

情報デザイン演習I 5.CSS実践

- ボックスモデル
- 3-9 幅と高さを指定しよう
- 3-10 余白を調整しよう
- 3-11 線を引こう
- 3-12 リストを装飾しよう
- IDとクラス
- 3-13 クラスとIDを使った指定方法
- 3-15 デフォルトCSSをリセットしよう
- おまけ `animtate.css`

前回のおさらい

- Chapter 3 Webのデザインを作る！CSSの基本
 - 3-1 CSSとは
 - 3-2 CSSを適用させる方法
 - 3-3 CSSファイルを作ろう
 - 3-4 CSSの基本の書き方を身につけよう
 - 3-5 文字や文章を装飾しよう
 - 3-6 Webフォントを使おう
 - 3-7 色をつけよう

とやりました。

3-8 背景を彩ろう

ざっと紹介するに留めます。

理由は、覚えられないので、何ができるか知っているだけで十分であり、使いたい時に見返したり、検索すれば十分です。

背景関係のプロパティもどんどん増えている気がします。

ボックスモデル

シラバスでは

4. CSSの基本 CSSの基本文法・クラス/ID・継承・ボックスモデルについて学修する。

となっていますが、クラス/ID, ボックスモデルについては触れませんでした。

教科書に入る前に、BOXモデルについて触れます。(クラス/IDは3-13で触れます。)

先に説明してから、教科書の3-9,10,11について説明していきます。(内容は同じこと)

