

情報デザイン演習I 4.CSS基本

1. CSSの基礎

- i. Chapter03-01 CSSの基本文法
- ii. Chapter03-02 CSSを記述する場所とCSSの検証について
- iii. Chapter03-03 CSSの「継承」と「ボックスモデル」の考え方
- iv. Chapter03-04 ブラウザーのデフォルトCSS

前回のおさらい

教科書ベース+αでやってみました。

- Webデザインの基礎知識
- HTMLの基礎
 - HTMLの基本文法
 - 見出しと段落を指定
 - 画像の種類と特性について
 - データの位置
 - 画像を配置する指定
 - 他のページにリンクする指定
 - 同じページ内のリンクを指定

今日やること

- Chapter 3 CSSの基礎(P.87-136)

までやっていこうと思います。

テキストの重要ポイントを説明しながら適宜補足していきます。

CSSの基礎

Chapter03-01 CSSの基本文法

CSSとは(P.88-)

HTMLとCSSの役割分担

- 構造(HTML)
- 見た目のデザイン(CSS)

を明確に役割分担することはWebデザインにおいて非常に大事な考えとなります。

CSS:Cascading Style Sheet

日本語にすると「連鎖したスタイルシート」とでもなるでしょうか？

スタイルシートとは

文書データの見栄えに関する情報のみを記録・定義したデータやファイルなどのこと

Word, Illustrator, InDesignなどのアプリでも存在しています。

スタイルシートが別だと

- スタイルシートを置き換えるだけで全く別のデザインにすることが可能
- 制作時にも複数名で役割分担が可能

Cascading: 「連鎖した」とは？

```
<body>  
  <p>おはよう</p>  
</body>
```

bodyタグの中にpタグがあったとしましょう。

その時に、bodyタグの方に「フォントの色を赤にして」とCSSで指定してあげると、その指定がpタグの中にまで影響することを意味します。

CSS2.1とCSS3

バージョンについて詳しく知らなくても問題ありませんが、CSS2.1を基準として利用できることがモジュール(機能)単位で増えているのが現状のCSS3ということになります。

現在何が使えるかは

<https://caniuse.com/>

で確認するようにすると良いでしょう。

CSS3発表当初の機能

- 変形が可能
- 透明度を操作可能
- グラデーションを作成可能
- 角丸の表現が可能
- 影の指定が可能
- アニメーションが可能

等表現力の向上がCSS3により可能となりました。

CSSの新機能

次々に新しい機能が搭載されています。

このテキスト発行以降にでてきたCSSの新機能のうち重要なものとして

- CSS 数学関数
- CSS grid
- CSS flexbox

があるかと思います。次回以降で触れます。

Web業界で働く人は敏感でないと簡単に取り残されていきます。

演習1

ちょっと教科書から離れてCSSを試してみましょう。

- 「ID_root」 - 「ID_04」 フォルダを作成しましょう。
- 「ID_04」 の中にfirst_css.htmlを作成しましょう。

first_css.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>初めてのCSS</title>
</head>
<body>
  <div>
    <h1>初めての CSS</h1>
    <p>この味気ない文章を加工していくよ!<br>どの様にな変わっていくかな?</p>
  </div>
</body>
</html>
```

インデントに気をつかって入力していこう。インデントはTABキーでつけられたね。

divタグ

初めて使うかな。デザインの的な表現をアシストするためのタグです。

インラインCSS

次のように変えてみましょう。

```
<h1 style="color:#8800ff;">初めての CSS</h1>
```

紫色になりましたね。

色の表記について(P.126)

HTMLでは基本的には16進数を使ってRGBの数値を指定して色を表現します.

#8800ffとは「#」が16進数で表すよ、という宣言で

- 88 R(Red)の値
- 00 G(Green)の値
- ff B(Blue)の値

16進数では0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,fの16個の文字を使って数字を表す方法で、

- #000000 黒
- #ffffff 白

となります。Photoshop,Illustratorの色表記からも選べますし、次を参考にしましょう。

[Adobe Color](#)

色の表記について2(P.126)

他にも次の方法で指定することもできます。

キーワード

「black」 「red」 「blue」 など

RGB

「rgb(255, 42, 53);」 「rgba(255, 42, 53, 0.5);」 のように指定。透過度も指定できる

HSL

「hsl(0, 100%, 100%);」 「hsla(0, 100%, 100%, 0.5);」 のように指定。透過度も指定できる

インラインCSS

次のように変えてみましょう。

```
<h1 style="color:#8800ff;font-size:60px;">初めてのCSS</h1>
```

文字大きくなりましたね。

P.113-126にテキストとフォントについては詳しく書いてあります。

CSSの表記方法

CSSの表記方法は

