

# 情報技術基礎 I: WWW と HTML/CSS

## (現代の情報技術 8)

久野 靖 (電気通信大学)

2020.5.12

### 1 「情報技術基礎」について

#### 1.1 趣旨と内容

今回から 3 回にわたり、講座「情報技術基礎」として World Wide Web (WWW) の話題を取り上げます。WWW は今日最も広くつかわれているネットワークサービスの 1 つであり、世の中に情報を提供する手段として有用です。WWW で表示される個々のページ (Web ページ) は HTML、CSS と呼ばれる記法で作成されており、ブラウザ (WWW を見るソフト) はネットから HTML+CSS を受け取って画面に表示しています。

本講座では全 3 回で、この HTML+CSS によるページ作成を一通りマスターして頂きます。今日ではさまざまなソフトウェアを通じて HTML、CSS を知らなくてもページを作ることはできますが、Web ページでどのようなことができるのか、またソフトウェアがどのようにして Web ページを作り出しているのかを知っておくことはとても役に立ちます。

#### 1.2 進め方

HTML+CSS は講義で話だけ聞いても身につかないので、本講座では説明は最低限とし、実習を中心とします。そのため、テキストでは一通り解説が書いてありますが、授業時にはその一部だけを取り上げます。残りの内容については課題に必要と思う箇所を参照してください。

実習は毎回「このようなページを作ってください」という課題がゴールとなり、そこに向かって進むようにできています。課題の内容はレポート (講座が 3 回なのでレポートも 3 件) にしていただき、翌週授業前日を締め切りとして教務部に提出していただきます (提出方法は教務部の掲示を見てください)。本講座部分の成績はレポートに基づいてつけます。

課題は協力してやって構いませんが (グループでやる課題もあります)、レポートは必ず各自でそれぞれ書いてください。レポート中で協力して作ったページ内容やその画面表示については同じ内容になるでしょうが、説明や考察等は自力で書くということです。その部分が互いに同一のレポートはどれも等しく減点します。

久野は本学教員でないので、質問や連絡はメールをお願いします。アドレスは「y-kuno@uec.ac.jp」です。メールには件名を必ずつけ、その先頭に [情報技術基礎] とつけてください。また本名と学生証番号を本文中で名のつてから用件に入るようにしてください。また、掲示、実習用ページ、この講座のための資料などは次の場所に置きます。メモしておいて活用してください。

<http://tk2-262-40766.vs.sakura.ne.jp/hoshi20/>

#### 1.3 今回の目標

3 回の講座はそれぞれその回の目標があります。今回の目標は、次のものです。

- HTML+CSS がどのようなものか体験し理解する。
- 簡単な Web ページを自力で構想し書けるようになる。

自分で情報発信できるようになると嬉しいと思いますので、がんばってください。

## 2 ネットワークサービスとその構成

私達がネットワークを使うのは、メッセージを送るなど、何か「役に立つ機能」を使うためです。ネットワーク上でこのような機能を提供するものをネットワークサービスと呼びます。

ネットワークサービスはどんな形で提供されているのでしょうか？ 通信を行うためには、互いに通信する2つのプログラムが「あちら」(リモート側/ネットワークの向こう側)と「こちら」(ローカル側/自分の手もと)で動く必要があります。「こちら」は自分のマシンだから自由になりますが、「あちら」はどうでしょうか？ この問題に対する1つの解は、プログラムを次の2種類に分けることです(サーバが「あちら」、クライアントが「こちら」になります)。

- サーバ — サービスを提供するマシンで常時稼働していて、サービス要求があるまで待ち、要求があったらサービスを提供する。
- クライアント — サーバに要求を出して、そのサービスを受ける。

これをクライアントサーバ方式と呼びます(図1)。この方式では、上述の通信の問題が自然な形で解決できますし、サービス提供のために必要な資源(データ等)はサーバのところで一括管理できるので、各種サービスの実現が比較的簡単に行えます。このため、クライアントサーバ方式はネットワークの初期から今日に至るまでネットワークサービスの構成方式として広く使われています。

