

## 10 ホームページ作成

### 10.1 ホームページを公開するには

ホームページを公開するには、

1. 公開する場所を確保
2. ホームページのソースを作成
3. ホームページのソースを所定の場所に保存

が必要である。ソースは HTML(HyperText Markup Language) という言語で記述する。Web ブラウザで [表示][ページのソース] をクリックすると閲覧しているページの HTML による記述を見ることができる。

東京女子大の学生は、東女の Web サーバを公開場所に使える。プロバイダのサーバを利用した場合との比較を次表に示す。

公開する場所	東女	プロバイダ
ソースを作成	エディタを使って HTML で書く Web オーサリングソフトを使う	エディタを使って HTML で書く ワープロで作り HTML 形式で保存 ホームページ作成ソフトを使う
ソースの保存場所	~b08a001/WWW/ (インターネットで公開) ~b08a001/WWW-local/ (学内のみ公開)	プロバイダのサーバの所定のフォルダ
ソースの保存方法	html という拡張子をつけ、上記のディレクトリに保存	ファイル転送 (アップロードするという)

### 10.2 WWW と WWW-local

東京女子大の学生 b08a001 の場合

公開範囲	インターネット	学内ネットワーク
ディレクトリ	~b08a001/WWW/ (大文字の W を 3 つ)	~b08a001/WWW-local/ (W は大文字、local は小文字)
ソース	~b08a001/WWW/index.html	~b08a001/WWW-local/index.html
URL	http://www.twcu.ac.jp/~b08a001/ (英字はすべて小文字)	http://www-local.twcu.ac.jp/~b08a001/ (英字はすべて小文字)
留意事項	著作権とプライバシーに特に留意する	

本講義で作るのは、練習のためのホームページなので、

~/WWW-local/compIID/index.html

に作成する。学内のみから閲覧できる。インターネットで公開したい場合は、同じものを

~/WWW/index.html

にコピーすればよい。(画像ファイルなどもすべてコピーする。) ターミナルを起動し、

```
$ cp -r ~/WWW-local/compIID ~/WWW
```

CoPy Recursive

とすればよい。-r は再帰的 (recursive) からきており、ディレクトリをその中のファイルやサブディレクトリ毎コピーするオプションである。

## 10.3 ホームページの作り方

### 10.3.1 はじめて作る時

1. ターミナルを起動する。
2. ディレクトリ `~/WWW-local/compIID` を作る。(WWW と IID は大文字、その他は小文字)

```
$ mkdir WWW-local
$ cd WWW-local
$ mkdir compIID
```

