

情報基礎実習 第 8 回

2015 年 6 月 11 日 (木), 6 月 12 日 (金)

担当教員：逸村裕, 小泉公乃

TF：池田光雪

1. 本日の主な作業

普段、私たちがアクセスしている Web サイトは、ほぼすべてが様々な言語や技術を併用して運用されている。その中でも、今回は最も基礎的な言語である Hyper Text Markup Language (HTML) の仕組みを学ぶ。HTML 以外の言語は Cascading Style Sheets (CSS) のみ 9 章で軽く触れる。

第 7 回の授業では HTML を使って簡単な Web ページを作ったが、今回は、1)画像、他のファイルへのハイパーリンクなどを組み込んだ Web ページの作成、2)HTML の妥当性の検証をおこなう。演習課題として、3)Word で Web ページを作成し、メモ帳で作成したものとの比較なども用意している。

今回の内容は第 7 回の授業で扱った内容が前提である。また、テキストにはいくつかの演習問題を載せているが、やや発展的な演習には☆、発展的な演習には☆☆、非常に発展的な演習には☆☆☆を付与した。取り組む順序などの参考にせよ。また、この分野における専門用語は**ゴシック体の太字**で表した。

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

【出席課題】

第 7 回の出席課題で作成した first.html を次の条件(1)~(5)を満たすように修正せよ。その後、メモ帳で開きソースコードを印刷したものと、Internet Explorer から Web 経由で閲覧・印刷し用紙の左上に学籍番号と氏名を手書きで記入したもの、計 2 部を授業時間内 (2 限終了時まで) に TA もしくは TF まで提出せよ。なお、ファイルの位置は www フォルダ直下のまま変更しないこと。

- (1) title 要素の内容が、情報基礎実習第 8 回出席課題 であること
- (2) 画像が埋め込んであること。ただし、自身が著作者でない画像を埋め込む場合は必ず出典を SIST02 形式で明示し、自身が著作者であった場合は作成、あるいは撮影年月日を明記せよ

- (3) 他の Web ページへのハイパーリンクがあること
- (4) ul 要素を使った箇条書きがあること -『共通科目「情報(実習)」の手引き 2015年度』の p. 215 を参照せよ
- (5) 可能な限り、The W3C Markup Validation Service (<http://validator.w3.org/>) で Passed 判定となる Web ページにせよ。さらに、Another HTML-lint gateway (<http://cetus.sakura.ne.jp/htmlint/htmlint.html>) で 100 点を取ることが望ましい。どちらの場合も達成した場合は達成したということを HTML 文書中に明記せよ。

2. HTML の構造

Hyper Text Markup Language (HTML) 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 ではこれらを**要素**という単位で扱う。HTML では、要素を表すために**タグ**を用いている。図 1 に HTML 要素の各部名称を示す。

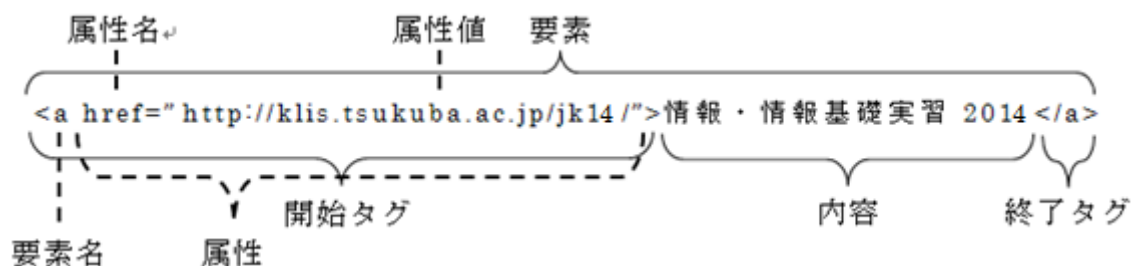


図 1. HTML 要素と各部名称

要素は通常、開始タグと終了タグで内容を挟むことにより表現される。ただし img 要素 (画像の埋め込み) など、内容を持たないため終了タグを必要としない要素もある。また、HTML では p 要素のような、終了タグを省略可能な要素がある。省略不可能なタグを省略してもブラウザが拡張解釈を行うため、大抵の場合は表示することは可能だが、省略しないことが望ましい。ブラウザの拡張解釈の詳細については第 7 章で詳しく述べる。

