

概要

教科書 P11~76

- Webサイトの基本
- 環境構築
- HTMLの基本
- 画像とリンクの表示

Webサイト作成によく使われる言語

プログラミング言語と言えば色々あるが、Webサイトを作れる言語は現状ほぼ一本化されている。それが、HTML、CSSとJavaScriptの組み合わせである。

HTML(HyperText Markup Language)

主にWebページの内容を「タグ」によってマークアップし、文書内の役割を指定するのに使われる。

```
<html>
  <h1>適当なHTML</h1>
  <p>本文</p>
  <br/>
  <p>本文2</p>
</html>
```

複数のバージョンがあるが、現在はHTML5(2014年勧告)が主流。

HTMLを中学や高校で習う人も多いだろうが、多分内容が古いのであまり役に立たない。

```
<marquee>横に動く</marquee> <!--HTML5では廃止されたよ!-->
```

あくまで文書の内容を整えるのに専念し、レイアウトや装飾は後述のCSSに任せるのが現代流。

CSS(Cascading Style Sheets)

Webページの見え目や構造などのスタイルを指定するための言語。

```
/*h1(見出し)に装飾を加える*/
h1 {
  color: #ff0000;
  line-height: 1em;
  font-size: 40px;
}
```

現在はCSS3が主流(CSS4も少しずつ勧告され始めている)。

JavaScript

Webサイトに動きを付けたり、振る舞いを与えるための言語。

```
//無限にアラートが出続ける
//逮捕案件
while(1)
{
    alert("無限アラート");
}
```

所謂プログラミング言語なので、計算などにも利用できる。

その他

PHP, Ruby等

Webサーバ上で動作し、動的にHTMLを生成することで動きのあるサービスを作ることができる。

PHPは[去年の資料](#)があるので興味があったら参考にしてね

React, Vue, Angular等

上記のJavaScriptを拡張し、利便性や生産性を高めたフレームワーク。言語もJavaScriptだけでなくTypeScriptなどで書くことができる。

本講習会では使わないです

環境構築

エディタ

既に好きなものがある人はそれを使おう。

ない人は<https://code.visualstudio.com/>からVisual Studio Codeをダウンロード！

よく分からない場合はブラウザ上でできる[CodePen](#)を使うのもあり

HTMLの基本

役割

HTMLの最大の特徴は「リンク」を持つことで、ハイパーテキスト同士のつながりを「ハイパーリンク」と言う。ハイパーリンクによって文書同士が繋がり、次のページに遷移することが可能になる。

また、文書内のテキストに対して適切な役割と意味を付与することをマークアップと言う。HTMLはタグで囲むことでマークアップしている。

なぜマークアップが必要なのか

A. コンピュータが文書を読めるようにするため

```
<!--SampleA.html-->
```

```
<!DOCTYPE html>
```

```
<html>
```

KCSの歴史

目次

- ・発足
- ・空白の10年間
- ・成長期
- ・全盛期
- ・衰退期

発足

慶應義塾大学では、1958年に創立100年の記念事業の一環として、10進法トランジスタ式電子計算機K-1を開発した。電気試験所のトランジスタ式電子計算機ETL Mark

IVをモデルに全トランジスタ化を行い、さらに浮動小数点演算機構および指標レジスタを加え、1語の桁数を12桁に拡張した。

内部記憶には高速磁気ドラムを、入出力装置は機械式テープリーダー、フレキシライタ、テープパンチを使用した。

...

```
</html>
```

これを表示すると**こうなる**

→人間には読めてもパソコンには生の文章じゃ分らない！

→マークアップしよう

基本の基本

先程の文章を簡単にHTMLで書くと以下ようになる。

```
<!--SampleB.html-->
```

```
<!DOCTYPE html>
```

```
<html lang="ja">
```

```
<h1>KCSの歴史</h1>
```

```
<h2>目次</h2>
```

```
<ul>
```

```
<li>発足</li>
```

```
<li>空白の10年間</li>
```

```
<li>成長期</li>
```

```

    <li>全盛期</li>
    <li>衰退期</li>
</ul>
<h2>発足</h2>
<p>
    慶應義塾大学では、1958年に創立100年の記念事業の一環として、10進法トランジスタ式電子計算機K-1を開発した。電気試験所のトランジスタ式電子計算機ETL Mark IVをモデルに全トランジスタ化を行い、さらに浮動小数点演算機構および指標レジスタを加え、1語の桁数を12桁に拡張した。
    <br/>
    内部記憶には高速磁気ドラムを、入出力装置は機械式テープリーダ、フレキシライタ、テープパンチを使用した。
  </p>
  <p>...</p>
</html>