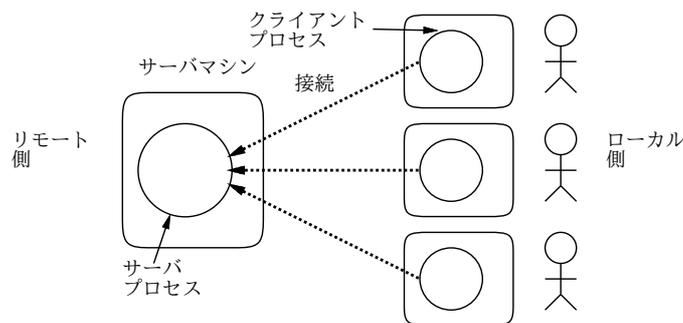


図 1: クライアントサーバ方式

クライアントはサービスを利用するユーザの数だけ動きます。ユーザはサービスを使うときにクライアントを起動し、そのプログラムがサーバと接続しサービスを受けます。つまりクライアントサーバ方式では、クライアントが各ユーザのマシンで動き、サーバはサービスを提供するための専用マシンで動く形になります。

1つのマシンで複数のサービスを提供することも普通にあるので、クライアントは特定のサービスを利用するのにサービスを提供するマシンの IP アドレスに加えて「ポート番号」と呼ばれる番号を指定します。WWW やメールなどの標準的なサービスは、標準化されたポート番号を使うようになっていますが、サービスやクライアントによっては初期設定でこれらの情報を個別に設定する必要がある場合もあります。

## 3 World Wide Web

### 3.1 WWW とリンク

今日、最も広く利用されているネットワークサービスは **WWW**(World Wide Web ないし Web) です。皆様もそのクライアントである **Web ブラウザ**を使わない日はないと思います。しかし、その見慣れた画面の裏側の仕組みを考えたことのある人は、多くないかも知れません。

Web の基本的な枠組みはハイパーテキストです。ハイパーテキストとは次のようなものです。

- 計算機の画面上にテキストや画像などの内容(コンテンツ)が表示されている。

- コンテンツの中には、他のコンテンツやその特定箇所を「指し示して」いる箇所が埋め込まれている。これをリンクという。
- リンクの箇所を何らかの方法で選択すると、画面はそのリンク先の内容に切り替わる。
- このようなリンクで互いに結び合わされたコンテンツの集まりは、リンクを自由にたどりながら読み進んでいくことができる。

ハイパーテキストの概念そのものは WWW よりずっと前から存在しましたが、それをネットワークサービスの形に組み立てたのが WWW なのです。

WWW ではネットワーク上にある様々なモノ (資源ないしコンテンツ) を指す手段として **URL** (Uniform Resource Locator) と呼ばれる形式を使用しています。URL は一般に次の形をとります。

スキーム: アドレス

スキーム中で最も一般的なのは、Web サーバからコンテンツを取り寄せるプロトコル **HTTP** (HyperText Transfer Protocol) を使う場合で、次の形を取ります。<sup>1</sup>

`http://ホスト指定/ディレクトリ/.../ディレクトリ/ファイル`

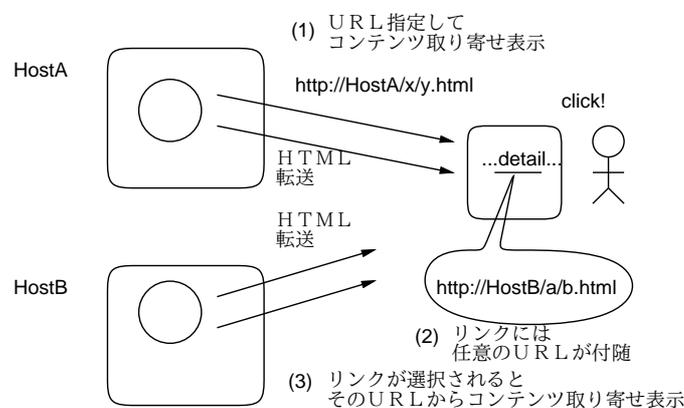


図 2: リンクの仕組み

なぜここで URL について長々と説明しているのかというと、URL こそが WWW の「肝」だからです。WWW でコンテンツを伝送しているのはごく簡単なプロトコルであり、ブラウザがやっていることは基本的には次の 2 点だけです。