3. Emacs で `index.html` を作成する。

- (a) カレントディレクトリをファイルを作成するディレクトリに移動する

```
$ cd ~/WWW/compIID
```

- (b) 雛形をコピーする

```
$ cp ~/asakawa/WWW/template.html ./index.html
```

- (c) Emacs を起動してファイルを読み込む。

- (d) 適当に書き換える。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="ja">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >
    <meta http-equiv="Content-Style-Type" content="text/css" >
    <title>東京子のページ</title>
    <style type="text/css">
    <!--
      -->
    </style>
  </head>
  <body>
    <h1>東京子のページへようこそ</h1>
    <p>
      準備中
    </p>
    <hr>
    <address>
      東京子 <mailto:b08a001@cis.twcu.ac.jp><br>
      2008年12月17日 <br>
    </address>
  </body>
</html>
```

(e) HTML の文法にしたがって、雛形を埋めながらソースを書く。

演習 10.1 1. 上記の 1-3(c) までを行え。

2. 本文に「準備中」(と同様の文)を加えよ。前後を<p>と</p>により囲むこと。
3. 保存をして、Opera で見てみよ。
4. Opera の文法チェック機能を利用して正しく HTML が記述できているかどうかを確認せよ。

注 10.1 インターネットでホームページを公開するときは、住所、電話番号(携帯を含む)は記載しない方がよい。

### 10.3.2 修正や追加を行うとき

1. Emacs でファイル `~/WWW-locl/compIID/index.html` を開く。
2. HTML の文法にしたがって、ソースを修正する。
3. 上書き保存する。
4. Opera で確認する。
5. 追加修正は 2 に戻る。

## 11 HTML 入門

### 11.1 HTML の概要

ホームページのソースは HTML(HyperText Markup Language) または XHTML(eXtended HyperText Markup Language) という言語で記述する。

HTML は直訳すると、ハイパーテキストを対象としたマークアップ言語である。ハイパーテキストとは、文書から別の文書や画像、音声などを自由に参照できる文書である。マークアップ言語とは、文章中に組み指定 (markup) をつけて論理構造をつくる言語である。

HTML は、見出し、段落、引用文、箇条書き、表、画像の表示、強調、リンクなどの論理構造を要素により表現する。

例えば、`<h1>東 京子のページへようこそ</h1>` は、最上位の見出しを表わす h1 要素 (element) で、`<h1>` は、h1 要素の開始タグで、`</h1>` は、終了タグである。「東 京子のページへようこそ」は h1 要素の内容 (content) である。ブラウザは「東 京子のページへようこそ」を最上位の見出しとして表示する。フォントや大きさはブラウザが決めるので、ブラウザによって見え方は異なる。

また、`<html lang="ja">` から `</html>` までを html 要素という。`<html lang="ja">` は html 要素の開始タグ、`lang="ja"` は属性、`lang` は属性名、`ja` は属性値である。

### 11.2 HTML の構成

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

は HTML4.01 Strict の文書型宣言である。HTML4.01 Strict の 使用とその DTD の場所が書いてある。

html 要素は、head 要素と body 要素からなる。

<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"</code>	文書型宣言
<code>"http://www.w3.org/TR/html4/strict.dtd"&gt;</code>	文書型宣言の続き
<code>&lt;html&gt;</code>	ここから html 要素
<code>&lt;head&gt;</code>	ここからヘッダ要素
<code>&lt;/head&gt;</code>	ここまでヘッダ要素
<code>&lt;body&gt;</code>	ここから body 要素
<code>&lt;/body&gt;</code>	ここまで body 要素
<code>&lt;/html&gt;</code>	ここまで html 要素

HTML のタグは小文字で書くべきである。

注 HTML 文書では段下げなどの空白や改行は、ソースを読みやすくするために用いる。ブラウザは半角の空白 1 個とみなし表示する。

### 11.3 head 要素

head 要素はタイトルおよびサーチエンジンが利用するキーワードや本文に表示されない事項を記述する。title 要素、meta 要素、style 要素、link 要素などからなる。

meta 要素 文字コード、著者、キーワードなどを表す。

title 要素 title 要素は必ず必要で、ブラウザのタイトルバーに表示され、ブックマークしたページはタイトル要素が登録される。

style 要素 色、フォント、表示する位置などの見え方の指定はスタイル要素、またはスタイル要素を独立のファイルにしたスタイルシートで表わす。

例 h1 要素は紺色の文字で背景は水色で中央揃え、body 要素の背景は白、文字は黒で表わすには、title 要素の前に

