



# CSS



# CSS の概要

## CSS とは

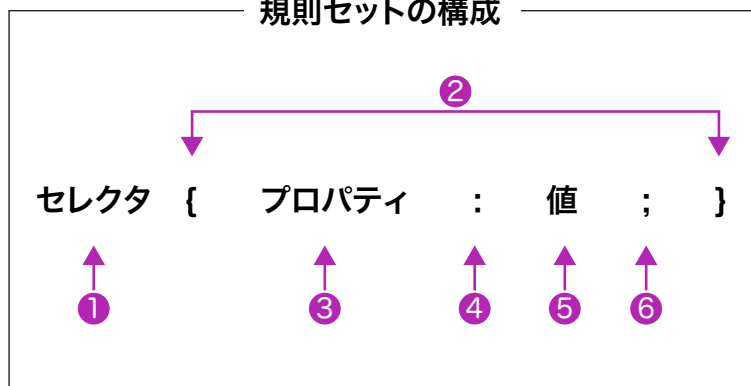
CSS は Cascading Style Sheets の略語で、HTML のようにプログラミング言語ではなく、またマークアップ言語でもありません。CSS はスタイルシート言語と呼ばれ、HTML の要素を選択的にスタイル（見た目）を設定するためのものになります。

## CSS の構文

CSS は、ルールベースの言語です。Web ページの特定の要素かグループに適用するスタイルのルールを定義します。

ルールは、スタイリングしたい HTML 要素（セクタ）から始まり、次に「{ }」（波括弧）を書き、「{ }」の中に「:」（コロン）の前にプロパティの名前を「:」の後にプロパティの値を書きます。このプロパティ名とプロパティの値のペアで作られたを宣言 1 つ以上指定します。

規則セットの構成



```
h1 {  
  color : tomato;  
  font-size : 32px;  
}
```

①      ②      ③

### ① セクタ

スタイルを設定する要素を指定します。要素の指定には、要素名を直接指定する方法以外にも様々な種類が用意されています。

### ③ プロパティ

セクタに適用するスタイルの種類を小文字で記述します。

### ④ コロン

プロパティと値の間には、「:」で区切ります。「:」は「=」を意味します。

### ② 宣言ブロック

「{ }」（波括弧）で囲った部分は、セクタへ 1 つ以上のスタイルを指定するブロックになります。

### ⑤ 値

スタイルの内容を数値やキーワードを指定します。

### ⑥ セミコロン

1 つの宣言の終了を意味します。宣言ブロック内の最後尾の宣言は、「;」による区切りを省略することができますが、宣言を追記した際などのエラーを防ぐためにも、「;」は必ず付けるようにしましょう。

## 複数の宣言を指定する

宣言ブロック内で、「;」（セミコロン）に続けて、宣言を記述すれば、1つのセレクトアに対して複数の宣言を同時に指定することができます。

```
h1 { color: tomato }  
h1 { font-size: 32px }  
h1 { text-align: center }
```

```
h1 {  
  color: tomato;  
  font-size: 32px;  
  text-align: center;  
}
```

上の2つは、1つのセレクトアに対して宣言を個別に指定するのと、複数の宣言を同時に指定するのと同じ結果になります。

## セレクトアのグループ化

セレクトアを「,」（カンマ）で区切って記述することで、複数のセレクトアに同じ宣言を指定することができます。

```
html ,  
body {  
  width: 100vw;  
  height: 100vh;  
}
```

# CSS セレクタ

CSS では、HTML の要素にスタイルを設定するためにセレクタが使用されます。CSS セレクタにはさまざまな種類があり、きめ細かにスタイルを設定する要素を選択することができます。

## セレクタとは

CSS のもっとも基本的なルールとなるもので、要素やその他を選択してスタイルシートを適用する要素をブラウザに伝えるものです。

## セレクタの種類

セレクタにはいくつかのグループがありますが、ここではタイプセレクタ、全称セレクタ、ID セレクタ、クラスセレクタ、子孫セレクタなどの基本となるセレクタを紹介いたします。

### タイプセレクタ

指定した要素にスタイルを適用する

要素名をセレクタに指定するもっとも単純なセレクタとなります。

```
<body>
<div id="container">
  <h1 class="head">Web World</h1>
  <p class="head">HTML CSS JavaScript</p>
</div>
</body>
```

HTML

```
h1 {
  color: #ff33cc;
}
```

CSS

### 全称セレクタ（ユニバーサルセレクタ）

すべての要素にスタイルを適用する

「\*」（アスタリスク）をセレクタに指定すると、すべての要素を対象としてスタイルを適用します。

```
<body>
<div id="container">
  <h1 class="head">Web World</h1>
  <p class="head">HTML CSS JavaScript</p>
</div>
</body>
```

HTML

```
* {
  box-sizing: border-box;
}
```

CSS

## ID セレクタ

指定した ID 名を持つ要素にスタイルを適用する

「#」（ハッシュ）に続けて、要素に指定した ID 名を入力します。要素への ID 名の指定は、グローバル属性である ID 属性を使っておこないます。ID 名は、原則 Web ページ内で重複しない一意である必要があります。

```
<body>
<div id="container">
  <h1 class="head">Web World</h1>
  <p class="head">HTML CSS JavaScript</p>
</div>
</body>
```

