

# コンピュータリテラシ# 13 – HTML/CSSによる Web ページ記述

久野 靖 (電気通信大学)

2017.5.19

## 1 今回の目標

今回の目標は次の通りです。

- HTML による Web ページのマークアップについて学ぶ — Web サイトは研究その他の活動において情報を共有/公開する有力な手段でありサイトがすぐ作れるようになっておくに有益です。
- CSS による表現指定について学ぶ — 論理構造と表現の分離という考えは情報技術において重要であり、またページの表現を調整できるとサイトの視認性のために有効です。
- リンク・画像などの扱いについて学ぶ — サイトに画像を入れることで分かりやすさやアピールが高まり、リンクを使うことで多くの情報をまとめたサイトが作れるようになります。

## 2 HTML と CSS

### 2.1 Web とマークアップ exam

LaTeX でマークアップの仕組みは理解したけれど、見たまま (WYSIWYG) 方式が分かりやすい、と思う人もいるかも知れません。しかし実は、見たまま方式が「使えない」場合も存在します。しかも、皆様がいつも目にしている Web ページがまさにそうなです。なぜでしょう？

ワープロであれば、出力する紙サイズを決め、紙に合わせて配置や文字詰めやフォントを決めて「見たまま」で作業できます。しかし Web ページはどうでしょう？ Web には「紙サイズ」は存在せず、「1行何文字」も決められません。マシンによって使えるフォント違ってきますから、「ここは MS 明朝」と指定しても、そのフォントがないかも知れません。

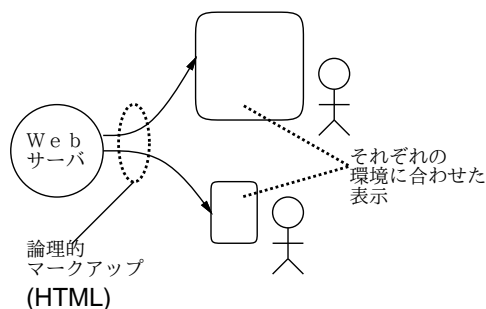


図 1: Web と HTML による論理的マークアップ

ではどうすればいいのでしょうか？ できることは「ここは表題」「ここは段落」「ここは箇条書き」など、文書の「構造を」指定しておいて、「ブラウザが画面に表示する時に」窓の幅や使えるフォントに合わせて整形してもらうことです。つまり意味づけ方式の (論理的な) マークアップを行う必要があるわけです (図 1)。そのための書き方が **HTML**(HyperText Markup Language) です。

しかし LaTeX に続いてまた別のマークアップを学ぶのか、と思う人もいるかも知れません。今日では、WYSIWYG ではないけれど GUI でページを作成できるソフトや CMS (contents management system) も多く使われていますから、HTML なしでもサイトは作れます。しかし、Web アプリケーションなどでプログラムから HTML を生成する場合には HTML を知っている必要があります。また、ソフトも何もなくても、少し練習すれば Web サイトが簡単に作れるようになり、便利です。

## 2.2 HTML の基本部分 exam

ではここで、HTML とはどんなものが簡単に説明し、練習用ページを作ってみます。sol では各自のホームディレクトリの下に `public_html` というサブディレクトリが用意されていて、そこに HTML ファイルを (誰でも読める保護設定で) 置くと、次の URL で学内限定で公開されます。

```
http://www.edu.cc.uec.ac.jp/~ユーザー名/ファイル名.html
```

たとえば Emacs で「`cd; cd public_html; emacs mypage.html &`」により編集を開始し、次のような内容を入れてみてください (〇〇のところは自分の名前をどうぞ)。<sup>1</sup>

