

# 情報デザイン演習I 9.プロトタイプ作成

HTMLを利用したプロトタイプを作成する。

## 1. 今日の内容

- i. レスポンシブデザイン(再度！)
- ii. デザインをコーディングしてHPにしてみよう。

## 前回のおさらい

- Webレイアウトの応用
  - float
  - Flexbox
  - CSS Grid

教科書から離れて、今時のレイアウトについて紹介しました。

# 今日の内容

## レスポンシブデザイン(再度！)

## レスポンシブデザイン

さて、7回目に

- レスポンシブデザイン
- Bootstrap

についてやりましたね。

前回の内容(CSS Grid)を踏まえると、レスポンシブデザインのやり方も変わってきそうですね。

## Bootstrap

Bootstrapが素晴らしいことは変わりますが、

- Flexbox
- CSS Grid

を学んだ今、Bootstrapを利用する必要はどこにあるのでしょうか？

## Bootstrapの機能

レスポンシブデザインのための機能を保持していることは当然ですが、他にも

- Formのレイアウト
- コンポーネント
- Bootstrap Icon

等いろいろあります。ちょっとみてみましょう。

- <https://getbootstrap.jp/docs/5.0/forms/overview/>
- <https://getbootstrap.jp/docs/5.0/components/accordion/>
- <https://getbootstrap.jp/docs/5.0/extend/icons/>



## CSS フレームワーク

様々な機能を提供するCSSフレームワークはたくさんあります。

時代と共に利用のされ方も変わっていきます。少し覗いてみましょう。

- <https://0115765.com/archives/6867>

## Webアイコンフォントサービス

アイコンも簡単に利用できるようなサービスがたくさんあります。

- <https://webdesign-trends.net/entry/11656>

## Bootstrapを使わないレスポンシブデザイン

```
@media screen and (min-width: 640px){  
  CSSの指定  
}
```

というメディアクエリーについて学びましたね。

これと、前回にやった...方法Bで統一しようかな

```
grid-template-rows: 100px 50px;  
grid-template-columns: 150px 1fr;  
grid-template-areas:  
  "areaA areaB"  
  "areaA areaC";
```

のところを組み合わせればできそうですね!

## やってみよう

前回のファイルを使って、(ない人はWebからダウンロードできます。)

[test\\_cssgrid\\_3column-B.html](#)

どうやったら、

```
max-width: 640px;
```

で縦に並べられるか考えてみよう。

## 答え

```
@media screen and (max-width: 640px){  
  .container {  
    grid-template-columns:none;  
    grid-template-rows:none;  
    grid-template-areas:  
      "header"  
      "nav"  
      "main"  
      "ad"  
      "side"  
      "footer";  
  }  
}
```

## なぜ方法Bか？

方法Aでは、要素の方に

どこからどこを使いたい！

と指定しましたが、方法Bだと、

\*\*\*という名前のところに配置して！

と言っているだけなので、親要素を変更するだけで良いのがメリットとなります。

**デザインをコーディングしてHPにしてみよう。**

ここからは配布資料を元に作業を進めていきます。

## 終わり

ID\_09を圧縮したzipファイルをmanabaに提出してください。