

Web講習会2021

ワールドワイドウェブ基礎

第3回: CSS (1)



Arthur

この回の目標

- セレクタの記法について知る
- CSSによる装飾の適用方法を理解する
- CSSについて自習するときの注意点を知る

※今日からの内容はそれなりに難しいかもしれません。

動画の視聴だけでなく、**実際にコーディングすること**を強く推奨します。

前回の復習

HTML

= HyperText Markup Language



視覚表現や文書構造を記述する言語

視覚表現は画像(と改行)だけ?

(旧)ブロックレベル要素
のデフォルトスタイル

段落

段落1です。こんにちは。

段落2です。こんにちは。

見出し

好きなもの

果物

キウイです。

リスト

- りんご
- みかん
- ぶどう

番号付きリスト

1. りんご

[再掲] 次回予告

これらの要望や疑問にお答えします

- 色やフォントを変えたい!
- 元インライン要素だけどブロックレベル要素のように改行を入れたい!
- divはまだしも、spanなんて何の使いみちがあるの?



CSSとは



styleタグ

head内でstyleタグを使うと、
文書の装飾ができる

見た目は文書の本体ではなく、
メタデータ

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>helloworld</title>
    <style>
      .....
    </style>
  </head>
  <body>
    <p>Hello, World!</p>
  </body>
</html>
```

装飾用の言語で記述

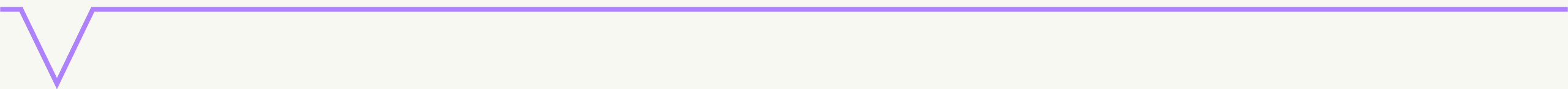
CSS

CSS = Cascading Style Sheets

Webページのスタイルの指定のために主に使われる言語



CSSの文法



CSSの文法

セレクタ

どの要素に装飾を適用するか

p

{

color: red;

background: yellow;

}

プロパティ名

装飾の種類を示す

値

プロパティに対して、具体的な色や数値などを決める

すべての段落について:

文字色を赤にする

背景色を黄色にする

CSSをHTMLに組み込む

styleタグの中に、先程のコードを書く

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>helloworld</title>
    <style>
      p {
        color: red;
        background: yellow;
      }
    </style>
  </head>
  <body>
    <p>Hello, World!</p>
  </body>
</html>
```

Hello, World!

コメント

/* */で囲んだ内容は無視される
C言語系と同じだが、// は使えない

```
/* 段落に対する装飾 */  
p {  
    color: red; /* 文字を赤く */  
    background: yellow; /* 背景を黄色く */  
}
```

基本のセレクタ

要素型セレクタ・classセレクタ・idセレクタ・全称セレクタ・子孫セレクタ

要素型セレクタ

セレクタ … どの要素に装飾を適用するかを示す
CSSだけでなく、JavaScriptなどでも同じ記法

要素型セレクタ → **タグ名**

タグ名で指定したすべての要素を指す

```
p {  
  color: red;  
}
```

```
<p>  
  赤い段落です。  
</p>
```

赤い段落です。

グローバル属性: class

※このページはHTMLの話題です

グローバル属性

どの要素でも使用できる属性

class属性

要素のクラスを空白区切りで列挙
分類

```
<p class="warning">  
  注意事項です！  
</p>
```

[再掲] 要素

開始タグ

終了タグ

`<p class="greeting">Hello, World!</p>`

タグ名 属性名 属性値

タグ名

このひとかたまりを要素という
属性名と属性値はオプション

classセレクタ

classセレクタ → .クラス名

指定したクラスを持つ全ての要素を指す

要素型セレクタとの併用も可能 ex) `p.warning`

`p` かつ `.warning`

```
.warning {  
  color: red;  
}
```

```
<p>  
  新プランは月額3000円！  
</p>  
<ul>  
  <li class="warning">7ヶ月目以降は月額7000円</li>  
  <li class="warning">税別です</li>  
</ul>
```

