

Slide URL

<https://vu5.sfc.keio.ac.jp/slide/>

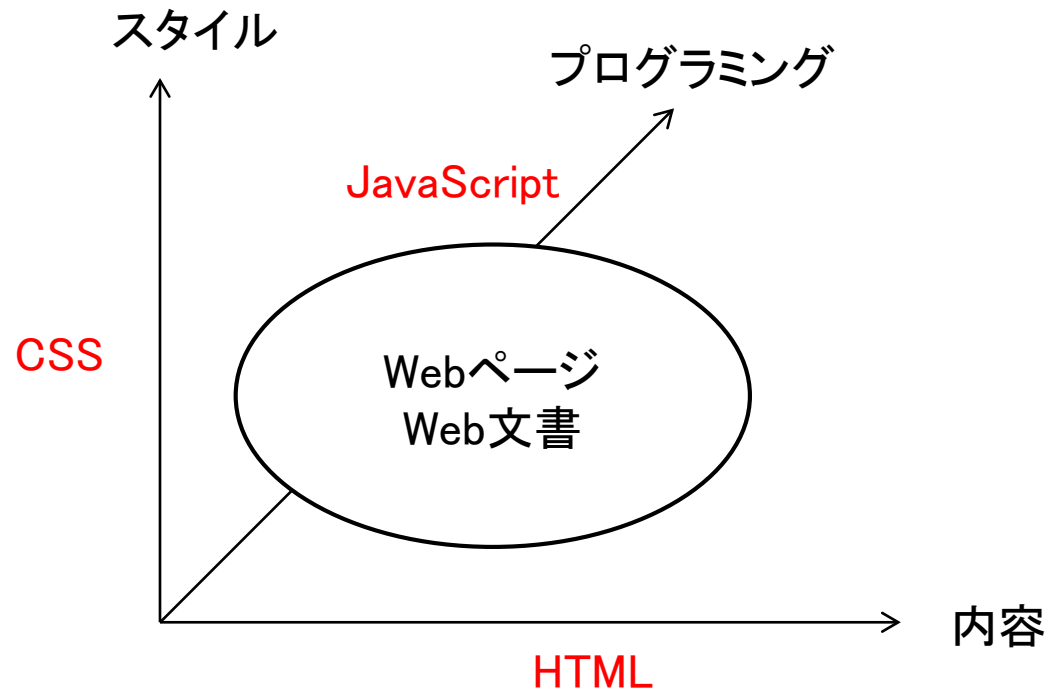
Web情報システム構成法

第5回 CSS入門

萩野 達也 (hagino@sfc.keio.ac.jp)

Webページの構成要素

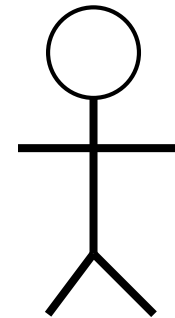
- ▶ 直交技術を組み合わせる
 - ▶ 内容
 - ▶ スタイル(表現方法)
 - ▶ プログラミング



