セレクタのおさらい

CSSで設定する上で必要なセレクタについての復習

セレクタとは

CSS 上で特定の要素に対してスタイルを設定したい場合、特定の場所を指定する方法をセレクタといいます。

```
div { font-size: 1.6em; }
```

セレクタ　この場合 div 要素全てにスタイルが適用される

セレクタを上手く利用することで、コードの節約や管理のしやすさに繋るためセレクタの種類を知ることはとても大切です。上手な使い方や第三者への管理方法などは今後の目標とし、まずはセレクタの種類を把握しましょう。

要素に対して指定するセレクタ

Universal selector
全称セレクタ

*

文書内すべての要素に対してスタイルを指定する

exsample

```
*{
        color: #FF88EE;
}
```

Type Selector
型セレクタ

E

文書内にある特定の E 要素に対してスタイルを指定する

exsample

```
p{
        color: #FF88EE;
}
```

Class selector
クラスセレクタ

E.classname

E 要素の class 属性の値が「クラス名」の要素に対してスタイルを指定する。

exsample - HTML

```
<div class="attention">
        <h1 class="attention">CSS Style</h1>
        <p>CSS について以下の説明 ....</p>
</div>
```

exsample - CSS

```
.attention{
        color: #FF88EE;
}
h1.attention{
        color: #00FFCC;
}
```

上記の場合、クラス名全体に文字色（#FF88EE）が適用されるが、先頭に要素などを指定した場合、h1 要素を含む attention クラスとなるため、範囲に差が出ます。注意してください。

ID selector
ID セレクタ

E#IDname

E 要素の id 属性の値が「id 名」の要素に対してスタイルを指定する。

exsample - HTML

```
<div>
        <h1 id="about">CSS Style</h1>
        <p>CSS について以下の説明 ....</p>
</div>
```

exsample - CSS

```
#about{
        font-size: 2em;
}
h1#about{
        font-size: 3em;
}
```

ID 属性値は、HTML 文書内で固有のものに限りユニーク（唯一）であることが条件です。そのため、上記 CSS の場合どちらも同じ場所を指定することになるため、後書きの方が優先されます。

Groups of selectors
セレクタのグループ化

E,F

同じスタイルでセレクタが異なる場合、セレクタ同士をグループにする。

exsample - HTML

```
<div>
        <h1 id="about">CSS Style</h1>
        <p>CSS について以下の説明 ....</p>
</div>
```

exsample - CSS

```
h1,p{
        font-size: 2em;
}
```

Child combinator

E 要素の子要素である F 要素

E > F

親要素に含まれる子要素にのみスタイルを指定する。

exsample - HTML

```
<div class="attention">
        <h1>CSS Style</h1>
        <p>CSS について以下の説明 ....</p>
        <section>
                <h2>Selector</h2>
                <p> セレクタとは ....</p>
        </section>
</div>
```

exsample - CSS

```
.attention > p{
        color: #FF88EE;
}
```

上記例では、attention クラスの子要素に対する p 要素の為、section 要素内の p は、attention クラスから見ると孫要素にあたる為、スタイルは適用されません。

Descendant combinator

E 要素の子孫要素の関係である F 要素

E␣F

親要素に含まれる子孫要素にスタイルを指定する。

exsample - HTML

```
<div class="attention">
        <h1 class="attention">CSS Style</h1>
        <p>CSS について以下の説明 ....</p>
</div>
```

exsample - CSS

```
.attention p{
        color: #FF88EE;
}
```

Pseudo-element

疑似要素 最初の行の 1 文字目

E::first-line

E 要素の最初の 1 文字目にのみスタイルを指定する。

exsample - HTML

```
<div class="attention">
        <h1 class="attention">CSS Style</h1>
        <p>CSS について以下の説明 ....</p>
</div>
```

exsample - CSS

```
h1::first-line{
        color: #FF88EE;
}
```

疑似クラスと表記の違いを表す為に、疑似要素はセミコロンが２つ（::）になりましたが、１つ表記でも可です。

Pseudo-element

疑似要素 最初の行の 1 文字目

E::first-line

E 要素の最初の 1 行目にのみスタイルを指定する。

exsample - HTML

```
<div class="attention">
        <h1 class="attention">CSS Style</h1>
        <p>CSS について以下の説明 ....</p>
</div>
```

