

phase 3 : CSS

いなにわうどん* (@kyoto.inaniwa)

IPC / jsys 合同 Web 研修 2024

2024 年 6 月 8 日 (土)

本フェーズでは、**CSS (Cascading Style Sheets)** を用いた Web サイトのスタイリングについて学習します。

1 CSS の基本

本章では、スタイルシートおよび CSS に関する、基本的な概念および記述方法を説明します。

1.1 文章構造とスタイルシートの分離

CSS は、文書（多くの場合 HTML 文書）にスタイルを付与するために使用される言語です。文書にスタイルを付与する仕組みはスタイルシートと呼ばれます。

Web サイトを作成する際は、文書構造とスタイルを分けて記述することが重要です。これにより、アクセシビリティを向上させたり、複数箇所にスタイルを効率的に適用したりすることが可能となります。そのためにも、スタイルシートの利用は欠かせません。

以下に、スタイルシートがもたらす恩恵の例を示します。例えば、図 1 の HTML が存在し、この見出しに下線を付与する場合を想定します。

図 1: HTML 文書の例

```
<h2>はじめに</h2>
```

```
<p>内容内容</p>
```

```
<h2>スタイルシート</h2>
```

```
<p>内容</p>
```

*<https://github.com/inaniwaudon>

この際、すべての h2 要素に u 要素（下線を付与する）を付与すると、h2 要素の数だけ修正を加える必要が生じるため、要素数が多い場合は大変面倒な作業となります。ここで、図 2 に示すスタイルシートを適用すると、すべての h2 要素に対して一括で下線を付与することができます。このように、文章構造とスタイルシートを適切に分離することで、開発および保守が容易な Web サイトの構築が実現されます。

図 2: CSS の例

```
h2 {  
    text-decoration: underline;  
}
```

1.2 スタイルシートの記述

HTML 文書に対して、スタイルシートは以下の方法にて記述することができます。これらを組み合わせた例を、図 3 に示します。

1. 拡張子 .css のファイルに記述し、link 要素を通じて読み込む。複数のページに同一のスタイルシートを適用したい場合に採用する。
2. style 要素に記述する。単一のページ（トップページ等）にのみスタイルシートを適用したい場合に採用する。
3. style 属性に記述する。単一の要素にのみスタイルシートを適用したい場合に採用する。プロパティ: 値; を列挙する。

1.3 セレクタ、プロパティ

CSS では、セレクタを用いてスタイルの適用対象を指定し、プロパティと値を組み合わせ、スタイルの適用内容を指定します。プロパティと値は常にペアとなり、これらを合わせて宣言と呼びます。また、複数の宣言が連なったものは宣言ブロックと呼ばれ、セレクタと宣言ブロックが組み合わせるとルールになります。これらの関係性を図 4 に示します。

図 3: HTML に CSS を記述する例

```
<html>
  <head>
    <!-- 共通の common.css を読み込み -->
    <link href="common.css" rel="stylesheet" />

    <!-- このページにのみスタイルを適用 -->
    <style>
      h1 {
        font-size: 2rem;
      }
    </style>
  </head>
  <body>
    <h1>タイトル</h1>

    <h2>見出し 1</h2>
    <p>本文 1</p>

    <!-- この要素にのみスタイルを適用 -->
    <h2 style="color: red;">見出し 2</h2>
    <p>本文 2</p>
  </body>
</html>
```

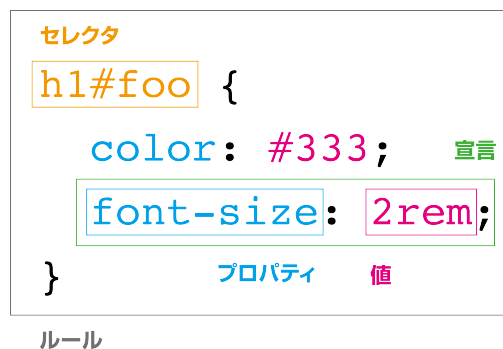


図 4: ルール、セレクトア、宣言、プロパティ、値の関係性

1.3.1 セレクトア

セレクトアは、スタイルを適用する HTML 要素を選択するために使用します。例えば、セレクトアに `h2` と指定した場合は、すべての `h2` 要素にスタイルを適用します。また、`h2#foo`