内容と表現方法

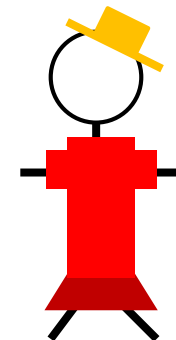
▶ 内容

- ▶ 情報
- ▶ データ
- ▶ Webページのもっとも重要な部分
- ▶ 意味のある部分



▶ 表現方法

- ▶ スタイル
- ▶ 飾り
- ▶ デザイン
- ▶ 内容をいかに見せるか

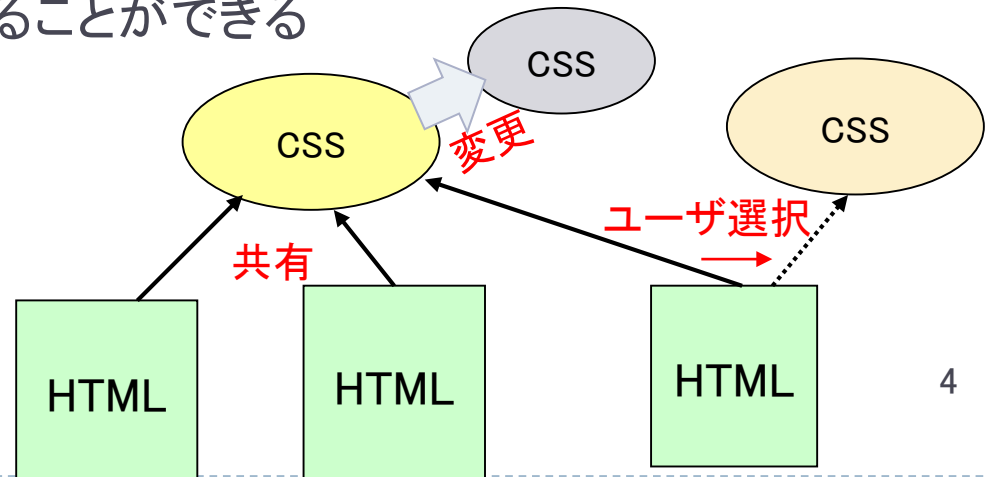


内容と表現の分離

CSS Zen Garden

<http://www.csszengarden.com/>

- ▶ 分離する利点
 - ▶ HTMLの役割がはっきりする
 - ▶ 内容を変えずにスタイルだけを変えることができる
 - ▶ 複数の文書で同じスタイルを共有できる
 - ▶ サイト全体を統一することができる
 - ▶ ユーザがスタイルを変えることができる
 - ▶ アクセシビリティの向上



CSSとは？

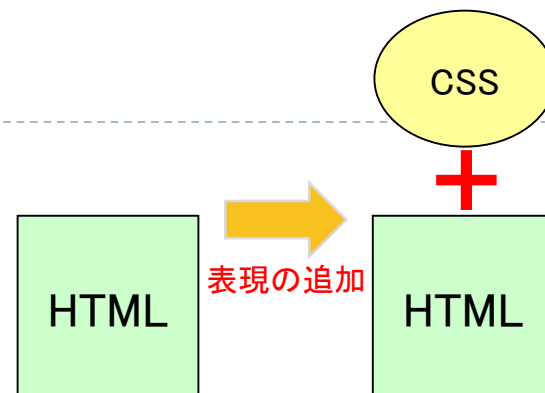
- ▶ 構造化文書に表現を与える
 - ▶ HTML文書に主に用いられる
 - ▶ XMLアプリケーション一般に使うこともできる。

▶ 内容と表現の分離

- ▶ 内容を書きやすい
- ▶ サイトの管理が容易になる
- ▶ アクセシビリティが上がる

▶ CSSのレベル

- ▶ CSS1 (Cascading Style Sheets, level 1) 1996年12月
 - ▶ ボックスモデル
- ▶ CSS2 (Cascading Style Sheets, level 2) 1998年5月
 - ▶ メディアを追加
- ▶ CSS2.1 (Cascading Style Sheets, level 2 revision 1) 2011年6月
 - ▶ 不明瞭部分を改定
- ▶ CSS3 (Cascading Style Sheets, level 3) 2011年6月～
 - ▶ モジュール化
 - ▶ 縦書きも可能
- ▶ CSS4 (Cascading Style Sheets, level 4)
 - ▶ モジュール化



バージョンではない

互換性を維持

どのレベルまでサポートするか

対応モジュールの選択

HTMLでCSSの与え方

▶ HTMLのheadに記述する

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bach's home page</title>
    <style type="text/css">
      h1 { color: blue }
    </style>
  </head>
  <body>
    <h1>Bach's home page</h1>
    <p>Johann Sebastian Bach
      was a prolific composer.</p>
  </body>
</html>
```

▶ style属性を用いる

