

**タグ付きテキストのお話。**

**1 時間目**

**WEB でのまーくあっぷらんげーじ**

うさこ（うえだうさこ）

usakororin

- 1999年 iMacDV 購入。web 制作にハマる。
- 2001年 印刷会社に入社
- 2003年 web 制作会社トゥーウェイズ立ち上げ
- 2004年 大阪芸術大デザイン学科入学
- 2010年 電子書籍の勉強会 イーパ部 in 大阪を立ち上げ



**タグって、マークアップって  
どういうこと？というのを  
WEBを例にお勉強**

**タグって、なあに？**

**タグで、コンテンツに意味を持たせる**  
ブラウザは、どのようにwebページを  
理解する？

「ここはページのタイトルですよ！」  
とわかる、目印を付けてあげる。  
その目印が**タグ**

# WEBの世界ではHTMLでタグ付け

## 基本形

<タグ>コンテンツ</タグ>

目印を付けたい箇所をタグで挟む。

# WEBページの材料について

# Web ページの材料

HTML : ページの内容

このふたつは、単なるテキストです。

HTML ファイルさえあれば、ページは作れます。

CSS : HTMLに装飾をするもの

その他

- 画像、動画、FLASH

- JavaScript これもテキストです。



**デモ1**

**HTMLだけで書かれたコンテンツ**

# デモ2

## CSSをあててみる

# デモ3

## 別のCSSをあててみる

# HTMLについて

DTPということなので、web にありがち  
のリンクやフォームなどではなく、文書、  
レイアウト部分にしぼってお話ませう。

**HTML**ってひっくるめていうけど  
細かくいうと、

**HTML**とか**XHTML**とかある。

バージョンも色々ある。

拡張子は **.html** ときに**.htm**

場合により**.xhtml**

# HTMLとは

はいぱーてきすと まーくあっぷ らんげーじ

**H**yper**T**ext **M**arkup **L**anguage

コンテンツをタグによって意味付けしたものの。

現在使われているのは  
HTML4.01とかHTML5かな？  
XHTMLの方が多い気がします

# XHTMLとは

いくすてんしぶる

**Extensible HyperText Markup Language**

HTMLをXMLのルールに合うようにした  
たもの。

ほんのり書き方がシビア。

現在使われているのは  
HTML1.0とか1.1 そろそろ5も?  
2は消えちゃいました。

# HTMLファイルの構成



```
<html>
```

```
<head>
```

表に見えない箇所

```
</head>
```

```
<body>
```

ページのコンテンツ

```
</body>
```

```
</html>
```

ざっくり書くとこんな感じ。

全体が<html>と</html>ではさまれ、その中にheadとbodyがある。

head内には、タイトル、みんな大好きな文字コード指定、など  
ちなみに現在ではUTF-8が主流ですが、ちょっと前はShift-JISとか  
プログラム絡む場合はEUC-JPなんかも使われていました。  
webサーバはLinuxということが多いので、最終的に改行コードはLFかな

# ページコンテンツのタグ付け まずは段落から

**<p>** これは段落を表します。 **</p>**  
開始タグ    内容    終了タグ

## 要素

コンテンツをタグで挟み込む。  
要素名は基本的に英単語の略  
p : paragraph (段落)

## 見出し

HTMLは、見た目ではなく、  
コンテンツの意味によって  
タグ付けを行います。

`<h1>` これは見出しです `</h1>`

**h1 : heading (見出し)**

**h1～h6まであります。**

※InDesignでは「段落」スタイルでひとまとめだけど、HTMLでは「段落」と「見出し」は区別されます。

ルビ

読めないんだもの

```
<ruby>  
<rb>有紗子</rb>  
<rp> (  
<rt>うさこ</rt>  
<rp>)</rp>  
</ruby>
```

## 表示例

---

有紗子 (うさこ)