と指定した場合は、id が foo の h2 要素にのみスタイルを適用します。

セレクタには様々な種類が存在しますが、説明が複雑になるため、本資料では割愛します。代表的なセレクタは下記の Web ページにまとめられているため、必要に応じて参照することをおすすめします。以下の中でも、特に要素名セレクタ、全称セレクタ、id / class セレクタ、子孫セレクタ、子セレクタ、隣接セレクタは頻出です。

- HTML クイックリファレンス. セレクタの種類 — CSS の基本. <https://www.htmq.com/csskihon/005.shtml>.

1.3.2 プロパティ，値

プロパティは適用するスタイルの種類を指定します。適用可能なスタイルは仕様書によって明示的に定義されており、color (文字色)、font-size (フォントサイズ)、background (背景色、背景画像) 等が存在します。

値には具体的なスタイルの内容を記述します。値の記述にはいくつかのデータ型が用いられます。特に、数値、色、大きさ、比率等が代表的です。例えば、要素の幅やフォントのサイズを指定する場合は、大きさおよび比率のデータ型が、背景色の記述には色のデータ型が使用されます。これらのデータ型を組み合わせで記述する場合も存在します。

また、プロパティと値は：(コロン)によって区切られます。宣言は；(セミコロン)によって区切られます。

以下に、代表的な CSS プロパティを示します。これらのすべてを説明するにはあまりにも紙面が少ないため、プロパティが有する意味はご自身で調べてみてください。

代表的な CSS プロパティ

テキストに関するプロパティ

line-height, text-align, font, color

ボックスモデルに関するプロパティ

width, height, margin, padding, border, background

要素の配置に関するプロパティ

position, top, left, bottom, right, flex, display, z-index

1.4 カスケード，継承

同一の要素に競合するスタイルが適用された場合に、どのスタイルを適用するかで問題が生じる場合が存在します。単一のページで問題となる場合は稀ですが、例えば複数ペー

ジから構成される Web サイトを構築する場合に、異なるページ用のスタイルシートが干渉する等の例が想定されます。ただし、近年では CSS Modules¹等の導入によって、開発者がこれらを意識する必要性は減少しつつあります。

この際に登場する概念がカスケードです。カスケードは、複数のスタイルを組み合わせるためのアルゴリズムです。スタイルに優先順位を付与し、スタイルが競合した場合は、最終的に優先順位が最も高いスタイルを適用します。カスケードは、大まかに以下の順序で優先順位を付与します。

1. CSS の提供元による優先順位。ユーザ > サイト作成者 > ブラウザの順に優先される。
2. 詳細度による優先順位。後述。
3. 記述順位による優先順位。style 属性 > style 要素 > 外部 CSS の順に優先される。同一箇所に記述された場合、後に記述されたスタイルが優先される。

1.4.1 詳細度

詳細度は、セレクタが持つ「重み」を指します。ルールが持つ詳細度が大きいほど、そのスタイルは優先されます。詳細度は id セレクタ > class セレクタ = 属性セレクタ > 要素名セレクタ といった具合に計算されます（実際にはその他にも仕様が存在します）。厳密には詳細度とは異なる概念ですが、特定の宣言を優先的に適用したい場合は、値の後ろに !important を付与する手法も存在します。ただし混沌を招くため、乱用は禁物です。

なお、詳細度の計算方法は、以下のページで詳細に解説されています。

- MDN. 詳細度 - CSS: カスケーディングスタイルシート. <https://developer.mozilla.org/ja/docs/Web/CSS/Specificity>.