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>〇〇's page</title>
</head>
<body>
<h1>〇〇です。</h1>
<p>…挨拶ないし自己紹介を書く…</p>
</body>
</html>
```

作成した後にブラウザで表示したようすを図 2 に示します (ファイル名は Emacs での指定と同じに)。



図 2: HTML をブラウザで表示

冒頭の「`<!DOCTYPE html>`」は以下のファイルが HTML であることを宣言します。以後、HTML では `<名前>…</名前>` という形でさまざまな種類のものを指定します。これを要素、`<名前>` を開始タグ、`</名前>` を終了タグと言います。開始タグから終了タグまでが 1 つの要素。ただし終了タグのない単独の要素もあります。また、開始タグの中に「`<名前 属性名="値" 属性名="値" …>`」のように 1 つ以上の追加指定 (属性) を書く場合もあります。

なお、このように HTML では「`<`」「`>`」「`"`」は特別な意味を持つので、この 3 つの文字を使いたい場合は代わりに「`&lt;`」「`&gt;`」「`&quot;`」という書き方を使う必要があります (この書き方を文字エンタリと言います)。

<sup>1</sup>ファイル名は好きしてよく、`index.html` にするとファイル名なしで (「/」までで) 表示されるページとなります。

要素の中には別の要素が入ることももあります (入れ子構造)。上に出て来る HTML の要素 (タグ) の意味は次の通り。

- `<html>…</html>` — HTML 文書全体をあらわす。
- `<head>…</head>` — ヘッダ (この文書に関する情報を記述する部分) をあらわす。
- `<meta charset="符号化">` — ファイルの符号化を指定 (閉じタグのない単独のタグ)。
- `<title>…</title>` — 文書のタイトルをあらわす。
- `<body>…</body>` — 文書本体 (ブラウザの窓の内側に表示される内容全体) をあらわす。
- `<h1>…</h1>` — レベル 1 の見出し (大見出し) をあらわす。さらに小さいレベルの見出しとして `<h2>…</h2>` ~ `<h6>…</h6>` まで使うことができる。
- `<p>…</p>` — 通常の段落をあらわす。

段落等の中はすべて窓の幅に合わせて詰め合わせられますから、空白や改行を入れても意味がありません。空白や改行で整形したい場合は `pre` 要素を使います。

- `<pre>…</pre>` — この内側は空白や改行をそのまま残す。

次のような形のを末尾 (ただし `</body>` の前) に入れてみると図 3 のようになります。