```
<h1 style="color: blue">Bach's
home page</h1>
<p>Johann Sebastian Bach
  was a prolific composer.</p>
```

▶ CSSファイルをリンクする

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bach's home page</title>
    <link rel="stylesheet" href="bach.css"
      type="text/css" />
  </head>
  <body>
    <h1>Bach's home page</h1>
    <p>Johann Sebastian Bach
      was a prolific composer.</p>
  </body>
</html>
```

CSSファイル

```
h1 {
  color: blue;
  text-align: center;
}
```

CSSの書き方

▶ セレクタにより適応箇所を指定

- ▶ 要素名, ID, パターン

▶ 属性と値を並べて書く

▶ 宣言的

```
body {  
  font-family: "Gill Sans", sans-serif;  
  font-size: 12pt;  
  margin: 3em;  
}
```

セレクタ

プロパティ

値

宣言的 vs 手続き的

```
セレクタ {  
  プロパティ : 値 ;  
  プロパティ : 値 ;  
}
```

▶ 継承とカスケード

- ▶ 子要素は親要素の属性を継承する
- ▶ 複数のスタイルシートをカスケードする

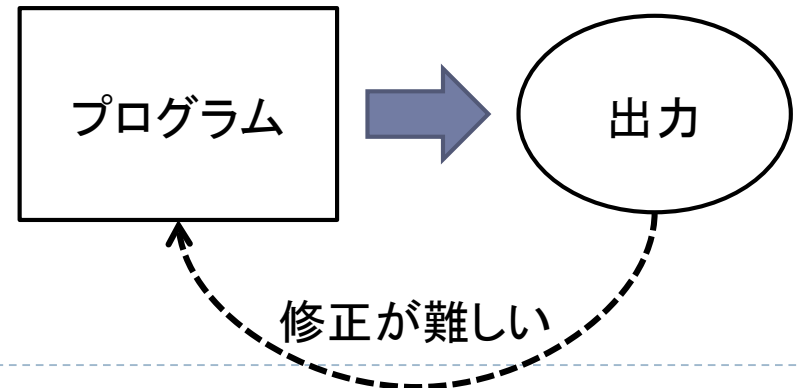
宣言的 vs 手続き的

▶ 宣言的

- ▶ 単純に記述する
- ▶ 規則を並べる
- ▶ 編集が可能
- ▶ 細かく書かないといけないので面倒

▶ 手続き的

- ▶ 手続きを書く
- ▶ プログラム
- ▶ 編集が難しい
- ▶ プリミティブは少なくて済む



セレクタ

▶ スタイルを適用する要素を指定

▶ パターンマッチにより指定

セレクタ	意味
*	すべての要素を指定
E	要素 E を指定
E F	要素 E の子孫である要素 F を指定
E > F	要素 E の子要素 F を指定
E:first-child	最初の子要素 E を指定
E + F	要素 E の直後の要素 F を指定
E, F	要素 E および要素 F を指定

セレクタ	意味
E:lang(en)	自然言語が en であるときの要素 E を指定
E[foo]	属性 foo のある要素 E を指定 (属性の値は問わない)
E[foo="warning"]	属性 foo の値が warning である要素 E を指定
E[foo~="warning"]	属性 foo の値が空白区切りの文字列のリストで、その中に warning が含まれている要素 E を指定
E[lang ="en"]	属性 lang の値がハイフンでつながれた文字列のリストで、それが en で始まる要素 E を指定
E.warning	E[class~="warning"] と同じ
E#myid	id 属性が myid である要素 E を指定

セレクタ	意味
E:link E:visited	:link は、まだたどられていないハイパーリンク要素 E を指定 :visited は、すでにたどられたハイパーリンク要素 E を指定
E:active E:hover E:focus	:active は、要素 E がクリックや押されて発火したとき :hover は、要素 E のマウスカーソルなどが上に来たとき :focus は、要素 E がフォーカスされたとき

