

情報デザイン応用演習I 10.CSSでレイアウト

目次

1. 初めに

i. float

ii. Flexbox

iii. CSS Grid

2. まとめ

初めに

前回の内容

CSSでレイアウトデザインするときに

- float
- flexbox
- CSS Grid

の3種類があり、使いわけについて説明しました。

レスポンシブデザイン

レスポンシブデザイン

PCでもスマホでもタブレットでも全てで綺麗に見えるレスポンシブデザインというのがありましたね。

ウィンドウの幅を変化させるとブレイクポイントでデザインが変わるものでした。

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せんが、

- 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昨年学びましたね。

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

来週は

- 終わってなければ続き
- 終わっていたらWordpressに行こうかな...

の予定です。