新プランは月額3000円！

- 7ヶ月目以降は月額7000円
- 税別です

グローバル属性: id

※このページもHTMLの話題です

id属性

要素の一意識別子

classと異なり複数列挙は不可

id名が同文書内で被ってはいけない

```
<h2 id="favorite_food">  
    好きな食べ物  
</h2>
```

ページ内リンク (前回の練習問題3) にも利用される

idセレクトタ

idセレクトタ → #id名

指定したidを持つ要素(高々1つのはず)を指す

要素型セレクトタとの併用も可能 ex) `h2#favorite_food`

```
#favorite_food {  
  color: red;  
}
```

```
<h2>  
  好きなゲーム  
</h2>  
<h2 id="favorite_food">  
  好きな食べ物  
</h2>
```

好きなゲーム

好きな食べ物

汎用要素の使い道

`div`や`span`は、HTMLの文書構造として特別な意味を持たないかたまりに対して装飾を適用したいときに利用できる

```
.text-red {  
  color: red;  
}
```

```
<p>  
  <span class="text-red">赤色</span>が好きです。  
</p>
```

赤色が好きです。

全称セレクタ

全称セレクタ → *

すべての要素を指す

```
* {  
  color: red;  
}
```

```
<h1>Hello, World!</h1>  
<p>The world would be dyed red.</p>
```

Hello, World!

The world would be dyed red.

セレクトタの列挙

セレクトタは , で列挙できる

複数のセレクトタに対して同じ装飾を適用したいときに便利

```
h1, h2 {  
  color: red;  
}
```

```
<h1>大見出し</h1>  
<h2>小見出し</h2>
```

大見出し

小見出し

子孫セレクタ

半角スペース!

子孫セレクタ → 親のセレクタ 注目要素のセレクタ
ある要素の子孫(内部)の要素を指す

```
div span {  
  color: red;  
}
```

```
<div>  
  春は<span>あけぼの</span>  
</div>  
<div>  
  <p>  
    夏は<span>なんだっけ?</span>  
  </p>  
</div>
```

春はあけぼの

夏はなんだっけ?

装飾の継承と上書き

継承 (inherit)

先祖の要素に指定された装飾は(だいたい)子孫に引き継がれる

```
div {  
  color: red;  
}  
span {  
  background: yellow;  
  /* color red; と書いてないけど… */  
}
```

```
<div>  
  Hello. <span>How are you?</span>  
</div>
```

Hello. How are you?

継承イメージ図

- 赤い文字

```
div {  
  color: red;  
}
```

```
<div>  
  Hello. <span>How are you?</span>  
</div>
```

- (赤い文字)
- 黄色い背景

```
span {  
  background: yellow;  
}
```

※(hoge): inherited style

継承の上書き

継承先で、同じプロパティの装飾が指定されていると、上書きされる

```
div {  
  color: red;  
}  
.bg-yellow {  
  background: yellow;  
}  
.text-blue {  
  color: blue;  
}
```

```
<div>  
  Hello.  
  <span class="bg-yellow">  
    How  
    <span class="text-blue">are</span>  
    you?  
  </span>  
</div>
```

Hello. How are you?

上書きイメージ図

- 赤い文字

```
div {  
  color: red;  
}
```

- (赤い文字)
- 黄色い背景

```
.bg-yellow {  
  background: yellow;  
}
```

```
.text-blue {  
  color: blue;  
}
```

```
<div>  
  Hello.  
  <span class="bg-yellow">  
    How  
    <span class="text-blue">are</span>  
    you?  
  </span>  
</div>
```

- ~~(赤い文字)~~
- 黄色い背景
- 青い文字


装飾の競合と詳細度

セレクタの多様性

ある要素を指すセレクタは無数に存在する

[例]

```
<div>
  Hello.
  <span class="bg-yellow">
    How
    <span class="text-blue">are</span>
    you?
  </span>
</div>
```



- span
- .text-blue
- span.text-blue
- span span
- div span
- body .text-blue
- *

...

