

デザインをコーディングして HP にしてみよう

2022 年 6 月 10 日

概要

今どきの CSS レイアウトについて Flexbox, CSS Grid を学びました。これを使って、HTML, CSS でデザインをコーディングしていきましょう。

目次

1	はじめに	2
1.1	読み間違えないでね	2
1.2	注意	2
2	デザインの入手	2
2.1	Web デザイン	2
2.2	プロトタイピングツールとは	2
2.3	XD	3
2.4	UI キット	3
2.5	デザインを入手しよう	3
3	HTML の構造を考える	3
4	HTML/CSS でレイアウトしていこう	5
4.1	HTML の構造 (おおまかな)	5
4.2	おおまかなレイアウトを CSS Grid で指定してみよう。	6
4.3	.menubar のレイアウトしてみよう	7
4.4	header のレイアウトしてみよう	8
4.5	.trending のレイアウトしてみよう	8
4.6	.happening-now のレイアウトしてみよう	8
5	画像	9
5.1	header 背景画像, HappeningNow 背景画像	9
5.2	Trending 画像, HappeningNowSmall 画像	10
6	まとめ	11

1 はじめに

1.1 読み間違えないでね

ソースコード 1 読み間違えないでね

```
1 数字: 0123456789
2 小文字:abcdefghijklmnopqrstuvwxyz
3 大文字:ABCDEFGHIJKLMNOPQRSTUVWXYZ
4
5 1:イチ
6 l:小文字のエル
7 i:小文字のアイ
8 !:ビックリマーク
9 |:バーティカルバー。Shift と ¥ を押したもの。
10
11 0:ゼロ
12 o:小文字のオー
13 O:大文字のオー
14
15 .:ピリオド
16 ,:コンマ
```

1.2 注意

- これから出てくるソースコードには、左に「行番号」と呼ばれる番号が出てくるけど、入力する必要はないからね。

2 デザインの入手

2.1 Web デザイン

実際の現場では、一人でやることもあります

Web デザイナー デザインする人

Web コーダー HTML, CSS, Javascript など再現する人

に分かれて作業します。

Web デザインは、かつては Photoshop, Illustrator を用いていましたが、最近では挙動もデザイン段階で確認するため、プロトタイピングツールが利用されます。

2.2 プロトタイピングツールとは

「プロトタイプ」とは、Web サイトやアプリなどの試作品を早い段階から作成する開発手法、およびその過程を意味します。UI デザインや UX デザインのブラッシュアップに用いられることが多いです。プロト

タイプを作成するツールを「プロトタイピングツール」と呼びます。^{*1}

プロトタイプは、システム開発工程の一部として組み込まれることもあり、主に以下のメリットがあります。

1. 欠陥の早期発見・改善

プロトタイプを用いて検証を行うことで、バグや欠陥などを早い段階で発見でき、早期に改善できる。

2. 関係者間での認識のズレの解消

機能やアイデアなどを試作品として形にすることで、関係者やユーザーから早い段階でフィードバックが得られ、関係者間での認識のズレを早期に解消できる。

2.3 XD

Adobe XD は、Adobe 社のプロトタイピングツールで、web サイトや、モバイルアプリなどのデザインに適した、オールインワンの UX/UI ソリューションです。デザイン、プロトタイプ、共有、全てを XD でおこなえます。また、Adobe XD は、共同作業を促進するパワフルで使いやすいプラットフォーム。web サイトやモバイルアプリ、音声インタフェース、ゲームなどのデザイン制作をチーム全体でスムーズにおこなうことができます。

XD の使い方について詳しくは、後期に学んでいきましょう。

2.4 UI キット

Adobe XD では参考となる UI キットを配布しています。今回はそれを参考に HP をコーディングしていきます。

2.5 デザインを入手しよう

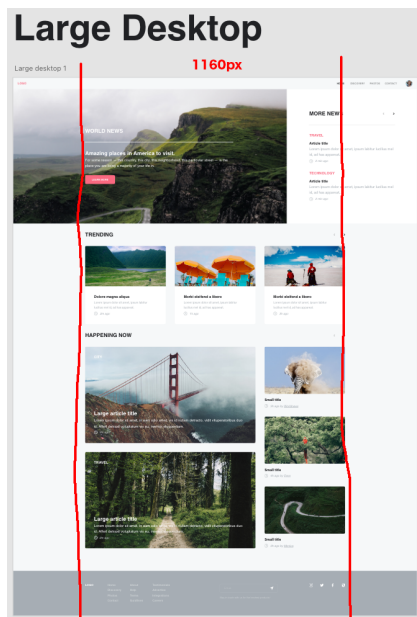
1. <https://www.adobe.com/jp/products/xd/features/ui-kits.html>
2. 「Responsive Design」の「キットを入手」をクリック
3. ダブルクリックで解凍して「xd-resources-responsive-design-ui.xd」を開く

