CSSのキホン

前の資料で、HTMLを使ってWebページの骨組みを作る方法を学びましたね。HTMLだけだと、文字や画像が並んでいるだけで、ちょっと寂しい見た目になってしまいます。

そこで登場するのが **CSS（Cascading Style Sheets）** です！ CSSは、HTMLで作った骨組みに色を塗ったり、文字の大きさを変えたり、要素の配置を調整したりして、Webページの **見た目をデザイン** するための言語です。

HTMLが「家の骨組み」だとしたら、CSSは「壁紙の色」や「家具の配置」を決める役割、と考えると分かりやすいかもしれません。

CSSを学ぶと、こんなことができるようになります。

- 文字の色や大きさを自由に変える
- 背景に色や画像を設定する
- 要素の周りに余白をつけたり、枠線を引いたりする
- 要素を好きな場所に配置する（レイアウト）
- Webサイト全体でデザインを統一する

1. CSSの書き方：誰に、何を、どうする？

CSSは、**「どのHTML要素に」「何を」「どのように変えるか」** を指定するルールを書いていきます。基本的な形は次のようになります。

```
セレクタ {
  プロパティ: 値;
}
```

- **セレクタ（Selector）**: 「どのHTML要素に」スタイルを適用するかを指定します。例えば、「すべての `<p>` タグに」とか、「`title` という名前を付けた要素に」といった具合です。
- **プロパティ（Property）**: 「何を」変えたいかを指定します。例えば、「文字の色（`color`）」や「背景色（`background-color`）」、「文字の大きさ（`font-size`）」など、たくさんの種類があります。
- **値（Value）**: 「どのように」変えるかを指定します。例えば、`color` プロパティなら `red`（赤色）や `#0000ff`（青色）、`font-size` プロパティなら `20px`（20ピクセル）のように指定します。
- **`{ }`（波括弧）**: この中に、適用したいスタイル（プロパティと値のセット）を書きます。
- **`:`（コロン）**: プロパティと値の間に入れます。
- **`;`（セミコロン）**: プロパティと値のセットの終わりに入れます。複数のスタイルを指定する場合に必要です。

例：すべての `<p>` タグの文字色を赤にする

```
p {
  color: red;
}
```

例：すべての `<h1>` タグの文字サイズを 30px に、文字を中央揃えにする

```
h1 {
  font-size: 30px;
  text-align: center; /* 中央揃え */
}
```

このように、`{ }` の中に複数の「プロパティ: 値;」を書くことで、一つのセレクタに対して複数のスタイルを同時に指定できます。

2. CSSをHTMLに適用する3つの方法

書いたCSSルールをHTMLに適用するには、主に3つの方法があります。

方法1：外部スタイルシート（推奨！）

一番おすすめの方法です。CSSのルールだけを別のファイル（拡張子 `.css`、例えば style.css）に書き、HTMLファイルからそのCSSファイルを読み込みます。

style.css（CSSファイル）

```
/* style.css */
h1 {
  color: blue;
}

p {
  font-size: 16px;
}
```

index.html（HTMLファイル）

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>CSSの練習</title>
  <!-- <link>タグでCSSファイルを読み込む -->
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>これは青色になります</h1>
  <p>この文字サイズは16pxになります。</p>
</body>
</html>
```

`<head>` タグの中に `<link rel="stylesheet" href="style.css">` と書くことで、`style.css` ファイルに書かれたCSSルールがHTML全体に適用されます。

メリット:

- HTMLとCSSを分離できるので、コードが見やすくなる。
- 複数のHTMLファイルで同じCSSファイルを使い回せるので、デザインの統一や修正が楽になる。

方法2：内部スタイルシート

HTMLファイルの `<head>` タグの中に `<style>` タグを書き、その中に直接CSSルールを書く方法です。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>CSSの練習</title>
  <style>
    /* <style>タグの中にCSSを書く */
    h1 {
      color: green;
    }
    p {
      font-size: 14px;
    }
  </style>
</head>
<body>
  <h1>これは緑色になります</h1>
  <p>この文字サイズは14pxになります。</p>
</body>
</html>
```

