

# CSS の基本

## 目次

第1章	スタイルシートの基本.....	2
1.1	スタイルシートとは .....	2
1.2	CSS の書き方 .....	4
1.3	<span>タグ .....	11
1.4	<div>タグ .....	12

## 第1章 スタイルシートの基本

### 1.1 スタイルシートとは

HTML を補助して柔軟なデザインを可能にするのがスタイルシート（CSS）です。これによって、HTML でテキストのタグ付けや画像、テーブルなどのタフ付けを行って、タグで指定した部分に関連させて CSS がテキストの書式、特定の範囲の幅、位置、マージンなどを指定してデザインを行います。

Web ページの見栄えを整えるにあたり、CSS では次の3つの要素を用います。

**セレクトア** - どの HTML タグに対して見合えを調整するか

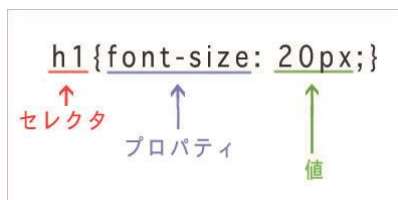
**プロパティ** - どういった内容のデザインを施すか

**値** - プロパティはどの程度か

簡単に言えば、「何に、どのように、どのくらい」デザインを施すかということを示します。

説明だけではわかりにくいと思うので、実際にスタイルを作ってみましょう。

たとえば、「h1 タグ」に対して、文字の大きさを 20 ピクセルにしたいときは次のように設定します。



コードの間に半角スペースや tab、改行を入れても問題はありません。

全角スペースは絶対に使わないようにしましょう

#### プロパティを複数指定したいとき

ひとつのセレクトアに対して、プロパティは複数指定することができます。

たとえば、フォントサイズを 20px・文字の色を青にしたいときには、

`h1{font-size: 20px, color: blue;}` のように書きます。

また、コードには空白部分は無視されるという性質があるので、`{ }` で囲んだ設定を見やすく並列に並べてもいいでしょう。

```
h1{  
  font-size: 20px,  
  color: blue;  
}
```

## 1.2 プロパティの種類

プロパティにはほかにも、文字幅の調整や背景色の指定など目的別にたくさんの種類があります。「CSS リファレンス」などで検索すれば、様々なサイトが出てくるのでいろいろ試してみましょう。

下記の表は、CSS を書く上でよく使われるプロパティをまとめています。プロパティ自体は他にも多くありますが、紹介しているものは覚えておくと作業効率も上がるでしょう。

プロパティ	説明文
color	文字色を指定する
background	背景の指定をする
background-color	背景の色を指定する
font-family	フォントの種類を指定する
font-weight	フォントの太さを指定する
line-height	行の高さを指定する
text-align	行の揃え位置を指定する

width	幅を指定する
height	高さを指定する
margin	マージンの指定をする
padding	パディングの指定をする
border	ボーダーの色や太さを指定する
position	要素の配置方法を指定する
display	要素の表示形式を指定する
float	左か右に寄せて配置する
z-index	要素の重なるの順序を指定する


## 2 CSS の書き方

CSS には書き方が 3 通りある。

- 「style 属性」として記述 (h, p タグなど「タグの中」に記述)
- 「style タグ」の中に記述 (HTML ファイルの「head タグの中」に記述)
- 外部ファイルに記述 (link タグで呼び出す)

### 1.3.1 「style 属性」として記述

HTML ファイル内の特定の要素に直接 CSS をあてる方法です。

 サンプル - Microsoft internet Explorer	<b>ソース(sample1_3_1.html)</b>  <pre> &lt;html&gt; &lt;head&gt; &lt;meta charset="UTF-8"&gt; &lt;title&gt;サンプル&lt;/title&gt; &lt;/head&gt; </pre>
--	---

## 第一章 HTMLの基本

### 1.1 タグについて

## 第二章 文字の装飾

### 2.1 段落を指定する

### 2.2 改行する

### 2.3 線を引く

クラス指定も使えます。

```
<body>
```

```
<h1 style="color: orange; background-color: gray;">
```

```
第一章 HTMLの基本</h1>
```

```
<h2 style="color: green; font-size: 20pt;">
```

```
1.1 タグについて</h2>
```

```
<h1 style="color: orange; background-color: gray;">
```

```
第二章 文字の装飾</h1>
```

```
<h2 style="color: green; font-size: 20pt;">
```

```
2.1 段落を指定する</h2>
```

```
<h2 style="color: green; font-size: 20pt;">
```

```
2.2 改行する</h2>
```

```
<h2 style="color: green; font-size: 20pt;">
```

```
2.3 線を引く</h2>
```

```
<p style="border: double; color: purple;
```

```
background-color: yellow; font-size: 16pt;">
```

```
クラス指定も使えます。</p>
```

```
</body>
```

