CSSで画面を装飾しよう

メンバープロフィール

NAME

仁和 泰也

こんにちは。仁和泰也といいます。友人からはにわりんと呼ばれているので、にわりんと呼んでください。住まいは新宿で、大学4回生です。趣味は、写真を撮ることが好きなので、休日は山に登り、風景や草花をカメラに収めています。また、自粛生活が続いたとき、料理くらいできないとと思い、夏から料理教室に通っています。得意料理はカレーライスです。

生年月日	2000年10月10日
血液型	A型
出身地	東京都
所属・部署	東京工業大学 情報理工学院
星座	天秤座
趣味	写真・料理
ニックネーム	にわりん

CSSってなに？

Cascading Style Sheets

HTMLのスタイルを定義する言語。
HTMLの構造に対して、段階的にスタイルを定義できる。

どうやって書くの？

方法は3つあります。

- 要素に直接埋め込む
- ページのヘッダに埋め込む
- 別ファイルで定義したものを読み込む（推奨）

要素に直接埋め込む

```
<header style="background-color: red;">
```

ページのヘッダに埋め込む

```
<head>
    ～中略～
    <style type="text/css">
        main {
            background-color: blue;
        }
    </style>
</head>
```

別のファイルに切り出す

HTMLと同じ場所に、style.cssというファイルを作ります。

HTMLに読み込みの定義をします。

```
    <head>
        ～中略～
        <link rel="stylesheet" href="style".css">
        ～中略～
    </head>
```

CSSファイルにスタイルを定義してみましょう。

```
footer{
    background-color: yellow;
}
```

スタイルの指定方法

```
header {
    background-color: red;
    color: white;
}
main {
    background-color: blue;
}
```

- セレクタ（スタイルの適用対象）に対して、プロパティ（指定できるスタイル名）と値のセットを指定する
- プロパティと値のセットごとに、区切りとしてセミコロンをつける

セレクタの書き方

```
#greeding {
    font-size: 20px;
}
.icon {
    color: gray;
}
header, footer {
    background-color: lightgray;
}
p {
    background-color: lightblue;
}
.description p {
    color: #800000;
}
```

- タグのほか、HTMLに属性をidやclassを追加して指定できる
  - idは画面内で同じ値が重複してはいけない。CSSでは頭に「#」をつけて指定する。
  - classは画面で同じ値を複数指定してよい。CSSでは頭に「.」をつけて指定する。
- スペースで区切って、要素の構造で指定できる
- カンマで区切って、複数の要素を指定できる

文字を装飾する

以下のような、文字を装飾するプロパティがあります。

```
.description p {
    color: #800000;
    font-size: 14px;
    font-weight: bold;
    font-family: "Meiryo UI", "Hiragino Maru Gothic ProN";
}
```

上記は一例であり、たくさんありますので、

どんなことができるかリファレンスを検索して眺めてみましょう。

ブロックを整形する

以下のようにブロックレベルの要素を整形するプロパティがあります。

```
.description {
    border: 1px solid black;
    margin: 10px;
    padding: 10px;
    width: 500px;
    height: 200px;
}
```

ブロックをレイアウトする

ブロックをレイアウトしてみましょう。

profile

overview

メンバープロフィール

仁和 泰也

こんにちは。仁和泰也といいます。 友人からはにわりんと呼ばれているので、にわりんと呼んでください。 住まいは新宿で、大学4回生です。趣味は、写真を撮ることが好きなので、 休日は山に登り、風景や草花をカメラに収めています。 また、自粛生活が続いたとき、料理くらいできないとと思い、 夏から料理教室に通っています。得意料理はカレーライスです。

detail

生年月日	2000年10月10日
血液型	A型
出身地	東京都
所属・部署	東京工業大学 情報理工学院
所属・部署	東京工業大学 情報理工学院
所属・部署	東京工業大学 情報理工学院
星座	天秤座
趣味	写真・料理
ニックネーム	にわりん

```
<main>
    <div id="profile">
        <div id="overview">
            <h1>
                メンバープロフィール
            </h1>
            ～中略～
            <div class="description">
                ～中略
            </div>
        </div>

        <div id="detail">
            <table>
                ～中略～
            </table>
        </div>
    </div>
</main>
```

