

## 情報基礎実習 第 8 回 HTML の仕組み

2012 年 6 月 13 日, 6 月 14 日

担当教員：逸村裕

TF：池田光雪

HTML は Web の基盤を担う言語の 1 つである。実際のほとんどの Web サイトは HTML だけでなく、他の様々な技術を併用して運用されているものが大多数であるが、本演習では最も基礎的な言語である HTML (Hyper Text Markup Language) の仕組みを学ぶ。やや発展的な内容についてはオプションとして 9 章で触れるので、興味がある人は「それまでの基礎部分を修得してから」チャレンジしてみよう。

第 7 回では HTML を使って簡単な Web ページを作ったが、今回は画像、他のファイルへのリンクなどを組み込んだ Web ページを作成する。また、Word などでも Web ページを作成し、自身がメモ帳で作成したものとどう違うかも比較する。

さらに、オプションとして HTML の検証と Cascading Style Sheets (CSS) についても触れる。

### 今回の内容

作業の前にプリントをよく読んで、作業の全体像を把握し時間配分を決めよう。また、Web ページ作成にとりかかる前に必ず 8 章「Web ページ公開の注意事項」を読むこと！

- 画像を埋め込む
- リンクをつける
- Word, Excel から Web ページを作る
- Word を使いデザイン文字、数式、画像、リンクの入った Web ページを作る
- ☆ 正しい HTML ファイルが書けたかを検証する (オプション)
- ☆ CSS を使ってレイアウトなどを変える (オプション)

### 今回の出席課題

前回の実習で作成した first.html を次の条件(1)~(4)を満たすように修正し、Internet Explorer から Web 経由で閲覧・印刷して、左上に学籍番号・氏名を記入して TA に提出せよ。なお、ファイル名は変えず、ファイルの位置も www フォルダ直下のままにすること。

- (1) 自分自身が撮影した、あるいは Microsoft Paint など自分自身が作成したものなど、著作権や肖像権の問題が無い画像が埋め込まれている
- (2) ul 要素を使った箇条書きがある
- (3) 他の Web サイトへのリンクがある
- (4) title 要素の内容が **情報基礎実習第 8 回出席課題** である

## 今回のレポート課題

最低限以下の条件(1)~(6)をすべて満たす、ファイル名を **welcome.html** とする HTML ファイルを 1 つ作成し、<http://www.u.tsukuba.ac.jp/~s13xxxxx/welcome.html> で Web に公開しなさい (13xxxxx の部分は各自の学籍番号下 7 桁とする)。レポート課題の採点対象は **welcome.html** のみとするため、CSS を使うなど、何らかの工夫をする場合は外部ファイルを作りそれを参照するのではなく、**welcome.html** 内に全て記述すること。なお、The W3C Markup Validation Service で Passed 判定が出る、さらには Another HTML-lint gateway で 100 点満点を取る Web ページは加対象とする (達成した場合は、それぞれ達成できたことがわかるように Web ページ中に明記すること)。

- (1) 内容を **情報基礎実習第 8 回レポート課題** とする title 要素を含む
- (2) ul 要素を使った箇条書きを含む
- (3) 写真が img 要素で埋め込まれている (著作権・肖像権などに配慮すること)
- (4) PowerPoint で作成したイラストが img 要素で埋め込まれている
- (5) 他の Web ページへのリンクを含む
- (6) ブラウザにおいて拡大・縮小、さらにはブラウザのウィンドウの高さ・幅を変えても表示が大きく崩れない