```
(CSSプロパティ):(値);
```

複数であれば

```
(CSSプロパティ):(値);  
(CSSプロパティ):(値);  
...
```

となります。コロン・セミコロンの違いに気をつけましょう。

インラインCSSとは？

インラインCSSという言葉を送りげなく使ってきました。
HTML タグの中に style 属性を使って記載する方法です。

お勧めしません

理由としては、一つ一つに定義しないといけないからです。

h1 が複数あったら、全てのタグに同じ style 属性をつけないといけないですよ。そして、変更があったら全てを変更しなくてはならない、ということを意味します。

内部スタイルシート

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>初めてのCSS</title>
  <style>
    h1 {
      color:#8800ff;
      font-size:60px;
    }
  </style>
</head>
<body>
  <div>
    <h1>初めてのCSS</h1>
    <p>この味気ない文章を加工していくよ!<br>どの様にな変わっていくかな?</p>
  </div>
</body>
</html>
```

内部スタイルシートとは？

head 内に style タグで記載する方法です。

これも基本的にはお勧めしません

なぜでしょう？

あらためてCSSとは？

HTMLとCSSの役割は明確に異なり、「文書構造」「見た目」を分離すると伝えました。なのに、同じファイルで記載してはその意味が薄れてしまいます。

- htmlファイル「～.html」
- cssファイル「～.css」

としてファイルを別にする事で、明確に分離することができます。このことにより、制作分担することが可能となります。

HPは大体複数ページでできている。

ランディングページ等1ページで作成されるHPもありますが、多くの場合複数ページから構成されています。

この場合、前述したようにCSSファイルを外部に置くことで、全てのページの「見た目」を統一することが可能となります。

デザインに一貫性を持たせることを「トーン&マナー」ともいい、CSSはそれの実現に非常に役に立ちます。

外部スタイルシートを使うには...

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>初めてのCSS</title>
  <link rel="stylesheet" type="text/css" href="first_css.css">
</head>
<body>
  <div>
    <h1>初めてのCSS</h1>
    <p>この味気ない文章を加工していくよ!<br>どの様にな変わっていくかな?</p>
  </div>
</body>
</html>
```

外部スタイルシートを作ろう

新規に「first_css.css」を作成して以下のようにしてみよう。

```
h1 {  
  color: #8800ff;  
  font-size: 60px;  
}
```

ライブプレビュー時の注意

CSSファイルを選んで「Go Live」をしても、見ることはできません。
ブラウザは何を表示してよいかわからないからです。

- htmlファイルからlinkタグで外部スタイルシートを指定
- htmlファイルを「Go Live」

とすることに気をつけましょう。

CSS 記述場所まとめ

インラインCSS

bodyタグ内の要素に直接style属性で指定

内部スタイルシート

headタグ内のstyleタグの中に指定

外部スタイルシート

別のcssファイルを作成し、linkタグによって指定

CSSの基本文法を理解する

P.90に戻りましょう。

プロパティ 値については説明しました。

セレクタという概念があります。

先ほど、h1 というセレクタ を用いましたが、ここをクリアするには「ID」「クラス」という概念が必要になってきます。

ID

前回、「同じページ内のリンクを指定」(P.60) で ID を設定しましたね。そして、#をつけてリンクすることを確認しました。

これと同じでCSSでも次のように指定できます

```
<style>
  h1#s1{
    color: #8800ff;
  }
</style>
...
...
<h1>影響を受けない見出し1</h1>
<h1 id="s1">影響を受ける見出し1</h1>
```


IDの特徴

IDはHTML文書の中に1度だけ現れます。

```
<h1 id="s1">影響を受ける見出し1</h1>  
<h1 id="s1">影響を受ける見出し1</h1>
```

このような書き方は禁止されています。

それに対して、クラスはHTML文書の中で複数使うことができます。

クラス

```
<p class="boy">男の子の会話1</p>  
<p class="girl">女の子の会話1</p>  
<p class="boy">男の子の会話2</p>  
<p class="girl">女の子の会話2</p>
```

CSSでのクラス指定

```
<style>
  p.boy{
    color: blue;
  }
  p.girl{
    color: red;
  }
</style>
```

このようにピリオドを使って指定します。

- IDは#
- クラスは.

やってみよう first_class.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>初めてのクラス</title>
  <style>
    p.boy{
      color: blue;
    }
    p.girl{
      color: red;
    }
  </style>
</head>
<body>
  <div>
    <h1>初めてのクラス</h1>
    <p class="boy">今度ご飯行こうよ</p>
    <p class="girl">えー、何食べるの?</p>
    <p class="boy">ケーキ</p>
    <p class="girl">コロナで自粛してるからいけないー。ご飯じゃねーし</p>
  </div>
</body>
</html>
```