装飾の競合

同じ要素に対して複数の装飾が適用されていたらどうする？

※先程の「**継承・継承の上書き**」とは異なる概念であることに注意

```
* {  
  color: black;  
}
```

```
.text-blue {  
  color: blue;  
}
```

```
span {  
  color: red;  
  background: yellow;  
}
```



```
<span class="text-blue">What is my color?</span>
```

😄 プロパティが競合していない
→ 統合すればよい

😞 プロパティが競合している
→ 2つのルールで競合を解決

- 詳細度
- Source Order

color: ???; 😞
background: yellow; 😄

詳細度

詳細度 … セレクタの戦闘力を表す

セレクタを構成する以下のセレクタにつき、詳細度が高まる

idセレクタ → 100

classセレクタ → 10

要素セレクタ → 1

全称セレクタ → 0

※本当はもう少し複雑です

https://developer.mozilla.org/ja/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance#specificity_2

詳細度の計算例

セクタ	100	10	1	詳細度の合計
*	0	0	0	0
span	0	0	1	1
p	0	0	1	1
.text-blue	0	1	0	10
span.text-blue	0	1	1	11
#id-blue	1	0	0	100

装飾の競合

競合した場合、詳細度が高いセレクトタの装飾が優先される

詳細度: 0

```
* {  
  color: black;  
}
```

詳細度: 10

```
.text-blue {  
  color: blue;  
}
```

詳細度: 1

```
span {  
  color: red;  
  background: yellow;  
}
```



```
<span class="text-blue">What is my color?</span>
```

color: blue; 😞
background: yellow; 😊

Source Order

詳細度が同じなら、後に定義されているものが優先

```
h1 { /* 詳細度: 1 */  
  color: red;  
}  
h1 { /* 詳細度: 1 */  
  color: blue;  
}
```

Headline 1

カスケードの原則を破りたいとき

プロパティの文の最後に **!important** を入れると、ルールを無視してオーバーライドできる

ただし、**最後の手段**と考えること(やむを得ない場合を除き、使わない)

```
h1 { /* 詳細度: 1 */  
  color: red !important;  
}  
.text-blue { /* 詳細度: 10 */  
  color: blue;  
}
```

```
<h1 class="text-blue">Headline 1</h1>
```

Headline 1

CSSのプロパティ(1)

テキストと背景

color: 文字色

color: 文字の色を決める

```
<p>  
  文字が赤くなったよ  
</p>
```

```
p {  
  color: red;  
}
```

文字が赤くなったよ

色の指定方法

定数のように定義されている色を利用

ex) red / yellow / green / blue / black / white

c.f.) WEB色見本 原色大辞典 <https://www.colordic.org/>

カラーコードで指定

光の3原色それぞれの成分の量を16進数で表現 (00～ff)

ex) #ff0000 → R=255, G=0, B=0 (すなわち、赤)

他にも以下のような方法も存在

rgb(_, _, _) / hsl(_, _, _) / rgba(_, _, _, _)

background: 背景

background: ボックスの背景を決める

本当は記述方法や設定できる内容にバリエーションがあるが、割愛

```
<p>  
  背景が黄色くなったよ  
</p>
```

```
p {  
  background: yellow;  
}
```

背景が黄色くなったよ

font-size: 文字の大きさ

font-size: 文字の大きさを決める

```
<div>
  Normal text
</div>
<p>
  Large text
</p>
```

```
p {
  font-size: 32px;
}
```

