

情報基礎  
CSSを用いたWebページ作成

CSSとは

- ◆ Cascading Style Sheetの省略表記。シーエスエスと読む
- ◆ Webページのレイアウト(視覚的構造)を定義する「スタイルシート」の規格の一つ
- ◆ Webの標準化団体であるW3C(World Wide Web Consortium)によって標準化
- ◆ 現在ひろく使われているのはCSS2 (version 2)

W3Cで推奨される考え方

- ◆ 論理構造: マークアップ言語
  - ◆ HTML, XHTML, XMLなど
- ◆ レイアウト(見た目): スタイルシート言語
  - ◆ CSSなど

HTML&CSSの経緯

- ◆ 1990 Web登場
  - ◆ HTMLはWebページの論理構造を記述するもので、レイアウトを記述するものではない
  - ◆ レイアウトを記述するためのスタイルシート言語は規定されなかった
- ◆ 1993頃以降 ブラウザ普及
  - ◆ サイト製作者は見た目をよくしたい
  - ◆ ブラウザにあわせてレイアウト記述
    - ◆ 例) 文字を大きくしたいときに<h3>を使う
  - ◆ ブラウザによって異なる独自要素も登場 例) <font>
  - ◆ HTMLの混乱と複雑化

HTML&CSSの経緯

- ◆ 1996 レイアウトを記述する枠組としてCSSが規定(CSS1)
- ◆ 1997 HTML3.2: 折衷的なHTML
  - ◆ 暫定的措置としてレイアウトに関する独自要素の取り入れ
- ◆ 1997 HTML4.0: 論理構造とレイアウトの分離
- ◆ 1998 CSS2
- ◆ 1999 HTML4.01
- ◆ 2000 XHTML1.0
- ◆ 2001 XHTML1.1

HTML&CSSの経緯

- ◆ 現在: 依然として混沌状態
  - ◆ CSSを使わないページが多い
  - ◆ CSSを使う場合: HTML4.01 or XHTML1.0 or 1.1 + CSS2が一般的
- ◆ 注意: CSS規定以前に開発されたブラウザはCSSに対応していない

## CSS利用のメリット

- ◆ 詳細なレイアウトを記述できる
  - ◆ HTMLよりレイアウト記述力が高い
  - ◆ 例) テキストや画像を好きな位置に配置できる
- ◆ HTMLから、レイアウトに関する記述を除去できる
  - ◆ HTMLがシンプルに
  - ◆ 文法間違いを減らせる→情報を正しく伝達できる、表示の間違いを減らせる、維持管理が楽に
  - ◆ アクセシビリティ向上
    - ◆ 例) HTMLのtable要素を使ってレイアウトすると、音声読み上げソフトを利用してWebページを聞く視覚障害者に正しく情報が伝わりにくい

## CSS利用のメリット

- ◆ レイアウトに関する記述を外部CSSファイルにできる
  - ◆ サイトの維持管理が楽に(後述)
- ◆ ユーザが自分の好きなスタイルシートでページを見ることができる
- ◆ 正しい(推奨される)考え方

## CSSのメリットの例(1)

- ◆ 「New」という文字列の色を赤から黄色にかえたい場合
- ◆ 100ページ(100個のHTMLファイル)のWebサイト

## HTMLのみで記述

- ◆ HTMLファイル

```
◆ <font color="red">New</font>
```

100個のHTMLファイルで、redをyellowに変更

## HTML+CSSで記述

- ◆ HTML

```
◆ <span class="attention">New</span>
```

“attention (名前は自由)”というclassを設定  
100個のHTMLファイルは変更不要

- ◆ CSS(別ファイル)

```
◆ .attention
```

```
◆ color: yellow;
```

1個のCSSファイルにおいて“attention”の色を  
redからyellowに変更

## CSSのメリットの例(2)

- ◆ <h1>の下に下線をひきたい場合
- ◆ 100ページ(100個のHTMLファイル)のWebサイト

## HTMLのみで記述

### ◆HTMLファイル

- ◆<h1>適当な見出し</h1>
- ◆<hr color="lime" size="5">

100個のファイルHTMLで<hr>の  
colorとsize設定変更

- ◆注:HTMLでは、<hr>には本来color, size属性はなく、ブラウザ独自要素

## HTML+CSSで記述

### ◆HTML

- ◆<h1>適当な見出し</h1>

<hr>は不要  
100個のHTMLファイルは変更不要

### ◆CSS(別ファイル)

- ◆h1
  - ◆border-bottom: solid 5px lime;

1個のCSSファイルの<h1>の設定変更

## 課題

- ◆1. トップページをCSSを用いて作り直す
- ◆2. Konqueror, Mozilla(Linux), Internet Explorer(Windows)で確認
- ◆3. Another HTML LintまたはW3C HTML ValidatorでHTML文法チェック, W3C CSS ValidatorでCSS文法チェック

## 課題(余裕のある人)

- ◆4. 画像ファイルをimagesディレクトリにまとめる
- ◆5. 2ページ目以降もCSSを用いて作り直す

## 一般的なディレクトリ構成

- ◆public\_html(ディレクトリ)
  - ◆css(ディレクトリ)
    - ◆index.css(名前は自由)
  - ◆images(ディレクトリ)
  - ◆index.html
  - ◆他のHTMLファイル