### ● 提出物 :

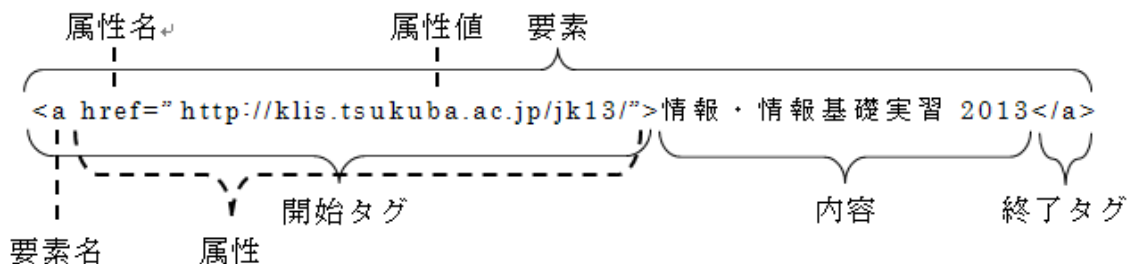
- 次の(A)~(C)を A4 片面モノクロ印刷し、この順で左上 1 箇所をステイプルしたもの
- (A) Lab2013-1.docx を元に、次の (ア) ~ (ウ) 全てを記した Word 文書
    - (ア) Web ページ作成に当たって工夫を凝らした点 (レポートの採点にあたっては内容より技術面に評価の重点を置く)
    - (イ) Word など生成した HTML ファイルと、自分で HTML タグを打ち込み作成した HTML ファイルそれぞれの特徴や長所、短所等の比較
    - (ウ) 第 8 回の演習内容で印象に残った点、及び感想
  - (B) **welcome.html** をメモ帳で開き印刷したもの。ただし、フォントはデフォルトの MS ゴシックの標準、12 ポイントのままとせよ
  - (C) **welcome.html** を Internet Explorer 9 で開き印刷したもの

- 締切 : 木曜日組 : 6 月 19 日 (水) 15:00 ; 金曜日組 : 6 月 20 日 (木) 15:00 厳守
- 提出先 : 春日エリア 2 階学務前レポートボックス
- 課題名 : 木曜日組 : 情報基礎実習 0613 ; 金曜日組 : 情報基礎実習 0614
- 備考 :

レポートを作成した後、**welcome.html** はレポートが返却されるまで一切更新しないこと。また **index.html** のような、ピリオドより前のファイル名が **index** となるファイルは **www** フォルダ直下には置かないこと (理由は 8 章の⑤参照)。

## 1. HTML の構造

HTML (HyperText Markup Language) は文書を構造化するマークアップ言語の一種である。一般に、文書は「見出し」「本文」「箇条書き」「斜体部分」など様々な単位に分割可能であるが、HTML ではこれらの単位を**要素**として扱う。HTML では、要素を示すために**タグ**を用いている。



HTML においては、一部属性値を除き、要素名・属性名・属性値に大文字・小文字の区別はない。そのため、どちらを使っても構わない(たとえば、title 要素の開始タグは<title>、<TITLE>どちらの表記でも構わない) が、HTML の後継にあたる XHTML (詳しくは 9.2. で触れる) では小文字にしなければならないため、HTML でも小文字を使う事が望ましい。以下、本テキストでは全て小文字を使って記述するものとする。

## 2. 相対パスと絶対パス

HTML においてファイルの場所を表すには**パス**を用いる。パスは**絶対パス (絶対 URL)**と**相対パス (相対 URL)**の2種類があり、絶対パスは `http://` から始まる URL を書き、web のどこに対象とするファイルが位置するのかを指定する方法、相対パスはその表記がある箇所から対象とするファイルが相対的にどこに位置するのかを記述する方法である。筑波大学春日エリアの住所に例えれば、絶対パスは「つくば市春日 1-2」という書き方、相対パスは「つくば駅から北北西に 300m」という書き方である。

学籍番号が 201399999 の学生が、www フォルダに図 1 のようにファイルとフォルダを配置した場合を考える。first.html から example.png の位置を絶対パスで表すと `http://www.u.tsukuba.ac.jp/~s1399999/example.png` となり、相対パスで表すと `./example.png` となる。相対パスにおいて、同じフォルダは「./」(ピリオドとスラッシュ)

