

G's Academy CSS講座



CSSって？

■CSS

Webサイトの「見た目・装飾・レイアウト」を定義する言語。

【CSSを書いていく準備】

- 1、「○○.css」というファイルを作る。
- 2、ファイルの頭に「@charset "utf-8";」と記入
- 3、HTMLファイルに<link rel="stylesheet" href="CSSの読み込み先を記載" />

■CSSはここを抑えろ！！！！



①命令をしたい「場所」を指定して

②命令の「内容」を書く！！！！

これはJavaScriptでも、非常に重要な考え方！！！！

■CSSの記述の基本

【セレクトタ】

CSSを適用したい場所の目印のこと。

(「このHTMLの文字色を赤くしたいよ〜！」の「**ここ**」につける目印)

【プロパティ】

目印をつけた場所に適用したい装飾の内容のこと。

(「このHTMLの文字色を赤くしたいよ〜！」の「**赤くしたいよ〜！**」の部分がプロパティに該当)

■基本的な書き方



```
セクタ {  
    プロパティ名: 指定したい内容;  
}
```

※例

```
h1{  
    background-color:#f00;  
    /*h1タグの背景色を赤色にする、  
    という命令です*/  
}
```



CSSを書こう...のその前に！！

ブラウザってChrome,Safari,Firefox,IE,edgeなど色々ありますが...

HTMLだけの状態で見るとわかりますが...

ブラウザによって見た目のなんと異なること
とか！！！！

■リセットCSS



各ブラウザ間が元々持っている表示の差異をいちいち考えていると

とてつもない労力がかかってしまう...

全ぶ一旦チャラにして、
全てフラットな状態にしてから、
CSSを適用できるようにしよう！

そんな救世主の代表がreset.cssと呼ばれるものです。
(他にもnormalize.cssとかいろいろな種類がある)



レッツトライ！CSSはじめての1歩

（何か1個CSSを書いてみよう）

■セクタ(目印)の種類



CSSを適用するために、HTMLタグを目印にすると
そのタグがHTMLの中に1つとは限らない！

つまり、意図しないところにも装飾が適用されてしまう可能性
が！！

■IDとClass

【ID】(HTML上では「id="aim"」 CSS上では「#aim」みたいに表記)

そのHTML内で1ヶ所のみ使用可能。

CSS適用の目的で使用するより、JavaScript絡みで使うことが多い。

【Class】(HTML上では「class="aim"」 CSS上では「.aim」みたいに表記)

そのHTML内で複数箇所使用可能。

かつ、同じ箇所に複数のclassを使用可能。(半角スペースで区切る！)

(<p class="section-title text-center"> </p>)

似てる！
classの使いどころ！

NEWS

お知らせ

- 2016.03.03 工事現場での写真管理を限りなくスマートにする 工事写真共有ホスティングサービスPhotoruction中島貴春氏×えふしん氏【前編】
- 2016.01.06 ジーズアカデミーは、圧倒的な熱量が高まり続けていく場所 LandSkip下村一樹氏に聞く【後編】
- 2016.01.04 世界の風景を指先ひとつで連れてくる。Web時代の新しい窓 LandSkip下村一樹氏【前編】

ニュース一覧を見る

FEATURE

特長

■ブロック要素とインライン要素

htmlの各タグは、ある種、積み木やブロックのようなものです。

まずは、この2つをdeveloper-toolsを使って理解する！！

【block要素】

特に指定しない限り、横幅一杯に幅(width)を取る要素。

【inline要素】

特に指定しない限り、中身に含まれる要素の大きさ分しか幅を取らない要素。

高さや上下余白の指定ができない。

■よく出るCSSプロパティ(文字編)

■文字関連

font-size・・・文字の大きさ(px、em、rem、%などで指定可能)

color・・・文字の色(#ff9900などのカラーコードやredなどの色名)

font-family・・・文字の種類

text-align・・・中央揃え、左揃え、右揃えなどの配置を指定

※text-alignは、中央揃えだったり、左揃えだったりにしたいテキストや画像を囲うタグに対して適用してあげます。

■よく出るCSSプロパティ(背景編)

