

## 1. スタイルシートとは (教科書 第 3 章 P.47)

### 1.1. スタイルシート(CSS)とは何か

CSS は、Cascading Style Sheets の略で「スタイルシート」あるいは単に「スタイル」とも呼ばれます。

CSS はウェブページの見栄えを設定するための言語であり、現在、多くのサイトで使用されています。

CSS は HTML と組み合わせて使用されます。HTML ではページ内の各要素の意味や構造を設定しますが、CSS ではページを装飾するための情報を設定するよう、住み分けがなされています。

CSS では例えば、ウェブページが画面に表示される際の 色・サイズ・レイアウト といった表示の詳細や、プリンタなどで印刷される際のレイアウト、音声で読み上げられる際の再生情報などを指定できます。

### 1.2. なぜ CSS を使うのか

HTML にも見栄えを設定する項目があります。例えば<font>タグや background 属性など、数多くの設定が存在します。また、表(table タグ)でページをレイアウトすれば、HTML だけで文章の並べ方(レイアウト)を設定することもできます。

しかしウェブページの見栄えを HTML で設定するのは、HTML 本来の目的と異なる使い方です。HTML は本来、情報構造を定義するためのしくみとして開発されました。しかし見栄え重視のために、表をレイアウト枠として使うなど奇抜な使い方をすると、文書の情報構造は乱れます。人やコンピュータにとって分析しづらい文書構造になってしまえば、せっかくの文書を有意義に活用しづらい状態になります。

そこで、HTML では文書構造のみを定義し、ページの見栄え(スタイル)については HTML と別に管理することが推奨されるようになりました。この見栄えの設定に用いられるのが CSS なのです。

※ HTML の新しいバージョンである HTML5 では、<font>や<center>タグなど、いくつかの見栄えに関するタグが廃止されています。

### 1.3. CSS のメリット

#### (1) 文書構造を保ったままページの見栄えを設定できる

HTML で見栄えを設定しようとして文章の内容や意味にそぐわない HTML を使用すると、文章構造が難解になります。そこで CSS を使うと文章構造(HTML)への影響を最小限にしながらページの見栄えを設定できます。ページの構造と見栄えを分離することで、HTML の記述をスッキリさせられます。

#### (2) 作業しやすくなる

例えば HTML 文章中の見出し(<h1>)の色やサイズを変えようとする、<h1>タグが使われているすべての箇所を修正する必要があります。CSS を使うと、このような見栄えの設定を一括管理できます。

また複数ページ間で同じデザインを共有できるので、サイト全体のデザイン変更などが容易になりデザインの大規模な編集が楽になります。

#### (3) SEO 効果やアクセシビリティ向上

見栄えに関する記述を CSS にまとめ、HTML から排除することで、Google 等の検索エンジンにとって解釈しやすいページとなり、ページの分析処理なども軽量化されます。これによって閲覧者がサイトを適切に検索できるようにする SEO 効果が期待できます。

※SEO (Search Engine Optimization) とは「検索エンジン最適化」という意味です。つまり Google や Yahoo などの検索サイトで検索をした際に、よりの確に(より上位に)表示されるための対策のことです。

## 2. CSS の書き方 (教科書 第 3 章 2)

### 2.1. 基本書式 [ セレクタ プロパティ 値 ]

CSS で設定を記述する際は、どの部分に対して(セレクタ)、どのスタイルを(プロパティ)、どのように(値)適用するかを記述します。例えば、以下のように記述すると、<h1>タグ範囲に対して、文字色を、赤くする効果となります。

```
h1 { color:red; }
```

原則として、セレクタ {プロパティ名:値; プロパティ名:値; …… } のように記述します。

**セレクトア(selector)**とは、見栄え(スタイル)を適用する対象を指定するもので、基本は **HTML** タグ名を書きます。この例では<h1>タグが対象であることを意味します。セレクトアを変えることで見栄えを様々な対象に設定できます。

**プロパティ(property)**とは、設定する見栄え(スタイル)の種類のことです。**値**とは、**プロパティ**に対する具体的な設定値です。この例では文字色を指定する **color** プロパティに対して、**値 red** を指定しています。**CSS** では様々な**プロパティ**が用意されており、使い分けることでさまざまに見栄えを設定できます。

**プロパティ:値;** 部分は { ~ } で囲んでください。なお{ ~ }の範囲全体を宣言ブロック (declaration block)、**プロパティ:値;** の部分をスタイル宣言 (declaration) と呼びます。

## 2.2. 複数の宣言 (プロパティ:値;) を一度に記述

一つのセレクトアに対して複数のスタイル宣言をする場合は、**セミコロン( ; )**で区切れば連続して記述できます。**セミコロン ( ; )** はスタイル宣言を区切るための記号なので、最後のスタイル宣言末尾の**セミコロン**は省略できます。ですが文法ミスを起こしやすいため、最後の**セミコロン**も省略しないことを推奨します。

```
p { color:red; background-color:green; }
```