- a. 指定された URL からコンテンツを取り寄せる。
- b. 取り寄せたコンテンツを適切な形で画面に表示する。

コンテンツを取り寄せたとき、それが **HTML** (HyperText Markup Language) 形式であれば、中にリンクが含まれていることがあります (そこに別コンテンツを指す URL 情報が含まれています)。そしてユーザがリンクを選択すると、ブラウザはリンク先のコンテンツを取り寄せて表示し (上記 a+b の動作)、結果的に「別のページへ飛ぶ」わけです (図 2)。これだけで済むのは、「リンク先」が URL の形で表され、世界中のどのサーバのどのコンテンツでも自由に指し示せるから、なのです。

## 4 HTML と CSS

### 4.1 Web とマークアップ

普段ワープロソフトなどで文書を作成するときは、文字の大きさ、フォント、配置などを画面上で確認しながら、メニューなどを使って変更・設定していきます。これを見たまま方式 (**WYSIWIG**,

<sup>1</sup>既に説明したように、サーバとブラウザの間で暗号化通信を行うために TLS/SSL を使うこともでき、その場合はスキームとして `http:` の代わりに `https:` を使います。

What You See Is What You Get) と呼びます。しかし、WWW ではこの方法は使えません。なぜでしょうか。

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。

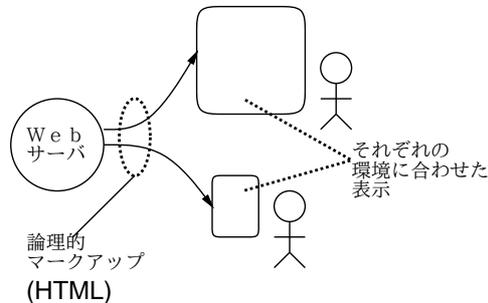


図 3: Web と HTML による論理的マークアップ

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。

これを実現するため、WWW ではマークアップ (markup) 方式を使います。この方式では、通常のコンテンツ (内容) と一緒に、この部分は何である、のような「目印 (マークアップ)」をファイルに書き込みます。WWW のためのマークアップの形式を **HTML**(HyperText Markup Language) と呼びます。そして、Web サーバからは HTML 形式のファイルが送られてきて、ブラウザが自分の環境に合わせて整形し表示する、というやり方をします (図 3)。

マークアップ言語は「どのような書き方をするか」を学んで使いこなす必要があるため、WYSIWYG よりハードルが高くなります。実は今日では、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、ソフトも何もなくとも、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。そのようなことから、本講座では HTML を学んで頂きます。

## 4.2 HTML の基本部分

これから HTML とはどのようなものか学んで行きますが、すぐ実践してみるために「練習ページ」を使用します (図 4)。このページの下方にコピー用のテンプレートがありますので、それをコピーして先頭の入力欄に貼ってから修正してください。表示してみたい時はいつでも Run ボタンを押すことで、そのページを整形したようすがすぐ下の領域に表示されます。

では、最初に作成してもらう内容を見ていただき、説明します。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"> ← Windows で作る時は shift_jis かも
<title>〇〇's page</title>
</head>
<body>
<h1>〇〇です。 </h1> ←ここを好きに作る
```

