

## 情報基礎実習 第 8 回

2014 年 6 月 12 日 (木)、6 月 13 日 (金)

担当教員：逸村裕、高久雅生

TF：池田光雪

普段我々がアクセスする Web サイトは、ほぼ全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、最も基礎的な言語である Hyper Text Markup Language (HTML) の仕組みを今回は学ぶ。カリキュラムの制約上、HTML 以外の言語は Cascading Style Sheets (CSS) のみ 9 章で軽く触れるに留める。

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、HTML の妥当性の検証も扱う。

### 今回の内容

第 7 回の授業で扱った内容を理解していることを前提としている。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すること。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
  - Microsoft Word 2010, Microsoft Excel 2010 から Web ページを作る
  - Microsoft Word 2010 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
- ☆ 正しい HTML ファイルが書けたかを検証する
- ☆ CSS を使ってレイアウトなどを変える

### 【出席確認課題】

第 7 回の出席課題で作成した first.html を次の条件(1)~(4)を満たすように修正し、Internet Explorer 9 から Web 経由で閲覧・印刷して、用紙の左上に学籍番号と氏名を記入して提出せよ。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) 画像が埋め込んであること。ただし出典の表示は第 5 回レポートに準ずるものとする
- (2) ul 要素を使った箇条書きがあること
- (3) 他の Web サイトへのハイパーリンクがあること
- (4) 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

## 今回のレポート課題

- 締め切り
  - 木曜クラス：6月18日（水）15:00
  - 金曜クラス：6月19日（木）15:00
- 内容

最低限次の条件(1)～(6)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/~s14xxxxx/welcome.html` のような URL から閲覧できるよう Web に公開せよ(14xxxxx 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 Web ページは加点対象とする。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) 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（出典の表示は第 5 回レポートに準ずるものとする）
  - (4) 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
  - (5) 他の Web ページへのハイパーリンクを含む
  - (6) 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
- 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 箇所をステイプラー（針無しは不可）で左上 1 箇所を綴じたもの

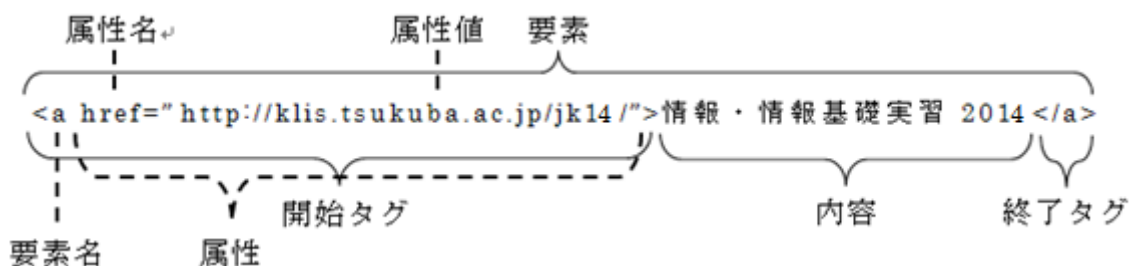
  - A) Lab2014-1.docx を元に、次の(ア)～(ウ)全てを記した Word 文書
    - (ア) Web ページ作成に当たって工夫を凝らした点
    - (イ) Word など生成した HTML ファイルと、自分で HTML タグを打ち込み作成した HTML ファイルそれぞれの特徴や長所、短所等の比較
    - (ウ) 第 8 回の授業内容で印象に残った点、及び感想。またもしあれば残りの授業における要望
  - B) `welcome.html` をメモ帳で開き印刷したもの。ただし、表示設定は初期設定（フォントの MS ゴシックの標準、12 ポイント）のままとせよ
  - C) `welcome.html` を Internet Explorer 9 で開き印刷したもの
- 提出先

春日エリア 7B 棟 2 階 学務前レポートボックス
- 備考
  - レポートの採点は Web ページにおいてどのような技術を使っているか、および使いこなしているかに重点を置く
  - レポート課題の採点対象は `welcome.html` の HTML ファイルのみとする。たとえば CSS を外部参照している場合、CSS 部分は採点の対象としない。この場合は