## 課題の手順(準備)

- ◆0. 見本の確認
  - ◆<http://www.ex.media.osaka-cu.ac.jp/~harumi/mihon/>以下
  - ◆index.html, profile.html, hobby.html
  - ◆css/index.css
  - ◆images/photo.gif
- ◆コピーして使用してもよい

## 課題の手順

- ◆1. トップページをCSSを用いて作り直す
  - ◆1.1 cssという名前のディレクトリをpublic\_htmlの下に作成
  - ◆1.2 index.cssという名前(名前は自由)のファイルをcssの下に作成
  - ◆1.3 index.htmlをtest.htmlという名前で保存(バックアップのため)
  - ◆1.4 test.htmlからindex.cssを参照
    - ◆<link rel="stylesheet" type="text/css" href="css/index.css">

## 課題の手順

- ◆1. トップページをCSSを用いて作り直す
  - ◆1.4 ボックスのレイアウトを考える(見本参照)
  - ◆1.5 ボックスとその内容の記述をtest.htmlに行う(HTML)
  - ◆1.6 ボックスの見た目をindex.cssに記述する(CSS)
  - ◆注)1.5, 1.6 は、上から順番に少しずつ行う
  - ◆注)CSSで記述しやすいようにデザインを変更してもよい

## 課題の手順

- ◆1. トップページをCSSを用いて作り直す
  - ◆1.7 完成したらindex.htmlをindex.html.bakという名前で保存し(バックアップのため)、test.htmlをindex.htmlの名前で保存

## 課題の手順

- ◆2. Konqueror, Mozilla, Internet Explorerで確認

## 課題の手順

- ◆3. HTMLとCSSの文法チェック
  - ◆3.1 Another HTML Lint or HTML Validator
  - ◆3.2 CSS Validator

## 課題の手順(余裕がある人)

- ◆4. 画像ファイルをimagesディレクトリにまとめる
  - ◆4.1 public\_htmlの下にimagesディレクトリを作成
  - ◆4.2 画像ファイル(.jpg, .gifなど)をimagesディレクトリに移動
  - ◆4.3 HTMLファイルの中の画像へのパスを変更して保存
    - ◆例) をに

## 課題の手順(余裕がある人)

- ◆ 5. 2ページ目以降もCSSを利用して書き直す
  - ◆ 手順は1.と同じ

## 参考

- ◆ Webページを作ろう
  - ◆ <http://www.ex.media.osaka-cu.ac.jp/~nagata/tutorial/>
- ◆ とほほのスタイルシート入門
  - ◆ <http://www.tohoho-web.com/css/basic.htm>

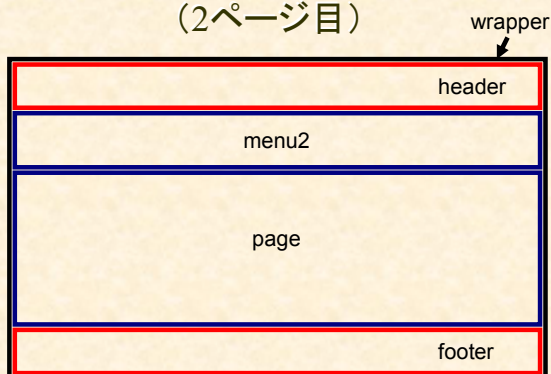
## 参考

- ◆ Another HTML Lint
  - ◆ <http://openlab.ring.gr.jp/k16/htmlint/htmlint.html>
- ◆ W3C HTML Validator
  - ◆ <http://validator.w3.org/>
- ◆ W3C CSS Validator
  - ◆ <http://jigsaw.w3.org/css-validator/>

## 見本の基本レイアウト (トップページ)



## 見本の基本レイアウト (2ページ目)



## marginとpadding



## marginとpaddingの記述方法

- ◆margin: 0; 上下左右が0
- ◆margin: 0 auto; 上下が0 左右は中央揃え
- ◆margin: 0 0 0 0; 上、右、下、左が0
  
- ◆個別に指定するときは
  - ◆margin-left, margin-right, margin-top, margin-bottom
  - ◆例) margin-left: 100;

## CSSファイルの見方

- ◆HTML:ブラウザで「ドキュメントのソースを表示」
- ◆CSS:HTMLを見てCSSファイルのURLをブラウザで入力

## CSS利用の問題

- ◆HTMLより仕様が複雑
- ◆CSSに対応していないブラウザもある
- ◆ユーザがスタイルシートを利用しないこともある
- ◆ブラウザによってCSS対応状況が異なり、バグもある
- ◆CSSの文法通りに記述しても正しく表示されないことがある。ブラウザの対応度合いの差はHTMLのより大きい
- ◆複数のブラウザで美しく正しく表示させるためには、多くの手間と時間とノウハウが必要

## どうすればいいか

- ◆100%正しい、という答えはない
- ◆見た目に全くこだわらないのであれば
  - ◆CSSを使わず、HTMLにレイアウトに関する記述を書かない、という選択肢もある
- ◆見た目に多少でもこだわるのであれば
  - ◆ブラウザの対応状況を考慮しながら、できる範囲でCSSを用いたページ作成が望ましい