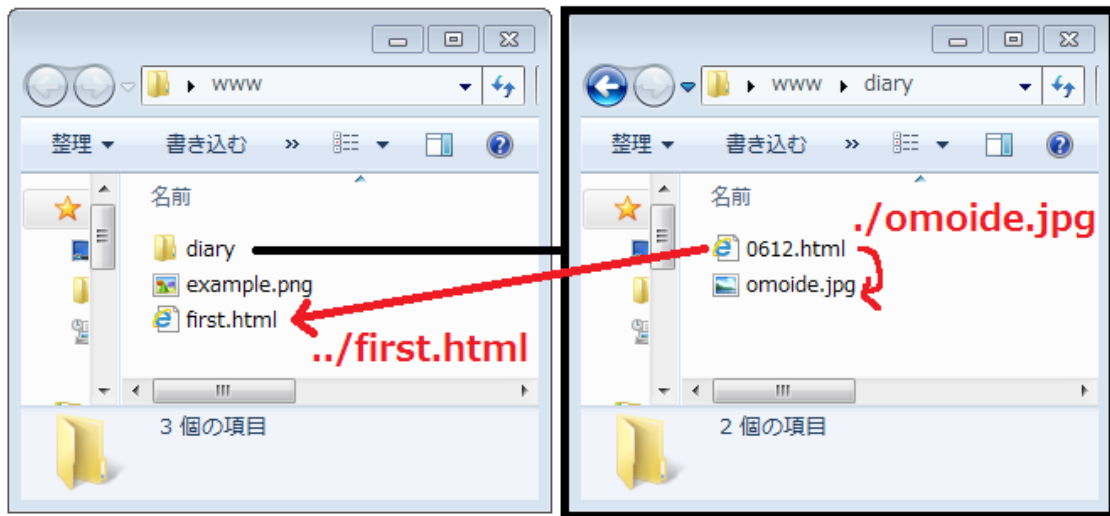


図 1. 相対パスの記述例

で表すが、省略可能であり単に `example.png` としてもよい。次に、`www` フォルダ内の `diary` フォルダにある `0612.html` から、`www` フォルダ直下にある `first.html` の位置を記述する。この場合、絶対パスで記述すると `http://www.u.tsukuba.ac.jp/~s1399999/first.html` となり、相対パスで記述すれば `../first.html` となる。相対パスにおいて、「`../`」（ピリオド2つとスラッシュ1つ）はそのフォルダを持つフォルダ（1つ上のフォルダ）を意味する文字列であり、たとえば2つ上のフォルダにある `test.html` を表すには「`../../test.html`」と書く。

絶対パスと相対パスはどちらを使ってもよい。使い分けの目安として、他の Web サイトにあるファイルは絶対パスでしか記述することができない。また、絶対パスで記述するとフォルダごと移動する（例えば図1において `www` フォルダの下に `jk13` というフォルダを作り、`diary` フォルダ、`example.png` と `first.html` を `jk13` フォルダに入れる）と、絶対パスで移動したフォルダ・ファイルを参照していた場合、全て書き換えが必要になる。しかし、相対パスで記述していた場合、移動したファイル・フォルダ内の関係性は変わらないため、書き換えの必要は無い（`0612.html` から `first.html` フォルダを参照するには `../first.html` のままでよい）。

### 3. 画像を埋め込む

HTML 文書中に画像を埋め込むには `img` 要素と `src` 属性を用いる。たとえば同じフォルダにある `example.png` という画像を、相対パスを使って埋め込むには

```

```

と記述する。`img` 要素は `width` 属性と `height` 属性を使って表示する大きさを変えることができる。それぞれの属性値はピクセル数またはブラウザに対する表示率（%）を取る。`example.png` が幅 `400px`、高さ `300px` の画像の場合、縦横を2倍にした幅 `800px`、高さ `600px` の画像として表示するには

```