```
<pre>
  柿食へば
      鐘が鳴るなり
          法隆寺
</pre>
```



図 3: `pre` 要素は空白と改行が残る

最後にリンクのつけ方と、数個追加のタグを説明します。

- `<a href="URL">リンクテキスト</a>` — 「リンクテキスト」の部分がリンクとして表示され、そこを選択すると `URL` のページを表示する。
- `<h2>…</h2>` — 第 2 レベルの見出し (中見出し) を表す (1~6 まで使える)。
- `<blockquote>…</blockquote>` — 引用文を表す。
- `<div>…</div>` — デイビジョン (一連の範囲) を表す。

このように、`a` 要素の開始タグは「`href=...`」という指定 (属性と呼びます) がつけられ、それによってリンク先を指定します。これを利用して、「リンク集」を作ってみましょう。1 項目を 1 段落とします (リンクは段落等の中に入れることになっています)。例を示しましょう (図 4)。

<h2>私のリンク集</h2>

<p><a href="http://www.fujitv.co.jp/">フジテレビ</a>  
のサイトはよく見に行きます。ドラマ好き。</p>

<p><a href="http://www.nhk.or.jp/">NHK</a>はあまり  
見ません。</p>



図 4: リンクを入れたページ

**演習 1** sol に好きなファイル名で Web ページを作成しなさい (まず最小限のもので表示してみて、あとは編集するごとにブラウザのリロードで表示を更新するのがよいです)。表示できたら、次のテーマから 1 つ以上選んで作成・検討してみなさい。

- 見出しのタグは<h1>~</h1>から<h6>~</h6>まで 6 レベル用意されていますが、それらがどのように見目を違えているか (大見出しはより大見出しらしいか) 調べてみなさい。できれば、複数のブラウザでやってみて比較できるとなおよいです。
- 段落のタグ<p>~</p>はどのような機能を持っているか (たとえば字下げとか複数並べたときの様子とか)、検討してみなさい。ディビジョンのタグ<div>~</div>や引用文のタグ<blockquote>~</blockquote>についても同様に調べ、比較しなさい。
- (チャレンジ問題) 自分のページ内にリンクを作りなさい。URL はブラウザの URL 窓からコピーするのが間違えにくくていいでしょう。ページによってはうまくリンクできないものもあると思いますが、どのようなものはうまく行ってどのようなものはうまく行かないか検討してみなさい。

## 2.3 構造と表現の分離、スタイルシート exam

ここまで HTML 要素をいくつか見てきましたが、ページを作ってみると白地に黒で変わりばえしません。しかし世の中の Web ページを見ると、色や配置などの表現がざまなまに工夫されています。自分のページでもこれらの表現を行なうにはどうすればよいでしょうか? 実は過去には HTML にも「色をつける」「フォントを変える」「中央そろえ」など表現を指定する要素がありました。しかし、HTML 4.0 からはこの種の機能はすべて「非推奨」になり、代わりに「スタイルシート」と呼ばれる方法で表現を指定するようになりました。

なぜでしょうか? たとえば HTML では「大見出しを全部集めて来て一覧を作る」などの作業は (grep などのツールを使って) 簡単に行なえますが、そのとき見出しの中に「ここは青い色」など別のタグが混ざっているとうまく取り出せなかったり、または取り出したものにタグが混ざるなど、面倒なことが起きます。

それに、大見出しを青い色にするとしたら、全部の大見出しをそのように統一したいわけですが、すべての大見出しの所に余分に「ここからここまで青」というタグをつけて行くのも無駄な話です。

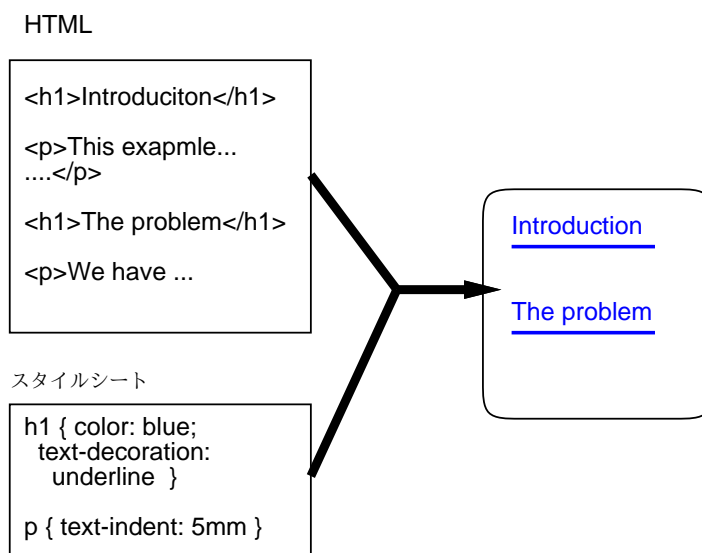


図 5: スタイルシート の概念

コンピュータで処理するのだから、「すべての大見出しは青」と「ひとつ」言えは済むようであるべきではないでしょうか (図 5)。スタイルシートとはちょうどそのように、つまり文書の構造のそれぞれについて「このような部分はこのような表現」という形で表現を指定する機能なわけです。

HTML と組み合わせるスタイルシート指定言語としては **CSS** (Cascading Style Sheet) が使われます。HTML に CSS の指定を追加する方法としては、次の 3 通りがあります。

- (1) CSS 指定を次のように **style 要素**の内側に書く。style 要素はヘッダ部分に入れる必要がある。

```
<style type="text/css">
CSS 指定…
…
</style>
```

- (2) CSS 指定を別ファイルに入れ、HTML のヘッダ部分に次のような **link 要素**を入れる (ここでは CSS 指定が `mystyle.css` というファイルに入っているものとしました)。この方法はやや複雑だけど、1 つの CSS ファイルを複数の HTML ページに適用させられる。

```
<link rel="stylesheet" href="mystyle.css" type="text/css">
```

- (3) HTML の各要素に **style 属性**を指定し、その値として CSS 指定の本体部分を書く。特定の要素だけに表現を指定する場合に使う。