セレクタの例

▶ グループニング

- ▶ h1, h2, h3 { font-family: sans-serif; }

▶ 子孫

- ▶ h1 { color: red; }
- ▶ em { color: red; }
- ▶ h1 em { color: blue; }

```
<h1>慶應義塾大学</h1>  
<p>SFCは<em>湘南藤沢キャンパス</em>の事です. </p>  
<h1>慶應義塾大学<em>湘南藤沢キャンパス</em></h1>
```

強調の違い

▶ 子供

- ▶ body > p { line-height: 1.3; }
- ▶ div.cool ol > li { color: red; }

```
<div class="cool">  
  <ul><li>項目</li></ul>  
  <div>  
    <ol>  
      <li>箇条書き</li>  
    </ol>  
  </div>  
</div>
```

▶ 隣接

- ▶ math + p { text-indent: 0; }
- ▶ h1 + h2 { margin-top: -5mm; }

```
<h1>第1章 はじめに</h1>  
<h2>HTMLとは</h2>  
<p>HTMLは...</p>  
<h2>CSSとは</h2>
```

隙間の調整

セレクタの例 (つづき)

▶ 属性

- ▶ `h1[title] { color: blue; }`
- ▶ `span[class="example"] { color: blue; }`
- ▶ `a[rel~="copyright"] { color: blue; }`
- ▶ `*[lang|=“en”] { color : red; }`

▶ class属性

- ▶ `*.pastoral { color: green; }`
 - ▶ `.pastoral { color: green; }`
 - ▶ `*[class~="pastoral"] { color: green; }`
 - ▶ `div.pastoral { color: green; }`
 - ▶ `p.pastoral.marine { color: green; }`
- } 同じ指定

▶ id属性

- ▶ `h1#chapter1 { text-align: center; }`
- ▶ `#section1 { text-align: left; }`

疑似クラスと疑似要素

▶ 疑似クラス

- ▶ `div > p:first-child { text-indent: 0 }`
- ▶ `a:link { color: red } /* unvisited links */`
- ▶ `a:visited { color: blue } /* visited links */`
- ▶ `a:hover { color: yellow } /* user hovers */`
- ▶ `a:active { color: lime } /* active links */`
- ▶ `:lang(fr) > q { quotes: '«' '»' }`

▶ 疑似要素

- ▶ `p:first-line { text-transform: uppercase }`
- ▶ `p:first-letter { font-size: 200%; font-style: italic; font-weight: bold; float: left }`
- ▶ `h1:before { content: counter(chapno, upper-roman) ". " }`
- ▶ `body:end { content: "The End" }`

カスケード

- ▶ 複数のスタイルシート
 - ▶ 著者のスタイルシート
 - ▶ 利用者のスタイルシート
 - ▶ ブラウザ(ユーザエージェント)のスタイルシート

優先度

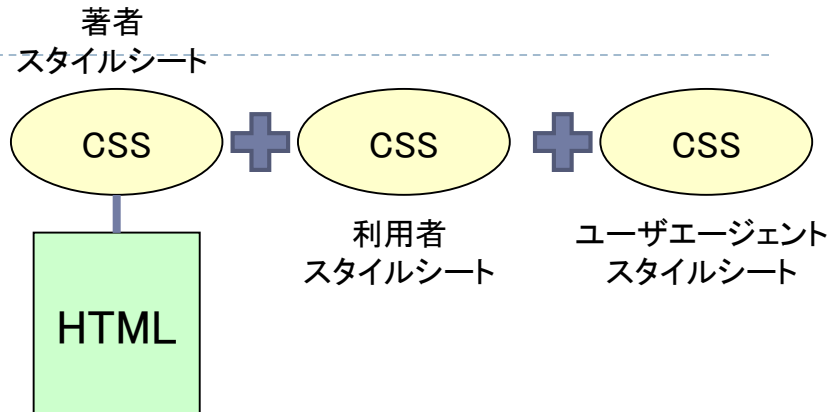
1. 利用者が !important と指定したもの
2. 著者が !important と指定したもの
3. 著者の指定
4. 利用者の指定
5. デフォルト

