

Web 講義 4 ~Tailwind CSS~

Tailwind とは？

CSS とは html の見た目の部分を制御するもの。
まず、「MDN CSS の基本」および「MDN CSS の構造」を参照し、すべて読んでください。

- ・ style.css

```
p{/* <-セレクター */  
  color: red; /* プロパティ:値 */  
  font-size: 20px;  
}  
.bold{/* bold:太字 */  
  font-weight: bold;  
}
```

とくにセレクターのはじめにドットがついた場合、クラスセレクターといい、以下のようにして使用できる。

- ・ index.html

```
<p class="bold">  
  太字の文章  
</p>
```

課題

Next.js のプロジェクトを開き、上のような bold クラスを p タグに適用させてみよう。page.tsx と同階層の global.css に css を記入できる。

ただし、jsx 記法では `class="bold"` ではなく、`className="bold"` となることに注意せよ。

このようにして css を使えば、サイトをどのような見た目にもすることが可能だが、自分でクラス名を考えるのは意外と難しい。

そこで事前にクラスをたくさん書いてくれているフレームワークを使用する。Bootstrap や bulma などがあげられるが、Next.js と相性の良いモジュールは Tailwind である。

今回の講義では Tailwind の公式サイトを常に開いておくこと。

<https://tailwindcss.com>

CSS の値と単位

「MDN CSS の値と単位」を参照。

px,rem,deg,s,vw,vh,lvw,lvh,%,calc について理解すること。

Tailwind CSS によるレイアウトの方法

- ① やりたいことをどのように CSS で再現すればいいのか調べる。
- ② Tailwind 公式サイト内を検索する。(Ctrl+K)

Tailwind CSS におけるレスポンス対応

sm,md,lg,max-sm,max-md,max-lg

Tailwind CSS におけるカスタム設定

- ・ 任意の値を設定する
w-[100px]や ml-[30vw]のように設定する。
- ・ tailwind.config.ts を設定する
->参考

課題(終わらないと思うのでお家で頑張りましょう)

- ・ CSS 初心者の方は MDN の CSS の記事を最初から読んでいく。(「MDN CSS の基本」から続きを)
- ・ ある程度 CSS についての知識がある人は「MDN テキストの装飾」「MDN CSS レイアウト」を読みつつ、実際に Tailwind で書きながら試していこう。
- ・ CSS の基本的なプロパティを理解できた人は阿部寛のホームページまたは「MDN 基礎的なレイアウトの理解」のページを再現してみよう。
- ・ 再現できたら、デベロッパーツールを用いてスマホ用のサイトにもなるようにレスポンス対応をしてみよう。

CSS のポイント

- ・ block,inline,hidden
- ・ relative&absolute
- ・ translate
- ・ flex,items-center,justify-center,gap
- ・ grid

- border
- shadow

以下参考 Next.js の機能と深く関わる部分

CSS Modules(参考)

- page.tsx

```
import styles from
"@styles/test.module.css"

export default function Page() {
  return (<main>
    <div className={` ${styles.test}`} />
  </main>)
}
```

- styles/test.module.css

```
.test{
  color: red;
  font-size: 20px;
  font-weight: bold;
  text-align: center;
}
```

実際のページで確かめるとわかるが class 名が test-(6文字のランダムな文字列)のようになっており、クラス名が被ることがなくなる。

Tailwind.config.ts の設定(参考)

```
import type { Config } from "tailwindcss";

const config: Config = {
  content: [
    "./pages/**/*. {js,ts,jsx,tsx,mdx}",
    "./components/**/*. {js,ts,jsx,tsx,mdx}"
  ],
  theme: {
    extend: {
      backgroundImage: {},

```

```
colors: {
  theme: "#FF9209",
},
fontFamily: {
  avianosans: ["aviano-sans"],
  zen_kaku_gothic_new: ["var(--font-zen-kaku-gothic-new)"],
},
},
plugins: [],
};
export default config;
```

詳細は Tailwind の Customization を参照(そちらの方がわかりやすい)

このように設定すると bg-theme や font-theme のようにして #FF9209 を呼び出すことができる。

また、font-avianosans や font-zen_kaku_gothic_new のようなクラス名も作成される。

この二つのフォントで設定方法が違うが、これはフォントの読み込み方が違うからである。Avianosans は adobe font を経由して読み込んでいるためそのままクラス名 aviano-sans を指定している。だが全角ゴシックフォントについては Next.js の最適化モジュールを使用して読み込んでいるため、別に variable というプロパティを別に指定する必要がある。

- layout.tsx

```
import { Zen_Kaku_Gothic_New } from
"next/font/google";
const zen_kaku_gothic_new =
Zen_Kaku_Gothic_New({
  subsets: ["latin"],
  weight: "400",
  variable: "--font-zen-kaku-gothic-new",
});
```