



CSS

- Webページの視覚表現をコントロールするための言語。
- XHTMLは文書の「構造」を指定する言語であり、CSSを利用することにより、Webページの「表現」についてコントロールすることが可能。

CSSの書式

- セレクタ { プロパティ: 値; }
- 主に「h1」や「p」などのXHTML上の要素名で指定する。
- class名やid名で適用対象を絞り込み、さらに細かな設定も可能。
- `body { color: black; }`
 - bodyがセレクタ、colorがプロパティ、blackが値。
- `h1, h2, h3 { color: green; }`
 - 複数のセレクタを指定する場合は、半角カンマ(,)区切りで並べる。
- XHTML文書でCSSを使用する場合、全てを小文字で書くべき。
- テキストは引用符で囲み指定する
 - `font-family: "MS ゴシック";`

単位

- 0の場合、単位は省略できる。
 - 絶対単位
 - pt (ポイント)
 - cm (センチメートル)
 - pc (パイカ)
 - mm (ミリメートル)
 - in (インチ)
 - 相対単位
 - px (ピクセル ディスプレイの解像度の設定により大きさが変化するため)
 - em (font-sizeプロパティを1とする。)
 - ex (小文字「x」の高さを1とする。)

色

- RGB値 (16進数 6桁)
 - #に続けて、16進数 (0～f) で指定。
 - #ff0000
- RGB値 (16進数 3桁)
 - #に続けて、16進数 (0～f) で指定。
 - #f00
- RGB値 (パーセンテージ)
 - 「rgb()」を利用し、各値の0%～100%を半角カンマ区切りで指定。
 - rgb(100%, 0, 0)
- キーワード
 - 標準デフォルトカラー17色の固有カラー名
 - black (#000000)
 - silver (#c0c0c0)
 - gray (#808080)
 - white (ffffff)
 - maroon (#800000)
 - red (#ff0000)
 - purple (#800080)
 - fuchsia (#ff00ff)
 - green (#008000)
 - lime (#00ff00)
 - olive (#808000)
 - yellow (ffff00)
 - navy (#000080)
 - blue (#0000ff)
 - teal (#008080)
 - aqua (#00ffff)
 - orange (#ffa500)
 - ユーザーのGUI環境で使用される28種類のシステムカラー
 - ActiveBorder (アクティブウィンドウの枠線)
 - ActiveCaption (アクティブウィンドウのタイトル)
 - AppWorkspace (MDIの背景)
 - Background (デスクトップの背景 画像ではない!)
 - ButtonFace (ボタンの表面)
 - ButtonHighlight (ボタンのハイライト)
 - ButtonShadow (ボタンの影)

- **ButtonText** (ボタンのテキスト)
- **CaptionText** (タイトルのテキスト)
- **GrayText** (選択不能のテキスト)
- **Highlight** (選択された項目)
- **HilightText** (選択された項目のテキスト)
- **InactiveBorder** (非アクティブウィンドウの枠線)
- **InactiveCaption** (非アクティブウィンドウのタイトル)
- **InactiveCaptionText** (非アクティブウィンドウのテキスト)
- **InfoBackground** (ツールチップの背景)
- **InfoText** (ツールチップのテキスト)
- **Menu** (メニューの背景)
- **MenuText** (メニューのテキスト)
- **Scrollbar** (スクロールバー)
- **ThreeDDarkShadow** (ディスプレイ上の立体表現の中の暗い影)
- **ThreeDFace** (ディスプレイ上の立体表現の中の表面)
- **ThreeDHighlight** (ディスプレイ上の立体表現の中のハイライト)
- **ThreeDLightShadow** (ディスプレイ上の立体表現の中の明るい影)
- **ThreeDShadow** (ディスプレイ上の立体表現の中の影)
- **Window** (ウィンドウの背景)
- **WindowFrame** (ウィンドウの枠)
- **WindowText** (ウィンドウのテキスト)

適用方法と優先順位

- head要素内でスタイルシート言語を指定
 - `<meta http-equiv="Content Style-Type" content="text/css" />`
- head要素内でlink要素により外部スタイルシートを参照
 - `<link rel="stylesheet" type="text/css" href="/css/style.css" />`
- 後から読み込んだスタイルを優先的に適用する。
- 個別性の高いスタイルを優先的に適用する。
- id数 (x100) class数 (x10) 要素数 (x1) 数値が高い方が個別性が高いと見なされ優先される。
 - `* {}` (0) idなし、classなし、要素なし
 - `ul + ol {}` (2) idなし、classなし、要素x2
 - `ul + ol li {}` (3) idなし、classなし、要素x3
 - `li.num {}` (11) idなし、classx1、要素x1
 - `ul + ol li.num {}` (13) idなし、classx1、要素x3
 - `li.num.com {}` (21) idなし、classx2、要素x1
 - `#get {}` (100) idx1、classなし、要素なし

- `p#get {}` (101) `idx1`、`class`なし、要素`x1`
- 要素の属性での指定 (`align`、`valign`など) より、`CSS`が優先される。