- style 要素や style 属性で指定するなどの方法を探ること
- レポート中のあらゆる箇所において手書きは不可とする
- 提出先を間違った場合、原則として採点の対象外とする
- 課題名はふさわしいものを各自で考案し記述すること
- 提出後におけるいかなるレポートの差し替えも認めない
- 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することを禁ずる。もしファイルの最終更新日時が締め切りを超過していた場合、大幅な減点をするか、もしくは採点の対象外とする
- 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 フォルダ以下には置かないこと

## 1. 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 ではこれらの単位を**要素**として扱う。HTML では、要素を表すために**タグ**を用いている。



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 (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。属性の値は引用符 (" 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 (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 では全て小文字で記述しなければならない制約が加わっているため、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で記述する。

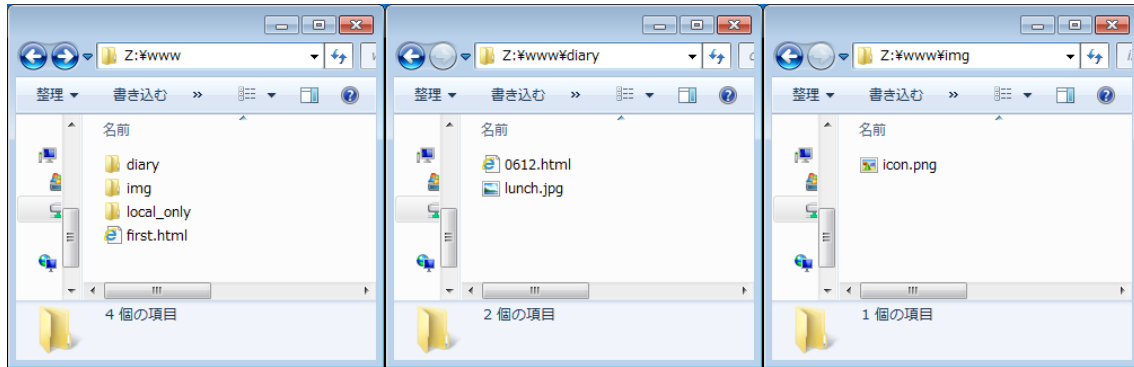


図 1. フォルダとファイル配置の例

## 2. 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、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たとえば、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 201499999 の学生が `www` フォルダに図 1 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/~s1499999/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<sup>1</sup>。例として、図 1 において `diary` フォルダ内の `0612.html` から同ファイルの `lunch.jpg` を参照

<sup>1</sup> 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することも可能である

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/~s1499999/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/~s1499999/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。

### 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。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。たとえば `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、たと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ふことと、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せよ。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すること。

#### 【演習 1】

画像を `first.html` 中に埋め込め。

### 4. ハイパーリンクを張る

他の Web サイト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

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

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<sup>2</sup>。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、その場合クリックできなくなることに注意せよ。

### 【演習 2】

first.html から他の Web ページへのリンクを張れ。

## 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、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全て半角文字にせよ。たとえば元のファイル名が「情報基礎実習第 7 回.docx」であれば、jk13\_07.html のようなファイル名にすることが望ましい。

### 【演習 3】

- (1) 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
- (2) 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 をよく理解していなくても 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。

### 6.1 デザイン文字（ワードアート）を描く（実習の手引き p.190）

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

---

<sup>2</sup> 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

## 6.2 数式を入力する（実習の手引き p.191-194）

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。

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

## 6.3 図や写真を入れる（実習の手引き p.183-184）

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なので、Word で長い文書を作成するときには積極的に活用せよ。

## 6.4 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。「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 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/jk14/> 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/jk14/>