■背景関連

background-color・・・背景色

background-image・・・背景画像

background-repeat・・・背景画像等の繰り返し

background-size・・・背景画像の大きさを指定。最近はcoverなど多使用。

background-position・・・背景画像の位置指定

※まとめたの指定も可能

background:●●●;

■よく出るCSSプロパティ(レイアウト編)

■ボックスモデル関連

margin・・・外余白

padding・・・内余白

border・・・境界線(border:solid 1px #f00;のように、種類・太さ・色と指定)

width & height・・・幅と高さをそれぞれ指定

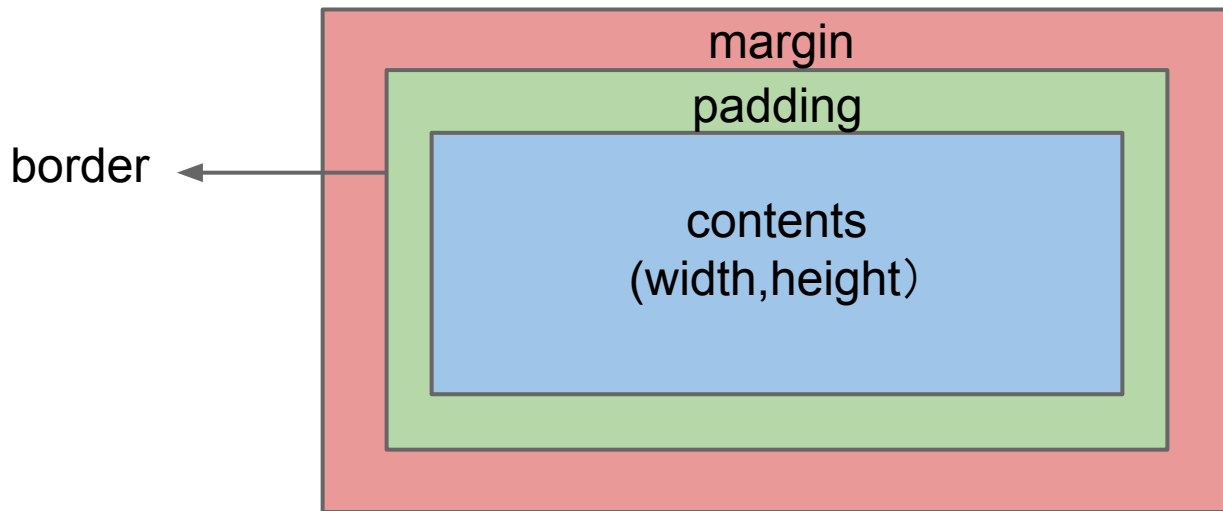
※まとめた指定も可能

margin:10px・・・上下左右まとめて指定

margin:10px 20px 30px・・・上、左右、下と指定

margin:10px 20px・・・上下、左右と指定

■ボックスモデルの理解



■疑似クラス

「要素にマウスが乗ったとき」

「ラジオボタンがチェックされたとき」

など通常と違う状態の時に見た目を変えるためにセレクタに付けるもの。

※例

```
a { color:#f00; }
```

```
a: hover { color:#00f; }
```

通常時、aタグの色は赤、マウスが乗った時はaタグの色は青になる。



CSSレイアウト編

■ここまで作ってみて

HTMLは、何もしないと左上から、下に向かってブロックが並べられるように、縦にレイアウトされていきます。

(左寄せ)

でも、最近あんまり左寄せになっているサイトやサービス
見ないですね〜〜

■CSSレイアウト



例えば、「横並び」

これは、CSSを駆使してこそ実現可能になるものです！

CSSを使いこなせるようになって、自由にレイアウトできるようになるう！

レイアウトを制するものは、CSSを制す！



CSSレイアウトをする前に
知っておきたいこと

■divの使いどころ

下図のように、右寄せにしたい(横並びにしたい)ものがまとめている場合、
右寄せにしたいタグ1つ1つにCSSを指定するより、
一括でまとめて一つのタグで囲んであげて、1箇所にCSS指定した方が楽！