この例では<p>タグの範囲に対して、文字色(color)を赤、さらに背景色(background-color)を緑に設定しています。

## 2.3. コメントの記述

**/\*** と **\*/**で囲んだ範囲は**コメント**となり、**CSS** では**無視**されます。一時的に **CSS** を無効にしたり、**CSS** ソース内にメモを残したりする際に便利です。なお **/\*** **/\*** **\*/** **\*/** のように二重に使うことはできません。

```
p { color:blue; line-height:1.5; } /* 段落の文字色を青くし、行の高さを 1.5 倍にする */
```

## 2.4. スペースや改行について

**CSS** では 半角スペース、改行、タブ は結果に影響を与えません。

|                                                                    |                                                          |
|--------------------------------------------------------------------|----------------------------------------------------------|
| <pre>blockquote {   color: red;   background-color: green; }</pre> | <pre>blockquote{color:red;background-color:green;}</pre> |
|--------------------------------------------------------------------|----------------------------------------------------------|

上の例は改行やスペースが異なりますが、どちらも全く同じ動作になります。見やすいように編集してください。

## 2.5. 大文字と小文字について

通常の **HTML** では、**CSS** について、大文字・小文字を区別しません。大文字で書いても小文字で書いても全く同じ動作になります。しかし拡張版 **HTML** である **XHTML** に **CSS** を使う場合は、**小文字**で記述する必要があります。

よって **CSS** を記述する際、汎用性を考えるならば、要素名や属性名は**小文字**で記述するのがおすすめです。

## 2.6. CSS の記述ミス

**CSS** では書き間違いなどで誤った記述をすると、その部分は**無視され、効果を発揮しません**。画面にはエラーメッセージなどが表示されないのが、誤りに気づきにくいです。設定を行ったのに変化がない場合は、**CSS** の書き間違いの可能性がありま。例えば **color: red;** を設定して文字の色が変わるかなど、簡単な設定でテストしましょう。

## 2.7. 設定の重複

**CSS** では、同じ効果を持つ設定を複数回書くと、原則、後に書いた設定が有効になります。(設定の上書き)

```
h1 { color:red; background-color:brown; } /* h1 タグの文字の色を赤色に 背景色を茶色に */
h1 { color:blue; } /* h1 タグの文字の色を青色に (文字色の設定重複) */
```

上の例の場合、<h1>タグの文字色は 青色 (blue) になります。背景色は茶色 (brown) のままです。

## 3. HTML への組み込み方 (教科書 第 3 章 4)

**HTML** ページに **CSS** を適用するには、大きく分けて 3 つの方法があります。

- ① 外部 **CSS** ファイルを読み込む ( <link>タグ ) おすすめ!
- ② ファイルごとに設定する ( <style>タグ )
- ③ 局所的に設定する ( style 属性 )

### 3.1. 外部 CSS ファイルを読み込む ( <link>タグ )

別途 CSS を記入したファイルを用意して、HTML ファイルに読み込ませる方法です。

そのためには HTML 文章の<head>タグ内に、以下のように<link>タグを記述し、CSS ファイル名を指定します。

HTML ファイルの内容

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="xxx.css" >
</head>
<body>
<h1>第一章 はじめに</h1>
</body>
</html>
```

CSS ファイル「xxx.css」の内容

```
h1 { color: red; } /* h1 見出しの文字色を赤にする */
```

上の例では、HTML ファイル内の全ての<h1>見出しの文字色が赤色になります。

※ もちろん、CSS ファイル名を間違えたり、ファイルの保存場所を間違えたりすると読み込めません。注意してください。

※ 複数<link>タグを書けば、複数の CSS ファイルも読み込めます。設定は混じり合い、重複する設定項目に関しては、原則、後から読み込んだ設定が優先されます。

修正作業のしやすさなどを考慮すると、この方法で CSS を読み込むのがおすすめです。

### 3.2. ファイルごとに設定する ( <style>タグ )

HTML の<head>タグ内に<style>タグを追加して CSS を設定する方法です。

この場合、書き込んだファイル内だけで CSS 設定が有効なので、特定ページにのみ特別な設定をしたい時に便利です。

