# 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におけるレスポンシブ対応

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初心者の人は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シブ対応をしてみよう。

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}",

    "./app/\*\*/\*.{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",

});