要素の開始タグには、その要素の性質などを表す**属性**を記述することができる。属性には、他の Web ページなどへのハイパーリンクの URL (a 要素の href 属性)、フォントの色 (font 要素の color 属性) などを指定するものがある。1 つの要素に複数の属性を記述することも可能で、その場合は半角スペースで属性間を区切る。**属性値**は引用符 (" あるいは ') で囲む。HTML では値によっては引用符を省略可能だが、値の範囲を明確に定めるためにこれも省略しないことが望ましい。

HTML においては一部の属性値を除き、要素名・属性名・属性値において大文字・小文字の区別をしない。したがって、どちらを使っても構わないが、HTML の後継にあたる XHTML では全て小文字で記述しなければならない。従って、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で記述する。

¹ 例えば、title 要素の開始タグは<title>、<TITLE>どちらの表記でも構わない。

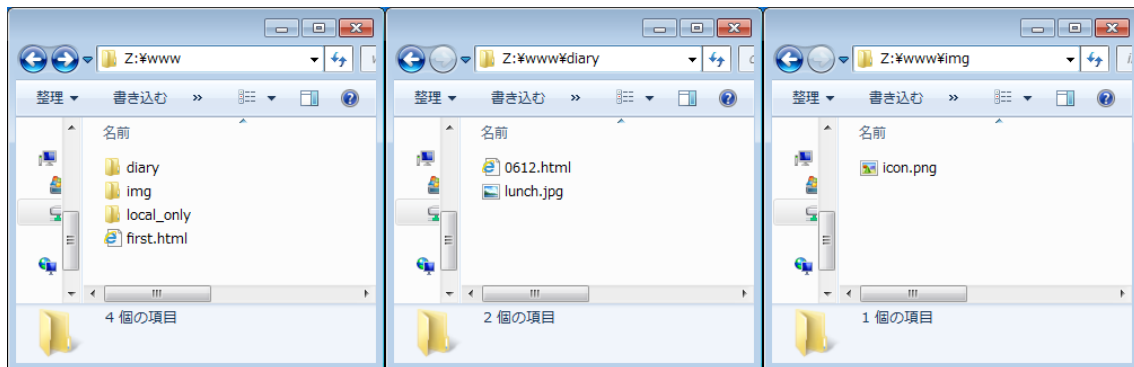


図 2. フォルダとファイル配置の例

3. 相対パスと絶対パス

HTML においてファイルやフォルダの場所を表すには**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である。一方、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を表すことに例えれば、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」という書き方に相当する。

学籍番号 201599999 の学生が `www` フォルダに図 2 のような構成でファイルとフォルダを配置した場合を考える。このとき `first.html` から `diary` フォルダの `0612.html` を参照するには、絶対パスでは `http://www.u.tsukuba.ac.jp/~s1599999/diary/0612.html` と、相対パスでは `diary/0612.html` と記述する。

次に、`0612.html` から同じフォルダにある `lunch.png` を参照する場合を考える。このとき、相対パスでは `./lunch.png` あるいは `lunch.png` と記述する。相対パスにおいて、同じフォルダは `./`（ピリオド 1 つとスラッシュ 1 つ）で表すが、これは `lunch.png` のように省略可能である。

最後に、`diary` フォルダにある `0612.html` から `img` フォルダにある `icon.png` を参照する場合を考える。このとき、相対パスでは `../img/icon.png` と記述する。相対パスにおいて `../`（ピリオド 2 つとスラッシュ 1 つ）はそのファイルやフォルダが格納されているフォルダ（1 つ上のフォルダ）を意味する。

絶対パスと相対パスは正しく記述できてさえすればどちらを使ってもよい。ただしその仕組み上、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。