セレクタの順

1. IDによる指定 (インラインを含む)
2. 属性による指定
3. 要素による指定

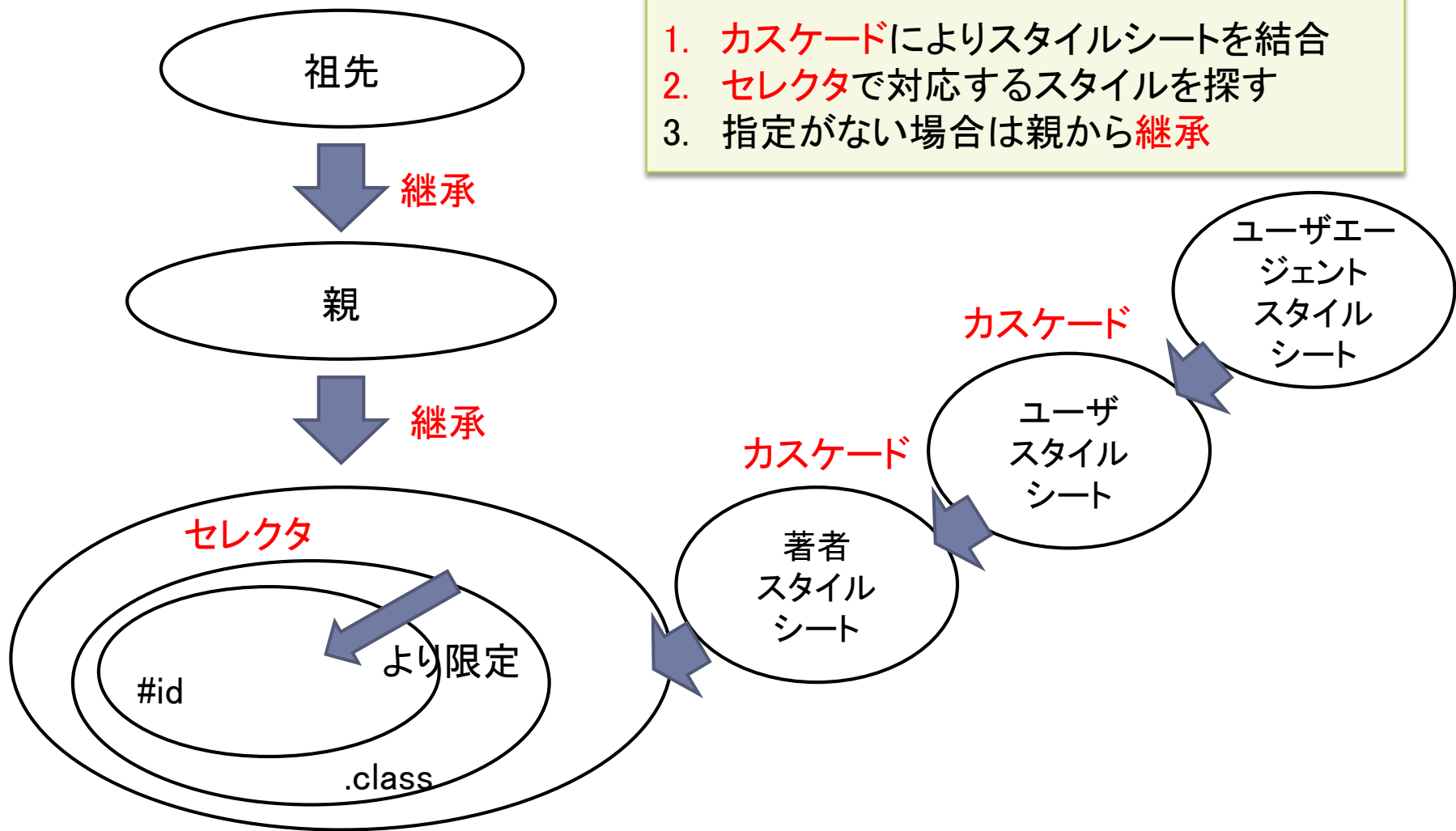
同順の場合

- ▶ より限定されたものを順位が高いと考える
- ▶ @import で取り込まれたものは最初に取り込まれたと
考え優先度は低い



```
/* From the user's style sheet */  
p { text-indent: 1em ! important }  
p { font-style: italic ! important }  
p { font-size: 18pt }  
  
/* From the author's style sheet */  
p { text-indent: 1.5em !important }  
p { font: 12pt sans-serif !important }  
p { font-size: 24pt }
```