```
<p style="color: blue">この段落は青い。</p>
```

以下では (1) の方法を使うようにします。CSS 指定を入れたページの例を示します (図 6)。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>sample</title>
```

```

<style type="text/css">
body { background: rgb(220,200,255) }
p { background: rgb(200,255,240) }
p { padding: 3mm 5mm }
</style>
</head>
<body>
<h2>スタイルシートとは</h2>
<p>HTML は文書の「どこからどこまでが何を意味するか」を指
定しますが、その見え方(色やあけ方など)は指定しません。</p>

<p>現在では、HTMLを補完する形で「スタイルシート」と呼ば
れる機能を組み合わせることで見え方を指定します。</p>
</body>
</html>

```

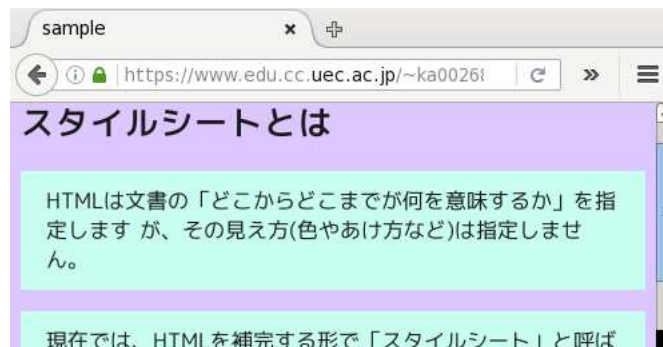


図 6: スタイルシートを使ったページ

## 2.4 CSS の指定方法

順序が逆になりましたが、CSS の指定方法について説明しましょう。まず、CSS の指定は「規則」の集まりで、1つの規則は次の形をしています。

セレクタ { プロパティ: 値; プロパティ: 値; … }

セレクタは、とりあえず HTML のタグとってください。つまり「この要素はこう表現する」という指定です。プロパティは、色や字下げなどです(すぐ後で説明します)。そして、それに対する値を指定します。値の指定方法は次の通り。

- 文字サイズの指定方法: 12pt (ポイント数)、xx-small、x-small、small、medium、large、x-large、xx-large、百分率(下記。本来のサイズの何倍/何%という形で指定)。
- 長さの指定方法: 1px(画面上の点)、1cm(センチ)、1em(文字「m」の幅1個ぶん)などがある。
- 百分率: %をつけた数値。ページ幅に対する割合などの指定に使用。
- 色の指定方法: black、blue、gray、green、maroon、navy、olive、purple、red、silver、white、yellow、rgb(赤、緑、青)ただし赤/緑/青は3原色の強さを0~255の数値で表す。
- ファイルや URL: url(ファイル名)、url(URL)。

CSS プロパティの代表的なものとしては次のものがあります。

- color: 色 — 文字色を指定。
- background: 色 — 背景色を指定。
- margin: 長さ — 要素の周囲のマージン (余白) 幅を指定。4 つの長さを指定すると「上、右、下、左」の長さを指定したことになる。2 つだと「上下、左右」、1 つだと 4 周全部がその長さに。また個別に指定したければ margin-bottom、margin-top、margin-left、margin-right という指定もできる (padding、border も同様)。
- padding: 長さ — 要素の周囲のパディング (詰めもの) 幅を指定。指定方法は margin と同様。
- border: 形状 色 長さ — 要素の枠を指定。形状として、solid(均一)、dashed(点線)、double(2重線)、ridge(土手)、groove(溝)、inset(くぼみ)、outset(出っぱり) 等が指定できる。色は枠の色、長さは枠の幅を指定。
- text-indent: 長さ — 段落先頭の字下げ幅を指定。
- text-align: 種別 — left、right、center で左そろえ、右そろえ、中央そろえを指定。
- text-decoration: 文字飾り。underline(下線)、blink(点滅) 等を指定。
- font-style: 傾き。normal、italic、oblique 等を指定。
- font-size: 文字の大きさを指定。