```

と記述する。属性値として 200%を指定しても、それは HTML ファイルを表示しているブラウザのウィンドウサイズに対して 200%で表示ということであり、この場合 800px を意味するわけではないということに注意すること。width 属性か height 属性の両方を記述した場合、元の画像の縦横比と異なってもその通りに表示されるが、どちらか一方を省略した場合は縦横比が維持されるようにもう片方が自動で計算される。また、両方を省略した場合は画像の元の大きさのまま表示される。

img 要素を使えば技術上はあらゆる画像を埋め込むことが可能であるが、著作権・肖像権等に注意すること。情報基礎実習で作成する Web ページにおいては、画像の作成者が権利を放棄（パブリックドメイン化）していない限りは必ず画像の近くに出典を明記すること。

(演習) 画像ファイルを first.html 中に表示せよ

#### 4. リンクをつける

他の Web サイトやファイルへのリンクを作成するには、a 要素と href 属性を使って

```
<a href="相対パスもしくは絶対パス">見出しとなる内容</a>
```

のように記述する。この要素をブラウザで見ると、アンダーラインがついて色が変わり、

[見出しとなる内容](#)

のように表示される（初期状態では、リンク先がそのブラウザ上で既読であれば色は紫、未読で青色になる）。ここをクリックすれば、絶対パスか相対パスで指定したファイルへ移動できる。見出しとなる内容は自由に設定可能であり、この中に img 要素を記述することで画像をリンクとすることも可能である。リンク先は他の HTML ファイルのみならず、テキストファイルや PDF ファイル、画像ファイルのようにどのようなファイルでも構わない。ブラウザ上で表示ができないファイル（たとえば.xls）にリンクを張ることもできるが、その場合、クリックするとそのファイルのダウンロードを行う。

(演習) first.html から他 Web ページへのリンクを作成せよ

#### 5. Word, Excel を使って Web ページを作る

全学計算機システムで Web ページを公開するには、HTML ファイルを www フォルダに置くだけでよい。本テキストのここまでの内容ではメモ帳を使って自分自身で要素を打ち込み HTML ファイルを作成してきたが、これまでの演習で扱った Word や Excel でも Web ページを作成することが可能である。

Word、Excel の場合は、[ファイル]タブから[名前をつけて保存]を選択し、[ファイルの種

類(T:)を「単一ファイル Web ページ (\*.mht;\*mhtml)」や「Web ページ(\*.htm; \*.html)」として保存することで Web ページを生成できる。このとき、ファイル名が全角文字（たとえば情報基礎実習第 7 回.docx）であれば、半角文字（たとえば jk13\_07.docx）としよう。すると、複数のファイルが生成されるはずなので、それぞれブラウザで開き、どのように表示されるのかを確認してみよう。このように、Word や Excel からは（レイアウトなど細かいことに拘らなければ）簡単に Web ページを作成することが可能である。

（演習）

- Word 文書を Web ページとして保存し、どのようなファイルやフォルダが作られたか（名称やファイルサイズなど）を調べよ
- Excel の 1 シートを Web ページとして保存せよ。その際、ページタイトルが「Excel で Web ページ作成」となるように設定せよ

## 6. Word を使いデザイン文字、数式、画像、リンクの入った Web ページを作る

Word を使うことで、HTML 要素を知らなくとも Web ページを作ることができる。第 1 回で扱わなかった Word の機能を含んだ様々な機能を使って Word 文書を作り、それを Web ページにしてみよう。

- デザイン文字（ワードアート）を描く（『共通科目「情報（実習）」の手引き』 p.188）

# 情報基礎実習

- 数式を入力する（『実習の手引き』 p.189）

$$\Delta\Psi + \frac{2m}{\hbar^2} (E - V)\Psi = 0$$

- 画像（写真）を入れる（『実習の手引き』 p.181-182）
- リンクをつける

[挿入]タブ内真ん中やや左の「リンク」から[ハイパーリンク]を選択する。「表示文字列(T):」が 4 章の a 要素でいう「見出しとなる文字列」、「アドレス(E):」が href 属性の属性値にあたる。相対パスで別のファイルへのリンクを作成するには、目的となるファイルをクリックすると自動的に「アドレス」が設定される（図 2）。他の Web サイトへのリンクを作成するには、「アドレス(E):」に「<http://klis.tsukuba.ac.jp/jk13/>」のように絶対パスを打ち込めばよい。また、URL を Word に貼り付けると

