可能である。つまり、ページレイアウトとは組み合わせが重要なのである。

なお、本教材では Web ページの表現方法について追求はしていない。しかし、必要最低限の CSS 技術を身に付けた場合、ページレイアウトにおいて次に学習すべきことはレイアウトテクニックだと考える。レイアウトテクニックとは、CSS でレイアウトを施すにあたっての各要素の効率的な使い方にあたるが、これは、CSS の学習というよりは Web ページ作成の概念に近い。しかし、CSS とは本来 Web ページのデザインを施す言語である。従って、CSS の使い方と Web ページの作成を平行して取り上げるのが好ましいと言える。今後は CSS による、Web ページの作成に加え、Web ページのクオリティを挙げる表現方法について学習できるような教材があれば、更に高い CSS 技術を身に付けることができるであろう。

注

- (1) Cascading Style Sheets の略。Web ページデザインを施すための言語。
- (2) Hyper Text Markup Language の略。Web ページを記述するための言語。
- (3) <http://www.tohoho-web.com/www.htm>
- (4) Extensible Hyper Text Markup Language の略。HTML を XML で再構築した言語。
- (5) Extensible Markup Language の略。文書を構造化する言語。

文献表

- 2005 年度卒業論文「Web 標準デザインの初心者用オンライン教材の作成」坪倉由起 http://tiber.que.ne.jp/otani/html_css/sources/index.html
- 2007 年度卒業論文「CSS を用いた Web デザイン・応用教材の作成について」小松知世 <http://tiber.que.ne.jp/otani/CSSkyozai/index.html>
- 「Web 標準の教科書」益子貴寛 2005 年
- 「とほほの WWW 入門」<http://www.tohoho-web.com/www.htm>
- 「HTML 色見本」<http://www.hi-ho.ne.jp/douton/htmlcolor.html>