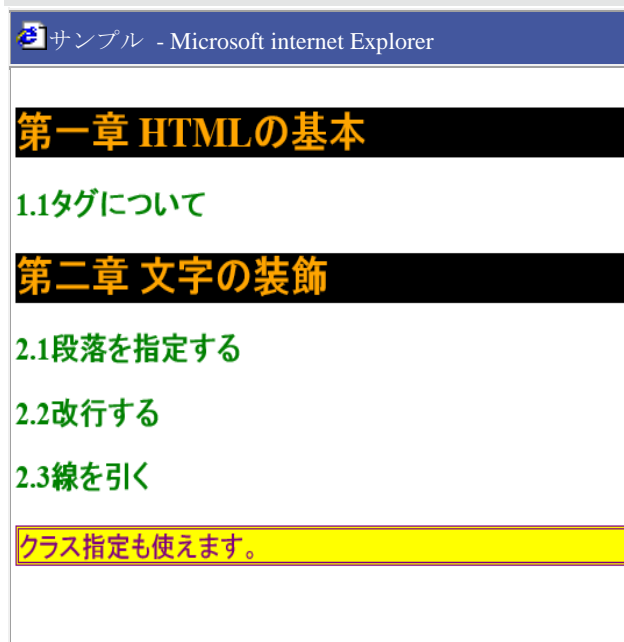
```
</html>
```

### 1.3.2 「style タグ」の中に記述

HTML ファイル内の head タグの中に、style 要素として CSS を記述する方法です。

#### ■ 最も基本的な書き方

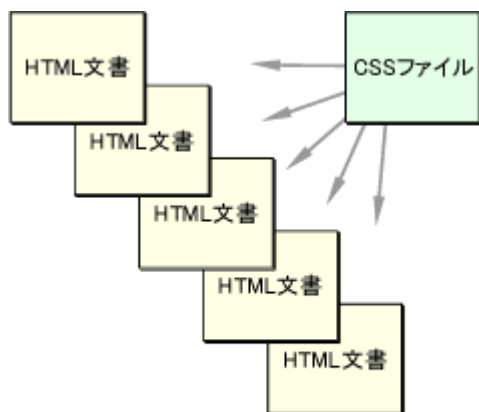
セレクトア { プロパティ : 値 ; }



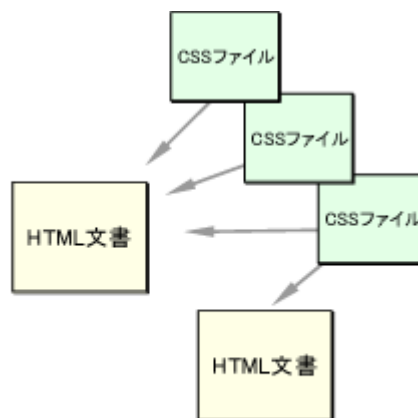
ソース(sample1_3_2.html)
<pre>&lt;html&gt; &lt;head&gt; &lt;meta charset="UTF-8"&gt; &lt;title&gt;サンプル&lt;/title&gt; &lt;style type="text/css"&gt; h1 {     color: orange;     background-color: black; }  h2 {     color: green;     font-size: 20; }  p {     color: purple;     border: double;     background-color: yellow;     font-size: 16pt; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;     &lt;h1&gt;第一章 HTMLの基本&lt;/h1&gt;     &lt;h2&gt;1.1タグについて&lt;/h2&gt;     &lt;h1&gt;第二章 文字の装飾&lt;/h1&gt;     &lt;h2&gt;2.1段落を指定する&lt;/h2&gt;     &lt;h2&gt;2.2改行する&lt;/h2&gt;     &lt;h2&gt;2.3線を引く&lt;/h2&gt;     &lt;p&gt;クラス指定も使えます。&lt;/p&gt; &lt;/body&gt; &lt;/html&gt;</pre>

### 1.2.3 外部ファイルに記述

上記2つの方法では、HTML ファイルの中に CSS を記述していたが、3つ目のこの方法では「外部ファイル」として拡張子は、.css を記述し、複数のページに適用します。



一枚のCSSファイルで  
複数の文書のスタイルを一括管理できる



CSSファイルを分割して  
文書ごとに必要なスタイルを適用できる

■ <link rel="stylesheet" href="URL (.css) " type="text/css">

サンプル - Microsoft Internet Explorer	スタイルシートファイル (mydesign.css) のソース
<b>第一章 HTMLの基本</b>	<pre>@CHARSET "UTF-8";</pre>
<b>1.1タグについて</b>	<pre>h1 {</pre>
段落は16ptで、青色の文字です。	<pre>color: orange; background-color: black; }</pre>
<b>第二章 文字の装飾</b>	<pre>h2 {</pre>
<b>2.1段落を指定する</b>	<pre>color: green; font-size: 20pt; }</pre>
<b>2.2改行する</b>	<pre>p</pre>
<b>2.3線を引く</b>	<pre>color: blue; font-size: 16pt; }</pre>
クラス指定も使えます。	<pre>.mystyle { border: double 5pt; color: pink; }</pre>