<http://klis.tsukuba.ac.jp/jk13/>

のようにその URL に対し自動的にハイパーリンクが付与される。

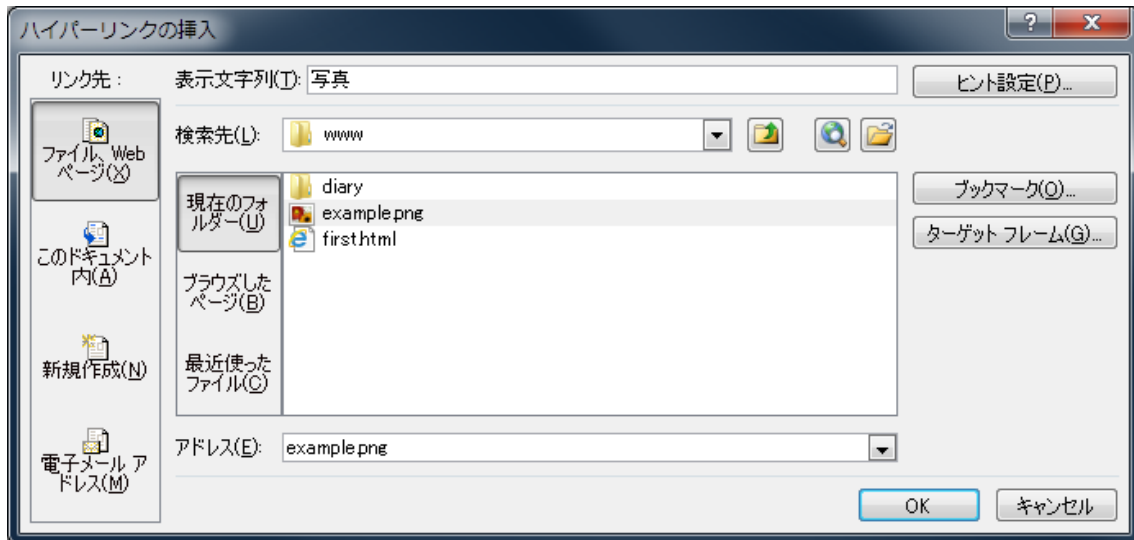


図 2. 「ハイパーリンクの挿入」 ウィンドウ

#### (演習・オプション)

デザイン文字、数式、画像、リンクの入った Word 文書を Web ページとして保存し、どのようなファイルやフォルダが作られたか（名称やファイルサイズなど）を調べよ

## 7. HTML ファイルが正しく書けたかを検証し、署名する

HTML ファイルをブラウザで開くと、ブラウザは HTML 要素を解釈し、整形して表示している。この解釈・表示のことを**レンダリング**と呼び、それを行うソフトウェアを**レンダリングエンジン**と呼ぶ。Internet Explorer には Trident、Firefox には Gecko というレンダリングエンジンが採用されている。ブラウザによっては複数のレンダリングエンジンを採用しているものがあり、切り替えることも可能である。

HTML の書き方（構文）が本来定義されたものとは違った、誤ったものであっても、ほとんどの場合ではレンダリングエンジンが拡大解釈をしてくれるのでブラウザ上では何らかの表示がされる。しかし、人によって使うブラウザ・レンダリングエンジンは異なるため、自分自身が使うブラウザでは望むように表示されていても、他の人が使うブラウザでは全く異なった表示になる可能性がある。そのため、できる限り構文は正確に、省略せず書いてレンダリングエンジンが拡大解釈をする余地を少なくすることが望ましい。正しく HTML の構文で書かれた HTML ファイルを**妥当 (Valid)** な HTML ファイルと呼ぶ。

