

CSS

[CSS3][[レスポンシブ Web デザイン](#)]

適用方法

■ style 属性で個別に指定

```
<p style="margin: 15px">
```

■ head 要素の範囲内で style 要素によって指定

```
<style type="text/css">
p {margin: 15px}
</style>
```

■ head 要素の範囲内で link 要素によって外部スタイルシートを参照する

```
<link rel="stylesheet" type="text/css" href="./css/style.css"/>
```

■ head 要素の範囲内で @import によって外部スタイルシートを参照する

```
<style type="text/css">
@import url("./css/style.css")
</style>
```

セレクトと疑似クラス、疑似要素

[[jQuery](#)]

タイプセレクトとユニバーサルセレクト

■タイプセレクト

- ・要素名のみをセレクトにする

```
body {color:black;}
```

■ユニバーサルセレクト

- ・すべての要素に対してスタイルを指定できる

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

- ・ユニバーサルセレクトは、他のセレクトが伴う場合は省略できる
 - ・ *.note と .note は等しい
 - ・ *#note1 と #note1 は等しい
 - ・ *[abbr] と [abbr] は等しい

class セレクトと id セレクト

■ class セレクト (要素名 .class 名)

- ・ 文章内で複数の要素にスタイルを適用する

```
p.note{color:red}
⋮
<p class="note">...</p>
```

- ・ ユニバーサルセクタを利用、要素を省略することで、すべての要素に対して適用できる

```
.note{color:red}
```

■ id セクタ (要素名 #id)

- ・ 文書内で、唯一の要素にスタイルを適用するのに利用する。

```
div#gnavi{font-size:small}
⋮
<div id="gnavi"> ... </div>
```

属性セクタ

- ・ 特定の属性値や属性名をもつ要素に対してスタイルを適用する

■要素名 [属性名]

- ・ 属性名を持つ要素にスタイルを適用する

```
h3[id] {color:olive}
⋮
<h3 id="r508">...</h3>
```

■要素名 [属性名 =" 属性値 "]

- ・ 属性名と属性値を持つ要素に対してスタイルを適用する

```
h3[id="r508"] {color:olive}
⋮
<h3 id="r508">...</h3>
```

■要素名 [属性名 ~=" 属性値 "]

- ・ 属性名を持ち、属性値が空白区切りで複数指定されている場合、属性値が含まれている要素にスタイルを適用する

```
p[class~="note"]{color:red}
⋮
<p class="w3c wcag note">...</p>
```

■要素名 [属性名 | =" 属性値 "]

- ・ 属性名を持ち、属性値がハイフン区切りで複数指定されている場合、属性値が含まれている要素にスタイルを適用する

```
[lang|="en"]{font-family:Veradana }
⋮
<span lang="en-US">...</span>
```

```
<span lang="ja">...</span>
```

セレクタの組合せ

■セレクタ セレクタ (空白文字区切り)

- ・子孫セレクタ (descendant selector) といい、親要素に含まれるすべての子孫要素にスタイルを適用する

```
div.note p {color:red}
```

絞り込みも可能

```
div.note p abbr {text-decoration:underline }
```

孫要素以下に適用する例

```
div * p {color:blue}
```

■セレクタ > セレクタ

- ・子セレクタ (child selector) といい、親要素の直接の子要素にスタイルを適用する

```
blockquote.wcag > p {font-size:small}
```

絞り込みも可能

```
blockquote.wcag > p > abbr {font-size:small}
```

■セレクタ + セレクタ

- ・同一の親要素をもつ並列関係にある要素について、前にある要素を兄要素、後に続く要素を弟要素と呼ぶ
- ・隣接セレクタ (adjacent sibling selector) はある要素の直接の弟要素にスタイルを適用する

```
h1 + h2 {color:green}
⋮
<h1>...</h1>
<h2>...</h2>
```

セレクタのグループ化

- ・半角カンマでグループ化し、複数の要素にスタイルを適用することができる

```
th,td {font-size:small}
```

疑似クラス

- ・「状況」によってスタイルを適用

■ :first-child 疑似クラス

- ・ 親要素の範囲内で最初に登場する子要素のみにスタイルを適用

```
div > p:first-child { text-indent:0 }
```

■ リンク疑似クラス (:link、:visited)

:link 疑似クラス

- ・ 未訪問のリンクに適用

:visited 疑似クラス

- ・ 既にかいたリンクに適用

例

```
:link {color:blue}
a:link {color:blue}
a[href]:link {color:blue}
:visited {color:blue}
a:visited {color:blue}
a[href]:visited {color:blue}
```

ダイナミック疑似クラス

- ・ ユーザーのアクションに適応して表示を変えるブラウザでのスタイルをコントロール

:hover 疑似クラス

- ・ カーソルが当たっている状態

:active 疑似クラス

- ・ アクティブな状態

:focus 疑似クラス

- ・ フォーカスを受けている状態

例

```
a:hover {color:yellow}
```

疑似要素

:first-line 疑似要素

- ・要素の1行目のみにスタイルを適用

:first-letter 疑似要素

- ・要素の1文字目のみにスタイルを適用

:before 疑似要素と :after 疑似要素

- ・有る要素に含まれる内容の前後に内容を生成

Tips

■ CSS 背景画像の設定、透明度の設定

■ ブラウザ用デバッグツール

■ 縦書き

```
writing-mode:tb-rl
```

■ リンク色を個別に変える

```
a.type1 { color:blue; }  
a:hover.type1 { color:red; }  
a.type2 { color:blue; }  
a:hover.type2 { color:orange; }
```

■ テーブルレイアウト

```
table-layout:auto  
テーブル（表）全体を読み込んでから、  
各縦列の幅を決定して表示します。これが初期値です。
```

```
table-layout:fixed  
最初の横一行を読み込んだ時点で、  
各縦列の幅を決定して表示を開始します。
```

■ 条件付きコメント

- ・IEの独自拡張である「条件付きコメント」は、以下のような書式で HTML 中に記述
- ・ <http://allabout.co.jp/internet/hpcreate/closeup/CU20070412A/index2.htm>

```
<!--[if IE]><script src="../../other_libs/excanvas_r3 /excanvas.js" type="text/javascript"  
charset="utf-8"></script><![endif]-->
```

■ important ルール

- ・ <http://www.xml.vc/kiso/!important.html>
- ・ CSS のスタイルシートはブラウザのデフォルト・スタイルシート→ユーザー・スタイルシート → ページ作者のスタイルシートの順に優先度が高くなります。また、ひとつのス

スタイルシートにおいても後から指定したスタイルが優先されます。

- ・「後から」とは同じセレクタのスタイルが重複した場合は最後（CSS ソースの下）に指定したスタイルが優先されること。
- ・普通のスタイルよりはクラスセレクタを使ったスタイルが優先
- ・クラスセレクタよりは ID セレクタを使ったスタイルが優先

!important ルールを使うとスタイルを優先させることができます。「プロパティ: 値」の後に半角スペースで区切り、!important を配置します。ただし、「プロパティ: 値」ごとに指定する必要があります。

```
body { background-color: white !important; }
```