**演習 2** 先の例の HTML+CSS をまずそのまま表示し、次に段落や本体の背景色を変更してみなさい。その後、以下の課題から 1 つ以上やってみなさい。

- h1~h6 の見出しの要素から 1 つ以上選び、自分がかっこいいと思うデザインになるように CSS で表現を工夫してみなさい。たとえば次のような調整が可能だと思われます (どうするかは自分で選択)。
  - 枠線で囲んだりする。下だけ、または下と上だけ枠線にするなども。
  - 周囲のあきを調節したり、色や文字サイズを変えたりする。
  - 書体を傾けたり、文字を点滅させるなど。<sup>2</sup>
- p 要素、blockquote 要素、div 要素から 1 つ以上選び、本文の文章をこれらの要素として表示されたときにかっこいいデザインになるように CSS で表現を工夫してみなさい。たとえば次のような調整が可能だと思われます (どうするかは自分で選択)。
  - 段落先頭の字下げ量や周囲のあきを調整する。
  - 枠線で囲んだりする。左だけ、または左と右だけ枠線にするなども。
  - 書体を傾けたり、文字を点滅させるなど。
- (自由課題) レポート課題で提出する Web ページについて、CSS を調節して全体として「統一された、美しい見栄え」になるように工夫してみなさい。

## 3 より進んだ HTML と CSS の指定

### 3.1 ブロック要素とインライン要素 exam

HTML は文書の意味に従ったマークアップをするので、「どの要素の中にはどの要素が入られるか」ということがきちんと定まっています。たとえば h1 要素を使うと文字が大きくなりますが、では p 要素 (段落) の中に h1 を入れていいかということそれは駄目です。h1 はあくまでも大見出しであり、段落の途中に大見出しが挿入されるというのは文書としてあり得ないからです。

間違った HTML を書かないために、次のような原則は理解しておきましょう。

---

<sup>2</sup>点滅は普通は「サイテー」だとされます。確認してみてもは。

- ブロック要素とは、段落などと同じ位置に置けるもので、p、h1~h6、blockquote、table、div、ul、ol、dl などがある。
- ブロック要素の中にはブロック要素とインライン要素と文字が入られる (p だけ例外で、中にブロック要素が入られない)。
- インライン要素は、ブロック要素の中に入っている必要があり、中にインライン要素と文字が入られる。

このほか、li は ul と ol の中、th と td は tr の中など (いずれも後述)、いくつか要素の機能から決まっている規則がありますが、それは原則というより個別の規則ですね。

インライン要素としてはこれまで a 要素しか出てこなかったため、他の主なものを挙げます。

- <br> — ここで行かえをする。単独のタグ。
- <em>…</em> — この範囲を強調表示する。
- <b>…</b>、<i>…</i>、<code>…</code>、<var>…</var>、<sup>…</sup>、<sub>…</sub> — それぞれ、ボールド、イタリック、プログラムコード、変数、肩字、添字のスタイルとする。
-  — 埋め込み画像 (後述)。
- <span>…</span> — 任意のインラインの範囲 (後述)。

### 3.2 id と class exam

ここまでは CSS のセレクタとして要素 (タグ) の名前を使用して来ましたが、それだと当然「全部の〇〇」の表現を変更することになります。それでいい場合も多いですが、「ここだけ」「このいくつかだけ」のような指定も行いたいですね。そのため、各要素の開始タグには次の 2 種類の属性が指定可能です。

- <タグ名 … id="固有名"> — 固有名を指定する。
- <タグ名 … class="クラス名"> — クラス名を指定する。

固有名 (id) は「この特定の箇所」を示すのに使うのもので、従って同じものを 2 つ以上指定してはいけません。クラス名は複数指定することができます。これらを CSS で指定するときはセレクタとして次の書き方を用います (その先の書き方はこれまでと同じ)。

```
#固有名 { ... }
.クラス名 { ... }
```