または

う さ こ  
有紗子

※ブラウザによる表記の違い。

※めっちゃ面倒！

改行  
から  
閉じタグがない空要素タグ



`<p>`コツコツ。`<br />`  
廊下から足音が聞こ  
える。`</p>`

## 表示例

---

コツコツ。  
廊下から足音が聞こ  
える。

**br** : break InDesignの強制改行  
空要素なので閉じタグ無し。

※XHTMLでは閉じタグが必須なので、タグの  
最後に「/」をつける。

XMLのルールでは<br></br>っていう、内容が空っぽ状態で良いらしいけど、その短縮系で書きます。

ブラウザの暴走?を避けるため、/の前には半角スペース。

# 属性

## 要素のこまごま

名前難しそうだよねぇ。そんなに身構えないでね。

## 属性とその値

```
<p class="lead">ここに内容</p>
```

要素に属性がつくことがある。  
つけられる属性は決まっている。  
要素によって、必須の属性もある。

画像

必須属性を持つ要素

```

```

**src** : 画像パス (場所)

**alt** : 代替文字

上記2属性は必須

※width、heightは任意だがよくつける

※空要素なので閉じタグ無し

「画像パス」はhtmlファイルからの相対パスまたは、サーバのルートからの絶対パス、上のまたはhttpではいまるurl(これも絶対パスと呼ぶ)の3パターンがあります。

## コメント

見せたかないけど書いときたい

アプリケーションが勝手に書いたりもするので、一応覚えておくと良いと思います。

<!-- ココは表示されません -->

```
<p>募集中！ </p>  
<!--<p>締め切りました。  
</p>-->
```

表示例

募集中！

一時的に隠しておきたい表記、ソース整理のため、アプリケーションが書き出すコメント等があります。

フォント

色やサイズを設定したい



**<font> ~ </font>**

```
<font color="#f00"
size="+1">久々に</
font>使うぜ。
```

**表示例**

---

久々に使うぜ。

**設定できる属性**

**size、color、faceなど**

※現在主流の HTML/XHTML では廃止。  
見た目は CSS で設定します。

# ブロック要素とインライン要素

また新しい言葉ですが、インラインで考えると簡単!

# ブロック要素

h1~h6、pなど

**表示** (cssでどうとでもなります)

---

左右いっぱいになる。

タグの前後で改行される。

## インライン要素

**strong**、**font**、**ruby**など

文章中の一部分に反映される

# インライン要素の例

## 強調 <strong>

<p>お申し込みは  
<strong>3月18  
日</strong>締め  
切りです！ </p>

### 表示例

お申し込みは**3月18**  
日締め切りです！

# InDesignで考えると解りやすい

段落スタイル → ブロック要素

文字スタイル → インライン要素

# div要素とspan要素

## 特別な意味を持たない要素

HTMLは意味付けだけが必要な  
場面があります



## div 要素 (ブロック要素)

```
<div id="header">  
<h1>見出し</h1>  
<p>文章</p>  
</div>
```

レイアウト目的などで一部コンテンツをまとめる。

※HTML5では、`header`などの要素があります。

## span 要素（インライン要素）

`<p>`例えば文章の一部を`<span>`赤く  
`</span>`したい時なんかに使えます  
ね！`</p>`

部分的に文字に変化をさせたいときに  
使います。

# **class属性とid属性**

イメージとしては、  
InDesignで、複数の段落に「段落スタイル」で設定したスタイルをかける

そのスタイル名の設定方法

## class 属性

ページに何度でも同じ名前のもので使える。

段落スタイル名と同じようなもの

```
<p class="photo">
```

```

```

```
</p>
```

ファイル名ですが、タグもクラス名も、全部半角英数字で書きます。代替テキストもコンテンツはもちろん日本語 Oko

## id 属性

大枠のレイアウトに利用されることが多い。

```
<div id="header">○○○</div>
```

**id="header"**を設定できるのは1ページに一カ所だけ

あとは、アンカー（ページ内リンク）の設定にも利用します。  
タグは他にも色々あるけど、よく使うのはあとはリストとテーブルくらいです。気軽にチャレンジ！

についてCSS

**CSSってなあに？**



# CSSとは

かすけーでいんぐ すたいる しー と  
**Cascading Style Sheet**

**HTMLに記述されたタグ等に対して、  
装飾を指定するもの**

**HTMLでも  
文字の色とか変えられたやん？**

## 役割分担

**HTML** : コンテンツに対する意味付け

**CSS** : 見え方を設定

**HTMLに装飾のデータを入れちゃう**

**と、データの流用がしにくい、更新性が悪いのです**

**CSSってどう書くの？**

```
p { color : #444 ; }
```

セレクタ プロパティ 値

セレクタを複数設定する  
場合

p,ul,li

など、カンマで区切る

スタイルを複数設定  
する場合

プロパティ:値;

を複数記述

CSS もテキストファイルです。

セレクタは「セレクトするもの」って考えたら解りやすいかも。

# CSSをHTMLに反映させるには？

# 別ファイルに記述して呼び出す

style.css内

```
p {color:#444; }
```

HTML側ヘッダ内

```
<link rel="stylesheet" type="text/  
css" href="style.css" media="all" />
```