HTML ファイルが妥当かどうかは、HTML を制定・勧告した W3C という団体が提供しているサービス The W3C Markup Validation Service (<http://validator.w3.org/>) を使えばよい。Web ページであればその URL を入力すればよいし、HTML ファイルをアップロードして、また直接 HTML ファイルの内容（**ソース**ともいう）を貼り付けて検証することもできる。検証の結果、問題がなければ Passed と表示される。

また、Another HTML-lint gateway (<http://cetus.sakura.ne.jp/htmlint/htmlint.html>) は構文がどの程度正しいかを 100 点からの減点方式で計算するサービスである。Another

HTML-lint gateway では The W3C Markup Validation Service よりさらに細かい検証を行っているため、The W3C Markup Validation Service で Passed となる Web ページを作るより Another HTML-lint gateway で 100 点を取れる Web ページを作ることの方が難しい。

The W3C Markup Validation Service で Passed となるか、Another HTML-lint gateway で 100 点を取ることができれば構文的には文句なしである。ただし、構文的に正しい Web ページは人間が見やすい Web ページとは限らないことに留意すること。理想的には Valid かつ人間にも見やすい Web ページを作ることがベストだが、実際に運用されているほとんどの Web ページには構文に何らかの誤りがある (**Invalid** な Web ページという)。

The W3C Markup Validation Service でエラーが出ない Web ページを作成できたら、その証として W3C のアイコンが入った署名をしよう。次の 2 つの p 要素を body 要素の終了タグ (</body>) 直前に入れよう (内容は適宜変えても良い) :

```
<p>
  作者 s13xxxxx<br>
  このページは情報基礎実習の課題として作成しました。<br>
  作成日 2013/6/13<br>
  最終更新日 2013/**/**
</p>
<p>
  <a href="http://validator.w3.org/check?uri=referer">
    
    </a>
</p>
```

#### (演習・オプション)

The W3C Markup Validation Service で Passed となる Web ページを作成し、W3C の署名を入れよ。この演習はそう簡単ではないため、余裕がある場合のみ取り組むこと。なお、body 要素内に多くの要素を入れれば入れるほど Valid な Web ページを作成することは難しくなるため、単に Valid にしたいだけであれば内容を極力削れば楽である。

#### (演習・オプション)

Another HTML-lint gateway で 100 点となる Web ページを作成せよ。この演習は Valid な Web ページを作ることよりさらに難しいため、かなり余裕がある場合のみ取り組むこと。まず Valid なページを作った上で、減点されている箇所とその説明をよく確認して 1 つ 1 つ改善することで比較的早く 100 点のページを作ることができる。



## 8. Web ページ公開の注意事項

### ① 『筑波大学における情報システム利用のガイドライン』

(<http://www.u.tsukuba.ac.jp/document/ja/UnivGuideline.pdf>) によると、全学計算機システムを利用して Web ページを公開する際は偽名・匿名を用いず、発信情報に関する責任の所在を明示しなければならないとしている。しかし、本演習で扱う内容では Web ページの URL に学籍番号が必ず含まれるため、改めて氏名等を書く必要は無いと解釈している。ただし、このガイドラインは掲示板や SNS への投稿にも適用されるため、一度目を通しておくこと。なお自分・他人のものを問わず、住所や電話番号などは絶対に書かないこと。

② Web ページを公開する際は、その内容が外部へ公開しても問題がないものであるかどうかを十分に確認すること。著作権を侵害していない、表現や内容に問題がない、内容に責任が持てる、大学生として恥ずかしくないなどが重要な観点である。ただし、内容の面白さ・つまらなさを読み手が判断することであり、法令や学内規則に反さない範囲であればどのような内容のものを公開してもよい。

③ Web ページは誰でも自由に公開できる。Web ページに限らず、Web 上の情報を参照する際は、そこに書かれている内容の真偽を自分で判断する必要がある。本や雑誌などの出版物には多少のチェックが入るが、Web ページには原則チェックが入らないため、閲覧者の判断力が必要とされる。