CSSのセレクト

P.91の上にあるように指定方法はいろいろあります。

- タイプセレクト
- IDセレクト
- クラスセレクト

をよく使います。divの状況によっては

- 子孫セレクト(自分の範囲の中全て)
- 子セレクト(直下を指定)

も使います。

リンクのセレクト

a タグの a: hover, a: active とかはよく使います。(P.93)

マウスが

- リンクの上に来た時
- ボタンを押した時

によってデザインを変更することがよくあるからです。

Chapter03-02 CSSを記述する場所とCSSの検証について

CSSを記述する場所(P.88-)

さっきやりましたね。

3種類あることを確認しましょう。(P.95上)

@import

サイトが大きくなると、CSSファイルを分割したくなることがあります。

そんな時には、

```
@import url();
```

などをCSSの冒頭に書くことで分割することができます。

検証

HTML や CSS はエラーが多少あってもたまたまうまく動作することがあります。バリデーションサービスがあるということを押さえておきましょう。(P.95)

ただ、CSSは進化が早いため、警告が多く出る傾向に現在はありま

[CSS Validation Service](#)

実習課題06(P.97)

やってみましょう。

- <https://www.scc-kk.co.jp/scc-books/b-408.html> からダウンロードしましょう (前回してますね)。
- Chapter3のTraining6を「ID04」にコピーしましょう。
- Sample-06-Complete.html, Sample-06-Complete.css を削除しましょう。
- Sample-06.htmlから2つのファイルを作成して、正しくCSSファイルにリンクを貼ってみましょう。

Chapter03-03 CSSの「継承」と「ボックスモデル」の考え方

カスケーディングスタイルシートとは(P.99)

先ほども「連結した」について説明しましたが、テキストで抑えておきましょう。

やってみよう cascading.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>継承の確認</title>
</head>
<body>
  <p>
    富士山は<em>日本で一番高い</em>山です
  </p>
</body>
</html>
```

headタグ内に追加しよう

```
<style>
  body {
    font-size:24px;
  }
  p{
    color: #888888;
  }
  em {
    color: #ff0000;
  }
</style>
```

どうなるか？

富士山は・山です

bodyを継承し「font-size:24px;」が設定され、pで定義された「color:#888888;」となっている

日本で一番高い

bodyを継承し「font-size:24px;」が設定され、pで定義された「color:#888888;」となっているが、emの「color:#ff0000;」の方に上書きされている。

Chromeのデベロッパーツールで確認しよう

- 表示 - 開発/管理 - デベロッパーツール(Option+Command+I)で表示させて、
- ElementsのStyle

を確認してみましょう。

- Elements で「<p>」を選ぶと Styles で「Inherited from body」として 24px が継承されているのがわかります。
- Elements で「 日本で一番高い 」のところを選択すると、 Styles で「Inherited from p」と書いてありますが、打ち消しされていて、em の設定に上書きされていることがわかります。

継承理解できたかな？

- 基本的には、上位の要素のスタイルが下位に引き継がれますが、下位で上書きできることがわかったでしょうか？
- これが、CSS(Cascading Style Sheet)..."連鎖する"スタイルシート、の意味となります。
- 下位の要素で指定された CSS の方が有効となり、以前の部分の指定は打ち消されます。

CSSファイルのどこで書いたかは関係なく、要素の上位・下位という関係性に影響があることを抑えておきましょう。

ただし、同じセレクトで2度指定した場合には2回目が有効となります。

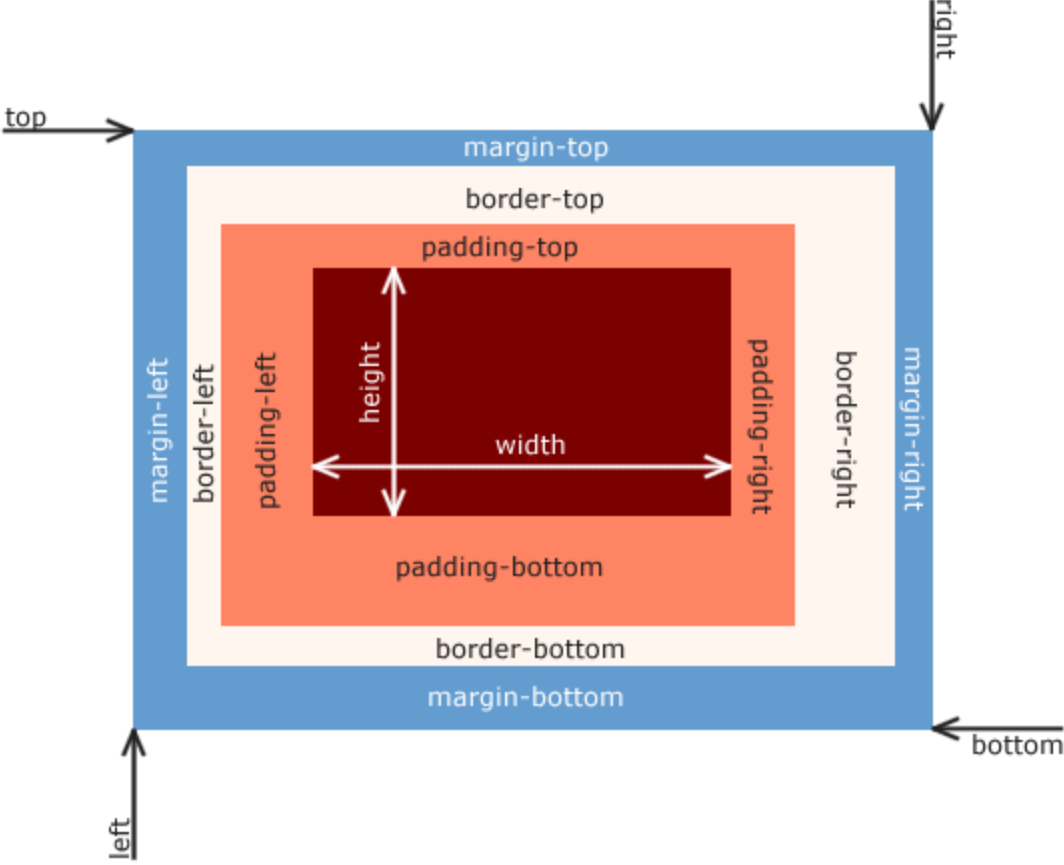
CSSボックスモデルとは？ (P.100)

CSSにおける要素とそのデザインに関する考え方

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

box-sizing

ボックスモデルの便利さに関しては、次回以降やっていきますが、CSS制作者にとって、少しめんどくさいものでもありました。そのため、width,heightの定義を少し変えるやり方が出てきました。(P.249)

- 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。

実習課題07

P.100から少しおさらいしてから

実習課題No.07に取り組みましょう。

Chapter03-04 ブラウザーのデフォルトCSS

まずはやってみよう

- 「ID_04」の中に「layout.html」「layout.css」というファイルを作成しよう
- bodyタグの中にheaderタグ・mainタグ・footerタグを配置しよう
- layout.htmlからlayout.cssにリンクしよう
- それぞれ以下のように設定してみよう
 - header 高さ:100px, 背景色: red;
 - main 高さ:400px, 背景色: green;
 - footer 高さ:50px, 背景色: blue;

背景色は次のように設定してみよう

