CSS の歴史

Cascading Style Sheet (CSS) はスイス・ジュネーブの欧州原子核研究機構 (CERN) が進めていたプロジェクトの中でノルウェー出身の (コンピューター科学者 : オペラ・ソフトウェアの最高責任者) ホーコン・ウィウム・リー氏によって, WWW が誕生した直後 (1994 年) に発表されています.

世界初のブラウザ Mosaic (HTML1.0) が 1993 年に開発されましたが, このときの設計段階においてすでに数種類のスタイル情報がブラウザの中に組み込まれていました. しかし HTML のソースの中で私たちがスタイルを定義できるようになったのは 1995 年 (HTML2.0) からです. とは言うものの, この時期にはまだ CSS は草案であったためにブラウザに組み込まれていたことは発表されておらず, 一部の関係者以外は CSS の存在を知りませんでした. その後 , CSS は W3C の中で結成された専門グループ (CSS Working Group) が議論しながら設計し, 1996 年になってやっと正式な発表となりました. Cascading Style Sheet が発表されて Netscape と IE が CSS の機能を取り入れましたが, 二つのブラウザは一部の機能だけをバラバラに組み込んでしまったため, 世間で CSS はまったく普及しませんでした. そして , ユーザーたちの間で「ブラウザがバージョン・アップされるときに将来普及されるであるはずの機能が隠れて入っている」という噂が広まり, CSS に関する情報は早い段階で混乱してしまいました. 実際のところ CSS の機能はバージョン・アップで取り消されたり, 内容が変更されてしまっている部分が多く, 追加された機能がどのモジュールに分類されているのかがわかりにくい状態が続いていました.

Cascading Style Sheets (カスケーディング・スタイル・シート)
CSS（読み方 : シー・エス・エス）
ホームページの見た目を整えるための技術

- **ページの横幅はどれくらいに**
- **文字の色は何色に / 書体は / 文字サイズは / 文字にドロップ・シャドー (スライディングの距離) は**
- **背景の色 / 画像を指定する**
- **文字や画像にカーソルを近づけたときの色の変化 / 動きを作る**
- **カーソルの形を選択する**
- **画面サイズに合わせて配置する画像の大きさは**
- **表の縦と横の大きさを調節**

などをページ内に指定することができる.
すべての基準はグラフィック・デザインとの比較
CSS を使って Web デザイナーがコンテンツをモニタ内でどこまで読みやすく
(見た目を美しく / 操作しやすく) できるかを追求する.
現在では , Web デザイナーと Web エンジニア (プログラマー) はそれぞれ別の仕事を担当している.

HTML だけで記述した状態

HTML に CSS を追加した状態

これまでの CSS のバージョンは Level-1(1996 年) / Level-2(1998 年) / Level-2.1(2011 年) / CSS3 / CSS4(2013 年) / Level-2 Rvision2(2016 年) / CSS5(2020 年) となっており, CSS1 とは「CSS のバージョン 1」ではなく「Cascading Style Sheets, level 1」の略称であると発表されています. CSS3 で追加された機能がどのモジュールに分類されているのか, いくつか例を挙げてみます.

- アニメーション : CSS Animations Level 1
- 変形 (transform) : CSS Transforms Module Level 1
- フレックスボックス : CSS Flexible Box Layout Module Level 1
- 背景グラデーション : CSS Image Values and Replaced Content Module Level 3
- 透明度 : CSS Color Module Level 3

CSS が本格的に普及したのは XHTML の時代と言えるでしょう.

2013 年あたりから Media Query(メディア･クエリー) を利用するエンジニアが増え, 大量のデータを一度に抱えた CSS を取り扱うコーディング・スタイルが主流となってきました. さらに BootStrap / Pure / UIkit / Semantic UI / Foundation / Tailwindcss などのフレームワークも登場し,「CSS の書き方」そのものが変化しています. 2006 年にプログラミング言語 (CSS の拡張言語) を使ってひとまず独自のデータをコーディングし, 後で必要な部分だけを抜き出してシンプルな CSS にコンパイルする言語 Sass(Syntactically Awesome Style Sheets : サース) が開発されました (もしも WEB エンジニアがプログラマーから入ってきた人なら Sass は使いやすい). 効率よく CSS を書けることがメリットですが, デメリットはコンパイルする必要があることです (ブラウザは直接 Sass で作成された文章を読み取ることはできない). 最近 Sass の登場により, 通常の書き方で記述した CSS を「プレーン CSS」, Sass で作成した CSS を「コンパイル CSS」と分類するようになりました. Sass が登場した後に (Sass をコンパクトにまとめた) LESS という言語も生まれています. CSS の記述方式は日々発展しています.