これらの記法間で大きな差が生じるのは、パスで記述したファイルやフォルダを移動したときである。ファイルやフォルダの位置関係を変更した場合、絶対パスでの記述は必ず書き換えが必要となる。しかし相対パスで記述した場合は書き換えが必要ないことがある²。例として、図 2 において `diary` フォルダ内の `0612.html` から同ファイルの `lunch.jpg` を

² 相対パスの場合、別途 `base` 要素で基準となる URL を設定することでより高度な指定を

参照することを考える。前述の通り、絶対パスで記述すれば

`http://www.u.tsukuba.ac.jp/~s1599999/diary/lunch.jpg` , 相対パスで記述すれば `lunch.jpg` となる。ここで、`0612.html` と `lunch.jpg` を `www` フォルダ直下に移動したとする。この場合、絶対パスでは `http://www.u.tsukuba.ac.jp/~s1599999/lunch.jpg` のように表記が変わるが、相対パスは `lunch.jpg` のままであり変更の必要はない。

4. 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。例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。また、`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。例えば `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。ただし、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%` のサイズで表示という意味であり、この場合 `800px` を意味するわけではないということに注意せよ。`width` 属性と `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。なお、一般に HTML では属性間の順序は無視される。したがって、例えば上記の例において `width`, `height`, `src` の順で属性を記述したとしても表示は全く同じとなる。

`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、画像を技術的に埋め込めるということと、法律的（著作権、肖像権等）に埋め込みが許されるかは別問題であること注意すること。情報基礎実習で作成する Web ページにおいては、画像を埋め込む場合は原則として出典を明記せよ。また自らが著作者の画像を使用する場合は、不適切な他者の著作物の使用と明確に区別するため、作成年月日などを代わりに明記せよ。

【演習 1】

画像を `first.html` 中に埋め込め。また、その前後において `p` 要素を用い SIST02 形式でその画像の出典を示せ。自らが著作者の場合は代わりに作成年月日等を示せ。

5. ハイパーリンクを張る

他の Web ページやファイルへのハイパーリンクを張るには `a` 要素と `href` 属性を使って

することも可能である。

見出しとなる内容

と記述する。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

と表示される³。このとき、見出しとなる内容をクリックすれば、絶対パスか相対パスで指定した URL が開かれる。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。見出しとなる内容に何も書かないことも可能ではあるが、その場合、クリックが困難になること注意せよ。

【演習 2】

first.html から「情報基礎・情報基礎実習 2015(知識情報・図書館学類)」の Web ページ (<http://klis.tsukuba.ac.jp/jk15/>) へのリンクを張れ。

6. 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、所定の手続きをした後 HTML ファイルを `www` フォルダに置くだけでよい。ここまでの演習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、これまでの授業で扱った Word や Excel でも Web ページを作成することが可能である。

Word と 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、ファイル名に全角文字が使われているのであればできる限り全て半角文字にせよ。例えば元のファイル名が 情報基礎実習第 7 回.docx であれば、jk15_08.html のようなファイル名にすることが望ましい。

【演習 3】

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を調べ、まとめよ。

【演習 4】

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。

6.1 Word を使いデザイン文字や画像、リンクなどの入った Web ページを作る

前述の通り、Word を使うことで、HTML をよく理解していなくとも Web ページを作ることができる。第 1 回授業では扱わなかった 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。

³ CSS で明示的に設定していない場合はブラウザの設定に依存するが、初期状態ではリンク先がそのブラウザ上で未読であれば青色、既読であれば紫色になる。

6.1.1 デザイン文字（ワードアート）を描く（実習の手引き p.186）

Word では文字に対し様々なエフェクトを付与することが可能である。

情報基礎実習

6.1.2 数式を入力する（実習の手引き p.187-190）

数式エディタを使うことで、複雑な数式でも綺麗に表示させることが可能である。

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

6.1.3 図や写真を入れる（実習の手引き p.179-180）

本テキストのように、図や写真を入れることも可能である。また、図表番号を自動で振ることも可能である。Word で長い文書を作成するときは積極的に活用せよ。

6.1.4 ハイパーリンクを張る

[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。ここで、「表示文字列(T):」が 5 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して選択すればよい。すると自動的にアドレスが設定される (図 3)。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/jk15/> 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/jk15/>