```
<p>…挨拶ないし自己紹介を書く…</p> ←ここを好きに作る
</body>
</html>
```

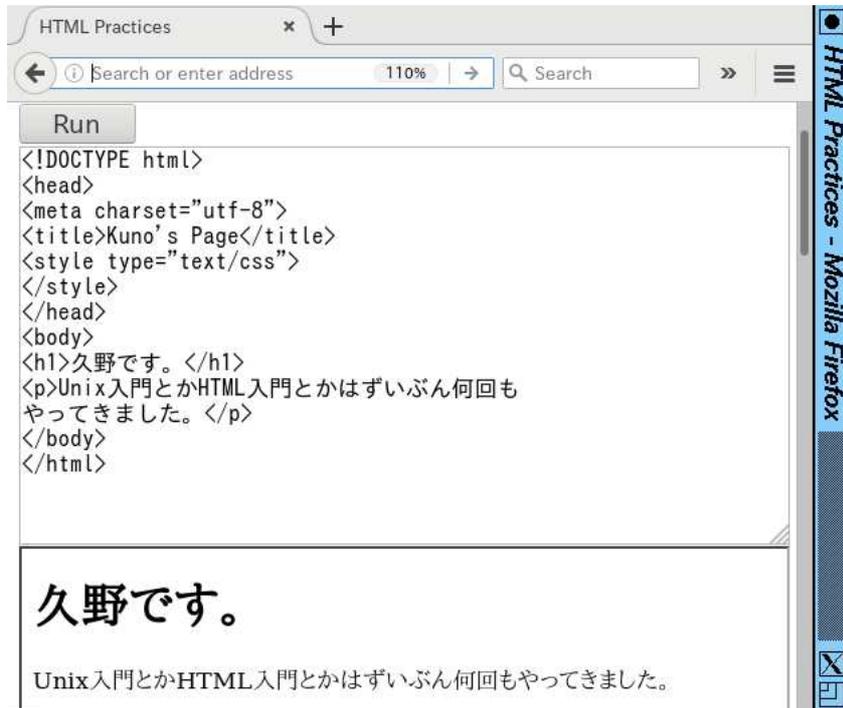


図 4: HTML 練習ページを使っているところ

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「<」「>」およびすぐに説明する機能に使う「&」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;」「&amp;」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トリまたは文字参照と言います)。

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。

次のような形のものを末尾 (ただし`</body>`の前) に入れてみると図5のようになります。最低限、`h1`要素 (見出し)、`p` 要素 (段落)、`pre` 要素 (そのまま) さえ覚えればいちおうページは作れます。

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



図 5: `pre` 要素は空白と改行が残る

次はリンクのつけ方と、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。
- `<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ておくことになっています)。例を示しましょう (図6)。

```
<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>
```

**演習 1** 好きな内容の 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、少しずつ書き足すのがよいです)。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た目を違えているか (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。デイビジョンのタグ`<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。
- 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。



図 6: リンクを入れたページ

### 4.3 構造と表現の分離、スタイルシート

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。

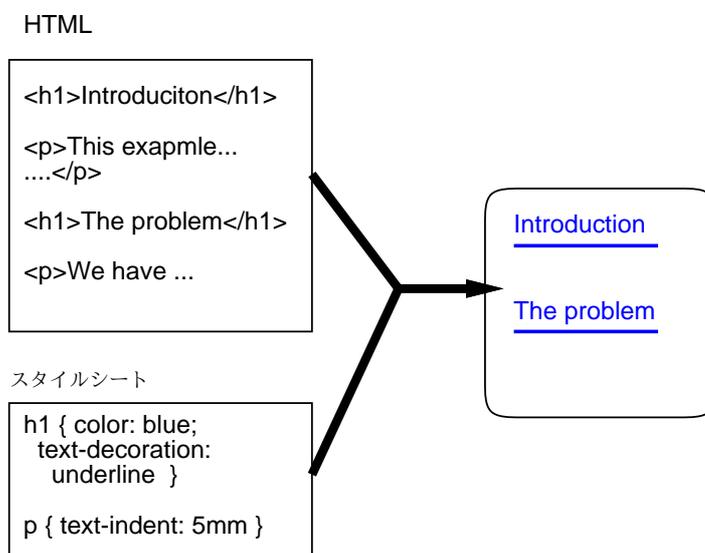


図 7: スタイルシート の概念

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。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「ひとつこと」言えれば済むようであるべきではないでしょうか？ (図 7)。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**の内側を書く。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しました)。この方法はやや複雑だけれど、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 (練習ページのテンプレートにも style 要素は用意してあります。先程作ったページの style 要素の中に次のものを入れて表示した様子を図 8 に示します。本体部分 (body 要素の中) は一切変更していないけれど、色々な表現ができていることが分かります。

…前略…