メリット:

- HTMLファイルだけで完結するので手軽。

デメリット:

- 他のHTMLファイルで使い回せない。
- HTMLファイルが長くなりがち。

方法3：インラインスタイル

HTML要素の開始タグの中に `style` 属性を直接書き、そこにCSSルールを書く方法です。そのタグにだけスタイルが適用されます。

```
<!DOCTYPE html>
<html lang="ja">
<head>
```

```
  <meta charset="UTF-8">
  <title>CSSの練習</title>
</head>
<body>
  <!-- タグに直接style属性を書く -->
  <h1 style="color: orange;">この見出しだけオレンジ色</h1>
  <p style="font-size: 18px; background-color: lightgray;">この段落だけ文字サイズ18px、背景グレー</p>
  <p>ここはスタイルが適用されない</p>
</body>
</html>
```

メリット:

- 特定の要素だけにピンポイントでスタイルを適用できる。

デメリット:

- HTMLの中にデザインの情報が混ざり、コードが読みにくくなる。
- 修正が大変になることが多い。
- 基本的にはあまり使わない方が良い方法です。

まずは、外部スタイルシートを使う方法に慣れていきましょう！

3. よく使うCSSプロパティを使ってみよう

CSSにはたくさんのプロパティがありますが、まずはよく使うものをいくつか紹介します。

文字の色を変える: `color`

```
p {
  color: blue; /* 青色 */
}
h2 {
  color: #ff0000; /* 赤色（16進数カラーコード）*/
}
```

背景色を変える: `background-color`

```
body {
  background-color: lightyellow; /* 薄い黄色 */
}
h1 {
  background-color: #333; /* 暗い灰色 */
  color: white; /* 背景が暗いので文字は白に */
}
```

文字の大きさを変える: `font-size`

```
p {
  font-size: 16px; /* 16ピクセル */
}
.small-text { /* "small-text" というクラス名がついた要素 */
  font-size: 12px;
}
```

単位には `px`（ピクセル）の他に、`em`, `rem`, `%` などもあります。 `em` や `rem` は相対的なサイズ指定で、親要素やルート要素に対する割合で指定します。`%` も同様です。 `px` は絶対的なサイズ指定で、画面上のピクセル数を指定します。（さらなる細かい説明は、自分でググってみてください！）

文字の太さを変える: `font-weight`

```
strong {
  font-weight: bold; /* 太字（デフォルト）*/
}
.normal-weight {
  font-weight: normal; /* 通常の太さ */
}
h1 {
  font-weight: 900; /* 数字でも指定可能（100〜900）*/
}
```

テキストの配置を変える: `text-align`

```
h1 {
  text-align: center; /* 中央揃え */
}
p {
  text-align: left; /* 左揃え（デフォルト）*/
}
.right {
  text-align: right; /* 右揃え */
}
```

外側の余白を設定する: `margin` 要素とその外側にある他の要素との間のスペースです。

```
p {
  margin: 20px; /* 上下左右に20pxの余白 */
}
h2 {
  margin-top: 30px; /* 上だけに30pxの余白 */
  margin-bottom: 10px; /* 下だけに10pxの余白 */
}
img {
  margin: 10px 20px; /* 上下10px, 左右20px */
}
```

内側の余白を設定する: `padding` 要素の枠線（border）とその中身（content）との間のスペースです。

```
div {
  background-color: lightblue;
  padding: 15px; /* 内側上下左右に15pxの余白 */
}
button {
  padding: 10px 20px; /* 内側上下10px, 左右20px */
}
```

枠線を引く： `border` 太さ、種類（実線、点線など）、色をまとめて指定できます。

```
img {
  border: 1px solid black; /* 1pxの黒い実線 */
}
div {
  border: 3px dashed red; /* 3pxの赤い破線 */
  padding: 10px; /* 枠線と中身の間に余白を入れると見やすい */
}

button {
  border: none; /* 枠線なし */
  background-color: lightgray; /* 背景色を指定 */
}
```