詳細度を意識することにより、効率的な CSS を記述することができます。ただし、スタイルの量が肥大化した際に、そのすべてを把握することは極めて困難となる場合がほとんどです。したがって、近年では Block Element Modifier (BEM) に代表されるような、クラス名のみを利用したスタイリングが重視されています。

1.4.2 継承

いくつかのプロパティは継承、すなわち子要素に引き継がれます。例えば、要素 A に `border: solid 1px #ccc; (1px のボーダを付与), font-size: 14px; (フォントサイ`

¹<https://github.com/css-modules/css-modules>

ズを 14px に設定) の宣言を指定する例を想定します。また、要素 A は子要素 B を持つとします。この際、子要素 B は枠線は付与されませんが、フォントサイズは 14px となります。これは「要素 A の width プロパティは子要素 B に継承されないが、font-size プロパティは継承される」という仕様を表しています。場合によっては意図せぬスタイルが継承されることもあるので、注意が必要です。

1.5 おわりに

本資料で扱ったのは氷山の一角に過ぎず、CSS には他にも様々な仕様が存在します。その一例を以下に示します。

本資料では扱えなかった仕様群

ボックスモデル、インライン/ブロックレベル、CSS アニメーション、フレックスボックス、グリッド、Web フォント、レスポンシブデザイン、メディアクエリ、CSS 変数、関数

CSS は、後に扱う JavaScript 等とはまた異なる難しさがありますが、複雑なプロパティを緻密に制御していくことで、多様な表現を実現することができます。拙作ながらその例として、図 5 に 2021 年度の雙峰祭 Web サイトを示します。ぜひ、デザインを楽しみながら CSS を学んでいきましょう！



図 5: 2021 年度の雙峰祭 Web サイト

2 演習問題

図 6 に示す Web ページのデザインを再現するための CSS ファイル `style.css` を記述してください。Web ページの文書構造を表す HTML は、図 7 に示すソースコードをそのまま使用してください（ソースコードは `\day1\phase2\samples\index.html` にも添付しています）。

問題の難易度はやや高めです。本資料で扱いきれなかった内容を含みます。

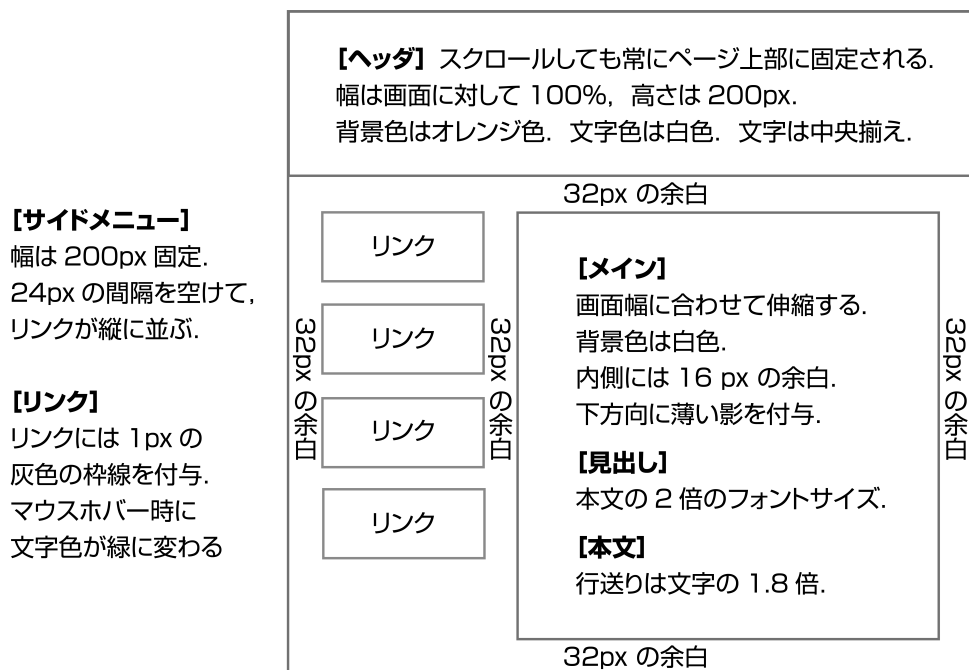


図 6: 作成する Web ページのデザイン指示

[illegible]