

# CSSで書籍組版を

(有)イー・エイド

藤島 雅宏

FormatterClub「CSS書籍組版」セミナー講演資料

開催日 2013年10月18日

会場 東京都中央区浜町区民館

# 簡単そう

- × 易しい →→→ 簡単なことしかできない
  - × CSSはWeb用であり、組版機能が劣る
- 
- コンテンツはHTMLなので、構造が簡単
  - デフォルトのスタイルでそれらしく組める
  - CSSによるスタイルシートは直截的に分かる

# HTMLコンテンツの構造

```
<html>
  <head>
    <title>HTMLコンテンツ</title>
  </head>
  <body>
    <h1>大見出し</h1>
    <p>テキスト</p>
    <h2>中見出し</h2>
    <p>テキスト</p>
    <h3>小見出し</h3>
    <p>テキスト</p>
  </body>
</html>
```

# Listの構造

<ul>

<li>...</li> <li>...</li>...

</ul>

<ol>

<li>...</li> <li>...</li>...

</ol>

<dl>

<dt>...</dt> <dd>...</dd>...

</dl>

# Tableの構造(1)

<table>

<caption>表題</caption>

<colgroup>

<col>...</col> <col>...</col>...

</colgroup>

<thead>

<tr><th>...</th> <th>...</th>...</tr>

</thead>

# Tableの構造(2)

```
<tfoot>
```

```
  <tr><td>...</td> <td>...</td>...</tr>
```

```
</tfoot>
```

```
<tbody>
```

```
  <tr><td>...</td> <td>...</td>...</tr>
```

```
</tbody>
```

```
</table>
```

# Bodyの構造(2)

```
<div>
```

```
  <h2>タイトル</h2>
```

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

```
</div>
```

```
<blockquote>
```

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

```
</blockquote>
```

```
<hr>
```

```
<address>...<br>...</address>
```

# CSSの記述が分かりやすい

- 整形されたコンテンツが相手だから・・・
- 目的の要素に対して必要な組み指定をしていけばよい
- 共通指定、固別指定、選択指定など様々な要素選別機能がある
- 複雑な親子関係、階層構造を取らない



# スタイルの指定方法

- デフォルト設定 AH Formatterのデフォルト
- `<link rel="stylesheet" type="text/css" href="css/styles.css" />`
- `<style>`  
    `@import "css/styles.css";`  
    `p { font-family: sans-serif; }`  
`</style>`
- `<p style="font-family: sans-serif;" >...</p>`

# セレクタ

- タイプセレクタ

h1 { font-size: 24pt; }

- 属性セレクタ

h1[class="display"] { color: red; }

## 部分一致の属性セレクタ

~="copyright"

|="en"

^="image"

\$=".html"

\*="hellow"

- IDで分類

h1#chapter1 { font-family: fantasy; }

# 疑似クラス

- 要素:root
- 要素:nth-child()      要素:nth-last-child()
- 要素:nth-of-type()    要素:nth-last-of-type()
- 要素:first-child      要素:last-child
- 要素:first-of-type    要素:last-of-type
- 要素:only-child      要素:only-of-type
- 要素:empty
- 要素:not

# 疑似要素

- 要素::first-line  
p::first-line { text-transform: uppercase; }
- 要素::first-letter  
p::first-letter { font-size: 3em; color: red; }
- 要素::before  
h1::before { content: counter(chapno) “.” }
- 要素::after  
h1::after { content: counter(chapno) “.” }

# 結合子

- 子要素または子孫要素  
親要素 子要素
- 子供要素  
親要素 > 子要素
- 隣接要素  
要素名1 + 要素名2
- 兄弟要素  
要素名1 ~ 要素名2

# 脚注・側注・リストマーカー

- 脚注呼び出し疑似要素  
要素名::footnote-call
- 脚注マーカー疑似要素  
要素名::footnote-marker
- 側注呼び出し疑似要素  
要素名::sidenote-call
- 側注マーカー疑似要素  
要素::sidenote-marker
- リストマーカー疑似要素  
list::marker

# @ルール

- ファイルの取り込み @import url(URL)
  - 対象媒体指定 @media
  - 文字コード指定 @charset
  - ネームスペース @namespace
  - フォント指定 @font-face
  - ページ指定 @page
- @page Name @page:blank
- @page:first @page:left
- @page:right マージンボックス
- @footnote @sidenote

# 関数(1)

- 属性関数 `attr(ident type? [, value]? )`
- 計算関数 `calc(16em + 6pt )`
- カウンタ関数 `counter( par-num, upper-roman)`
- 複合カウンタ関数  
`counters( ident, string, list-style-type )`
- リーダ関数 `leader(dotted)`
- 矩形関数 `rect( top, right, bottom, left)`
- 埋込矩形関数  
`inset-rect( top, right, bottom, left )`