CSSボックスモデルとは？

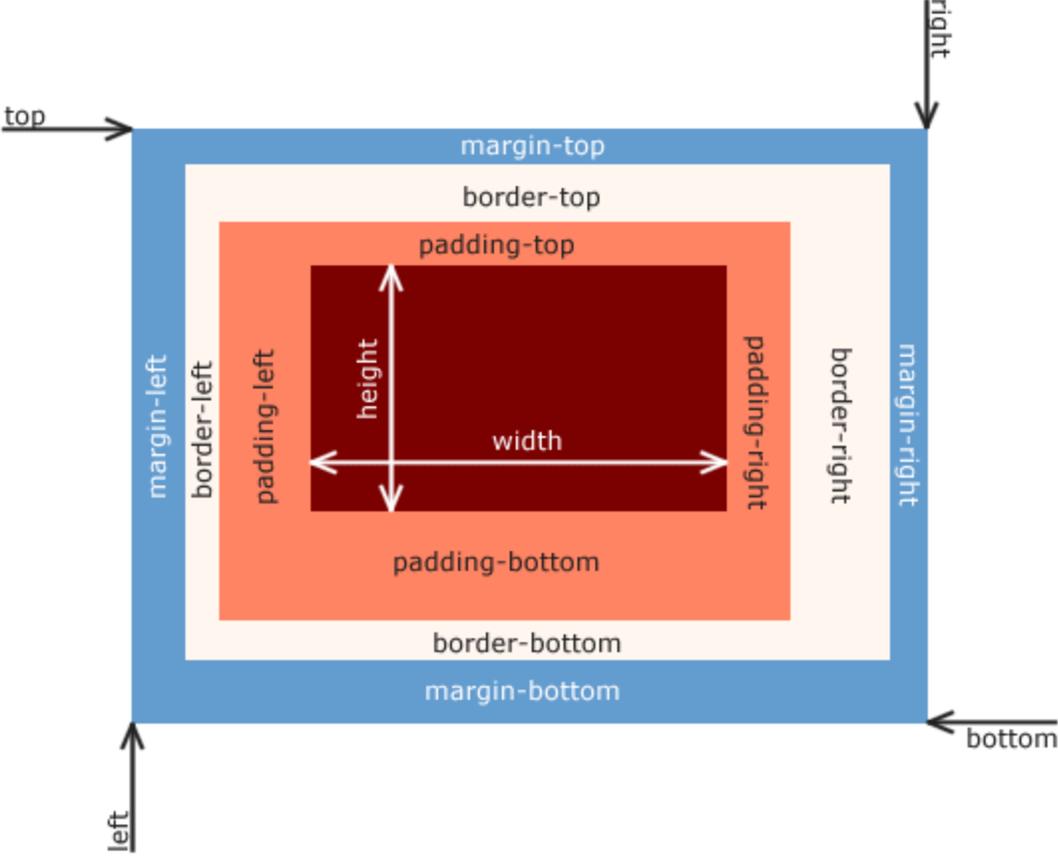
CSSにおける要素とそのデザインに関する考え方

教科書では単語が出てきませんが**ボックスモデル**という考え方は非常に大事です。

CSS では全ての要素にボックス (箱) を想定します。そして、箱の

- コンテンツの大きさ
- 箱の内側の余白
- 箱の枠線 (境界線)
- 箱の外側の余白

という概念を持ちます。



ボックスモデルのプロパティ

コンテンツの大きさ

height,width

箱の内側の余白

padding

箱の枠線(境界線)

border

箱の外側の余白

margin

ボックスモデルの確認

デベロッパーツールを利用するとどの様にボックスモデルが設定されているか確認することができます。

これは、CSS を作成する時には非常に重要となります。

CSSでは

- 文字の色やフォントサイズを指定すること

ももちろん大事ですが

- 要素をどのようにレイアウトするか

が非常に大事だからです。

読み方

読みづらいので...

- `width`
- `height` ...なぜかhiteになってますが、発音はあっています。

box-sizing

幅と高さをどう考えるか、という計算方法は以下の二通りあります。

デフォルトではcontent-boxになっていますが、レスポンシブデザインなどを考える時には、圧倒的にborder-boxにした方が楽になります。

- `box-sizing: content-box;`
paddingとborderの値をwidth,heightに含まない方法
- `box-sizing: border-box;`
paddingとborderの値をwidth,heightに含む方法

ある、ということだけ覚えておきましょう。

box-sizingの指定方法

```
* {  
  box-sizing: border-box  
}
```

と記載することで、HTMLの全ての要素に適応させることができます。

subgrid

まだ全然早いです、2023/9月末にsubgridというプロパティも利用できるようになっていきます。

忘れると嫌なので、ちょっとここで触れておきます。

- CSSのsubgridがすべてのブラウザにサポートされたので、カード内の見出しやテキストを簡単に揃えることができます

3-9 幅と高さを指定しよう

サイズを指定しよう「widthプロパティ」「heightプロパティ」

widthが幅で、heightが高さでしたね。

死ぬほど使います。このスペル、読み方は忘れないように。

やってみよう

- **ID_ROOT**の中に**ID_05**として今日の作業フォルダを作成しよう
- その中に**c3-09**フォルダを作成しよう
- P.130の右のHTMLを**c3-09-1.html**として作成しよう
- **style.css**を作成してP.130の左のCSSを記述してみよう。

htmlの書き方復習です。style.cssにリンクするやり方はわかるかな？

h1, h2, pはdivタグの中に入れてみましょう。

わからなかったら、4回目のコードをみましょう。

c3-09-01.html

bodyタグの中は以下のようになります。適当な文章を入力するときは「lorem20」とかでしたね。

```
<div>  
  <h1>猫の1日</h1>  
  <h2>ひたすら寝ています。</h2>  
  <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Repellat </p>  
</div>
```

幅をautoと指定すると

```
div {  
  width: auto;  
}
```

の意味です。これは、親要素と同じ大きさを意味し、

```
div {  
  width: 100%;  
}
```

と同じ意味になります。

style.cssを変えてみよう。

継承について

```
div {  
  width: 500px;  
  width: auto;  
  width: 100%;  
}
```

のように記載すると、**width:100%**が有効になります。後から書いたものが有効だからです。インスペクタでもみてみましょう。

Webサイト上で使う単位とは？

- 絶対単位：他の要素に影響を受けないして方法
 - px 最も一般的な単位
- 相対単位：ブラウザの表示領域や親要素等のサイズを基準にして指定する方法
 - % 親要素を基準とした問いの単位
 - em 親要素のフォントサイズを基準にした単位(印刷業界で用いられていた単位で、大文字の M を基準としたサイズ表記が由来)
 - rem ルート要素(html)のフォントサイズを基準にした単位(root+width)
 - vw ビューポート(Webサイトを閲覧している時の表示領域)の幅を基準とした単位で 1/100(viewport width)
 - vh ビューポートの高さを基準とした単位で、 1/100(viewport height)

