情報デザイン演習I 7.Webレイアウト基本

> float, Flexbox, CSS Gridによるレイアウトについて学修する。

1. 今日の内容
    i. レスポンシブ Web デザインの仕組みを理解する
    ii. CSS フレームワークについて理解する
    iii. ワイヤーフレームを描く
    iv. プロトタイプの作成 (1) ページの構造化
    v. プロトタイプの作成 (2) ページの視覚表現「固定幅レイアウト」
    vi. [プロトタイプの作成 (3)コンテンツの「幅」の調整

前回のおさらい

- Webレイアウトの基礎
  - 文書をHTMLで構造化する
  - HTMLアウトラインを確認する
  - セマンティックコーディングをしていこう
  - header/main/footer/aside/navタグ
  - article/sectionタグ

残っていること

基本的な話はほぼ終わりに近づいています。本格的なレイアウトを組むための

- Flexbox
- CSS Grid

についてやってみましょう。(floatもおさらいします)

今日の内容

FlexboxとCSS Grid

前回、floatについて触れましたが、これからはレイアウトにfloatは使いません。

画面をレイアウトするときに利用するのが、

- Flexbox
- CSS Grid

になります。Flexboxが先に一般的になり、CSS Gridがその後にできた技術となります。現在は、どっちを使うか、で議論が起きている気がしますが、

一般的には

- Flexbox: 1方向に並べる時(折り返しあり)
- CSS Grid: 2次元にレイアウトする時

と言われています。

float

floatの難しいところ

これまで2段組などレイアウトしていくときにはfloatを多用していました。

ただし、難しいところは、

> clear: both

としないと、レイアウトが簡単に崩れてしまうところです。

また、内容量によって、要素の高さが不揃いになるところも面倒くさい理由でした。

今日のPDF資料について

レイアウトの実験が目的なので

- 内部スタイルシートを利用している(styleタグで、CSSを実装)
- header, main, footer等のセマンティックコーディングはしていないことに注意してください。

やってみよう

「ID_07」フォルダを作成してから、

> float/FlexBox/CSS Grid入門

の「floatはレイアウトを組むのには厄介！」をやって見ましょう。

floatのデメリット

- clear: bothしないと、変な回り込みを起こす
- 要素の高さがバラバラ

Flexbox

Flexboxとは？

正式名称はFlexible Box Layout Moduleといいます。

今までよりも自由に、そして簡単に横並びのレイアウトを作ることができます。
フレキシブル（柔軟性のある）レイアウトができます。

Flexboxの長所

あくまでも基本は要素を横に並べるためのものですが、

- 要素の高さを自動で揃えてくれる
- 要素が多くなり、横並びできなくなると自動で折り返してくれる
- 余白の調整が簡単
- 並び順を自由に変えられる(逆とか)

Flexboxの使い方

要素が並ぶ箱に

> display: flex;
> を追加するだけです。

やってみよう

「Flexboxは横並びに便利」をやって見ましょう。

CSS Grid

CSS Grid

CSS Grid Layoutが正式名称となります。

Flexboxが1次元だったのに対し、CSS Grid Layoutでは2次元レイアウトを作成することが可能になります。

これからのWebデザイン

間違いなく

- Flexbox
- CSS Grid Layout

によるレイアウトに移行していきます。それではCSS Grid Layoutに入っていきましょう。

2次元レイアウト

それでは2次元レイアウトとはなんでしょうか？

今の所多くのデバイスは液晶ディプレイにWebを表示していますから、2次元ですね。

ということは、サイトデザインをこの方法でレイアウトすることが可能となります。

具体的には

マス目を用意しグリッドを作成し好きな順番に配置したりすることで様々なレイアウトが可能になります。

用語

HTML要素

- コンテナ グリッド全体を囲む要素
- アイテム コンテナの子要素

概念

- ライン グリッドを分ける垂直および水平線のこと。上下左右の端にも存在する
- トラック グリッドの行と列
- セル ラインで囲まれる最小単位
- エリア 一つ又は複数のセルが結合してできるセルの集まり

やってみよう

「CSS Grid Layoutは簡単！」をやって見ましょう。

まとめ

チートシート

- Flexboxチートシート
- CSS Grid Layoutチートシート

どう使い分ける？

float

純粋に画像などの回り込みに利用する

Flexbox

1方向に並べるときに利用する(改行あり)

CSS Grid

サイト全体をレイアウトする

詳しい記事

- FlexboxとCSS Gridの違いと使い分け｜よくあるレイアウトで理解する

詳しくみていきましょう。

Flexbox/CSS Gridの使い分けに関しては多少議論があるようです。

- 実例で学ぶFlexboxとCSS Gridの使い分け

教科書で復習しよう

- 3-16「レイアウトを組もう」P.154~P.159
- 3-17「CSSグリッドでタイル型に並べよう」P.160-P.166

です。

これで、3章まで終わりました。

レイアウトの基本で漏れていることが

- 4-10 レスポンシブに対応させよう(P.206~214)

となります。仕組みだけ伝えて終わりにしようと思います。

追記:

教科書ではmin-width, max-widthを使っていますが、これより「range-syntax」の方が直感的で分かりやすいです。

- ついにSafariも。 media queryの範囲指定をより直感的に書ける記法が全ブラウザ対応へ

サンプル html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Media Query</title>
    <style>
        div {height: 50vh;}
        @media screen and (width<800px) {
            div {background-color: green;}
        }
        @media screen and (800px <= width <= 1000px) {
            div {background-color: red;}
        }
        @media screen and (1000px < width) {
            div {background-color: blue;}
        }
    </style>
</head>
<body>
    <div>
        <p>Hello</p>
    </div>
</body>
</html>
```

レスポンシブ対応するには？

メディアクエリを利用して、gridのエリアを変更するだけですね！！！

これで余計なclear: both;など考えなくて良くなります。

終わり

ID_07を圧縮したzipファイルをNASに提出してください。