```
<style type="text/css">
  <!--
    h1{ color: navy; background: aqua; text-align: center }
    body{ background-color: white; color: black }
  -->
</style>
```

と書く。

<!-- と --> で挟んであるのは、スタイルシートに未対応のブラウザが、そのまま表示するのを防ぐためである。スタイルシートのコメントは、/\* と \*/ で挟む。

表 11: 表示する位置

属性の値	意味
left	左揃え
center	中央揃え
right	右揃え

表 12: 色 (HTML4.0 で名前の定義されている色)

色名	カラーネーム	色名	カラーネーム
黒	black	紺	navy
青	blue	緑	green
濃緑	teal	黄緑	lime
水	aqua	小豆	maroon
紫	purple	オリーブ	olive
グレー	gray	銀	silver
赤	red	紅紫	fuchsia
黄	yellow	白	white

表 13: 色 (HTML4.0 で名前の定義されていない色)

色名	カラーネーム	カラーコード
オレンジ	orange	#FFA500
スプリンググリーン	springgreen	#00FF7F
空	skyblue	#87CEEB
藤紫	royalblue	#4169E1
すみれ	violet	#EE82EE

- カラーネームによって指定してもブラウザによっては表示されないことがある。
- カラーコード (# のあとに 2 桁の 16 進数を 3 つ並べる) は color: #FFA500 のように指定する。
- 安全に利用できる 216 色のカラーネームとカラーコードは [http://html.eweb-design.com/0204\\_cn.html](http://html.eweb-design.com/0204_cn.html) あるいは <http://e-words.jp/p/r-colornames.html> などを見よ。

## 11.4 body 要素

文書の実際の内容を記述する。表示方法はスタイル要素に書く。

## 11.5 見出し

見出し (Heading) は、h1 要素から h6 要素までである。h1 要素が最上位の大見出し、h6 要素が最下位の見出しである。見出しの表示のさせ方は、ブラウザにより異なる。

見出しだけを抜き出すと目次が作れるような使い方を推奨する。大きな文字を表示するために h1 要素を用いるというような使い方はさけるべきである。

## 11.6 コメント

<!-- と --> で挟んだ部分はコメントになり、ブラウザでは表示されない。

## 11.7 段落、改行、空白

段落 (Paragraph) は、p 要素の内容とする。<p> と </p> で挟まれた部分に書く。) 一般的なブラウザでは、p 要素の前で改行され 1 行空白ができる。

強制的な改行 (forced line BReak) は br 要素である。<br /> とする。

空白をあげたいときは、&nbsp;を使う。例えば、&nbsp;&nbsp;&nbsp; とすると、半角 3 つ分の空白が入る。

注 & で始まり ; で終わる記号の列を文字実体参照という。

改行や空白をそのまま表示したいときは、pre 要素 (preformatted text=整形済みテキスト) を用いる。<pre> と </pre> で挟む。フォントは等幅フォントで表示される。

## 11.8 引用

長い引用文は、`blockquote` 要素の内容とする。終了タグ `</blockquote>` は省略できない。一般的なブラウザでは、左右がインデント (段下げ) されて表示される。

引用先は `cite` 属性で表わす。引用先はブラウザには表示されない。

## 11.9 リスト (箇条書き)

箇条書きには

- 番号なしリスト
- 番号つきリスト
- 定義リスト

がある。

### 11.9.1 番号なしリスト

番号なしリスト (Unordered List) は、`ul` 要素で表す。項目は `li` 要素である。終了タグはそれぞれ `</li>` `</ul>` である。

上の箇条書きのソースを示すと以下ようになる。

```
箇条書きには、
<ul>
  <li> 番号なしリスト </li>
  <li> 番号つきリスト </li>
  <li> 定義リスト </li>
</ul>
がある。
```

### 11.9.2 番号つきリスト

番号つきリスト (Ordered List) は、`ol` 要素で表す。項目は `li` 要素である。終了タグ `</li>`, `</ol>` は省略できない。

例 箇条書きの例。

```
<p>ホームページを公開するには、</p>
<ol>
  <li> 公開する場所を確保 </li>
  <li> ホームページのソースを作成 </li>
  <li> ホームページのソースを所定の場所に保存 </li>
</ol>
<p>が必要である。</p>
```

### 11.9.3 定義リスト

定義リスト (Definition List) は、dl 要素で表す。定義対象は dt 要素 (Definition Term)、定義対象の説明は dd 要素 (Definition Description) である。終了タグ `verb=|/dl|`, `</dt>`, `</dd>` は省略できない。

例

#### 定義リスト

その範囲が定義リストであることを示す。用語の説明以外に、シナリオ等にも利用できる。

#### 定義対象

定義する対象の他、シナリオの場合は役者名等を書く。

#### 定義対象の説明

定義対象の説明の他、シナリオの場合は役者のせりふ等を書く。

のソースは以下のとおりである。

```
<dl>
  <dt>定義リスト</dt>
  <dd> その範囲が定義リストであることを示す。
  用語の説明以外に、シナリオ等にも利用できる。 </dd>
  <dt>定義対象</dt>
  <dd> 定義する対象の他、シナリオの場合は役者名等を書く。 </dd>
  <dt>定義対象の説明</dt>
  <dd> 定義対象の説明の他、シナリオの場合は役者のせりふ等を書く。 </dd>
</dl>
```

## 11.10 テーブル (表)

表 (table) は、table 要素で表す。表の罫線の太さは、border 属性を使い、ピクセル単位 (0.3mm 程度) で表わす。(border 属性を省略すると、罫線のない表ができる。) 音声や点字など非視覚系メディアのため、summary 属性を用いることが推奨されている。表の見出しは、caption 要素で表す。(caption 要素はなくてもよい)。

横方向の一行 (Table Row 各行) は、tr 要素で表し、項目 (Table Data cell) は td 要素で表す。行あるいは列の見出しを表すヘッダセルは th 要素で表す。th 要素は通常のブラウザでは太字で表される。th 要素には abbr 属性で見出しの要約を付けることがアクセシビリティの向上になるとされている。

例 表

1 列の見出し 2 列の見出し

(1,1) (1,2)

(2,1) (2,2)

(3,1) (3,2)

のソースは次のようになる。