今回はこのデザインの「LargeDesktop1」を利用しましょう。

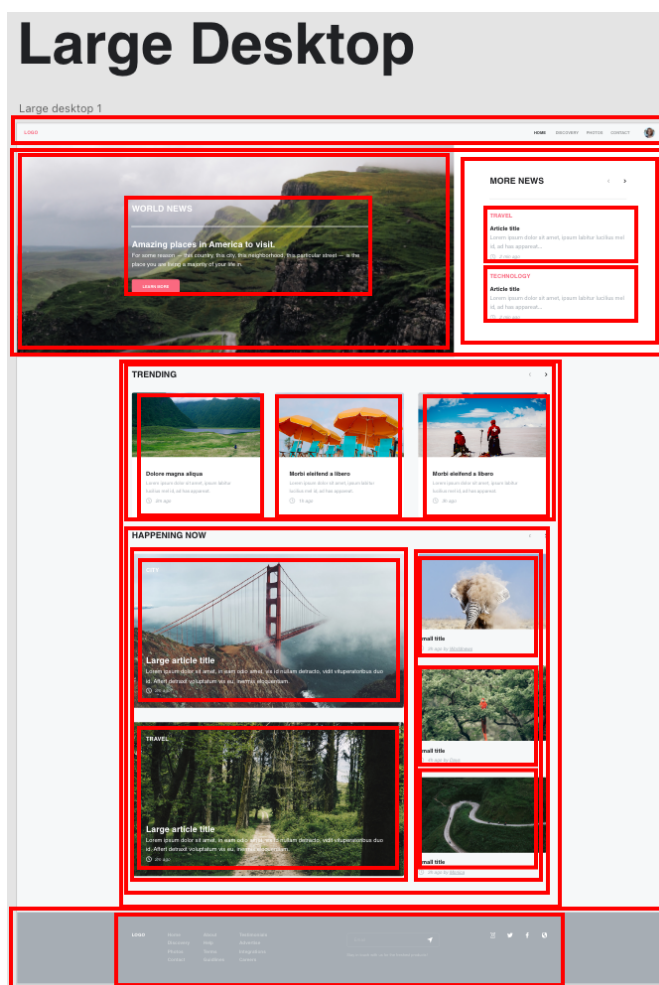
3 HTML の構造を考える

デザインを HTML にコーディングするにあたりテキスト P.141 にあるような構造を考える必要があります。「Large Desktop」と「Desktop」を見比べると、幅が広がっても、固定幅な部分があることに気づきます。

^{*1} 参考: <https://www.otsuka-shokai.co.jp/words/prototype-tool.html>



上の MORE NEWS は「Large Desktop」「Desktop」どちらをみても、このラインからはみ出してますね。WORLD NEWS はラインの中に入ってますね。(ちょっと厄介だなあ...)
これらからレイアウトの構造を考えてみましょう。



こんなふうに見えませんか？(細かいところはブロック分けできてないけど)

4 HTML/CSS でレイアウトしていこう

4.1 HTML の構造 (おおまかな)

今日のフォルダを用意して、large-desktop.html を準備しましょう。

タイピング練習ではないので、絵と見比べてブロックを想定して入力していきましょう。

なお、id に関してもなるべく使わない方向でコーディングしています。^{*2}

ソースコード 2 HTML 部分

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Let's Coding!</title>
8 </head>
9 <body>
10     <div class="contents">
11         <div class="menubar">
12             <h1>Logo</h1>
13             <nav>ナビゲーションエリア</nav>
14         </div>
15         <header>
16             <div class="worldnews">
17                 <h2>WORLD NEWS</h2>
18             </div>
19             <div class="morenews">
20                 <h2>MORE NEWS</h2>
21                 <div class="morenews-content">
22                     <div class="morenews-child">
23                         <h3>Article Title</h3>
24                     </div>
25                     <div class="morenews-child">
26                         <h3>Article Title</h3>
27                     </div>
28                 </div>
29             </div>
30         </header>
31         <main>
32             <section class="trending">
33                 <h2>TRENDING</h2>
34                 <div class="trending-content">
35                     <div class="trending-child"><h3>Trending1</h3></div>
36                     <div class="trending-child"><h3>Trending2</h3></div>
```

^{*2} <https://hsmt-web.com/blog/html-id-class/>