④ HTML は基本的には文書の構造を規定する言語である。文書の見にくさを整えようとして 1 行の文字数を決め、自分のブラウザできれいに表示されるようにしたとしても、違うパソコンやブラウザでその通りに表示されるという保証は一切無い。同様に、余白を入れたいがためだけに br 要素を連続して入れることは「構造」を表記することとはかけ離れたナンセンスな行為である。

⑤ index.html という名前の HTML ファイルを www フォルダ直下に置き、ブラウザの URL 欄に <http://www.u.tsukuba.ac.jp/~s13xxxxx/index.html> と入力した場合と、<http://www.u.tsukuba.ac.jp/~s13xxxxx/> と入力した場合でそれぞれどのような表示になるかを試してみよう。結論だけ言うと、全く同じ画面になるはずである。拡張子以外が index のファイル（多くの場合、index.html）はそのフォルダの「顔」として機能するようになっている。フォルダ内に index.html がない場合そのフォルダにアクセスする（www フォルダであれば <http://www.u.tsukuba.ac.jp/~s13xxxxx/>）と、そのフォルダ内の全ファイル・フォルダの一覧を見ることができるところがある（Web サイトによって設定は異なるが、全学計算機システムでは表示される設定である）。これはあまり見にくくがよくないため、できれば index.html のようなファイルを作ることが望ましい。ただし、採点でファイルの一覧を見る必要があるため、レポートが返却されるまでは index.html は作成したままにせず、一度試したら必ずファイルを消しておくこと！

## 9. HTML と CSS (オプション)

冒頭でも述べたように、Web サイトは HTML だけでなく、ほかの様々な技術を併用して構成されることが一般的である。例えば図 3 に示す Amazon.co.jp では次のような言語・技術が組み合わされている (はずである)。

- HTML : テキストや画像の表示
- CSS : レイアウトや色などの装飾の決定
- JQuery (JavaScript) : 「最近チェックした商品」の切り替えなど
- データベース : 商品データや、最近チェックした商品の登録
- PHP や Perl、Ruby など : データベースからデータの引き出し、注文システムなど

これらのうち、最後の 2 つは知識情報演習など、今後の授業・演習で取り組む。

CSS は Web ページにおけるレイアウトや色など、装飾を決定するための言語である。CSS を無効にした場合、どのような表示になるかを図 3 の右側に示す。フォントの色や文字の左寄せ、右寄せといったことは HTML にも対応する要素があるが、CSS を使えばより詳細に書くことが可能である。CSS を使うことの利点は次のようなものがある。

- 構造と装飾の分離

同じ Web ページでも、表示デバイスによって表示の仕方は異なるべきである。たとえば、全学計算機のモニターは比較的大きなものであるが、この大きさだけを前提としてレイアウトなどを決定してしまうと、iPad のような画面が小さいサイズでその Web サイトを表示する際に問題となる。iPhone に代表される、画面がより小さいスマートフォンであればなおさらである。

このような場合、構造と装飾が分離されていれば、装飾を変えることで様々な状況に対応可能となる。しかし、HTML において装飾までを指定してしまった場合、変更が困難になる。そのため、HTML では構造のみ記述し、CSS で装飾を指定すると良い。

- 複数の Web ページに適用可能

Web サイト全体で統一感を出すため、各 Web ページのデザインは原則として一緒にすることが望ましい。しかし、HTML で装飾を書いてしまうと、Web サイト全体のデザインの変更が困難となる。一方、CSS は外部ファイルとして記述することも可能であり、Web サイト中の各 Web ページから 1 つの CSS を参照すれば簡単にデザインを共有することができる。この場合、Web サイトのデザインを変更するには CSS ファイルのみを書き換えればよく、HTML ファイルを編集する必要はない。また、CSS は複数ファイルを同時に適用することも可能であり、メニュー用の CSS、本文用の CSS など複数の CSS を作り、それらを組み合わせるということもできる。

- より凝ったデザイン

CSS では HTML で定義されているものより、より凝った指定 (スタイル) が多く定義されている。CSS を使いこなすことができれば、Web 上のデザインはほぼ全てを再現することが可能である。