```

プレビュー

要素ごとに説明していく。

記述ルール①：タグは全て半角文字で記述する。

<>で囲まれたものがタグである。タグ部分の記号と英数字は全て半角文字で記述しなければならない。1箇所でも全角文字が入るとエラーの原因になるので注意。

またタグの名前は小文字小文字関係ないが、基本小文字で書くのがルール。

```

<!-- 良い例 -->
<p>半角</p>

<!-- 悪い例 -->
< p > 全角 < / p >

```

全角と半角の見分けが付きにくい文字も多いので、普段から半角を使う癖を身に付けよう！

記述ルール②：半角スペース、改行コード等は無視される

半角スペースと改行コードは無視される。そのためいくら空白を作っても改行しても出力される画面に影響はない。

コードが見やすくなるよう自由に使おう。

他になんの文字が無視されるのかよく分かりませんでした。

記述ルール③：開始タグと終了タグのセットで使う。

基本的にタグは開始タグと終了タグをセットで使い、それを適用したい要素を囲むように記述する。

```

<p>←開始タグ | 終了タグ→</p>

```

セットで使うことを忘れないように、中身の要素の最初に数個の空白を入れるとよい。(インデント)

```
<!--これはタブ1つで取ってる-->
<ul>
  <li>インデント</li>
</ul>
```

ただし改行タグbrのように中身の無いものはタグを一つしか使わない。

```
<!--どちらでもいい-->
<br>
<br/>
```

この場合は終了タグを付けてはいけない。

XHTMLの場合は最後に/を入れなければならない

SampleB.htmlの説明

1行目：これがHTMLの文章であることを宣言する。バージョンによって宣言が異なり、例はHTML5であることを示す。「DOCTYPE」を大文字で、「html」は小文字で書くのが通例。

```
<!DOCTYPE html>
```

2行目：HTMLの内容はhtmlタグで囲う必要がある(省略可能な場合も)。

HTML5的にはhtmlタグの中にはheadタグとbodyタグしか書いてはいけないので注意!

```
<html lang="ja">
  ...
</html>
```

属性(attribute)

htmlタグの中に書かれているlangを、htmlの属性という。

属性は個々のタグに対してなんらかの性質を与えることができ、その場合=に続けて値を渡す。

タグの種類によって付与できる属性は決まっている。

- 開始タグの名前から半角スペースを開けて属性名を書く
- 続けて=を書き、属性の値をダブルクォーテーション「"」で囲んで後ろにつける
- 複数属性を付ける場合はそれぞれ半角スペースを開ける

```
<a href="https://kcs1959.jp" target="_blank">KCSホームページ</a>
```

3行目：h1,h2,...タグは見出しを作る。数字の小さい方が大きな見出しになる。

```
<h1>KCSの歴史</h1>
```

結果：

KCSの歴史

5~11行目：箇条書き。全体をulタグで囲み、箇条書きのそれぞれをliタグで並べる。olタグで囲むと数字の箇条書きになる。

```
<ul>
  <li>発足</li>
  <li>空白の10年間</li>
  <li>成長期</li>
  <li>全盛期</li>
  <li>衰退期</li>
</ul>
```

結果：

- 発足
- 空白の10年間
- 成長期
- 全盛期
- 衰退期

ul: Unordered Listの略。黒丸で箇条書きを作る

ol: Ordered Listの略。数字で箇条書きを作る

li: Listの略

13~17行目：pタグで囲んだ部分は段落として認識される。

```
<p>
```

慶應義塾大学では、1958年に創立100年の記念事業の一環として、10進法トランジスタ式電子計算機K-1を開発した。電気試験所のトランジスタ式電子計算機ETL Mark IVをモデルに全トランジスタ化を行い、さらに浮動小数点演算機構および指標レジスタを加え、1語の桁数を12桁に拡張した。

```
<br/>
```

内部記憶には高速磁気ドラムを、入出力装置は機械式テープリーダー、フレキシライタ、テープパンチを使用した。

```
</p>
<p>...</p>
```

結果：

慶應義塾大学では、1958年に創立100年の記念事業の一環として、10進法トランジスタ式電子計算機K-1を開発した。電気試験所のトランジスタ式電子計算機ETL Mark IVをモデルに全トランジスタ化を行い、さらに浮動小数点演算機構および指標レジスタを加え、1語の桁数を12桁に拡張した。内部記憶には高速磁気ドラムを、入出力装置は機械式テープリーダ、フレキシライタ、テープパンチを使用した。

...