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。

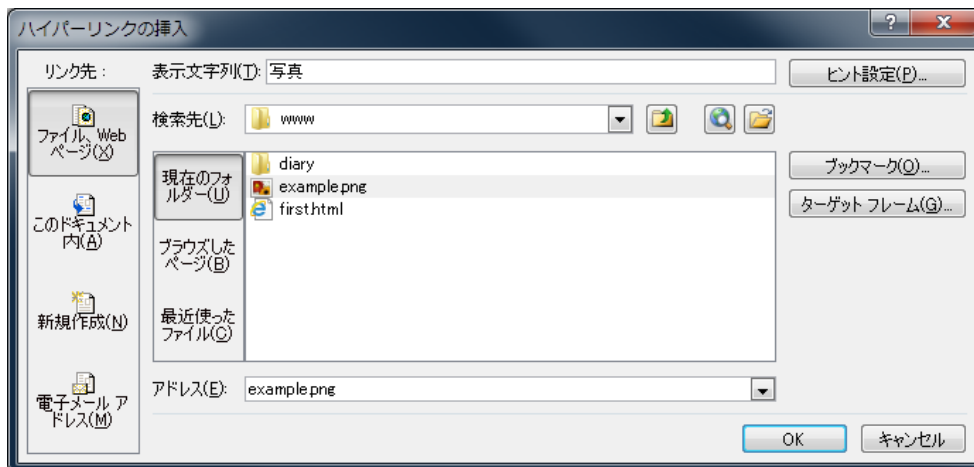


図 3. 「ハイパーリンクの挿入」 ウィンドウ

【演習 5】 ☆

デザイン文字、数式、画像、リンクなどが入った Word 文書を Web ページとして保存し、どのようなフォルダやファイルが作られたか、Word での表示と Web ページでの表示に相違があるかなどを調べ、まとめよ。

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、整形して表示する。この解釈・表示のことを**レンダリング**と呼び、それを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, Google Chrome には Blink というレンダリングエンジンがそれぞれ採用されている。世の中には数多くのブラウザが存在するが、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もある。

HTML の書き方（**構文**と呼ばれる）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レンダリングエンジンが拡大解釈を行うため、ほとんどの場合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や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極力少なくすることが望ましい。なお、正しい HTML の構文で書かれた HTML ファイルは**妥当 (Valid)** と呼ばれ、そうでない HTML ファイルは**非妥当 (Invalid)** と呼ばれる。

HTML ファイルが妥当かどうかは、HTML を制定・**勧告**⁴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/>) を使って**検証**することができる。Web ページであればその URL を入力するだけでよく、たとえ Web

⁴ 規格として批准することを**勧告 (Recommendation)** という。勧告されるまでには**作業草稿**や**勧告候補**、**勧告案**といったいくつかの段階が存在する。

ページとして公開していない HTML ファイルであってもファイルをアップロードする、HTML ファイルの内容（ソースコードともいう）を貼り付けるなどの方法で検証することが可能である。検証の結果、妥当であると認められれば Passed と表示される。

また、Another HTML-lint gateway (<http://cetus.sakura.ne.jp/htmlint/htmlint.html>) は HTML の構文がどの程度正しい、あるいは望ましい記述をしているかを 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 HTML ファイルを作るより Another HTML-lint gateway で 100 点を取れる Web ページを作るの方が真に難しい。

The W3C Markup Validation Service で Passed となる、さらには Another HTML-lint gateway で 100 点を取ることができる HTML ファイルは構文的には全く問題がない。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せよ。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最も望ましいが、実際に運用されているほとんどの Web ページは Valid ではなく Invalid である。

【演習 6】☆☆

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。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よい。