HTMLファイル内にいくつ設定してもOK  
メディアタイプは、色々設定できて面白いよ!  
screenとprintはよく使います。

# セレクタについて

**CSS Selectors Level3** <http://www.w3.org/TR/selectors/>



## ある要素に反映

```
p { color:#444; }
```

p要素に反映される

```
p span { color:#f00; }
```

p要素の中にある、span要素に反映される。

## id 属性に反映

```
#header { color:#444; }
```

id = "header"が設定された要素に反映される

```
div#header { color:#444; }
```

id = "header"が設定された div要素に反映される

## class 属性に反映

```
.photo { border:solid 1px #f00; }
```

**class = "photo"が設定された 要素に反映される**

```
p.photo { border:solid 1px #f00; }
```

**class = "photo"が設定された p要素に反映される**

セレクタは他にも沢山あって、色々な選択ができて、例えばドロップキャプとかも表現できます。

# レイアウト周り

# レイアウトでよく使うプロパティ

**width** : 幅、 **height** : 高さ

**margin** : マージン

**padding** : 内側余白

**display** : 表示形態

**float** : 浮かせて寄せる

**background** : 背景

**border** : 境界

※まだまだあるけど書ききれません

# 背景画像

デザイン上の装飾はHTML上に画像として置かずに、背景として設定する。

```
body {  
background-image: url(bg.png);  
}
```

※CSS3では複数の背景画像が設定できる

例えば飾り罫線とか、リストのポチっとしたアイコンなど

コンテンツとしてい身をもたないものは、HTMLに置くのではなくCSSに設定しちゃう

リストアイコンはcssで設定があるけど、空きなどがブラウザによってまちまちなので、背景にすることが多いです。モダンブラウザだと疑似クラスセレクタ :before なんかで代用もできます。

# 背景画像を使ったデモ

# 文字周り



## 文字表現でよく使うプロパティ

**font-size** : 文字サイズ、 **color** : 文字色

**line-height** : 行の高さ

**font-weight** : 文字の太さ

**text-decoration** : 文字装飾 (下線等)

**text-align** : 文字揃え

**text-indent** : インデント

※まだまだあるけど書ききれません

**どんなことができるかデモ**

**後で困らないように**

class名も意味でつけよう

「価格青字にしといて」

```
<span class="blue"> 500円</span>
```

```
span.blue { color: blue; }
```

「やっぱり赤で！」

```
<span class="blue"> 500円</span>
```

```
span.blue { color: red; }
```

後はメニューとか leftmenu が右にくる  
とか、ちょっと気持ち悪いことになります。

# タグ付けとコーディングの 重要性

# HTML、CSS バージョンの移り変わり

**2001年頃**      テーブルレイアウト

**2004年頃～**    XHTML+CSS    増える

MovableTypeの流行とSEO（検索エンジン最適化）  
NN4.7切り捨て

**2011年頃～**    HTML5+CSS3

スマートフォンの流行で軽量で装飾ができる  
CSS3の利用

**タグ付けしたら  
何かいいことがあるん？**



- 1) 内容を解釈できる
- 2) データを活用できる

ぶらうざが、アプリケーションが、グーグルなどの検索ロボットが  
印刷物に、電子書籍に、音声読み上げに。特に XHTML であると XML として扱える

まとめ

**HTMLとは**

**文書を論理的に記述するための文法**

**CSSとは**

**文書の装飾を司るもの**

**CSSがなくても意味が通じるHTMLを作るのがとっても大事。**

# 参考サイト

## HTMLは世界共通！

**W3C (仕様を決めてるところだから、一番正確)**  
<http://www.w3.org/>

**W3C Validation Service (適合してるかチェック)**  
<http://validator.w3.org/>

**w3schools.com (タグごとに調べるのに便利)**  
<http://www.w3schools.com/>

**HTMLクイックリファレンス (HTMLもCSSも。解りやすい)**  
<http://www.htmq.com/>

**HTML5.jp (HTML5の日本語情報ならココ。本も出てます)**  
<http://html5.jp/>

仕様は日本語訳されているものもありますが、日本語だとちょっと取り違えちゃうかも、って感じのものもあるので、仕様を読むなら英語をお勧めします。  
あと、一番したのは、本があるけど、それが解りやすくおススメ!

**おしまい。**