```
<html>
<head>
<style type="text/css">
<!--
  h1 { color: red; }
-->
</style>
</head>
<body>
<h1>第二章 CSS とは</h1>
</body>
</html>
```

この例では、ファイル内の全ての <h1>見出しの文字色が赤色になります。

※ CSS 部分が HTML のコメント表現<!-- -->でコメント化されているのは、CSS に対応していない古いブラウザでこのページを見た時に、<style>タグ内の文字がそのまま表示されてしまうのを防ぐためです。ただ近年のブラウザは、ほぼ CSS に対応しているので、<!--と-->は省略しても、おおむね問題ないでしょう。

### 3.3. 局所的に設定する ( style 属性 )

HTML タグにそのつど style 属性を追加して CSS を記述する方法です。style 属性を追加したタグの範囲だけに CSS 設定が有効です。

```
<p style="color:red; line-height:1.5;" >CSS のメリットとは</p>
```

上の例では、指定の<p>タグ範囲内のみ、文字色が赤、行間が 1.5 倍の高さになります。

なおこの方法は、局所的にスタイルを設定する際などには便利ですが、HTML 記述が複雑になりがちです。効率良く見栄えを管理するには、3.1 のように外部 CSS ファイルを読み込むのが推奨です。

## 4. セレクタの種類

```
セレクタ { プロパティ名: 値; }
```

セレクタは見栄え(スタイル)を適用する対象を示します。 代表的には以下のような種類があります。

| 名前      | セレクタの書式    | 設定対象        | 例                     |
|---------|------------|-------------|-----------------------|
| 要素型セレクタ | タグ名        | タグ名で指定した全要素 | p {color:blue;}       |
| クラスセレクタ | .クラス名      | クラスを付けた要素   | .sample {color:blue;} |
| id セレクタ | #id 名      | id を付けた要素   | #sample {color:blue;} |
| 複数セレクタ  | セレクタ, セレクタ | 複数のセレクタ     | h1, h2 {color:blue;}  |
| 子孫セレクタ  | セレクタ セレクタ  | 下の階層の子孫要素   | td b {color:blue;}    |

## 4.1. 要素型セレクタ

セレクタにタグ名を書きます。指定したタグがページ内に存在すれば全て CSS の適用範囲になります。

例えば p {color:blue;} ならば、ページ内の全ての段落(<p>)タグの文字色が青色になります。

## 4.2. クラスセレクタ

あらかじめ HTML タグに class 属性を追加して、好きなクラス名を決めておきます。(例) <h1 class="hoge" >特定のクラス名が付けられたタグに対してだけ CSS を適用する書き方が、クラスセレクタです。

CSS ではセレクタとして class 名の先頭に ドット[.] を付けて記述すると、クラスセレクタとして扱われます。

クラス名が同じであれば、複数箇所や異なる HTML タグに対しても同じスタイルを適用できます。

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
<h2 class="info">第三章 CSS のメリット</h2>
<p class="info">CSS と HTML でページをデザイン</p>
<p>HTML5 では文章の構造と見栄えを分離することが推奨されている。</p>
</body>
</html>
```

mystyle.css の内容

```
.info {color:blue; }
```

上の例では、クラス「info」を指定した<h2>と<p>タグの範囲だけに、CSS 設定(文字色:青)が適用されます。

## 4.3. 複数セレクタ

複数のセレクタをカンマ[, ]で区切って並べると、まとめて同じ CSS 設定を適用することができます。

```
h1, h2 {color:blue;} /* h1 タグと h2 タグの両方で文字色が青になる。 */
```

## 4.4. 子孫セレクタ

セレクタを半角スペースで区切ると、ある要素より下の階層にある子孫要素を対象にスタイルを適用することができます。

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
<table>
<tr><td><p>1 時間目</p></td><td><p>2 時間目</p></td></tr>
</table>
</body>
</html>
```

mystyle.css の内容

```
td p { margin:10px; }
```

この例では<td>タグの範囲内にある<p>タグのみ、余白(margin)が 10 ピクセルに設定されます。

#### 4.5. その他のセレクト

セレクトには他にも様々な種類があります。気になった方は「CSS セレクト 種類」でウェブ検索するなどして調べてみましょう。

### 5. スタイルシートの適用に便利なタグ

CSS は既存の HTML タグの範囲を目標に見栄えを設定できます。

しかし、もともと HTML タグが書かれていない場所に見栄えを設定したい場合もあります。

このような場合のために、HTML には <div>タグと<span>タグが用意されています。

#### 5.1. <div>タグ

<div>タグは、それ自身には効果がありませんが、<div>～</div>で囲まれた四角形の範囲に対して CSS を適用させられます。 ※ <div>範囲の上下には自動的に改行が入ります。

<div>タグは単体では効果が無いので、必ずクラスや style 属性と併用して使用します。

HTML 記述例