ソース(sample1_3_3.html)
<pre> &lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt; &lt;meta charset="UTF-8"&gt; &lt;title&gt;サンプル&lt;/title&gt; &lt;link rel="stylesheet" href="../style/mydesign.css" type="text/css" /&gt; &lt;/head&gt; &lt;body&gt;   &lt;h1&gt;第一章 HTMLの基本&lt;/h1&gt;   &lt;h2&gt;1.1 タグについて&lt;/h2&gt;   &lt;p&gt;段落は16ptで、青色の文字です。&lt;/p&gt;   &lt;h1&gt;第二章 文字の装飾&lt;/h1&gt;   &lt;h2&gt;2.1 段落を指定する&lt;/h2&gt;   &lt;h2&gt;2.2 改行する&lt;/h2&gt;   &lt;h2&gt;2.3 線を引く&lt;/h2&gt;   &lt;p class="mystyle"&gt;クラス指定も使えます。&lt;/p&gt; &lt;/body&gt; &lt;/html&gt; </pre>

## 1.3 CSS の指定方法

### 1.3.1 タグ名で適用先を指定

#### タグ名 {～}

ひとつめはタグで指定する方法です。タグで指定する場合は、タグ名をそのまま書けば OK です。a{～}やdiv{～}、img{～}といった具合です。

ソース(sample1_3_2.html)
<pre> &lt;html&gt; &lt;head&gt; &lt;meta charset="UTF-8"&gt; &lt;title&gt;サンプル&lt;/title&gt; &lt;style type="text/css"&gt;  p {     color: purple;     border: double;     background-color: yellow;     font-size: 16pt; }  &lt;/style&gt; &lt;/head&gt; </pre>



```
<body>
  <p>これは一段落です</p>
  <p>これは二段落です</p>
  <p>これは三段落です</p>
</body>
</html>
```

### 1.3.2 id 名で指定

#### #id 名{~}

HTML では<タグ名 id="id 名">のように id が指定され、そして同じ id 名はページ内に 1 回しか使えません。この id 名をデザインの適用先に指定する場合は #id 名 {} というように id 名の前に # をつけましょう。

ソース(sample1_3_2.html)
<pre>&lt;html&gt; &lt;head&gt; &lt;meta charset="UTF-8"&gt; &lt;title&gt;サンプル&lt;/title&gt; &lt;style type="text/css"&gt;  #p1 {     color: purple;     border: double;     background-color: yellow;     font-size: 16pt; }  &lt;/style&gt; &lt;/head&gt; &lt;body&gt;   &lt;p id=p1&gt;これは一段落です&lt;/p&gt;   &lt;p&gt;これは二段落です&lt;/p&gt;   &lt;p&gt;これは三段落です&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>

### 1.3.3 class 名で指定

#### class 名{~}

HTML では<タグ名 class="class 名">というように指定をし、class 名でセレクタを指定する場合は .class 名{~} のように、ドット(ピリオド)を前につけます。

試しに class=" myclass" の文字色をオレンジにしてみましょう。

ソース(sample1_3_2.html)
<pre>&lt;html&gt; &lt;head&gt; &lt;meta charset="UTF-8"&gt; &lt;title&gt;サンプル&lt;/title&gt; &lt;style type="text/css"&gt;  .p1 {     color: purple;     border: double;     background-color: yellow;     font-size: 16pt; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;     &lt;p&gt;これは一段落です&lt;/p&gt;     &lt;p class ="myclass"&gt;これは二段落です&lt;/p&gt;     &lt;p&gt;これは三段落です&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>

### 1.3 「class」と「id」は役割が異なる

この「class」と「id」は、どちらでもよい方法が2種類用意されているわけではありません。この2つは、明確に役割が異なります。そこで今回は、「class」と「id」の使い分けについてご紹介致します。

#### class 属性を使ったスタイルの適用例

スタイルを適用する方法として「class」を使う場合は、次のようなソースになります。

■HTML :

```
<p class=" mystyle">クラスを使った例</p>
```

■スタイルシート :

```
p.mystyle {
    border: double 5pt;
    color: pink;
```

}対象の要素名と class 名との間に「.」(ドット)記号を記述します。要素名は省略することもできます。

#### id 属性を使ったスタイルの適用例

スタイルを適用する方法として「id」を使う場合は、次のようなソースになります。

■HTML :

```
<p id=" mystyle">アイディーを使った例</p>
```

■スタイルシート :

```
p#mystyle1 { color: red; }
```

対象の要素名と id 名との間に「#」(シャープ)記号を記述します。この場合も、要素名は省略することもできます。

### class 属性と id 属性の違いと、使い分け方法

上記のサンプルソースは、スタイルシートの適用に「class」を使うか「id」を使うかの差だけで、ブラウザ上での表示効果は全く同じです。どちらも、対象の文字が赤色になります。