図 3. (左) Amazon.co.jp (右) CSS を無効にした Amazon.co.jp

CSS の記述方法は『実習の手引き』 p.219 以降に書かれているため、興味を持った人は熟読し、色々としてほしい。ここでは CSS の適用の仕方のみを述べる。

CSS を適用する手法には

1. style 属性を使って直接要素に指定する
2. style 要素を使って HTML ファイル全体に指定する
3. 外部ファイルを作成し、それを link 要素で読み込む

の 3 種類がある。『実習の手引き』に書かれている手法は 3. である。ここでは 1. の手法を説明する。

### 9.1. style 属性を使った装飾の指定

style 属性を使うことで、任意の要素に対し直接 CSS のスタイルを適用することが可能である。たとえば、HTML 中のある段落の文字色を赤くすることを考える。HTML 要素を使って段落全体の文字色を赤くするには、p 要素の中に font 要素を書き、font 要素に color 属性を記述すればよい。具体的には、

```
<p><font color="red">赤くしたい文章</font></p>
```

とする。同じことを CSS で表現するには、p 要素に style 属性を記述し、その値の中で color というスタイルを記述する。具体的には

```
<p style="color:red;">赤くしたい文章</p>
```

とすればよい。

### 9.2. HTML と XHTML

HTML は現在 4.01 まで勧告（多くの検証をして、これ以上変更は加えないとした段階）されており、現在は 5 の策定がされている。またそれとは別に、「人間だけでなくコンピュータにも処理しやすい形式」として XHTML が 1.1 まで勧告されている。XHTML では上記「構造と装飾の分離」を念頭に設計されており、HTML にあった font 要素のようなデザインに関する要素は一切無い。そのため、レイアウト等は全て CSS で記述する必要がある。

## 付録：

HTML 要素の書き方については 多くの Web サイト・書籍で紹介されているため、適宜参考にするとよい。最も有名な入門用の Web サイトは

- 杜甫々. “とほほの WWW 入門”. とほほの WWW 入門.

<http://www.tohoho-web.com/www.htm>, (参照 2013-06-10).

である。

また、有名な書籍として

- (株) アンク. HTML タグ辞典 第 6 版 XHTML 対応. 翔泳社, 2007, 386p.
- (株) アンク. スタイルシート辞典 第 4 版. 翔泳社, 2007, 401p.

などがある。

ただし、Web サイトや書籍を参照するにあたっては、特にその内容の正誤について注意すること。HTML は今後習得するであろうプログラミング言語に比べればかなり適当な記述が許される言語であること、2000 年前後は今と比べブログや掲示板、Wiki といった仕組みが今ほどは当たり前のもので存在せず、個人で Web ページを作成するのが一般的だった時代であり、特にその頃の Web サイトや書籍は主観を元にした、かなり適当であるものが多い。また、HTML にも 9.2. で触れたように複数のバージョンがあるため、古いバージョンを元に作成された Web サイトや書籍ということもありうる (古いバージョンでは存在した属性や要素が新しいバージョンでは廃止されているということがある)。このような理由から、HTML の学習ではあまり 1 つの情報を鵜呑みにせず、複数の情報源をあたるべきである。最も正確な情報源は HTML を策定した W3C の web ページ

(<http://www.w3.org/TR/html401/>) である。この Web ページに関しては有志が日本語に翻訳した (<http://www.asahi-net.or.jp/~sd5a-ucd/rec-html401j/cover.html>) ものもある。

HTML や CSS 上達のコツはいくつかあるが、次のような段階を踏むことをお勧めする。

1. 上記 Web サイト、書籍などを参考にあらゆる要素・スタイルを 1 度ずつ試す
2. 1. で得た知識を生かし、実際の Web ページがどのような要素・スタイルで表現されているかを分析する
3. デザインが気に入った Web サイトのデザインを自分で 1 から真似て作る

HTML 要素の習得にあたっては、Another HTML-lint gateway で 100 点を取ることを目指すことも上達の近道の一つである。興味を持った場合は色々と試して欲しい。