メインタグの中を、divタグを使ってブロックで区切ります。

profileの中に、overviewとdetailが並ぶという階層にします。

```
#profile {
    display: flex;
}
```

- display: 要素の表示形式を指定するプロパティ
    - flex：要素内のボックスを柔軟に並べて配置する。

画面はブロックが階層になっていたり、並んだりしてレイアウトされていきます。
この考え方は、しっかり身につけておきましょう。

他にも細かな指定ができたり、gridやfloat、positionなど
他のレイアウト手法もありますが、
ここでは割愛します。興味があったら、調べてみましょう。

CSSフレームワークを使おう

ここまでCSSの基本的な書き方を学んできました。
開発現場では、CSSをいちから自分で書き起こすのではなく、
CSSフレームワークというものを用いることも多いです。

標準的なデザイン一式があらかじめコーディングされており、
CSSフレームワークの決まりにのっとってHTMLを書くことで、
勝手にデザインが適用されます。

CSSフレームワークを使って、画面を装飾していきましょう。

CSSフレームワークの種類

- Bootstrap
  - https://getbootstrap.jp/
- Bulma
  - https://bulma.io/
- Tailwind
  - https://tailwindcss.com/

今回はTailWindを使います。
軽量でカスタマイズしやすいCSSフレームワークで、
どんどん人気があがってきています。

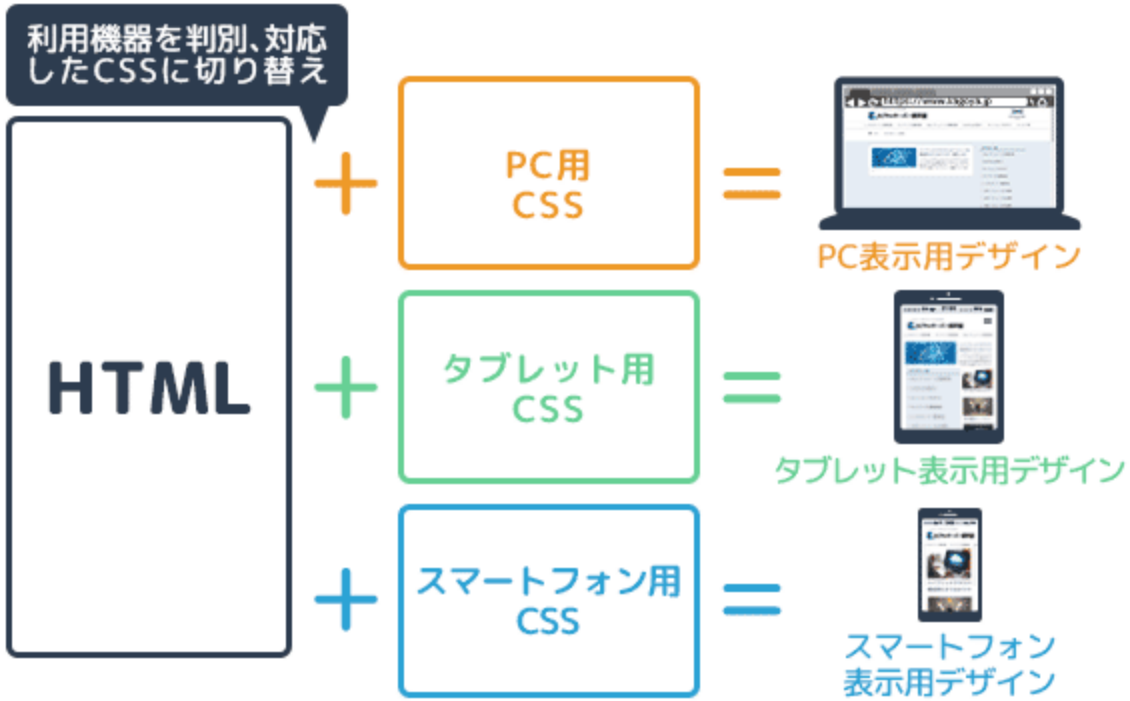
前段として

多くのCSSフレームワークで採用されていて、
デザインの仕方としても一般的な概念を二つ紹介します。

- レスポンシブウェブデザイン
- グリットレイアウト

レスポンシブウェブデザイン

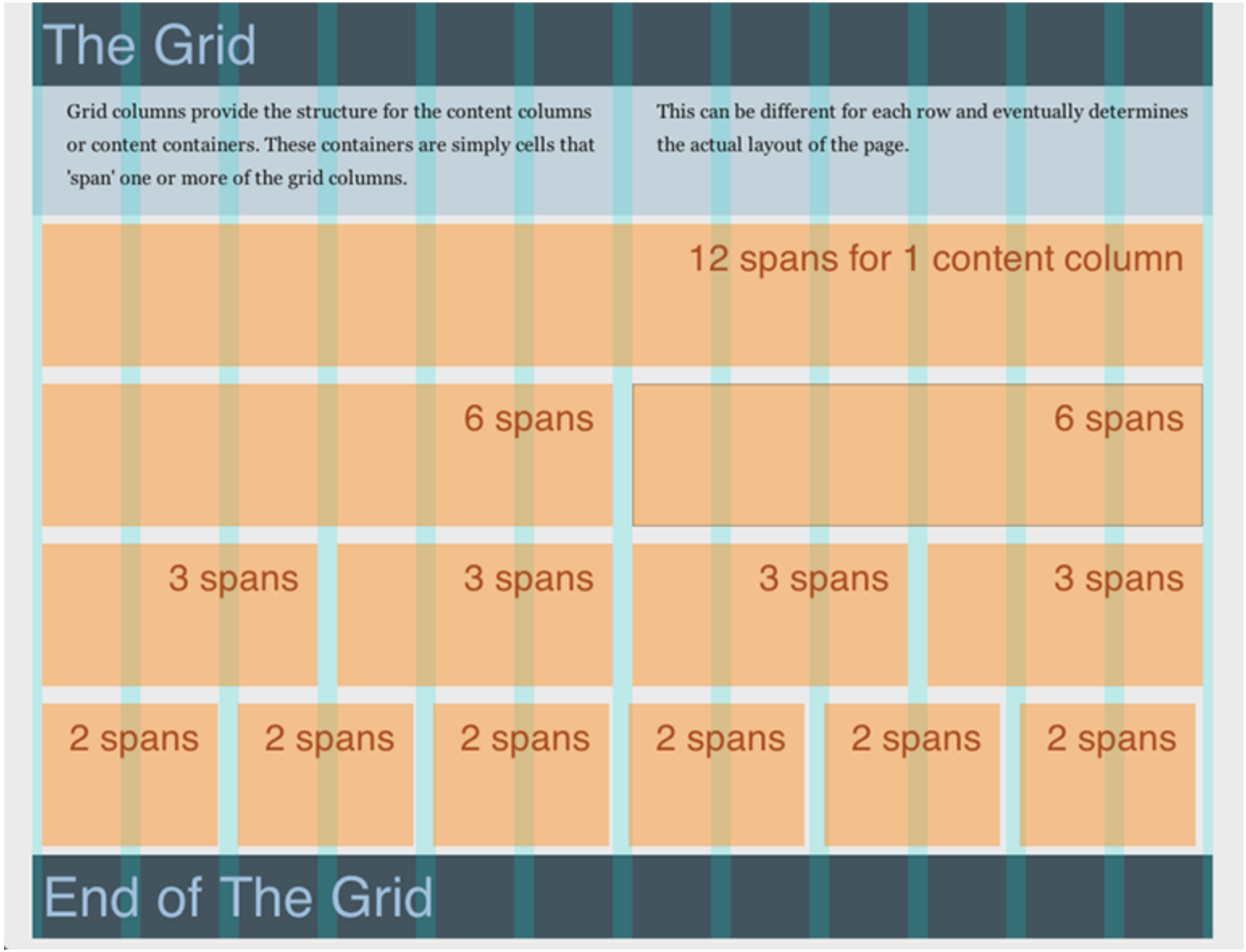
画面幅によって、適用するCSSを動的に切り替えることで、
様々な画面幅のデバイスでの表示に対応するという考え方です。