また、Passed と表示される 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せよ。具体的には、次の 2 つの p 要素を body 要素の終了タグ直前に入れよ（内容は適宜変えよ）。

```
<p>
  作者 s15xxxxx<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。
</p>
<p>
  <a href="http://validator.w3.org/check?uri=referer">
    
    </a>
  </p>
```

8. Web ページ公開の注意事項

8.1 筑波大学における情報システム利用のガイドラインについて

『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。しかし、本演習で扱う全学計算機システムにおいて Web ページを公開する場合、必ず URL に学籍番号が含まれるため、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。文字通り解釈すれば、例えば宿舎から大学の回線を使って掲示板に書き込む際に名前を付けないということは許されず、何らかの手段で責任の所在を明示しなければならない。

8.2 公開する Web ページの内容について

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検討・確認せよ。重要な観点として、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などがある。ただし、内容の面白さや有用性は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でもよい。

8.3 Web ページの内容の真偽について

Web ページは原則として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身で判断しなければならない。本や雑誌などの出版物には作成者以外による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。

8.4 Web ページの表示について

HTML は文書の構造を規定する言語であり、PDF のようにいつでも同じような見た目になるように表示を整えることには向いていない。したがって、文書の見た目を整えようとして 1 行の文字数を決め、自分の環境ではきれいに表示されるようにしたとしても、異なる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` 要素を連続して挿入することや、文字を大きく・太くしたいがために `p` 要素を `h1` 要素などに置き換えことは構造を表記するマークアップの本質とはかけ離れたナンセンスな行為であると言える。

8.5 index.html について

`index.html` というファイル名の 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/~s15xxxxx/index.html` と入力した場合と、`http://www.u.tsukuba.ac.jp/~s15xxxxx/` 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。結論を書くと、全く同じ表示になるはずである。その理由は、フォルダにアクセスした場合はファイル名のうち、拡張子以外が `index` であるファイルにアクセス

するという決まりがあるから⁵である。フォルダにアクセス（www フォルダであれば <http://www.u.tsukuba.ac.jp/~s15xxxxx/>）したとき、フォルダ内に index ファイルが無ければそのフォルダ内の全フォルダ・ファイルの一覧を見ることができる場合がある⁶。多くのファイルを公開するために意図的に見せる場合などは除き、フォルダ・ファイルの一覧はセキュリティの観点からできれば見せない方がよく、index.html のようなファイルを作ることが望ましい。

9. HTML と CSS

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 4 左に示す 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
- 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 : データベースへデータの問合せ、注文システムなど

これらのうち、少なくとも最後の 2 つは知識情報演習などの今後の授業で取り扱う。

Cascading Style Sheets (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。図 4 左に示した Amazon.co.jp で CSS を無効にした場合、どのような表示になるかを図 4 右に示す。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 を使うことで、一般には次のような恩恵が得られると言われている。

● 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。例えば、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、より画面が小さいスマートフォンであればなおさらである⁷。

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 において装飾までも指定してしまった場合、変更が

⁵ Web サイトの設定にもよるが、どのような拡張子でもよいというわけではない。また、index.html と index.htm のように複数の index ファイルが存在した場合、どれにアクセスするかも Web サイトの設定に依存する。

⁶ 見せない設定にしている Web サイトも数多く存在するが、全学計算機システムは初期状態では表示する設定となっている。

⁷ 近年ではレスポンシブ Web デザインと呼ばれる、ブラウザのサイズや視聴している機種によってレイアウトを大幅に変更する技法が流行している。例えば日本マイクロソフト社の Web サイト (<http://www.microsoft.com/>) はブラウザの横幅に応じて 4 回大きくレイアウトが変更される他、フォントサイズや画像の大きさも細かく変更される。

困難になる。したがって、HTML では構造のみを記述し、CSS で装飾を指定することが望ましいとされている。

- **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、各 Web ページから 1 つの CSS ファイルを参照することで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にする**