使える単位もまだまだ増えています。

- **【CSS】** ビューポート単位が新しく追加されました。

なぜこんなにいっぱい単位があるの？

簡単に言えば、楽になるからです。

どれを使うかの正解はありませんが、人によって使い分けをしている人もいます。

- **【px・em・rem・%】 CSS単位の使い分けについて考察**

とはいえ、最初はpx, %, emあたりから利用していけばいいと思います。

3-10 余白を調整しよう

余白の概念

ボックスモデルの説明でも触れましたが、

- margin: 要素の入れ物から他の入れ物までの余白
- padding: 要素の入れ物から中身(コンテンツ)までの余白

の二つがあります。

CSSでの書き方

上下左右全てに指定する場合には

```
div {  
  margin: 20px;  
  padding: 10px;  
}
```

のように記述します。

上下を20px, 左右を10pxにしたい場合には

```
div {  
  margin: 20px 10px;  
  padding: 20px 10px;  
}
```

のように記述します。(もちろん、marginとpaddingの値は違っていても構いません。単位も混在できます)

CSSでの書き方

上下左右全てバラバラに例えば上：10px、右：20px、下：30px,左：40pxにする場合には

```
div {  
  margin: 10px 20px 30px 40px;  
  padding: 10px 20px 30px 40px;  
}
```

の様に記述します。(もちろん、marginとpaddingの値は違っていても構いません。単位も混在できます)

スペースをはさんで、上から時計回りの順に指定すると覚えれば良いでしょう。

二つの場合には、上、右->上下、左右に読み替えばいいかな。

CSSでの書き方

一つ一つ個別に指定する場合には、専用のプロパティがあり

```
div {  
  margin-top: 10px;  
  margin-right: 20px;  
  margin-bottom: 30px;  
  margin-left: 40px;  
  padding-top: 10px;  
  padding-right: 20px;  
  padding-bottom: 30px;  
  padding-left: 40px;  
}
```

と記述できます。

今は知らなくていいけど

「マージン相殺」という言葉があるので、触れておきます。

- [CSSにおけるマージンの相殺を徹底解説](#)

簡単にいうと、上下のmarginに関しては、数字の大きい方が有効ということです。

余白を使ってグループ化する

(参考)デザインの4原則

4原則は「近接」「整列」「反復」「対比」ですが、

近接：関連する要素をグループ化し、整理することで情報を関連づける
をレイアウトするには余白(margin)が必要となります。

枠と文字の間に余白を作ろう

余白がないと読みづらくなりますが、この場合の余白は(padding)となります。

3-11 線を引こう

教科書ではborder-width,border-style,bordercolor

の順で説明してから

- border

を説明していますが、個人的にはborderだけを使うことが多いです。

そして、上下左右の特定の値を変えたい時には

border-top, border-right, border-bottom, border-left

を利用しています。

borderの例

例えば、見出し部分を目立たせたいな、というときに

```
h2 {  
  border-left: 10px solid black;  
  border-bottom: 1px solid black;  
  padding-left: 10px;  
}
```

とか簡単によく使います。

3-12 リストを装飾しよう

ざっと教科書に沿って説明します。

```
ul {  
  list-style-type: none;  
}
```

くらいしか自分はありません(メニューでよく使います。)

IDとクラス

非常に大事な考え方です。しっかり押さえましょう。

HTMLタグでclass,idを指定し、それに対してCSSで装飾をつけることができます。

クラス,IDの違い

- どちらもHTMLのタグを分類することができます。
- クラスによって同じHTMLタグを分けて指定することができるようになります。
- IDはページ内で同じID名を使うことができません。
- クラスはページ内で同じクラス名を複数回利用することができます。

クラス名,ID名のルール

- 空白を入れない
- 英数字とハイフン「-」、「_」を使う
- 一文字目は必ず英字(小文字が安全)