出典：【入門】レスポンシブWebデザインとは？概要と作り方を丁寧解説

グリッドレイアウト

画面を12や16といった列に分けて捉えて、レイアウトする考え方です。
画面幅によって、各要素で使う列数を変えるというふうに考えられるので、
レスポンシブウェブデザインと相性がいいです。



出典：グリッドレイアウトとブロークングリットレイアウト

準備

```
<head>
    〜中略〜
    <link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">
    <link rel="stylesheet" href="style.css">
<head>
```

headで、tailwindのcssの読み込みを宣言しましょう。

書き方

多くのCSSフレームワークでそうなのですが、
決められたclassを要素につけることで、
予め定義されたcssが適用されます。

画面大枠のデザインを指定しよう

```
<body>
    <div class="bg-white flex flex-col font-sans text-gray-600">
        <div class="container mx-auto">
            ～中略～
        </div>
    </div>
</div>
```

- bg-{色}: 背景色を指定する
- font-{ファミリー名}: フォントファミリーを指定する
- text-{色}-{濃さ}: テキストの色を指定する
- flex: flexのボックスであることを指定する
- flex-{方向}: フレックスボックス内でどの方向にコンテンツを配置するか指定する
- container: コンテンツ全体を囲むことで最大幅をいい感じにする
- m{場所}-{量}: マージンを指定する。autoを指定することで中央寄せになる

ヘッダーにデザインをあてよう

```
<header class="flex items-center justify-between relative pl-4 sm:pl-0 py-6">
    <p class="leading-none">
        <a href="#" class="text-xl sm:text-2xl font-bold text-blue-900 hover:text-blue-800">
            SAMPLE PROJECT
        </a>
    </p>
    <nav>
        <a href="list.html"  class="text-blue-900 hover:text-blue-600 py-3 px-6 text-sm font-bold">
            メンバーリスト
        </a>
        <a href="detail.html" class="text-blue-900 hover:text-blue-600 py-3 px-6 text-sm font-bold">
            メンバープロフィール
        </a>
    </nav>
</header>
```

- flex: flexのボックスであることを指定する
- justify-{寄せ方}: flexボックスに並べて行くときの寄せ方を指定
  -（並べる方向に対してどう寄せるか）
- item-{寄せ方}: flexボックスに並べて行くときの寄せ方を指定
  -（並べる方向と垂直にどう寄せるか）
- p{場所}-{量}: パディングを指定する
- leading-{高さ}: 行の高さを指定する
- sm/md/lg/xl: デザインを適用する画面サイズの条件指定
- text-{サイズ}: テキストのサイズを指定
- text-{色}-{濃さ}: テキストの色を指定
- hover/focus: デザインを適用する要素の状態の条件指定
- font-{重さ}: フォントの重さ（太さ）の指定

ナビゲーションをレスポンシブ対応しよう

```
<nav class="hidden md:flex text-lg">
    〜中略〜
</nav>
<button class="flex md:hidden flex-col absolute top-0 right-0 py-6 px-4">
    <span class="w-6 h-1 mb-1 bg-gray-500"></span>
    <span class="w-6 h-1 mb-1 bg-gray-500"></span>
    <span class="w-6 h-1 mb-1 bg-gray-500"></span>
</button>
```