角を丸くする： `border-radius`

画像やボタンの角を丸くすることができます。

```
img {
  border-radius: 10px; /* 角を10px丸くする */
}
button {
  border-radius: 5px; /* 角を5px丸くする */
}
```

幅と高さを指定する： `width`, `height`

```
img {
  width: 200px; /* 幅を200pxに */
  height: auto; /* 高さは自動調整（縦横比を保つ）*/
}
div {
  width: 50%; /* 親要素の幅の50% */
  height: 100px; /* 高さを100pxに */
  background-color: lightgreen;
}
```

影をつける： `box-shadow`

要素に影をつけて立体感を出すことができます。

```
div {
  box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.7); /* 右下に影をつける */
}
```

4. セレクタの種類：特定の要素にスタイルを適用する

「どの要素に」スタイルを適用するかを指定するセレクタにも、いくつか種類があります。

要素セレクタ（Element Selector）

HTMLタグ名をそのまま使います。そのタグすべてにスタイルが適用されます。

```
/* すべてのpタグ */
p {
  color: gray;
}
/* すべてのh2タグ */
h2 {
  border-bottom: 1px solid lightgray; /* 下線 */
}
```

クラスセレクタ（Class Selector）

HTML要素に `class` 属性で名前を付け、その名前を使ってスタイルを適用します。**同じクラス名を複数の要素に付けることができます。** CSSでは、クラス名の前にドット（`.`）を付けます。デザインの部品としてよく使われます。

HTML:

```
<p class="highlight">この段落は強調されます。</p>
<div>
  他のテキスト <span class="highlight">ここも強調</span> します。
</div>
<p>ここは強調されません。</p>
```

CSS:

```
/* "highlight" というクラス名がついた要素すべて */
.highlight {
  background-color: yellow;
  font-weight: bold;
}
```

IDセレクタ（ID Selector）

HTML要素に `id` 属性で固有の名前を付け、その名前を使ってスタイルを適用します。**`id` 属性の値は、一つのHTMLファイル内でユニーク（一意）でなければなりません。** つまり、同じID名を複数の要素に付けることはできません。CSSでは、ID名の前にシャープ（`#`）を付けます。ページの特定の部分（ヘッダー、フッターなど）を指定するのによく使われます。

HTML:

```
<div id="header">
  <h1>サイトタイトル</h1>
</div>
<div id="main-contents">
  <p>メインコンテンツです。</p>
</div>
<p id="footer">コピーライト</p> <!-- IDは一つだけ -->
```

CSS:

```
/* "header" というIDがついた要素 */
#header {
  background-color: skyblue;
  padding: 20px;
}
/* "main-contents" というIDがついた要素 */
#main-contents {
  margin: 20px 0;
}
/* "footer" というIDがついた要素 */
#footer {
  font-size: 12px;
  text-align: center;
  color: gray;
}
```

クラスとIDの使い分け:

- **クラス（`.`）**: 同じスタイルを複数の要素に適用したい場合。デザインの部品。
- **ID（`#`）**: ページ内の特定の、ただ一つの要素にスタイルを適用したい場合。ページの主要な区画。

5. まとめ：CSSでWebページを彩ろう

お疲れ様でした！これでCSSの基本的な書き方と、HTMLへの適用方法、よく使うプロパティやセレクタについて学ぶことができました。

- CSSはWebページの **見た目（デザイン）** を担当する言語です。
- `セレクタ { プロパティ: 値; }` の形でスタイルを指定します。
- CSSを適用するには、**外部スタイルシート（`<link>` タグ）**が推奨されます。

- `color`, `background-color`, `font-size`, `margin`, `padding`, `border` など、様々なプロパティで見た目を調整できます。
- **クラスセレクタ（`.`）**や **IDセレクタ（`#`）**を使うことで、特定の要素にスタイルを適用できます。

HTMLで骨組みを作り、CSSでデザインを整える。この二つを組み合わせることで、表現力豊かなWebページを作成することができます。最初は覚えることが多く感じるかもしれませんが、実際にコードを書きながら、少しずつ慣れていきましょう！