# 関数(2)

- 文字列関数 `string( ident )`
- 対象カウンタ関数 `target-counter()`
- 対象複合カウンタ関数 `target-counters()`
- 対象テキスト関数 `target-text()`
- URL関数 `url( string )`
- 拡張属性関数 `-ah-attr-from()`
- 拡張色指定関数 `-ah-rgb-icc()`

# 関数(3)

- 連載関数 `running()`
- 連載要素関数 `element()`
- 装置依存色指定関数 `device-cmyk()`
- 線形グラデーション関数 `linear-gradient()`
- 円形グラデーション関数 `radial-gradient()`
- 繰り返し線形グラデーション `linear-gradient()`
- 繰り返し円形グラデーション `radial-gradient()`

# 関数(4)

- 変形 (-ah-)transform
- 変形原点 (-ah-)transform-origin
- マトリックス関数 matrix()
- 平行移動関数 translate() X/Y
- 縮尺関数 scale() X/Y
- 回転関数 rotate()
- 傾斜関数 skew() X/Y

# 基本的な組版機能

- ボックス            margin border padding
- 文字設定            font line-height
- 段落設定            行揃え 段組み
- 日本語文字組版      ルビおよび圏点
- 表組            リスト
- 画像配置            背景色
- フロート
- 脚注、側注

# 複雑・多岐な組版機能

- 日本語組版機能の拡張仕様
- 縦組み・ルビ・圏点・多段組み等の機能充実
- フロート機能でボックスの位置決め機能
- トンボ・ツメ・塗足し等印刷固有の指定
- SVG・MathMLをサポート
- 目次と索引 手作りによる補助作業
- PDF出力による汎用性拡充

# AH FormatterのCSS拡張

- 値の拡張 dot-dash, dot-dot-dash, wave
- 関数の拡張 グラデーション関数など
- 単位の拡張 q, vmin, vmax, pvw, pvh, gr
- PDF出力における文書情報
- 脚注、行番号、行継続マーク、改定バー
- オーバーフローの拡張
- CSS3プロパティ
- V6.1拡張

# CSS指南 V2.α

- 可能な限り全てのプロパティを紹介
- 用例を豊富に掲載した
- Formatter V6.1の機能を収録
- 章別にページの振り直し 章番号 – ページ
- 爪の高さ位置を改段状に配置
- 巻末索引を付け、プロパティの参照を容易に
- 多言語組版の事例欠如——ゴメンナサイ

# 爪の作成

```
@page Chapter:left {  
  @left-middle {  
    margin-left: -3mm;  margin-right: 4mm;  
    background-color: #eee;  
    content: string(ChapterNo) "章";  
    font-size: 4mm;  line-height: 1.2;  
    font-family: sans-serif;  
    text-align: center;  
    padding: 1mm 1mm 1mm 4mm;  } }
```



# 階段状ツメの作成

```
@page Chapter:left {  
  @left-middle {  
    margin-top: calc( 0mm  
      + 20mm * ((counter(ChapterNo) - 1) mod 10) );  
    margin-bottom: calc(197mm  
      - 20mm * ((counter(ChapterNo) - 1) mod 10) );  
  }  
}
```

# その他事例紹介

- 章ごとの分冊組版
- レイアウトデザイン見本帳から
- ガリ版刷り創立記念誌の復刻
- . . . .

# これからの課題

- HTMLデータの構造化
- セレクタ・@ルール・関数の機能拡張
  - 指定の輻輳による混乱 —
- 組版機能の多国語対応拡充
- 組版機能の詳細化・特殊化
- 自動化の限界 — 手作業による補助
- 機能の多様化に伴い、学習が困難に
- 特殊記号の拡充

# HTML5の活用

## 構造化のすすめ

- section
- article
- aside
- hgroup
- header
- footer
- nav
- figure
- figcaption

# 将来性

- W3C規格のモジュール化した課題ごとの開発
- 必要な機能から随時取り込み
- W3CのWGでは連日多くの意見交換が
- 電子組版への展望を見据えた開発
- まだまだ先の開発目標

# CSS拡張メタ言語

- SCSS(Sass) Rubyで書かれたスタイルシート言語  
(Syntactically Awesom Style Sheet)
- LESS JavaScriptで書かれたスタイルシート言語
- メリット
  - CSSの効率的なコーディング
  - 統一された記述による可読性向上
- デメリット
  - 環境構築が必要
  - 開発チーム全員の再教育
  - クライアントへのデータ渡しの検討

# 終わりに

- 組版機能は拡張を含めて豊富にある
- カスタマイズ・ローカライズができる
- ウィットに富んだ構成を楽しむ
- データとスタイルの分離にこだわらない
- HTMLデータの作り方に工夫を
- 構造的なCSS構築の工夫を

有難うございました

(有)イー・エイド

藤島 雅宏