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 上のデザインのほぼ全てを再現することが可能である。



図 4. (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. 217-230 に書かれているため、興味を持った人は熟読し、色々試してみよ。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』には 3 の手法が説明されている。ここでは 1 の手法を説明する。なお、ある要素に対し 1~3 の複数の手法で異なる複数の指定をした場合、1 での指定が最も優先され、1 および 2 による指定がない時のみ 3 での指定が適用される。したがって、外部ファイルで包括的な定義を行い、個別に変更したい箇所だけ直接 HTML ファイル上で細かい指定を行うということがよく行われている。なお、各手法内でも同じ要素に対し

複数の指定がある場合、セクタ順⁸、出現順⁹の順序で優先度を決定する。

なお、CSS を本格的に学ぶのであればどの要素を対象とするか（**セクタ**）の書き方を習得する必要がある。詳しくは手引きなどを参照せよ。

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。例えば、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</span></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。

【演習 7】 ☆

いくつかの Web サイトで CSS を無効にし、表示がどのように変わるのかを試せ。なお、全学計算機システムにプリインストールされたブラウザでは、Firefox であれば[Alt]キー押しメニューバーを表示してから、[表示]、[スタイルシート]、[スタイルシートを使用しない]と順に選択することで簡単にスタイルシートを無効化できる。

【演習 8】 ☆☆

CSS を使って文書を装飾してみよ。その際、HTML 要素でも表現可能なこと（文字色、背景色、中央寄せの設定など）と、HTML 要素では困難、あるいは不可能なこと（余白、文字の回り込み設定など）の両方を試せ。

【演習 9】 ☆☆☆

同じ要素に対し複数のスタイルを指定した場合、どのような優先規則で適用されるかを実感し、使いこなせるようにせよ。

9.2 HTML と XHTML

HTML は現在 HTML5 まで勧告されている。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 は前述した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

⁸ 例えば、「a 要素」に対して指定した装飾よりも「body 要素中の a 要素」に対して指定した装飾が優先されるように、より詳細に指定したものが優先される。

⁹ 手法の違いやセクタでも優先度に差がつかない場合、後に指定した記述が優先される。

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。

10. さらなる自学自習のために

HTML 要素の書き方については 多くの Web サイトや書籍で紹介されているため、適宜参考にするとよい。入門用の Web サイトとして最も有名なものに

- 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 2015-06-01).

がある。

また、有名な書籍として

- 株式会社アंक. HTML タグ辞典 : XHTML 対応. 第 6 版, 翔泳社, 2007, 386p.
- 株式会社アंक. 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 は今後習得するはずの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¹⁰。また、HTML にも 9 章 2 節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¹¹。このような理由から、とりわけ 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。HTML の学習にあたり、最も信頼できる情報源は 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真似る、再現する

ただし、HTML や CSS の記述にはそのデザインなどに創作性がみられる場合、著作権が発生することには留意せよ。

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。

¹⁰ 例えば要素 (Element)、タグ (Tag)、内容 (Content) は HTML においてはそれぞれ明確に区別されるべき用語であるが、「エレメントの要素」のような本来の意味からではあり得ない使い方をしている書籍も数多くある。また、h1 要素の h はヘッダ (正しくは heading) とするような誤用も散見される。

¹¹ 例えば、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は頻繁に起こる。

【演習 10】

次の条件(1)~(5)をすべて満たす、ファイル名を `welcome.html` とする 1 つの HTML ファイルを作成し、`http://www.u.tsukuba.ac.jp/~s15xxxxx/welcome.html` のような URL から閲覧できるよう Web に公開せよ (15xxxxx の部分は各自の学籍番号下 7 桁である)。また、The W3C Markup Validation Service で Passed 判定が出る、さらには Another HTML-lint gateway で 100 点満点となるとなおよい。もし Passed 判定や 100 点満点となる Web ページを作成した場合は、それぞれ達成できたことがわかるような記述を Web ページ中に入れよ。

- (1) `ul` 要素を使った箇条書きを含む
- (2) 画像が `img` 要素で埋め込まれており、出典等も示されている
- (3) PowerPoint で自身が作成したイラストが `img` 要素で埋め込まれている
- (4) 他の Web ページへのハイパーリンクを含む
- (5) ブラウザにおいてページの拡大・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