```

37         <div class="trending-child"><h3>Trending3</h3></div>
38     </div>
39 </section>
40 <section class="happening-now">
41     <h2>HAPPENING NOW</h2>
42     <div class="happening-now-content">
43         <div class="happening-now-bigchilds">
44             <div class="happening-now-bigchild"><h3>HN Big1</h3></div>
45             <div class="happening-now-bigchild"><h3>HN Big2</h3></div>
46         </div>
47         <div class="happening-now-smallchilds">
48             <div class="happening-now-smallchild"><h3>HN Small1</h3></div>
49             <div class="happening-now-smallchild"><h3>HN Small2</h3></div>
50             <div class="happening-now-smallchild"><h3>HN Small3</h3></div>
51         </div>
52     </div>
53 </section>
54 </main>
55 <footer>
56     <div class="footer-content">
57         Footer Content
58     </div>
59 </footer>
60 </div>
61 </body>
62 </html>

```

HTML5 Outliner でどのように表示されるか確認しておきましょう。見た目だけでなく、HTML5 としてなるべく綺麗に伝わりやすく書くよう心がけましょう。

4.2 おおまかなレイアウトを CSS Grid で指定してみよう。

まずは、nav,header,main,footer を定義してみよう

ソースコード 3 おおまかな構造の CSS

```

1     <style>
2         * {
3             box-sizing: border-box;
4         }
5         body {
6             margin: 0px;
7         }
8         .contents{
9             display: grid;
10            grid-template-columns: 1fr 1160px 1fr;

```

```

11         grid-template-rows: 50px 600px 1fr 200px;
12         grid-template-areas:
13             "menubar menubar menubar"
14             "header header header"
15             ". main ."
16             "footer footer footer";
17
18     }
19     .menubar {
20         grid-area: menubar;
21     }
22     header {
23         grid-area: header;
24     }
25     main {
26         grid-area: main;
27     }
28     footer {
29         grid-area: footer;
30     }
31 </style>

```

エリアが分かりづらかったら、background-color を設定してみても構いません。DeveloperTools で見てみても構いません。

要素が多いとはみ出したりしていますが、エリアのレイアウトはできていることが確認できます。

- *のところは、幅と高さの計算方法をレイアウトしやすいようにするおまじないです。(幅や高さに境界と内側の余白を含む)
- body のところは、ブラウザのデフォルトのマージンをリセットしています。
- grid-template-areas の ". main ." のピリオドは何もアサインされないことを示しています。

4.3 .menubar のレイアウトしてみよう

H1 大きいし、ナビゲーションエリアがはみ出しているので、とりあえず float で左右に飛ばしましょう。

ソースコード 4 .menubar のレイアウト

```

1     h1 {
2         font-size: 16px;
3         float: left;
4     }
5     .menubar nav {
6         float: right;
7     }

```

4.4 header のレイアウトしてみよう

これから、色とかサイズなどは XD で調べられるところは調べて設定していきます。XD を見ると、「Large Desktop」「Desktop」とも MORE NEWS の方が幅 600px で WORLD NEWS がその残りになってくることがわかります。

ソースコード 5 header のレイアウト

```
1      header {
2          display: grid;
3          grid-template-columns: 1fr 600px;
4          grid-template-areas:
5              "worldnews morenews"
6      }
7      .worldnews {
8          grid-area: worldnews;
9      }
10     .morenews {
11         grid-area: morenews;
12     }
```

4.5 .trending のレイアウトしてみよう

ここって、同じものが3つ並んでるから、Flexbox が使えそうだね。

ソースコード 6 .trending のレイアウト