演習：HelloWorld

自分の環境でHTMLを表示してみよう

HTML5の基本

HTML5ではタグの種類も増え、役割が分散されていっている。各文の要素だけでなく文書の大きな構造もタグを使って表す。

```
<!--SampleC.html-->
<!DOCTYPE html>
<html>
  <head>
    <title>ページタイトル</title>
  </head>
  <body>
    <header>
      <h1>ヘッダ</h1>
      <nav>
        <ul>
          <li>ナビゲーション</li>
          <li>バー</li>
        </ul>
      </nav>
    </header>
    <main>
      <h1>メイン要素</h1>
      <section>
        <h2>中身1</h2>
      </section>
      <section>
        <h2>中身2</h2>
      </section>
    </main>
    <aside>
      <h2>あまり関係ない話</h2>
    </aside>
    <footer>
```

```
        <h2>コピーライトとか</h2>
    </footer>
</body>
</html>
```

プレビュー

- **headタグ** : ページタイトルなど、直接ページ表示には関係ない情報を記述する空間
- **bodyタグ** : ページの内容を書く空間
- **headerタグ** : ページのヘッダ部分(ロゴとかリンクとか)を記述する場所
- **navタグ** : サイト内ナビゲーションを作る
- **mainタグ** : ページのメイン要素を記述する場所
- **sectionタグ** : 見出しの付けられたひとかたまりを表す
- **asideタグ** : 本題と遠い話をする場所
- **footerタグ** : リンクやコピーライトを書くフッターを記述する場所

これらは主にコンピュータに何を書いているかを伝えるためのタグであるため、見た目にはほとんど影響しない。

追加でよく使うタグ

divタグ

```
<div>
  <h2>なにかの</h2>
  <p>ひとまとまり</p>
</div>
```

複数の要素をひとまとめにする役割を持つ。HTMLのみでは見た目に影響しない。CSSなどでまとまりを一気に扱いたい場合に役立つ。

spanタグ

適当な長い文章に一部装飾を入れたい

これもタグ自体に意味を持つことはないが、CSSと組み合わせることでその部分の見た目を変える。

head要素に書くもの

省略

//TOOD:演習問題

画像を表示する

imgタグとsrc属性を使う

- src属性：画像のパスを書く
- alt属性：画像が見つからなかった場合に表示するテキスト

HTMLファイルと同じディレクトリに写真を用意しよう！

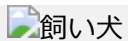
```

```

結果：



もし画像のパスを間違えている/存在しない場合は



のような表示になる。

サイズの調整

imgタグのwidth属性とheight属性を使う

```
<!--どちらも指定すると、縦横ピッタリの大きさになる-->  
  
  
<!--片方のみ指定すると、画像の元の比率で表示される-->  

```

結果：



リンクを設定する

aタグとhref属性を使う

- href属性：移動したいページのリンクを貼る

```
<a href="https://kcs1959.jp">KCSホームページ</a>
```

結果：

[KCSホームページ](https://kcs1959.jp)

パス

imgタグやaタグを使うときは使いたいものへのパスを書く。パスとはファイルへの行き方を示したようなもので、絶対パスと相対パスの2種類がある。

絶対パス

コンピュータのファイルシステムのルート(最上位)からファイルの場所を表現する方法。

```
/*例：Windowsの場合*/  
D:\Projects\Web2020HTML_CSS\day1\image.png
```

Web上のもので言えばURLにあたるかな？

どこから見てもファイルを探すことができるが、長くなってしまったり、ローカルのコンピュータ内のファイルは他から参照できなかつたりする。

相対パス

自分(ファイル)から見て目的のファイルがどこにあるかで表現する方法。

自分のいるフォルダは「./」で、それより上の階層を示すときは「../」で一つディレクトリを上げられる。

```
/*ここから見た感じ*/  
./../README.md
```

最初の「./」は省略してもよい

Webでは自分の管理するファイルを参照するときは、基本相対パスを使用する。

[SampleD.html](#)を見る

演習：

```
TopPage.html  
SubPage  
├image.png  
└SubPage.html
```

的な構成でページを作ってみよう！

おしまい

宿題

いや特に...