セレクタと継承とカスケード



実際の値の計算

1. 指定された値

- ▶ カスケードの値を用いる
- ▶ 親要素から継承された値を用いる
- ▶ デフォルト値を用いる

2. 計算された値

- ▶ 絶対的な値(px, cmなど)はそのまま
- ▶ 相対的な値(em, %など)は絶対的な値を計算する

3. 実際の値

- ▶ 可能な値にまるめる

CSSの色指定

▶ 名前による指定









- ▶ red, green, blue, magenta, yellow, cyan, blackなど
- ▶ 140種類が定義されている

▶ RGB値での指定

- ▶ `rgb(255, 0, 0)`
- ▶ `rgba(255,0,0,0.5)`

▶ 16進数で指定

- ▶ `#f00`
- ▶ `#ff0000`

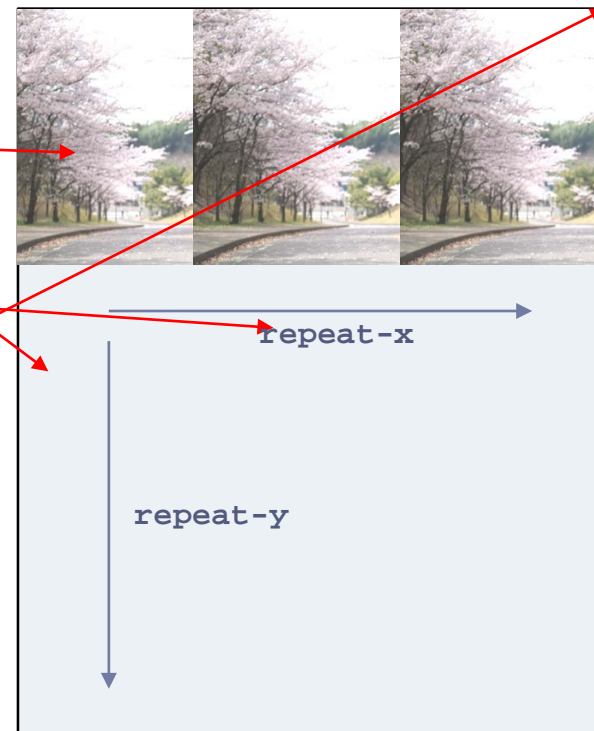
色	名前	RGB	16進数
	red	<code>rgb(255,0,0)</code>	<code>#ff0000</code>
	green	<code>rgb(0,255,0)</code>	<code>#00ff00</code>
	blue	<code>rgb(0,0,255)</code>	<code>#0000ff</code>
	cyan	<code>rgb(0,255,255)</code>	<code>#00ffff</code>
	magenta	<code>rgb(255,0,255)</code>	<code>#ff00ff</code>
	yellow	<code>rgb(255,255,0)</code>	<code>#ffff00</code>
	black	<code>rgb(0,0,0)</code>	<code>#000000</code>
	white	<code>rgb(255,255,255)</code>	<code>#ffffff</code>

利用例

```
p {
  color: blue;
  background-color: rdb(255,255,128);
}
em {
  color: #880022;
}
```