HTML

```
#container {
  width: 800px;
  margin: 0 auto;
}
```

CSS

## クラスセレクタ

指定した class 名を持つ要素にスタイルを適用する

「.」（ドット）に続けて、要素に指定した class 名を入力します。要素への class 名の指定は、グローバル属性である class 属性を使っておこないます。class 名は要素に関係なく重複して同じ値を指定することができます。

```
<body>
<div id="container">
  <h1 class="head">Web World</h1>
  <p class="head">HTML CSS JavaScript</p>
</div>
</body>
```

HTML

```
.head {
  color: #111;
  font-family: serif;
}
```

CSS

## 子孫セレクタ

子孫要素にスタイルを適用する

親要素に含まれるすべての子孫要素にスタイルを適用します。親要素と子孫要素の間は、半角スペースで区切って入力します。子孫セレクタを活用することで、スタイルを適用する要素の範囲を絞ることが可能になります。

子孫セレクタは、CSS を本格的に活用する上で、もっとも重要なセレクタといえます。

```
<body>
<div id="container">
  <h1 class="head">Web World</h1>
  <p class="head">HTML CSS JavaScript</p>
</div>
</body>
```

HTML

```
#container p {
  font-size: 12px;
}
```

CSS

# CSS を HTML に適用する方法

CSS を HTML に適用するには、一般的に3つの方法があります。それぞれの方法にメリット、スタイルの適用の優先順位などがありますので、特性を理解した上で使い分けていきましょう。

## link 要素を使って外部スタイルシートを読み込む

もっとも一般的な適用方法で、HTML ファイルとは別に用意した外部スタイルシート（CSS ファイル）を、HTML 文書の head 要素内に記述した link 要素で読み込みます。

1つの CSS ファイルを複数の HTML ファイルに読み込ませることによって、スタイルの統一性や変更を容易にすることができます。

```
<head>
  <meta charset="UTF-8">
  <title>HTML & CSS</title>
  <link rel="stylesheet" href="css/style.css">
</head>
```

HTML

link 要素は、rel 属性でスタイルシートがあることを、href 属性でスタイルシートが配置されている場所をブラウザに伝えます。

## style 要素を使ってスタイルを適用する

HTML 文書の head 要素内に記述した style 要素内に CSS を直接記述する方法になります。

外部スタイルシートよりスタイルの適用優先度は高くなりますが、HTML ファイルごとにスタイルの指定が分散してしまうため、スタイルの統一性や変更が大変になります。

```
<head>
  <meta charset="UTF-8">
  <title>HTML & CSS</title>
  <style>
    h1 {
      color: #ff33cc;
    }
  </style>
</head>
```

HTML

## style 属性を使ってスタイルを適用する

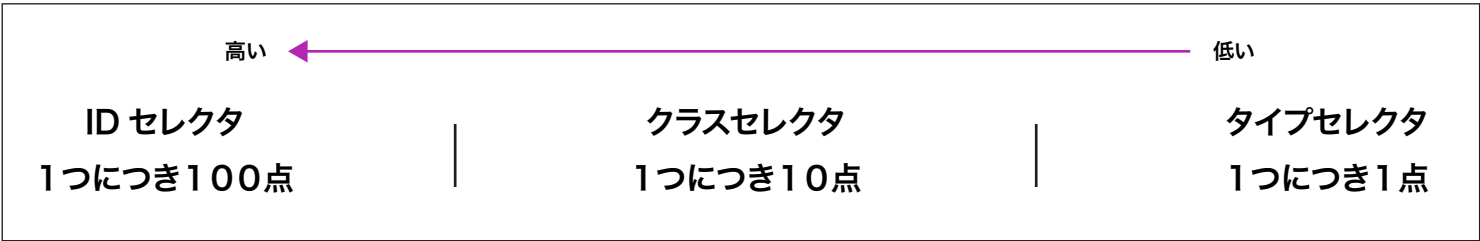
HTML のグローバル属性である style 属性を使うと、対象の要素にスタイルを指定することができます。

もっともスタイルの適用優先度は高くなりますが、要素ごとにスタイルの設定が分散してしまうため、スタイルの統一性や変更などが難しくなります。

HTML

```
<body>
  <h1 style="color: #ff33cc;">スタイルシートの適用優先度 </h1>
  <ol style="list-style: none;">
    <li>style 属性を使ったスタイルの適用 </li>
    <li>style 要素を使ったスタイルの適用 </li>
    <li>外部スタイルシート（CSS ファイル）を使ったスタイルの適用 </li>
  </ol>
```

### スタイルの適用優先度



# スタイルの継承

親要素に指定した CSS プロパティの値には、子要素に引き継がれるものと引き継がれないものがあります。子要素に CSS プロパティの値が引き継がれることを「継承」といいます。

```
<body>
<div id="container">
  <h1 class="head">Web World</h1>
  <p class="head">HTML CSS JavaScript</p>
</div>
</body>
```