HTML・CSSでのclass,idの表記方法

HTMLでは

```
<p id="important">段落</p>  
<p class="boy">段落</p>
```

の様に記述して指定します。CSSでは

```
#important {  
  
}  
.boy {  
  
}
```

のように、IDは「#」、クラスは「.」を使って記述します。

CSSでのタグとID or クラスの表記方法

```
p#id {  
  
}  
p.boy{  
  
}
```

のように記述することで、

pタグで、IDがid, またはクラスがboy

の様に指定することもできます。

クラスとIDの優先順位

クラスの指定よりIDの指定が優先されます。

IDによるページ内リンク

```
<a href="#top">トップに戻る</a>
```

のような書き方で、「id="top"」と指定されたところにリンクすることができます。

aタグの擬似クラス

ちょっと脱線しますが、aタグでは以下の様にして状態に応じてCSSを設定することもできます。

```
a {  
    /* aタグの装飾 */  
}  
a:hover {  
    /* マウスをオーバーした時 */  
}  
a:link {  
    /* リンク先に未訪問の時 */  
}  
a:visited {  
    /* リンク先が訪問済 */  
}
```

下線を消したい時には「text-decoration:none;」をaタグに、色を変えたいときはそれ以外の指定方法を使うことが多いです。

やってみよう

男の子と女の子の会話をちょっとステレオタイプ(色を青と赤)に作成してみましょう。

- c3_id_classフォルダをID_05の中に作成
- index.html, style.cssを作成
- index.htmlからstyle.cssへのリンクを作成
- bodyタグの中は
 - h1で見出しを作成、idをtopとしよう。
 - pタグでクラスがboy, girlのものを作成して、ページがスクロールするくらいコピーしよう。(文はlorem10とかでOK)
 - 最後にaタグでtopに戻る様に作成しよう。

やってみよう

- h1は

```
border-left: 10px solid gray;  
border-bottom: 1px solid gray;  
padding-left: 10px;  
color: #333;
```

- pの幅は30%に。
- boyクラス、girlクラスは文字色はそれぞれblue, redにして、周りを「1px dashed black」で囲もう
- 詰まっているので、pタグのpaddingを5pxにしよう。
- girlクラスは左のマージンを70%に設定しよう。
- aタグは文字色を白、背景を黒、下線を無くして、paddingは10pxにしよう。

おまじない

```
html {  
  scroll-behavior: smooth;  
}
```

とすると、ページ内リンクをクリックしたときに、アニメーションしながら動きます。

変なところないかな...

ページの左に余白あるけど、右に余白ないね。

30%と70%で100%だから右と左同じになりそうなのに。

なんでだろう。

Webインスペクタで考えてみよう。

答え

bodyのmarginが8pxになっているから

これは、ブラウザのデフォルトでそうになっています。

3-13 クラスとIDを使った指定方法

教科書を使ってもう一度復習してみましょう。

3-15 デフォルトCSSをリセットしよう

ブラウザはデフォルトである程度パラメータが決まっているのだけれど、これがブラウザによって異なるため非常に厄介となります。

それなら、設定をリセットするCSSを作ればいいじゃないか、ということで

- リセットCSS
- ノーマライズCSS

と呼ばれる方法が使われています。

どうすればいい？

CSSの機能が拡張するため、

リセットCSS 2023

等と調べるのがおすすめです。

- [2023年、現在の環境に適したリセットCSSのまとめ](#)

教科書では「[ress.css](#)」を推奨してる様ですが、このリストにありません(汗)

- [【2023年版】おすすめのリセットCSS \(reset.css\) 3選【CDNのコピペあり】](#)

こっちにはありました。そんなものです。

おまけ animate.css

CSSで簡単にアニメーションさせるライブラリがあります。

- [animate.css](#)

時間があまりそうなら、これを使って遊んでみましょう。

今日はこんなものかな？

2,3章は大体終わって

- 2-13 ブロック要素でグループ分けをしよう
- 3-14 レイアウトを組もう

はセットで、7回目に教える予定です。

- 3-16 よく使うCSSプロパティ一覧

は、参考資料扱いなので、必要な時にみましょう。

来週の予定

来週は、少し教科書離れた内容で

- セマンティックコーディング
- これまでの内容の演習

をやろうと思います。

お疲れ様