セレクトタ

- 要素名セレクトタ
 - 要素のみをセレクトタにする。
 - `body {}`、`p {}`
- ユニバーサルセレクトタ (*)
 - アスタリスクで記述することにより、すべての要素に対して指定
 - `* {}`
- `class`セレクトタ (要素名.`class`名)
 - 複数の要素にスタイルを適用するのに使用。
 - `p.note {} => <p class="note">`
 - 要素名を省略した場合は、`class`名で指定した全ての要素でスタイルを指定。
 - `.note {} => <p class="note"> <span class="note">`
 - 属性で要素を指定する場合。
 - `a[name="note"] => <a name="note">`
 - 複数指定する場合は以下のように指定
 - `p.note {} p.emph {} => <p class="note emph">`
 - `p.note {} p.emph {} = p.note.emph {} = p[class~="note"] [class~="emph"]`
 - `p.note {} = p[class~="note"]`
- `id`セレクトタ (要素名`#id`名)
 - 文書内で唯一の要素にスタイルを適用するのに使用。
 - 同じ`id`名を絶対に使ってはいけない。タイトルロゴ・グローバルナビゲーション・フッター情報など文書で一度しか出現しえないものには、`id`セレクトタでスタイルを適用する。
 - `div#note {} => <div id="note">`
- `class`名と`id`名で利用できる文字
 - (a~z)で開始し、アルファベット、数字、ハイフン、アンダーハイフンのみ。
- 属性セレクトタ
 - `h3[id] {} => <h3 id="r508">`
 - `h3[id="r508"] {} => <h3 id="r508">`
 - `p[class~="note"] {} => <p class="r508 wcag note">`
- セレクトタ セレクトタ
 - 子要素をターゲットにスタイルを適用する。
 - `div.note p {} => <div class="note"><p> (div要素のnoteクラス内のp要素)`
 - `div.note p span {} => <div class="note"><p><span> (div要素のnoteクラス内のp要素内のspan要素)`
 - `div * p {} => <div><div><p> (div要素内の別要素内のp要素)`

- 子セレクト 親 > 子
 - 親要素の直接の子要素をターゲットに適用。
 - `blockquote.wcag > p {} => <blockquote class="wcag"><p> (blockquote要素の wcag クラス内のp要素)`
 - `blockquote.wcag > p > abbr {} => <blockquote class="wcag"><p><abbr> (blockquote要素の wcag クラス内のp要素内のabbr要素)`
- プラス記号区切り
 - 同一の親要素をもつ並列関係にある要素への適用。
 - `h1 + h2 {} => <h1></h1><h2></h2>`
 - +の前の要素のすぐ後に+の後の要素が現れないと適用されない。 `<h1></h1><p></p><h2></h2>`
 - +の後の要素のみに適用。
- グループ化
 - 半角カンマ区切りでグループ化し、複数の要素に同一のスタイルを一度に適用。
 - `th, td {}`
 - `div#title dl, div#title p {}`

疑似クラス

- `:first-child`
 - 親要素の範囲内で最初に登場する子要素のみにスタイルを適用。
 - `p:first-child em {} =>`

```

<div>
  <p><em></em></p>
  <p><em></em></p>
</div>

```
- リンク疑似クラス
 - `:link`
 - 未訪問リンク `a:link {}`
 - `:visited`
 - 既訪問リンク `a:visited {}`
- ダイナミック疑似クラス
 - `:hover`
 - ロールオーバー `a:hover {}`
 - `:active`
 - アクティブの時(クリックした時など) `a:active {}`
 - `:focus`
 - フォーカス状態の時 `a:focus {}`
 - 複数指定する事も可能
 - `a:focus:hover {}`
- リンクとダイナミックの指定順序
 - `:link => :visited => :hover => :active`

- **:first-line**
 - ある要素の一行目だけ適用。
 - `p:first-line {} =>`

```
<p>
ああああああああああ
いいいいいいいいいい

</p>
```
 - どこが一行目になるのかは、ブラウザのウィンドウ幅やフォントサイズによって変わる。
 - ブロック要素のみしか適用できない。
 - 以下のプロパティ限定
 - color
 - word-spacing
 - 背景関連
 - text-trasform
 - フォント関連
 - clear
 - line-height
 - vertical-align
 - text-decoration
 - text-shadow
 - letter-spacing
- **:first-letter**
 - ある要素の一文字目のみ
 - `p:first-letter {}`

```
<p>あいうえお</p>
```
 - ブロック要素のみしか適用できない。
 - 以下のプロパティ限定
 - color
 - 背景関連
 - text-trasform
 - フォント関連
 - float
 - clear
 - line-height
 - vertical-align
 - text-decoration
 - text-shadow
- **:before、:after**
 - contentプロパティと合わせて、ある要素に含まれる内容の前後に何らかの内容を生成する。
 - 適用前 => `<p class="note">かきくけこ</p>`

```
p.note:before {content: "あいうえお";}
p.note:after {content: "さしすせそ";}

<p class="note">あいうえおかきくけこさしすせそ </p>
```