CSS背景

- ▶ **background-color: light-blue;**
 - ▶ 背景の色を指定
- ▶ **background-image: url("back.gif");**
 - ▶ 背景に画像を指定
- ▶ **background-repeat: repeat-x;**
 - ▶ 画像の繰り返しの方向を指定
 - ▶ repeat-x 横方向のみ繰り返し
 - ▶ repeat-y 縦方向のみ繰り返し
 - ▶ no-repeat 繰り返さない
- ▶ **background-position: right top;**
 - ▶ 画像の開始位置を指定
- ▶ **background-attachment: fixed;**
 - ▶ 画像をスクロールしないように固定する
- ▶ **background: #ffffff url("back.png") no-repeat right top;**
 - ▶ 全プロパティをまとめて指定
 - ▶ 色, 画像, 繰り返し, 固定, 位置の順に指定
 - ▶ 省略は可能



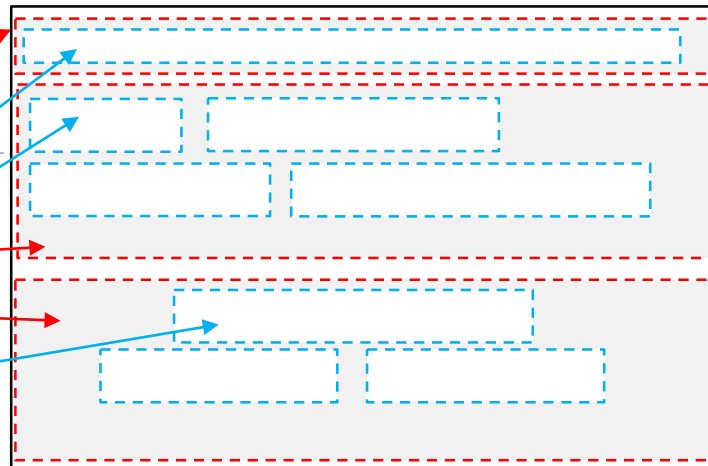
CSSボックスモデル

▶ ブロックボックス

- ▶ 段落用
- ▶ 垂直につながる

▶ インラインボックス

- ▶ 段落内の文書要素用
- ▶ 横につながる
- ▶ 親のブロックボックスをはみ出すと改行

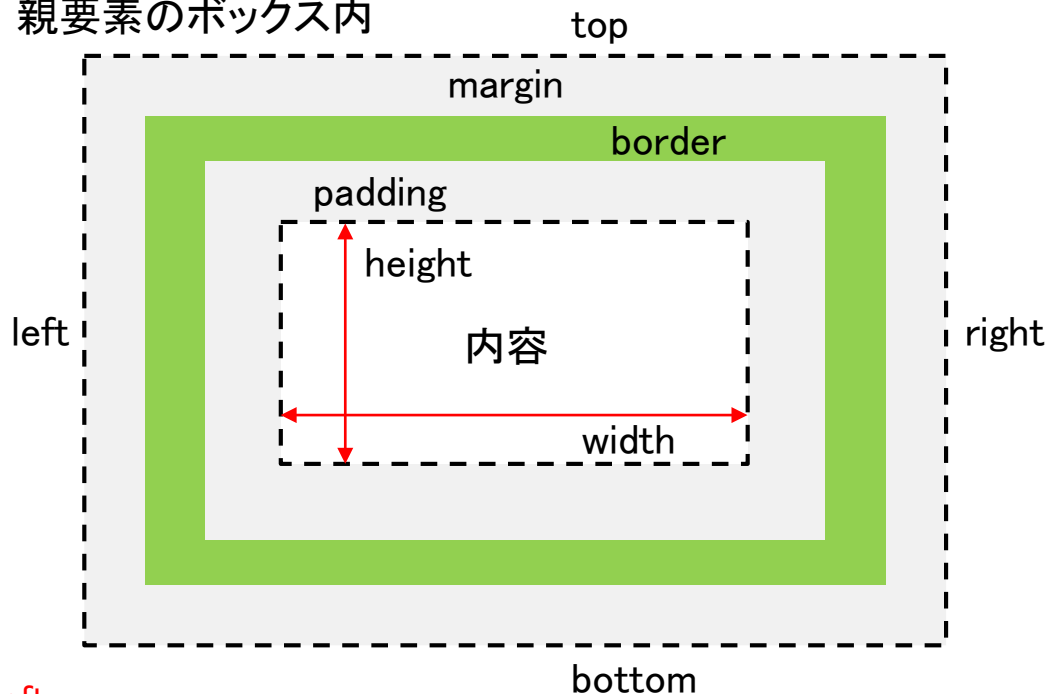


例

```
h1 {  
  margin: 25px;  
  border: 1px solid navy;  
  padding: 10px;  
}  
  
div.main {  
  margin-left: 200px;  
  padding: 10px 5px 15px 2px;  
}
```