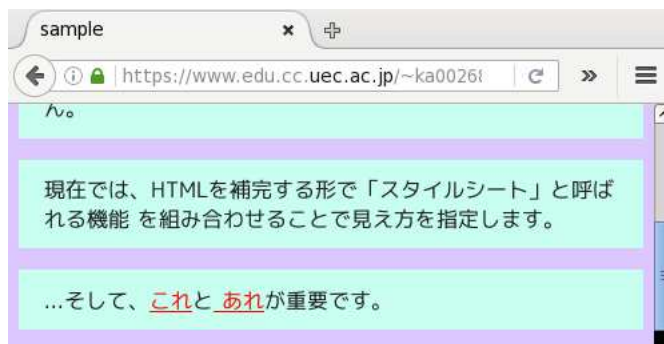


図 7: span とクラス指定による範囲の指定

そして、ここまですべて出て来た span 要素 (インラインの範囲) と div 要素 (ブロックの範囲) はいずれも、特定の表現はついていなくて、主に上記の方法で表現を設定するのに使われます。たとえば重要なところを赤字の下線つきにしたいとすると、HTML 側では span を用いて次のように記述します。



...そして、`<span class="imp">これ</span>`と`<span class="imp">あれ</span>`が重要です。

対応する CSS 側の記述は次のようにします。

```
.imp { color: red; text-decoration: underline }
```

### 3.3 箇条書と表 exam

HTML の追加として、LaTeX にもあった箇条書きと表について説明しておきます。まず番号なしの箇条書きは次のようになります。

```
<ul>
<li>一番目に大事なことは、事実に基づくこと。 </li>
<li>二番目に大事なものは、正確に推論する。 </li>
</ul>
```

`<ul>...</ul>`を`<ol>...</ol>`にすると番号が付きま (図 8)。LaTeX でいうと `ul` 要素は `itemize` 環境、`ol` 要素は `enumerate` 環境に相当します。箇条書きの項目はどちらも `li` 要素として含めます。

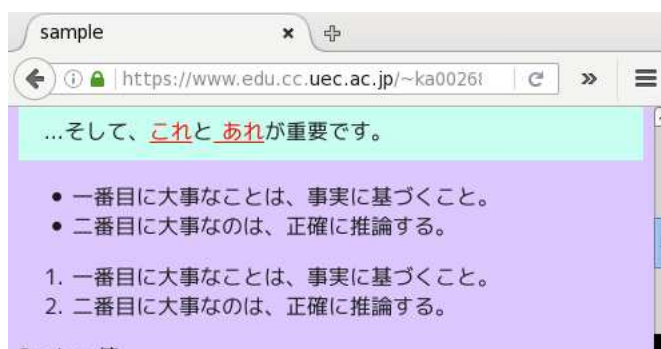


図 8: `ul` 要素と `ol` 要素による箇条書

一方、LaTeX にある `description` 環境に相当するものもあります。

```
<dl>
<dt>Boolean 値</dt>
<dt>論理値</dt>
<dd>「真偽値」とも呼ばれる。true/false のいずれか。 </dd>
<dt>整数値</dt>
<dd>一定の範囲の正・負・ゼロの整数。 </dd>
</dl>
```

こちらは LaTeX と異り、1つの項目に見出しを複数つけることもできます (図 9 上)。

次に表について説明しましょう。こちらはもう少し複雑で、表全体を表す `table` 要素、その中の 1 行ずつを表す `tr` 要素、そして行の中に入る各セルを表す `th` 要素または `td` 要素の 3 レベルから成ります。HTML の上で次のように縦横に揃えて書くと分かりやすいでしょう (「`border="2"`」という指定は表の罫線を幅 2 で描くという指定。無いと罫線も描かれない)。

```
<table border="2">
<tr><th>A</th><th>B</th><th>C</th></tr>
<tr><td>111</td><td>2222</td><td>3333</td></tr>
</table>
```

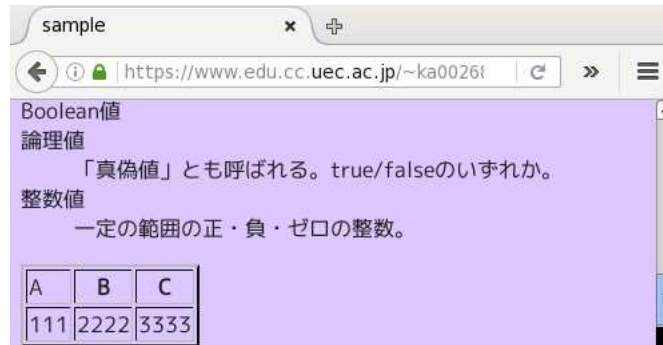


図 9: dl 要素の箇条書、table 要素

th と td の違いは、前者が見出し用、後者が一般用で、見出し用の方が少し強調表示になります (図 9 下)。また、th も td も「colspan="個数"」「rowspan="個数"」という属性が指定でき、これによって LaTeX と同様横や縦にまたがったセルが作れます。