Normal text

Large text

文字の大きさの指定方法

絶対指定

ピクセル

1px = 0.75pt 1pt = 1/72 in

ex) `font-size: 32px;`

相対指定 … 親要素の文字の大きさを基準にする

% … 基準に対する倍率(百分率)

ex) `font-size: 200%;` /* 親の2倍の大きさ */

em … 基準に対する倍率(n倍)

ex) `font-size: 2em;` /* 親の2倍の大きさ */

次回予告

CSS編はまだまだ続きます 😓

- CSSの外部ファイル化
- CSSのプロパティ(2) ブロック
- 疑似クラス、疑似要素
- レスポンシブデザイン

自学自習のヒント

[再掲] Webの仕様は移り変わる

他にも様々な要素が存在するし、オプションの属性値もたくさんある
→ Webで検索して必要に応じて学習

ただし、Webの仕様は定期的に変わるので、**現在の規格に準拠した情報であるか注意が必要**



CSS3

CSSの現在のバージョンは**CSS3**

厳密にはまだCSS2.1だが、CSS3に属するモジュールそれぞれ(の多く)が標準化されている

以下のような記述を見つけたら、古いバージョンかも…

- 疑似要素セレクタのコロンが1つ ex) div:after



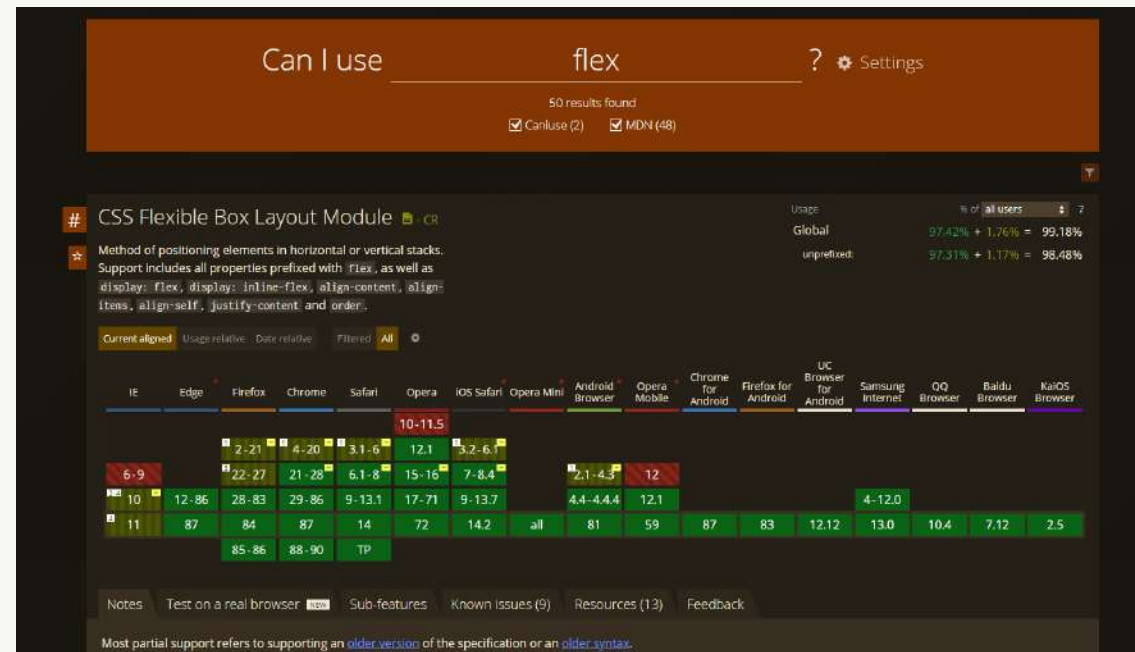
HTMLの場合と異なり、古い記法よりも新しすぎる記法のほうが問題になるかもしれない

Can I use _____ ?

Can I use...

<https://caniuse.com/>

Webの様々な仕様・機能が、どのブラウザで利用できるかを検索できる



開発者ツール

C言語のプログラムのデバッグにデバッガ(gdb, lldb...)を利用するように、Webにもデバッグツールが必要

右クリック→「検証(Chrome)」 「開発者ツールで調査する(Edge)」