Suss で記述したコード

```
$blue: #3bbfce
$margin: 16px

.content-navigation
  border-color: $blue
  color: darken($blue, 10%)

.border
  padding: $margin/2
  margin:  $margin/2
  border-color: $blue
```

コンパイルして CSS にした内容

```
.content-navigation {
  border-color: #3bbfce;
  color: #2b9eab;
}

.border {
  padding: 8px;
  margin: 8px;
  border-color: #3bbfce;
}
```

CSS の設置方法

■External CSS の場合
◯◯◯.css というファイルを用意し, head タグ内に <link> タグで CSS の指定 / 参照先 / 言語をを記述する (閉じタグは書かない).
例えばこのように <link rel="stylesheet"href="◯◯◯.css"type="text/css">

■Internal CSS の場合
head タグ内に <style> 〜 </style> を作り, この間に CSS 情報を記述する.

■Inline CSS の場合
HTML のタグに style という attribute を追加し, 直接 CSS を記述する.
CSS は最後に読み込んだ情報が優先されるため Inline CSS が最優先の指定として適用される.

CSS を作る手順

最初に Selector (セレクタ) を記述し, { (レフト・ブレース) の後に Property (プロパティ) と value(ヴァリュー) を指定して ; で区切る (最後に } (セミ・コロンとライト・ブレース) で閉じる).
Selector は ボディ (<body>) / ヘッド (<h1> 〜 <h6>) / パラグラフ (<p>) / ディヴィジョン ( ) などの HTML のタグ名で指定する.
さらに class や id を使って固有名の Selector を記述しながら適用する対象を絞り込み, もっと細かな設定を指定する.
例えば body{color:black;} と指定すれば body がセレクタ, color がプロパティ, blackがヴァリューとなる.

複数のセレクタを一度に指定することも可能ですが, その場合はそれぞれを 半角カンマ (,) で区切ること.
例えば h1, h2, h3{color:red;} のように

文字やボックスの大きさ / 距離などは次の単位で指定することができる (0 という値の場合のみ , 単位を省略することができる
絶対単位として
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)

相対単位として
px(ピクセル)
em(font-size プロパティを 1 としている)
ex(小文字の x の高さを 1 としている)

色に関する指定
RGB 値 (16 進数 3 桁, または 6 桁) は # に続けて 0 〜 f(Hx) で指定する.
例えば #f00 / #ff0000 のように

RGB 値を %(パーセンテージで指定することもできる
rgb() を利用して各値の 0% 〜 100% を半角カンマ区切りで指定する.
rgb(100%, 0, 0) のように

標準デフォルト固有カラー名も使用できる
green (#008000)
red (#ff0000)
blue (#0000ff)
yellow (#ffff00)
lime (#00ff00)
silver (#c0c0c0)
gray (#808080)
white (#ffffff)
aqua (#00ffff)
olive (#808000)
maroon (#800000)
purple (#800080)
orange (#ffa500)
のように

その他の項目
▶文字間の隙間を指定
quotes 要素の前後に挿入される引用符を指定
text-align テキストの水平方向への配置を指定
text-decoration テキストの装飾を指定
text-indent インデント（字下げ）の指定
text-transform 英数字の大文字, 小文字を変換して表示するかの指定
unicode-bidi　Unicode による文字の表記方向設定をどう扱うかの指定
vertical-align 行中のテキストや画像などの内容の, 上下方向の揃え位置を指定
white-space ソース中の改行やスペース, タブの扱いを指定
word-spacing 単語と単語の隙間を指定