コスゲタツヤの自己紹介



出身
埼玉
趣味
旅・音楽
Gs Academyを志した理由
世界を変えたいから
最近よく見るWEBサイト
[Schoo Google Apps Script入門講座](#)

コスゲタツヤの自己紹介



出身
埼玉
趣味
旅・音楽
Gs Academyを志した理由
世界を変えたいから
最近よく見るWEBサイト
[Schoo Google Apps Script入門講座](#)

■divの使いどころ

```
<h2>小菅の自己紹介項目</h2>
```

```
<dl>
```

```
<dt>名前</dt>
```

```
<dd>こすげたつや</dd>
```

```
<dt>出身</dt>
```

```
<dd>埼玉</dd>
```

```
<dt>ニックネーム</dt>
```

```
<dd>コスゲリアン</dd>
```

```
</dl>
```

これに適用
すればOK!



```
<div class="outer">
```

```
<h2>小菅の自己紹介項目</h2>
```

```
<dl>
```

```
<dt>名前</dt>
```

```
<dd>こすげたつや</dd>
```

```
<dt>出身</dt>
```

```
<dd>埼玉</dd>
```

```
<dt>ニックネーム</dt>
```

```
<dd>コスゲリアン</dd>
```

```
</dl>
```

```
</div>
```

■センタリングの基本を学んでdivの使いどころも体感！

【ブロック要素のセンタリング】

width:○○px;

margin:0 auto;

※横方向のmarginをauto(自動調整)

→左右のmarginが等しくなる

【インライン要素のセンタリング】

真ん中にしたい要素を囲むHTMLタグに対して

text-align:centerを適用

コスゲタツヤの自己紹介



出身	埼玉
趣味	旅・音楽
G's Academyを志した理由	世界を変えたいから
最近よく見るWEBサイト	Schoo Google Apps Script入門講座

名前

■float

floatを適用した要素は、左あるいは右寄せに配置されます。

HTMLだけだと、通常上から縦にレイアウトされていますが、floatなどを使うことによって、横並びが実現できるようになります。

■clearfix

floatを適用すると、文字通り、その要素は「浮きます」。

そのため、後に続く要素は、そのまま上にずれ込む形になり、高さの計算がおかしくなります。

また、floatを適用した要素の親要素の高さがなくなってしまいます。

それを防ぎ、正常なレイアウトを保つために使用するのがclearfixになります。

clearfixは、floatを適用した要素の親要素にclass="clearfix"という形で記載を実施します。

(overflow:hiddenを指定する方法もあります)