```

<table border="5" summary="4 行 2 列の表">
  <tr><th abbr="head">1 列の見出し</th><th abbr="head">2 列の見出し</th></tr>
  <tr><td>(1,1)<td>(1,2)</tr>
  <tr><td>(2,1)<td>(2,2)</tr>
  <tr><td>(3,1)<td>(3,1)</tr>
</table>

```

## 11.11 リンク

URL で指定されるページをリンク先に指定するには、a 要素 (anchor) の href 属性を用いる。<a href="リンク先の URL"> と </a> で挟んだ部分は、通常では青色で表示されアンダーラインが引かれる。この部分をダブルクリックするとリンク先に移動する。

例 東女のトップページをリンク先に指定するには、

```
<a href="http://www.twcu.ac.jp/">東女のトップページ</a>
```

とする。ブラウザは、東女のトップページを青色で表示しアンダーラインを引く。マウスカーソルを青色部分に移し、ダブルクリックすると東女のトップページが表示される。

リンク先が同じディレクトリにあるときは、HTML ファイルのパス名を指定する。

例 趣味のページの HTML ファイル hobby.html が、呼び出し元のファイル (index.html) と同じフォルダにおかれてあるとき、hobby.html をリンク先に指定するには、

```
<a href="./hobby.html">趣味</a>
```

とする。(./ は同じフォルダにあることを表わしている)。

また URL を使って

```
<a href="http://www-local.twcu.ac.jp/~b08a001/compIID/hobby.html">趣味</a>
```

としてもよい。

## 11.12 画像

指定した位置に画像を表示するのは、img 要素 (IMaGe) を用い、src 属性 (SouRCe) で画像ファイルのファイル名を指定し、画像が表示できないブラウザで閲覧する人や、目の不自由な人のために alt 属性 (ALternative) で説明をつける。標準的なブラウザでは、マウスカーソルを画像の上に置くと、alt で示した文字列が表示される。

画像の大きさを指定したいときは、width 属性で横幅をピクセル (pixel=画像上の点) 単位で指定し、height 属性で高さをピクセル単位で指定する。

例 表示したい画像 top\_image02.jpg を index.html と同じフォルダにおいてあるとき、

```
<IMG src="./top_image02.jpg" alt="東京女子大学トップページ画像">
```

とすると東京女子大学トップページ画像となり、画像の大きさを

```

```

と指定したとき、東京女子大学トップページ画像となる。

## 11.13 背景画像

index.html と同じフォルダにある abcd.gif を背景画像にするには style 要素を

```
<style type="text/css">
<!--
  h1{ color: green; text-align: center }
  body{ background-image: url("./abcd.gif"); color: black }
-->
</style>
```

と変更する。

## 11.14 色の変え方

### 11.14.1 文字列の色の変更 (スタイルシートの利用)

注意喚起を赤で表し、警告を黄で表すには、style 要素に

```
<style type="text/css">
<!--
  span.attention { color: red}
  span.warning { color: yellow}
-->
</style>
```

と書き (クラス名 attention, warning は自由に付けてよい)、本文では

```
<span class="attention">注意は赤</span>そして
<span class="warning">警告は黄</span>となる。
```

と書くと、注意は赤そして 警告は黄となる。

### 11.14.2 文字列の色の変更 (インラインスタイルシートの利用)

ある文字列を赤にしたいときは、span 要素の style 属性を用いて

```
<span style="color:red">文字列</span>
```

とする。

### 11.14.3 リンク先の色の変更

背景の色とリンクされている部分の色が同じ場合はリンクが表示されないが、リンクされている部分などの色を変更することにより表示することができる。

style 要素に