▶リスト
list-style-image リストのマーカーに使う画像ファイルを指定
list-style-position リストのマーカー位置を指定
list-style リストのスタイルを一括で指定
list-style-type リストのスタイルタイプを指定
counter-increment 要素の出現時に通し番号を増やす量の指定
counter-reset 要素の出現時に通し番号をリセット

▶表
empty-cells 内容が空のテーブルセルで枠線や背景を表示するかしないかを指定
caption-side キャプションの表示位置を指定
table-layout テーブル (表) のレイアウト方式を指定

▶幅・高さ
height 要素の高さを指定
max-height 高さの最大値を指定
max-width 幅の最大値を指定
min-height 高さの最小値
min-width 幅の最小値を指定
width 領域の幅を指定
余白 margin
上下左右のマージンを一括で指定 margin-right
右のマージンを指定 margin-left
左のマージンを指定 margin-bottom
下のマージンを指定 padding
上下左右のパディングをまとめて指定
padding-bottom 下のパディングを個別に指定
padding-left 左のパディングを個別に指定
padding-right 右のパディングを個別に指定
padding-top 上のパディングを個別に指定

▶枠線
border-width 上下左右のボーダーラインの太さをまとめて指定
border-top-width 上のボーダーラインの太さを指定
border-top-style 上のボーダーラインの種類を指定
border-top-color 上のボーダーラインの色を指定
border-top 上のボーダーラインの太さ, 色などを一括して指定
border-style 上下左右のボーダーラインの種類をまとめて指定
border-spacing 隣り合ったテーブルセルにおいて, 境界線の間隔を指定
border-right-width 右のボーダーラインの太さを指定
border-right-style 右のボーダーラインの種類を指定
border-right-color 右のボーダーラインの色を指定
border-right 右のボーダーラインの太さ, 色などを一括して指定
border-left-width 左のボーダーラインの太さを指定
border-left-style 左のボーダーラインの種類を指定
border-left-color 左のボーダーラインの色を指定
border-left 左のボーダーラインの太さ, 色などを一括して指定
border-color 上下左右のボーダーラインの色をまとめて指定
border-collapse 隣り合ったテーブルセルの間隔を指定
border-bottom-width 下のボーダーラインの太さを指定
border-bottom-style 下のボーダーラインの種類を指定
border-bottom-color 下のボーダーラインの色を指定

▶表示と位置
right 右端からの距離を指定 (基準点は position で指定)
left 左端からの距離を指定 (基準点は position で指定)
float フロート (後続する内容の回り込み) を指定

display 要素の表示形式を指定
direction 文章の表示方向を指定
clip 画像などの要素について切り抜き領域を指定
clear　float で設定した回り込み設定を解除
bottom 下端からの距離を指定 (基準点は position で指定)
overflow ボックス内に収まり切らない内容をどのように処理するかを指定
top 上端からの距離を指定 (基準点は position で指定)
visibility 表示 / 非表示の切り替え
z-index ボックスの重なり順序を指定

▶背景
background-repeat 背景画像の繰り返し方法を指定
background-position 背景画像の表示開始位置を指定
background-image 背景画像を指定
background-color 背景の色を指定
background-attachment スクロール時に背景画像を固定表示するかスクロールさせるかを指定
background 背景の指定

▶フォント
font-weight フォントの太さを指定
font-variant スモールキャップを使う (アルファベットの小文字を大文字と同じ形で小さく表示するか) の指定
font-style フォントスタイルを指定
font-size フォントサイズを指定
font-family フォントファミリーを指定
font フォントのスタイルや太さ, サイズなどをまとめて指定
color テキストの色を指定

▶コンテンツの生成
content 要素の前後に文字や画像などを挿入

▶ユーザインターフェイス
cursor マウスカーソルの形を指定
outline 輪郭線のスタイルや色, 太さをまとめて指定
outline-color 輪郭線の色を指定
outline-style 輪郭線のスタイルを指定
outline-width 輪郭線の太さを指定

▶印刷
page-break-before 印刷時の改ページ位置を制御
orphans 印刷時に要素が途中で改ページされる際, 前ページに残る最低行数
page-break-after 印刷時の改ページ位置を制御
page-break-inside 印刷時の改ページ禁止を制御
widows 印刷時に要素が途中で改ページされる際の, 次ページでの最低行数