exsample - CSS

```
p::first-line{
        color: #00FFCC;
}
```

Pseudo-classes

疑似クラス 基本擬似クラス

E:xxxx

リンク関係やユーザーアクションなどの基本擬似クラス

exsample - CSS

```
/* ハイパーリンクである要素の未訪問のもの */
a:link{ color: #FF0000; }
/* ハイパーリンクである要素の訪問済のもの */
a:visited{ color: #FF0000; }
/* 特定のユーザアクション（マウスカーソルが要素の上にある）状態 */
a:hover{ color: #FF0000; }
/* 特定のユーザアクション（マウスカーソルを押下した）状態 */
a:ative{ color: #FF0000; }
/* 特定のユーザアクション（要素にフォーカスした）状態 */
input[type="text"]:focus{ color: #FF0000; }
/* 特定のユーザアクション（ラジオボタンやチェックボックスなどの選択）状態 */
input[type="radio"]:checked{ color: #FF0000; }
/* 擬似クラスが有効状態にあるもの */
textarea:enabled{ color: #FF0000; }
/* 擬似クラスが無効状態にあるもの */
textarea:disabled{ color: #FF0000; }
```

擬似クラスを組み合わせることで、特定の状態化を作ることも可能です。例えば、:hover時の子要素の:hover時など。以下のサンプルも合わせて確認して下さい。

exsample - CSS

```
/*ul の子要素がホバー時の子要素である ul 要素 */
ul>li:hover>ul{
        display: block;
}
```

Adjacent sibling combinator

隣接セレクタ（隣接兄弟セレクタ）

E＋F

E 要素の直後に現れる F 要素に対してスタイルを指定する。

exsample - HTML

```
<div class="attention">
        <h1>CSS Style</h1>
        <p>CSS について以下の説明 ....</p>
        <p>CSS について以下の説明 ....</p>
        <section>
                <h2>Selector</h2>
                <p> セレクタとは ....</p>
        </section>
</div>
```

exsample - CSS

```
h1+p{
        color: #FF88EE;
}
```

上記例では、隣接セレクタの為、h1 要素の直後にくる p 要素になるので、section 内の p 要素はもちろん p 要素の次にくる p 要素にもスタイルは適用されません。

General sibling combinator

間接セレクタ（一般兄弟セレクタ）

E ～ F

ある要素の直後にある要素に対してスタイルを指定する。

exsample - HTML

```
<div class="attention">
        <h1>CSS Style</h1>
        <p>CSS について以下の説明 ....</p>
        <p>CSS について以下の説明 ....</p>
        <section>
                <h2>Selector</h2>
                <p> セレクタとは ....</p>
        </section>
</div>
```

exsample - CSS

```
h1~p{
        color: #FF88EE;
}
```

間接セレクタの為、h1 要素と兄弟にあたる p 要素２つにはスタイルが適用されます。section 要素内の p 要素に関しては、h1 要素と比較すると親子関係にある為スタイルは適用されません。

Attribute selectors
属性セレクタ

E[foo]

「属性名 foo」の属性を持つ要素に対してスタイルを指定する。

exsample - HTML

```
<form action="foo.php" method="post">
<p><input type="text"> 名前 </p>
<p><input type="password"> パスワード </p>
<p>
<input type="radio" name="sex" value="0"> 男性
<input type="radio" name="sex" value="1"> 女性
</p>
<textarea> 本文 </textarea>
</form>
<p>
<a href="#tel"> 電話でのお問い合わせ </a>
</p>
<p>
<a href="http://xxx.com"> 関連会社 A へのアクセス
</a>
</p>
<p>
<a href="http://yyy.com"> 関連会社 B へのアクセス
</a>
</p>
<p title="copyright ecc">
<small>copyright ecccomp.</small>
</p>
```

exsample - CSS

```
/* 属性セレクタケース 1*/
input[type]{
        colro: #FF0000;
}
/* 属性セレクタケース 2*/
input[type="text"]{
        font-size: 1.4rem;
}
/* 属性セレクタケース 3*/
p[title~="copyright"]{
        font-sioze: 0.8rem;
}
/* 属性セレクタケース 4*/
a[href^="#"]{
        color: #00FF33;
}
/* 属性セレクタケース 5*/
a[href$=".com"]{
        color: #003399;
}
/* 属性セレクタケース 6*/
a[href*="x"]{
        color: #993399;
}
```