```
<p>この文章はサンプル文章です。
```

```
<div style="border:dashed 1px red" >注意書き<br>これは重要な文章です。</div>
```

```
この文章はサンプル文章です。</p>
```

画面の出力例

```
この文章はサンプル文章です。
```

```
注意書き
```

```
これは重要な文章です。
```

```
この文章はサンプル文章です。
```

上の例では、<div>タグに style 属性を追加して、周囲に罫線(border)を引いています。

このように四角形の範囲を持つタグのことをブロック要素と言います。

なお 見出し(h1～h6)、段落(p)、リスト(ul,ol)などもブロック要素ですので、div と同じような結果になります。

また、ブロック要素は、width プロパティや height プロパティで四角形の大きさを変えられるのがポイントです。

色を塗って強調したり、ボタン類を表現したり、罫線をひいてレイアウト用の枠組みにしたりして、様々な用途で使えます。

#### 5.2. <span>タグ

<span>タグも、それ自身には効果がありませんが、<span>～</span>で囲まれた範囲に対して CSS を適用させられます。<div>タグと異なるのは、行内の一部が CSS 対象範囲となる点です。

HTML 記述例

```
<p>この文章はサンプル文章です。
```

```
<span style="border:dashed 1px red">注意書き これは重要な文章です。</span>
```

```
この文章はサンプル文章です。</p>
```

画面の出力例

```
この文章はサンプル文章です。注意書き これは重要な文章です。この文章はサンプル文章です。
```

上の例では、<span>タグに style 属性を追加し、周囲に罫線(border)を引いています。

<div>タグと異なり、行内の一部に罫線が引かれます。<span>範囲の前後には改行が入りません。

このような範囲を持つタグのことをインライン要素と言います。

なお 太字(<b>)、リンク(<a>)などもインライン要素ですので span と同じような結果になります。

インライン要素は、ブロック要素と異なり、width プロパティや height プロパティで大きさを変えられません。(width や height を使用しても無視されます)

このように、<div>タグや<span>タグを追加すれば、もともと HTML タグが書かれていない範囲にも CSS 効果を適用できます。

## 6. 様々なプロパティ (教科書 第 3 章 5)

CSS の基本的な書式は以下のとおりでした。

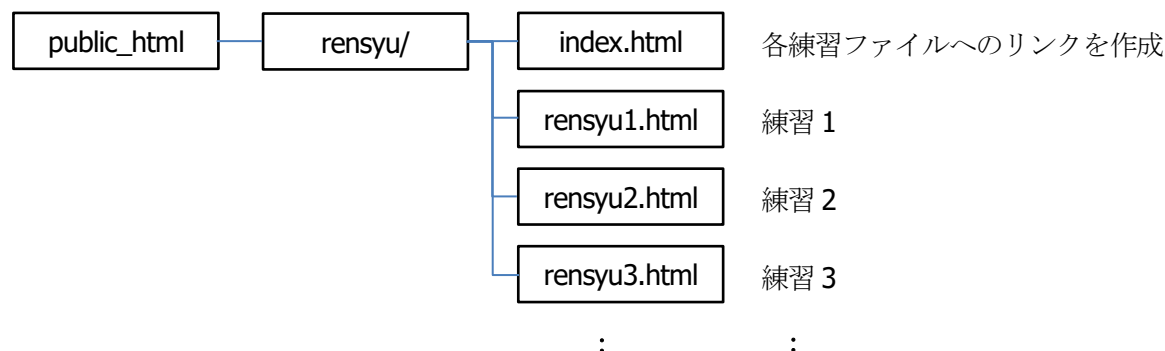
```
セクタ { プロパティ名: 値; }
```

プロパティにはたくさんの種類があります。教科書の第 3 章の 5 では代表的なプロパティの使い方を紹介していますし、巻末にはプロパティの一覧がありますので参考にしてください。もし教科書を持っていない場合は、「HTML クイックリファレンス <http://www.htmq.com/>」のページなど、インターネット上の解説サイトなども参考にしてください。

## 7. 課題

上記を踏まえて、コンピュータ基礎実習のページ (<http://www.cc.kyoto-su.ac.jp/~shimizu/>) に掲載されている[練習問題]を見ながら設問に回答し、その制作物の URL を moodle に記載してください。

以下のようなファイルを作成することになるはずです。



moodle に提出する URL は、以下のようになるはずです。

提出アドレス <http://www.cc.kyoto-su.ac.jp/~g0000000/rensyu/>

提出期限については moodle の画面を確認してください。

以上