HTML

```
#container {
  color: tomato;
  font-family: serif;
}
```

CSS



## グローバル値による継承の制御

CSS のすべてのプロパティは、継承を制御するための特別なユニバーサルプロパティの値を提供されています。

### inherit

要素に適用される CSS プロパティの値を親要素と同じものに設定します。「継承を有効にする」ことになります。

### initial

ブラウザが提供するデフォルトスタイルシート（初期値）にリセットします。デフォルトスタイルシートが設定されておらず、CSS プロパティの値が継承されている場合は、inherit が代わりに設定されます。

### unset

CSS プロパティを自然な値にリセットします。CSS プロパティが自然に継承される場合は inherit のように動き、そうでない場合は initial のように動きます。



# 長さ単位

## 長さの単位

利用できる単位には、指定した数値で大きさが決まる「絶対単位」と基準となる対象を持つ「相対単位」があります。

### 絶対的な長さの単位

単位	説明
cm	1 センチメートルに対応した単位。
mm	1 ミリメートルに対応した単位。
in	1 インチ 2.54cm に対応した単位。
pc	1 パイカ 12 ポイント に対応した単位。
pt	1 ポイント 1 インチの 1/72 に対応した単位。
px	1 ピクセルに対応した単位です。CSS の仕様では絶対単位に分類されていますが、ディスプレイの解像度によって、表示されるサイズは変化します。

### 相対的な長さの単位

単位	説明
em	親要素のフォントサイズを1とした単位。親要素のフォントサイズが 16px の場合、1em は 16px になります。
ex	要素のフォントの小文字「x」の高さを1とした単位。
ch	要素のフォントの「0」の幅を1とした単位。
rem	ルート要素（html 要素）のフォントサイズを1とした単位。
vw	ビューポートの幅の 1% を1とした単位。
vh	ビューポートの高さの 1% を1とした単位。
%	他の基準となる大きさに対する割合（基準は状況によって異なります）

※ 値が「0」の場合は、単位は省略することができます。

## 角度の単位

グラデーション関数やトランスフォーム系プロパティで、度、グラード、ラジアン、回転の値で表される角度の値に使用される単位になります。

単位	説明
deg	度数法（度）で表します。0 ～ 360 までの数値に deg を付けて角度を表します。
grad	グラード法で表します。0 ～ 400 までの数値に grad を付けて角度を表します。
rad	弧度法（ラジアン）です表します。正円 1 周分を $2\pi$ とした数値で角度を指定します。
turn	角度を回転数で表します。正円 1 周分を 1 ターンとした数値に turn を付けて角度を表します。

MDN Web Docs : <https://developer.mozilla.org/ja/docs/Web/CSS/angle>

## 時間の単位

トランジション系プロパティやアニメーション系プロパティで、秒単位、ミリ秒単位で表わされる時間の値に使用される単位になります。

単位	説明
s	秒単位の時間を表す単位。
ms	ミリ秒単位の時間を表す単位。

MDN Web Docs : <https://developer.mozilla.org/ja/docs/Web/CSS/time>

# 色の指定方法

最近のコンピュータで利用できる標準的な色のシステムは 24 ビットです。これは赤、緑、青のチャンネルでそれぞれ 256 の値を持てるため、およそ 1670 万色の色を指定できることになります。

CSS で色を指定する方法は、カラーキーワードまたは、カラーコードにて指定します。

## カラーキーワード

カラーキーワードによる色を指定する方法は、非常に単純で理解しやすいものになります。カラーキーワードは、その色の名前そのものを入力するだけになります。

## RGB モデル

### 16 進数 RGB

CSS による色指定のもっとも一般的な指定方法になります。「#」（ハッシュ）に続けて、6 桁の 16 進数の数を続けます。6 桁の数には、0 から f までの 16 個の文字を使用します。この値の二文字はそれぞれ赤、緑、青のチャンネルの値を表しており、それぞれのチャンネルに 0～255 の 256 通りの値を指定できます。

記述	説明
#RRGGBB	16 進数 0 ～ f の 6 桁の数値を指定します。「#0000ff」は青となります。
#RGB	赤、緑、青のチャンネルの 2 桁ずつが同値の場合は、3 桁で指定することが可能です。 「#0000ff」は「#00f」と同じになり、青となります。

## RGB と RGBA

関数記法として、rgb() と rgba() に 3 つのカラーを 10 進数の値で渡して、色を指定する方法になります。

rgb() は、各チャンネル（赤、緑、青）に、0～255 の値を指定します。

rgba() は、各チャンネルに合わせて、アルファ値（透明度）を 0～1 の範囲で色の透明度を指定します。

記述	説明
rgb()	0 ～ 255 までの数値を「,」（カンマ）で区切って 3 つ指定します。
rgba()	rgb() に加えて、4 つ目の値で透明度を指定できます。0 が完全な透明で、1 が完全な不透明です。rgba(0, 0, 255, 0.5) は、不透明度 50% の青になります。