```
1      .trending-content{
2          display:flex;
3          justify-content: space-between;
4
5      }
6      .trending-child {
7          width: 360px;
8          height: 350px;
9      }
```

4.6 .happening-now のレイアウトしてみよう

ここ、Flexbox/CSS Grid どっち使ってもいけそうなので、Flexbox で書いてみます。左右の縦は、普通に並べると縦に並ぶので特に Flexbox/CSS Grid 使いません。

ソースコード 7 .happening-now のレイアウト

```
1      .happening-now-content {
2          display: flex;
3          justify-content: space-between;
4          align-items: flex-start;
5      }
```



```

6      .happening-now-bigchilds {
7          width: 760px;
8      }
9      .happening-now-smallchilds {
10         width: 360px;
11     }
12     .happening-now-bigchild {
13         height: 430px;
14         margin-bottom: 40px;
15     }
16     .happening-now-smallchild {
17         height: 273px;
18         margin-bottom: 40px;
19     }

```

なんとなくのレイアウトができてきましたね。でも、画像とかなないと寂しいですね。

5 画像

XD では画像を選択して、Command-E とすると PNG 等で書き出すことができます。

作業フォルダの中に img フォルダを選択してその中に Header の画像を header-bg.png として書き出してみましょう。結構 XD で深く掘っていかないと (クリックしてグループ等の中に入らないと) 画像まで辿りつきません。

5.1 header 背景画像, HappeningNow 背景画像

テキスト P.130 あたりを参考に配置してみましょう。幅が大きくなっていいように max() を使って、どちらか大きい方に合わせるようにしています。

ソースコード 8 header 画像のレイアウト

```

1      .worldnews {
2          background-image: url(img/header-bg.png);
3          background-position: center;
4          background-size: max(100%,1220px);
5      }

```

HAPPENING NOW, の左の 2 枚も同様に happening-now-big1.png, happening-now-big2.png として書き出し、配置してみましょう。

HTML の class が上下とも同じで、これでは別々に指定ができないので、hn-big1, hnbig2 をそれぞれクラスに追加しておきましょう。

ソースコード 9 happening-now-big 画像のレイアウト

```

1      .happening-now-bigchilds {
2          background-position: center;
3      }
4      .hn-big1 {
5          background-image: url(img/happening-now-big1.png);

```

```
6      }
7      .hn-big2 {
8          background-image: url(img/happening-now-big2.png);
9      }
```

5.2 Trending 画像、HappeningNowSmall 画像

それぞれ記事に関する画像ですが、普通の順番だと

- 画像
- 見出し

の順になってしまいます。その書き方でも良いですが、あえて今回は

- 見出し
- 画像

の順に HTML を配置し、CSS Grid で順番を変えようと思います。

とりあえず、Trending 画像、HappeningNowSmall 画像を trendingX.png, happening-now-smallX.png として保存しましょう。これらは、画像の元サイズで書き出すのではなく、トリミングされた状態で書き出しましょう。

ソースコード 10 .trending-child の HTML 変更

```
1      <div class="trending-child">
2          <h3>Trending1</h3>
3          
4          <p>Lorem ipsum dolor sit amet, ipsum labitur lucilius mel id, ad has
              appareat.</p>
5          <p class="time">2m ago</p>
6      </div>
```

ソースコード 11 .trending-child のレイアウト

```
1      .trending-child {
2          display: grid;
3          grid-template-areas:
4              "trending-child-img" "trending-child-h3" "trending-child-p" "trending-
              -child-time";
5
6      }
7      .trending-child h3 {
8          grid-area: trending-child-h3
9      }
10     .trending-child img {
11         grid-area: trending-child-img
12     }
13     .trending-child p {
14         grid-area: trending-child-p
```

```
15     }  
16     .trending-child .time {  
17         grid-area: trending-child-time  
18     }
```

同様に残りの部分

- Trending2,3
- HN Small1,2,3

を修正してみましょう。

多分、ここで精一杯かな...

文字などの調整や Footer は来週に回しましょう。

6 まとめ

多少はみ出たりしてレイアウト崩れしているところがありますが、

- フォントサイズ
- マージン・パディング

等がまだ適切に設定されていないからです。

Flexbox/CSS Grid を利用するとレイアウトは非常に明快に指定することができ、大きなレイアウトからどんどん詰めていく、という流れが今日理解してほしいところでした。

以上