top right bottom left

親要素のボックス内



CSSテキスト

- ▶ **color: red;**
 - ▶ テキストの色の指定
- ▶ **text-align: center;**
 - ▶ テキストの位置揃え
 - ▶ center 中央揃え
 - ▶ left 左揃え
 - ▶ right 右揃え
 - ▶ justify 均等割り
- ▶ **text-decoration: underline;**
 - ▶ テキストの装飾
 - ▶ underline 下線
 - ▶ overline 上線
 - ▶ line-through 取り消し線
 - ▶ none なし
- ▶ **text-indent: 5em;**
 - ▶ テキストのインデント
- ▶ **line-height: 0.8;**
 - ▶ 行の高さ
- ▶ **vertical-align: middle;**
 - ▶ 縦位置
 - ▶ middle 中央揃え
 - ▶ top 上揃え
 - ▶ bottom 下揃え
- ▶ **font-family: "Time New Roman", Times, serif;**
 - ▶ フォントの指定
 - ▶ serif 髭あり, san-serif 髭なし
- ▶ **font-style: normal;**
 - ▶ italic イタリック
- ▶ **font-weight: normal;**
 - ▶ bold 太文字
- ▶ **font-size: 12px;**
 - ▶ フォントの大きさ

CSSでの大きさの指定

▶ 絶対指定

- ▶ **cm** センチメートル
- ▶ **mm** ミリメートル
- ▶ **in** インチ (1in = 2.54cm)
- ▶ **px** ピクセル
- ▶ **pt** ポイント (72pt = 1in)
- ▶ **pc** パイカ (1pc = 12pt)

```
body {
  font-size: 12px;
}

h1 {
  font-size: 24pt;
}

h2 {
  font-size: 150%;
}
```

▶ 相対指定

- ▶ **em** フォントの文字 M の幅
- ▶ **ex** フォントの文字 x の高さ
- ▶ **vw** viewportの幅の1%
- ▶ **vh** viewportの高さの1%
- ▶ % 親の要素との相対

```
p {
  text-indent: 4em;
}

div.main {
  margin-top: 1em;
  padding-bottom: 2ex;
}

div.large {
  font-size: 2.5vw;
}
```

課題：CSSでスタイルを付加

- ▶ 自分が生まれ育った町を紹介のトップページにCSSを付加しなさい
 - ▶ CSSは別ファイルとして用意し、リンクすること.
 - ▶ スタイルの内容は自由です.
 - ▶ 今回、説明したものを中心にCSSを書きなさい.
- ▶ HTMLとCSS
 - ▶ 日英でトップページを作っている場合には、とりあえず日本語のものにスタイルを付加しましょう.
 - ▶ CSSを **my.css** したときには、トップページの `mytown.html` (あるいは `mytown.html.ja`) の `head` 部分に以下を挿入してください.

```
<link rel="stylesheet" href="my.css" type="text/css" />
```

- ▶ 提出
 - ▶ <https://vu5.sfc.keio.ac.jp/kadai/>
 - ▶ スタイルを追加したHTMLのURLと作成したCSS自身を宿題登録システムから提出してください。
 - ▶ 締め切り: 5月19日正午

まとめ

- ▶ 原理

- ▶ 宣言的 vs 手続き的

- ▶ スタイルシート

- ▶ 内容と表現の分離

- ▶ CSS

- ▶ セレクタ
- ▶ カスケード
- ▶ 継承