```
<style type="text/css">
h1 { color: blue; text-decoration: underline }
p { text-indent: 5mm; background: rgb(180,200,255) }
pre { border: double green 4px }
</style>
```

…後略…

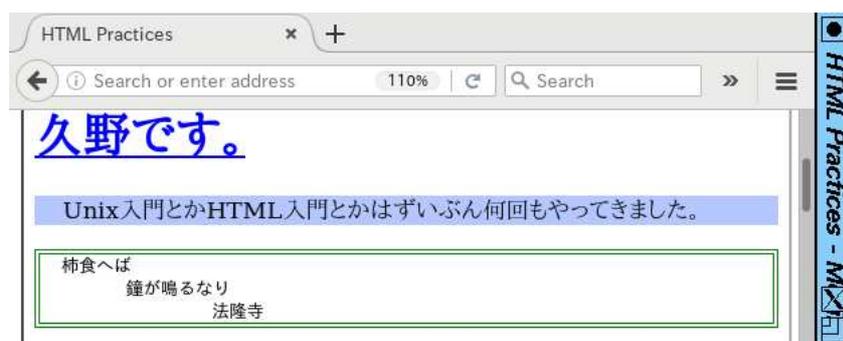


図 8: スタイルシートを適用した場合

#### 4.4 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 (本来のサイズの何%か)。
- 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)、overline(上線)、line-through(抹消線) 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** HTML の練習ページに先の例の CSS 指定を (1行ずつ順番に) 追加して効果を確認してみなさい。その後、以下の課題から1つ以上やってみなさい。

- a. 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、背景色や文字サイズを変えたりする。
- b. 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。
- c. 「p:hover { color: red }」のようにセレクトタに「:hover」を指定するとその要素上にマウスポインタが乗った時だけの表現を指定できるのでやってみる。<sup>2</sup>
- d. (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。

<sup>2</sup>さらにプロパティとして「transition: 3s」のように変化に要する秒数を指定すると色等がその時間かけてゆっくり変化する。戻るときもゆっくりにしたければ: hover をつけない側の指定にも transition を指定する。

## 5 より進んだ HTML と CSS の指定 option

### 5.1 予習: ファイルに HTML を作る

ここまでは「HTML 練習ページ」を使ってページを作ってきましたが、これではブラウザを止めたら内容が無くなってしまいますし、複数のページを作ることもできません。そこで、これまで作った HTML をファイルに保存して「普通の形で」ブラウザで表示してみましょう。保存場所としては「各自の USB メモリの中のフォルダ」を使用します。

- (1) エクスプローラで作業に使うフォルダを表示 (まだ無ければ「新規フォルダ」機能で作る)。
- (2) メニューの「アクセサリ」→「メモ帳」でメモ帳を開く。
- (3) 「HTML 練習ページ」で作ってあったソースを貼り付けるか、自分で HTML を打ち込む。
- (4) 「名前をつけて保存」で保存するが、保存時にファイルの種類を「すべての種類」にしてから、「mypage1.html」などの名前を (最後が「.html」) 保存。
- (5) エクスプローラ上の HTML ファイルをドラッグしてブラウザの窓にドロップすると開く。

このフォルダを保存するには、フォルダを USB メモリにドラッグしてコピーするか、または最初から USB メモリ上にフォルダを作って作業してください。

慣れてきたら「練習ページ」は使わずに、直接メモ帳で打ち込んだり直したりしてもよいです。ただ、練習ページを使うとすぐチェックできおかしなところが分かるという利点があります。

以下では基本的に今作業しているフォルダに色々なファイルを作ることになります。

### 5.2 ブロック要素とインライン要素

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。

- 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もので、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)。

### 5.3 id と class

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 { ... }
```

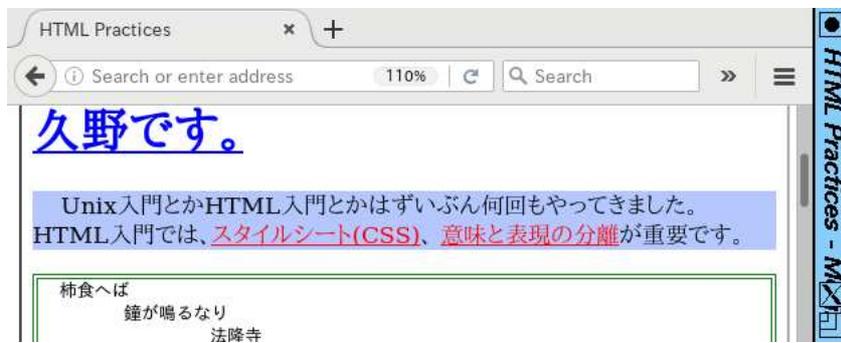


図 9: span とクラス指定による範囲の指定

そして、ここまでに出て来た **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。

```
HTML 入門では、<span class="imp">スタイルシート (CSS)</span>、  
<span class="imp">意味と表現の分離</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 }
```