### 3.4 CSS 指定の追加

最後に、主に table と div で活用する CSS 指定についても追加しておきます (最初のが table 用、残りは div でよく使います)。

- `border-collapse: collapse` — table 要素専用で、これを指定した表は隣接するセルの境界が 1 本の線になる (これを使う時は th や td の border も併せて指定することが多い)。
- `position:` — 位置指定。absolute(絶対位置を指定)、fixed(画面上での絶対位置を指定)、relative(本来あてはまる位置からのズレを指定) のいずれかが指定できる。
- `top:`、`left:`、`bottom:`、`right:` — 要素の上端、左端、下端、右端の位置を指定。
- `width:` 長さ、`height:` 長さ — この要素を整形する幅や高さを指定できる。div に対して使うことが多い。
- `float:` — 流し込みの指定。left(左に寄せて右に流し込む)、right(右に寄せて左に流し込む)、none が指定できる。
- `overflow:` — 高さを指定したとき、その中に入る内容が高さを超えたときの扱い。visible(そのまま見える)、hidden(見えなくする)、scroll(スクロールさせる) 等が指定できる。

**演習 3** 自分が作成した Web ページに関する説明をおこなう LaTeX 文書を作成しなさい。Web の画面を取り込むのは次の方法をおすすめします (が、別の方法でやっても取り込めていればかまいません)。

1. GUI の使える環境 (演習室等) で sol の窓から「firefox &」によりブラウザを起動する。
2. ブラウザで取り込みたいページ画面を表示させる。図が大きくなりすぎないようにブラウザの窓はかなり小さくした方がよい。
3. 「`import -compress rle fig1.eps`」のようにファイル名を指定して import コマンドを発行し (マウスカーソルが「+」形になる)、続いて取り込みたい窓の上をクリックするとその窓の画像が指定したファイルに取り込める。<sup>3</sup>
4. LaTeX のソースの中で図を入れたい箇所に次のように挿入指定 (幅とかファイル名は適宜変える。もっと別の入れ方でも入っていればよい)。

<sup>3</sup>-compress rle はランレングス符号化 (run-length encoding) による圧縮指定で、指定しないとファイルが大きい。

```
\begin{center}
\includegraphics[width=12cm]{fig1.eps}
\end{center}
```

## 本日の課題 **13A**

本日の課題は「演習 3」ですが、その文書の内容として「演習 1」「演習 2」に含まれる小問 (合計で 6 個) の中から 1 つ以上を選択し、結果をレポートとして報告するものとしてください。そして、LaTeX による整形結果を PDF ファイルにして、LMS の「レポート # 13」の箇所からアップロードしてください。以下の内容がこの順に含まれるようにしてください。

- 題名「コンピュタリテラシレポート # 13」、学籍番号と氏名、提出日付を書く。(グループでやったものはグループのメンバー全員の氏名も脚注などで別途書く。)
- 課題の再掲を書く (どんな課題であるかをレポートを読む人が分かる程度に要約する)。
- レポート本体の内容 (やったこととその結果) を書く。
- 考察 (課題をやった結果自分が新たに分かったことや考えたこと) を書く。
- 以下のアンケートに対する回答。

Q1. HTML によるページの記述はどれくらい知っていましたか。今回やってみてどうでしたか。

Q2. CSS による表現の指定はどれくらい知っていましたか。今回やってみてどうでしたか。

Q3. リフレクション (今回の課題で分かったこと) ・感想 ・要望をどうぞ。

なお、課題はグループでやって構いません。その場合も、(メンバー氏名を明記した上で) レポートは必ず各自で執筆してください。レポート文面が同一 (コピー) と認められた場合は同一であると認められた全員について点数にペナルティを科すことがあります。