- hidden: ボックスを非表示にする
- absolute: 絶対位置を指定して配置

メイン部分をレイアウトしよう

```
<h1>
    メンバープロフィール
</h1>
<div class="lg:w-11/12 mx-auto flex flex-wrap">
    <div class="p-4 lg:px-8 lg:w-1/2 w-full">
        <img src="パス" alt="仁和のプロフィール画像">
        ～中略～
    </div>
    <div class="p-4 bg-gray-100 mt-8 lg:w-1/2 w-full">
        <table>
        ～中略
        </table>
    </div>
</div>
```

w-{サイズ}: 幅の指定。1/2など、親要素に対する割合でも指定できる

flex-{折り返し方法}: flexボックス内の要素がはみ出るときの動きを指定する

見出しにデザインをあてよう

```
<h1 class="text-2xl sm:text-3xl text-blue-900 p-4 mb-4 md:mb-8 border-b">メンバープロフィール</h1>
〜中略〜
<h2 class="text-blue-900 text-2xl sm:text-3xl title-font font-medium mb-1">仁和 泰也</h2>
```

border-{配置}: ボーダーの表示を指定する

画像・名前の部分をレイアウトしよう

```
<div class="flex items-start">
    <div>
        <img src="https://images.unsplash.com/photo-1577409873799-29f8be1311dd" alt="仁和のプロフィール画像">
    </div>
    <div>
        <h2 class="text-blue-900 text-2xl sm:text-3xl title-font font-medium mb-1">
            仁和 泰也
        </h2>
        〜中略〜
    </div>
</div>
```

画像をカッコよく切り取ろう

```
<div class="shadow-lg h-20 w-20 sm:h-24 sm:w-24 border-white rounded-full overflow-hidden border-4 mr-4">
    <img src="https://images.unsplash.com/photo-1577409873799-29f8be1311dd?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1834&q=80" class="object-cover w-full h-full" alt="プロフィール画像" />
</div>
```

- shadow-{影の大きさ}: 影を指定する
- rouded-{大きさ}: 角の丸みを指定する
- overflow-{方法}: ボックス内の内容がボックスの大きさを超える場合にどうするか指定する

アイコンをレイアウトしよう

```
<div class="flex mb-4">
    <a href="#" class="ml-2 text-gray-500">
    <svg ～中略～ class="w-5 h-5">
        <path ～中略～></path>
    </svg>
    <a href="#" class="ml-2 text-gray-500">
    <svg ～中略～ class="w-5 h-5">
        <path ～中略～></path>
    </svg>
    </a>
    <a href="#" class="ml-2 text-gray-500">
    <svg ～中略～ class="w-5 h-5">
        <path ～中略～></path>
    </svg>
    </a>
</div>
```

自己紹介分にデザインをあてよう

```
<hr class="my-4 sm:my-8" />
<p class="leading-relaxed">
    本文
</p>
```

hrタグ: 水平線を引くタグです

テーブルにデザインをあてよう

```
<div class="p-4 bg-gray-100 mt-8 lg:w-1/2 w-full">
    <div class="shadow-md rounded overflow-y-auto">
        <table class="w-full text-md bg-white">
            <tbody>
                <tr class="border-b bg-gray-100">
                    <th class="p-4 whitespace-no-wrap text-left">生年月日</th>
                    <td class="p-4">2000年10月10日</td>
                </tr>
                〜中略〜
            </tbody>
        </table>
    </div>
</div>
```

- whitespace-{ルール}: 改行の扱い方を指定

フッターにデザインをあてよう

```
<footer class="p-4">
<p class="text-gray-500 font-xs text-center mt-5">
    provided by Guild Works Volunteers
    </p>
</footer>
```

最後に

CSSとTailwindの書き方を学びました。
たくさんのプロパティがでてきましたが、
だいたいどんなことができるかは想像できるようになったと思います。

これからはリファレンスをひきながら、
ページを作っていきましょう。