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。

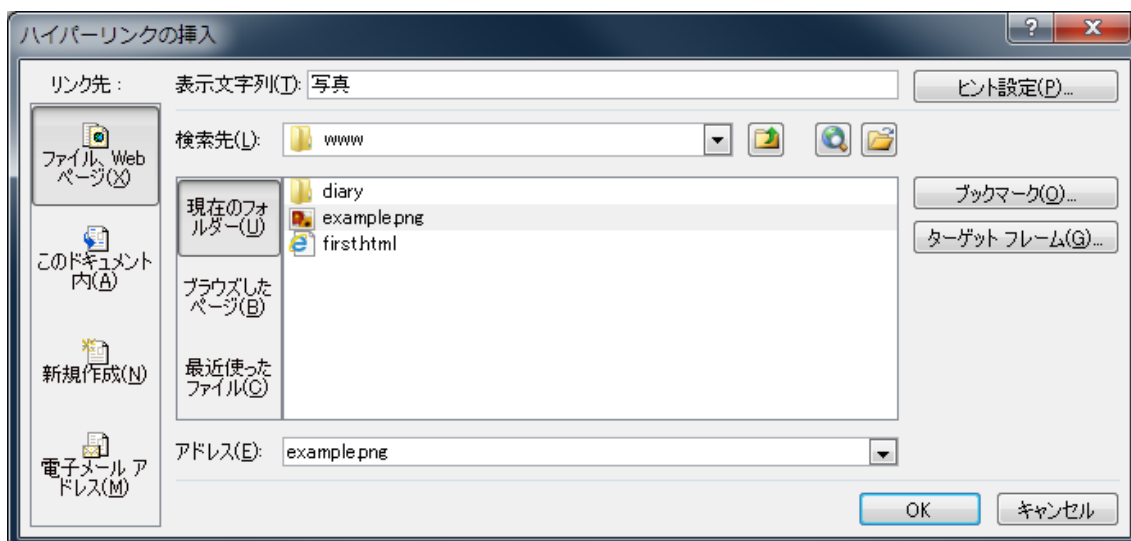


図 2. 「ハイパーリンクの挿入」ウィンドウ

### 【演習 4】☆

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があり、切り替えることも可能である。

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<sup>3</sup>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/>) を使って確認（**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 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**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/htmlhint/htmlhint.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。

The W3C Markup Validation Service で 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（内容は適宜変えても良い）。

---

<sup>3</sup> 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



```
<p>
  作者 s14xxxxx<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>
```

#### 【演習 5】☆☆

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。

#### 【演習 6】☆☆☆

Another HTML-lint gateway で 100 点となる Web ページを作成せよ。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

### 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ける Web ページの公開では URL に必ず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、たと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せねばならない。

### 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。重要な観点としては、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を

公開してもよい。

### 8.3 Web ページの真偽について

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、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。

また、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。

### 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/~s14xxxxx/index.html` と入力した場合と、`http://www.u.tsukuba.ac.jp/~s14xxxxx/` 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<sup>4</sup>からである。フォルダにアクセス (www フォルダであれば `http://www.u.tsukuba.ac.jp/~s14xxxxx/`) 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ときがある<sup>5</sup>。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。ただし、情報基礎実習第 8 回のレポート採点においては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 ファイルは作成してはいけない。index ファイルの役割を確認したあとは必ずファイル名を変更するか削除するかして、フォルダ・ファイルの一覧が閲覧できるような状態にせよ。

## 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

---

<sup>4</sup> 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

<sup>5</sup> 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

構成されることが一般的である。たとえば図 3 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 : データベースヘータの間合せ、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必ず取り扱う。

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 3 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 3 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 を使うことの利点に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、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<sup>6</sup>。

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。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**

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、ほぼ全て Web 上のデザインを再現

---

<sup>6</sup> 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。たと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

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.221-234 に書かれているため、興味を持った人は熟読し、色々と試してみよ。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 でも 2 でも指定がない時のみ 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<sup>7</sup>、出現順<sup>8</sup>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。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>
```

<sup>7</sup> たとえば、「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

<sup>8</sup> 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

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 → [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 HTML4.01 まで勧告されており、2014 年 6 月現在では HTML5 が策定されている。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. 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、適宜参考にするとうい。入門用の Web サイトとして最も有名なものは

- 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 2014-06-10).

である。

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<sup>9</sup>。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（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、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。

---

<sup>9</sup> たとえば要素 (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用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。