■position

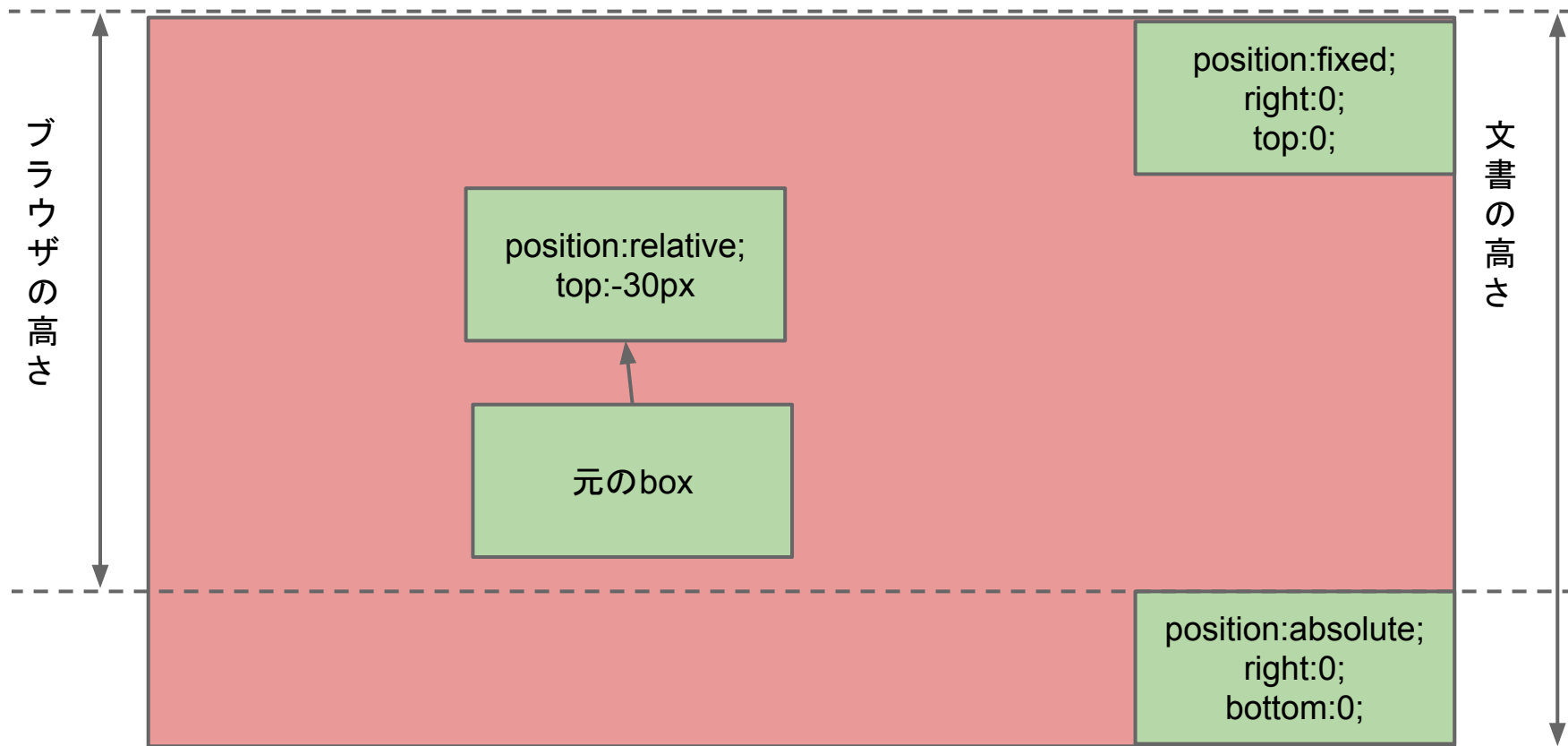
positionは、複雑なレイアウトが出来るCSSプロパティです。



- ・relative・・・相対配置
- ・absolute・・・絶対配置
- ・fixed・・・固定配置

なんでもかんでもpositionで配置すると、著しくメンテナンス性が悪くなり、すぐレイアウト崩れを引き起こしますので、多用すべきではないでしょう。

■positionの理解



■positionの理解

`position:relative;`

`position:absolute;`
`top:100px;`
`left:100px;`

■z-indexについて

CSSのposition(static以外)が適用された要素があると、重なるという概念が生じるため、z-indexが自動的に付与されます。

(何も指定しない場合はz-index:autoが自動付与)

position(static以外)とz-indexはセットといってもよい!



Tips

■CSSの注意点

- ・CSSの優先順位に注意する。

- 基本的には下に書かれたものが優先される。

- id,classなどにより「詳細度」が決まるので、それにより原則が変わる。

- ・Webサイトやアプリは基本的に同じ状態がずっと保たれないと思って作る。

- 文章が1行→2行になるだけで高さが変わる→height指定してると修正面倒。

- positionを無駄に使用する→やはり修正があるとき面倒。



デベロッパーツールを体験

ありがとうございました。