### 5.4 箇条書と表

HTML の追加として、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>スタイルシートは、表現の豊かなページを作るために必要。</li>  
<li>意味と表現の分離は、スタイルシートの適切な使用のために必要。</li>  
</ul>
```

<ul>...</ul>を<ol>...</ol>にすると番号が付き (図 10)。箇条書きの項目はどちらも li 要素として含めます。

一方、説明付きの箇条書きもあります。

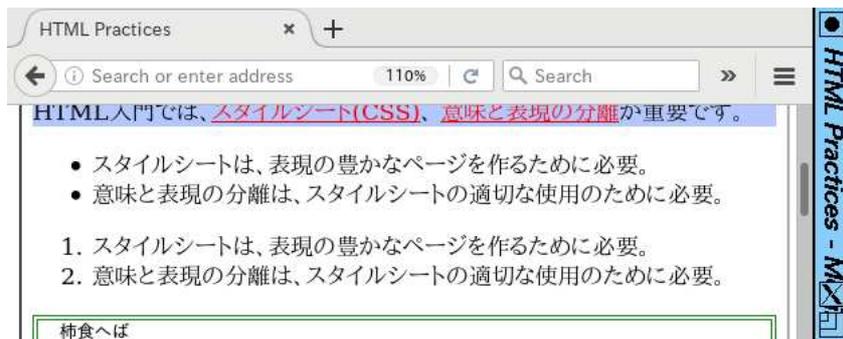


図 10: ul 要素と ol 要素による箇条書

```

<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 class="imp">整数値</dt>
<dd>一定の範囲の正・負・ゼロの整数。 </dd>
</dl>

```

1つの項目に見出しを複数つけることもできます(図 11 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 class="imp">C</th></tr>
<tr><td>111</td><td>2222</td><td>3333</td></tr>
</table>

```

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 11 下)。また、th も td も「colspan="個数"」「rowspan="個数"」という属性が指定でき、これによって横や縦に複数個ぶんまたがったセルが作れます。

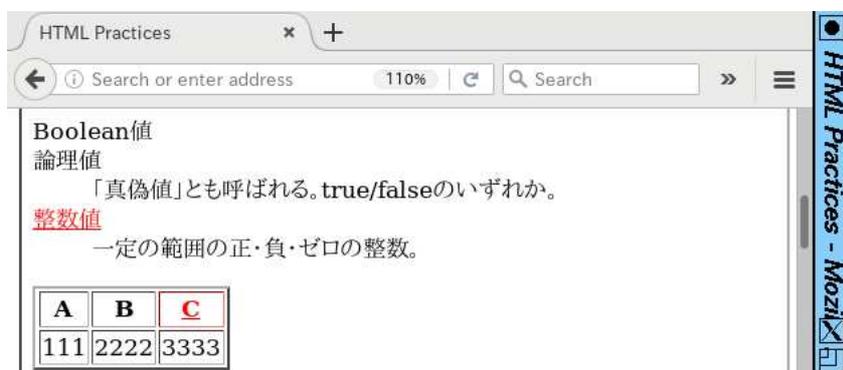


図 11: dl 要素の箇条書、table 要素

## 5.5 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(最初のもののが 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** 次のようなページを作ってみなさい。

- a. id 機能または class 機能を使って「特定の何かだけ表現が他と違う」ようにしてみる。
- b. 箇条書のどれかの形を使ってみる。
- c. 表を使ってみる。

## 本日の課題 **8A**

今回のレポート課題は「HTML」「CSS」の両方を使った Web ページを1つ以上作り、その内容を説明するレポートを作成することです。レポートは紙で提出なので、手書きでもコンピュータで作成でも構いませんが、HTML のソースやページの表示をおこなった様子を含めてください。これらはブラウザ等ソフトの印刷機能や画面プリント機能で出力された紙をはさんでもよいですし、画面キャプチャをレポートの中に図として貼りつけてもかまいません。

レポートは次回授業前日一杯までに教務部の指示に従って提出してください。

レポートには次の内容がこの順で含まれること。

- 表紙。題名「現代の情報技術 8A」、学籍番号と氏名、提出日付を書く。さらに、グループで協力してやった場合はグループの他のメンバー全員の学籍番号と氏名を書く。
- 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。今回の場合、作成した HTML とその説明、画面表示とその説明が含まれること。
- 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、課題はグループで協力してやって構いません。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。