```
background-color: red;
```

layout.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>継承の確認</title>
  <link rel="stylesheet" type="text/css" href="layout.css">
</head>
<body>
  <header>
    ヘッダー部分
  </header>
  <main>
    コンテンツ部分
  </main>
  <footer>
    フッター部分
  </footer>
</body>
</html>
```

layout.css

```
header{  
    height: 100px;  
    background-color: red;  
}  
main {  
    height: 400px;  
    background-color: green;  
}  
footer {  
    height:50px;  
    background-color: blue;  
}
```

なんか隙間があるね...

簡単なレイアウトを試してみたわけだが、
なんか隙間があるね。

デベロッパーツールでなぜ隙間ができるのか原因を調べてみよう。

borderのmargin

8になってるかな？

これを治すにはどうすれば良いだろう？

layout.cssに追加

```
body {  
    margin: 0px;  
}
```

あれ？

元々のmargin:8px;って誰が指定したんだろう？

これが、デフォルトCSSと呼ばれるもので、

ブラウザが元々持っているCSS

となります。

レンダリングエンジンによって、別々のCSSが元から定義されていることになります。よって、ブラウザによって見え方がバラバラになってしまいます。

これを回避するために

- リセットCSS
- ノーマライズCSS

という考え方が出てきました。

リセットCSS/ノーマライズCSS

リセットCSS

とりあえず一度全部ゼロに設定して、全てのブラウザで同じように見えることを保証した上で、上書きしてデザインしていこう、という考え方

ノーマライズCSS

見出しフォントの大きさなどデフォルトCSSを残しながらもブラウザの差異をなくしていこうという考え方

教科書に掲載のものを利用しても問題はありませんが、この辺は時代と共に進化する技術ですので「リセットCSS 2022」等とググって新しいものを取り入れていきましょう。

Chapter3の01-07までとりあえず触れましたね。

Chapter08は時間があれば...背景画像についてのCSSの説明なので知りたい時に見れば構いません。

というわけで、P.136まで終わったものとします。

余裕がある人は実習課題09,10,11もやってみましょう。

わからない人は質問受け付けます。

終わり

ID_04を圧縮したzipファイルをmanabaに提出してください。