```
a:link { color: まだ見ていないリンクされている部分の色 }
a:visited { color: すでに見たリンクされている部分の色 }
a:active { color: リンクされている部分をクリックしたときの色 }
```

を加える。

まだ見ていないリンクされている部分の色を blue、すでに見たリンクされている部分の色を purple、リンクされている部分をクリックしたときの色 (マウスボタンが押されている間の色) を red とする場合

```
a:link {color: blue}
a:visited {color: purple}
a:active {color: red }
```

### 11.14.4 表の色の変更

表の背景色を skyblue、セルの文字の色を navy にするには、style 要素に

```
table{ background-color: skyblue }
tr{ color : navy }
```

を加える。

### 11.14.5 問い合わせ先の色の変更

style 要素に

```
address {color: blue}
```

とすると、address は blue で表示される。

## 11.15 ブロックレベル要素とインライン要素

- body 要素が直接含むことができる要素はブロックレベル要素のみである。
- ブロックレベル要素はインライン要素および他のブロックレベル要素を内容に取り得る。例外もある。例えば、p 要素はブロックレベル要素であるが、他のブロックレベル要素を内容に取ることはできない。
- インライン要素はテキスト及び他のインライン要素を内容に取り得る。

ブロックレベル要素 address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframes, noscript, ol, p, pre, table, ul

インライン要素 a, abbr, acronym, applet, b, basefont, bdo, big, br, button, cite, code, dfn, em, font, i, iframe, img, input, kbd, label, map, object, q, s, samp, script, select, small, span, strike, strong, sub, sup, textarea, tt, u, var

center, dir, isindex, menu, noframes applet, basefont, font, iframe, strike, u は、Transitional または Frameset であり、将来廃止される可能性がある。

## 11.16 画像の入手方法

### 11.16.1 HTML で扱える画像

HTML で扱える画像ファイルは、gif, jpeg, png の 3 種類である。それぞれの、内容、仕組み、長所と短所は、[htmz.com](http://htmz.com) に分かりやすい説明がある。

### 11.16.2 画像ファイルの作成

画像ファイルの作成には、次の方法がある。作成にあたっては著作権にくれぐれも配慮すること。

- インターネット上のフリー素材を利用する。
- AppleWorks や gimp 等を使って自分で描く。
- Excel のグラフを PNG 形式にして利用する。
- カメラ付き携帯で撮影し、大学のメールアドレス宛に送信する。
- 写真などをスキャナーを使って大学のコンピュータに取り込む。
- 自宅の PC に保存している (デジカメで撮影した) 写真を大学のコンピュータに取り込む。

自宅の PC のファイルを大学の自分のディレクトリに移すのは

- USB メモリに保存し、端末から読み込む。
- 大学の自分のメールアドレス宛のメールに添付する。

### 11.16.3 フリー素材の取り込み方

他人作ったの画像ファイルは、フリーと謳っているもの以外は自分のページで公開すると制作者の著作権を侵すことになる。

フリーの素材の取り込み方を以下に示す。

1. 検索エンジンで、フリー素材を公開しているページを探す。
2. ブラウザで閲覧し、気に入った画像を探す。
3. 画像にマウスポインタを合わせる。
4. マウスの右ボタンをクリックする。
5. 画像を名前をつけて保存
6. 選択の窓にファイル名をつけて [OK]

## 11.17 font-family

style 要素を

```
<style type="text/css">
  <!--
    p#font1{font-family: serif}
    p#font2{font-family: sans-serif}
    p#font3{font-family: cursive}
    p#font4{font-family: fantasy}
    p#font5{font-family: monospace}
  -->
</style>
```

とし、body 要素に

```
<p id="font1">
  serif <br>
  ホームページのソースは HTML(HyperText Markup Language) または
  XHTML(eXtended HyperText Markup Language) という言語で記述する。
</p>
```

などとする。

## 11.18 DIV の例

style 要素を

```
<style type="text/css">
  <!--
    div#fbox{border: solid 3px aqua; padding-left: 10px; padding-left: 10px}
  -->
</style>
```

とし、body 要素に

```
<div id="fbox">
<p>
この部分は枠で囲まれる。
</p>

<p>
ここまで枠で囲まれる。
</p>
</div>

<p>
ここは枠の外である。
</p>
```

などとする。