属性セレクタケース 1

input 要素の属性名が type の要素

属性セレクタケース 2

input 要素の属性名が type であり、値が text の要素

属性セレクタケース 3

p 要素のうち title 属性の値に copyright の単語を含む要素

属性セレクタケース 4

a 要素のうち href 属性の値が # ではじまる要素

属性セレクタケース 5

a 要素のうち href 属性の値が .com で終わる要素

属性セレクタケース 6

a 要素のうち href 属性の値が x を 1 つ以上含む要素

属性セレクタは、組み合わせで不要な ID や Class を避けてスタイルを設定することが可能です。特にフォーム関連でのスタイルを設定する際やページ内リンクに対してのスタイルなど知っておくと不要なコードを避けることが出来るので、合わせておさえておきましょう。

Structural pseudo-classes

擬似クラス　最初の要素

E:first-child

E 要素の最初の子要素に対してスタイルを指定する。

exsample - HTML

```
<h1>HTML について </h1>
<section>
    <h1> マークアップ言語 </h1>
    <p> マークアップ言語とは ....</p>
    <h1> タグについて </h1>
</section>
```

exsample - CSS

```
section > h1:first-child{
        font-size: 2rem;
}
```

上記例では、section 要素の子であるはじめの h1 要素となる為、section 直後に出てくる h1 要素に関してはスタイルが適用されますが、その他 h1 要素に関しては適用されません。
また、仮にマークアップ言語と書かれた見出しが無い場合、section 要素の直後の子要素は p 要素となる為、この場合もタグについてと書かれた見出しはスタイルは適用されません。

Structural pseudo-classes

擬似クラス　最後の要素

E:last-child

E 要素の最後の子要素に対してスタイルを指定する。

exsample - HTML

```
<ol>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
</ol>
```

exsample - CSS

```
ol>li:last-child{
        color: tomato;
}
```

上記例では、ol 要素の子要素である最後の li 要素に対してスタイルが適用されます。first-child と last-child をそれぞれ使用することで、最初と最後の要素を指定することが出来るので、合わせて覚えておいてください。

Structural pseudo-classes

擬似クラス　an+b 番目の隣接する要素

E:nth-child(an+b)

指定した E 要素に隣接する an+b 番目の要素に対してスタイルを指定する。

exsample - HTML

```
<ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
    <li>PHP</li>
    <li>MySQL</li>
</ul>
```

exsample - CSS

```
/* 奇数の li 要素に対してスタイルを指定 */
ul>li:nth-child(2n+1){
        color: blue;
}
/* 偶数の li 要素に対してスタイルを指定 */
ul>li:nth-child(2n){
        color: green;
}
```

an+b の関係は、a または b は整数（正負または 0）で無ければならず、n は正の整数または 0 となります。
上記例では、2n+1 これは奇数を示し、2n（または 2n+0）は偶数を示します。また、奇数は odd 偶数は even で表すことが可能なので、偶数・奇数の表記のみであれば odd even で代用が可能です。

Structural pseudo-classes

擬似クラス　an+b 番目の（隣接に限らない）要素

E:nth-of-type(an+b)

指定した E 要素が an+b の兄弟を持つ要素にスタイルを指定する。

exsample - HTML

```
<div class="list">
    <h2> フロントエンド </h2>
    <p>HTML</p>
    <p>CSS</p>
    <p>JavaScript</p>
    <h2> バックエンド </h2>
    <p>PHP</p>
    <p>MySQL</p>
</div>
```

exsample - CSS

```
.list>p:nth-of-type(2n+1){
        color: tomato;
}
.list>p:nth-of-type(2n){
        color: skyblue;
}
```

上記例のような、p 要素の間に別の要素が挟む場合、隣接されていない要素（今回だと h2 要素以降の p 要素）は順番がリセットされてしまい、順番に指定しているスタイルが崩れてしまいます。そのため、nth-of-type() を使用することで、隣接されていない要素に対してもスタイルを指定することが可能です。