では、「class」と「id」の違いは何でしょうか？

実は、両者の役割(利用目的)には大きな差があります。

■id :

「固有の名前を割り当てる」

→ 同じ id 名は、1 ページ中に **1 度しか使えない**

class 属性が対象の「種類・部類」を表しているに過ぎないのに対して、id 属性は対象に「固有の名前」を付けて**一意に表す**ために使われます。ですから、1 ページ中に同じ id 名は1度しか使えません。

■class :

「種別名を割り当てる」

→ 同じ class 名を、1 ページ中に**何度でも使える**

#### ・スタイルシートの優先順位

もし同じ設定がされた場合は、**インラインでタグに直接スタイルを指定した場合が最も優先**され、次に<head>〜</head>間に<STYLE>タグを使って設定した場合、最後にスタイルシートファイルでの設定という優先順位になります。

## 1.4 <span>タグ

<span style="color:blue;font-size:12pt">〜</span>のように記入すればこの中に記入された文字はフォントの大きさが 12pt で青色になります。自分で指定したスタイル以外は何も効果がありません。

サンプル - Microsoft Internet Explorer

span を使えば、部分的にスタイルを設定できます。

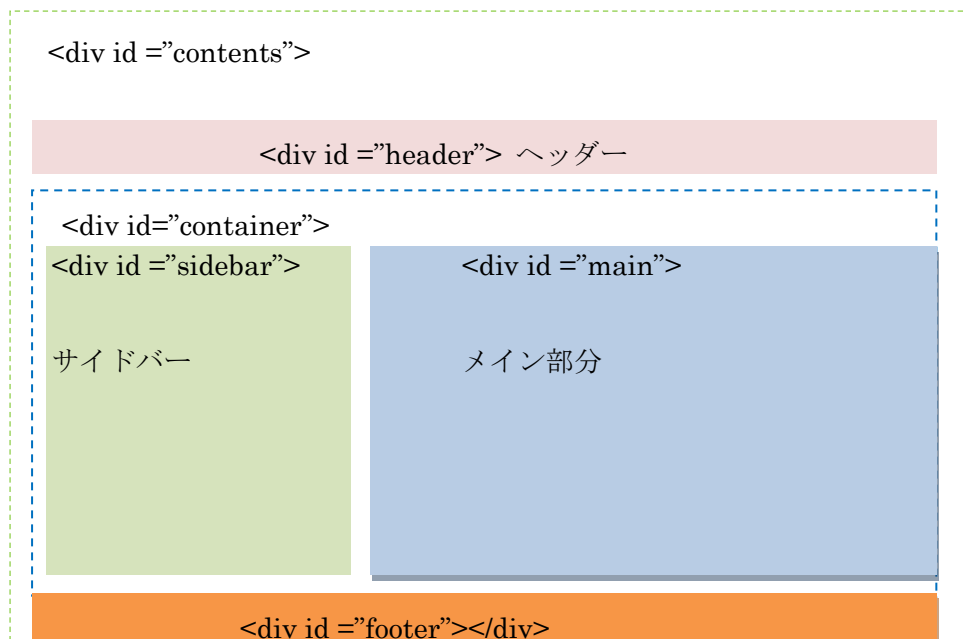
ソース (sample1\_3.html)

```
<html>
<head>
<meta charset="UTF-8">
<title>サンプル</title>
</head>
<body>
  spanを使えば、
  <span style="color: red;">部分的にスタイルを</span>設定できます。
</body>
</html>
```

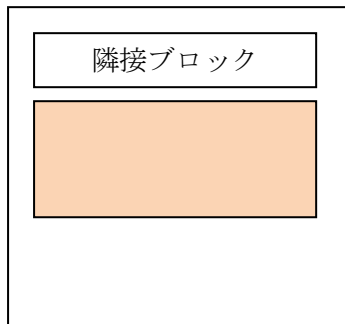
### 1.5 <div>タグ

<div id=" " ">~</div>

<div class=" " "></div>

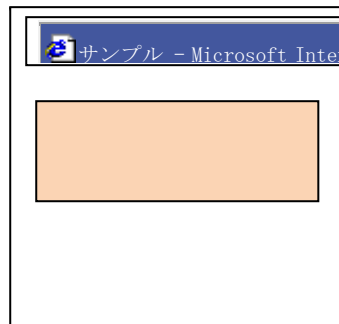


position:static



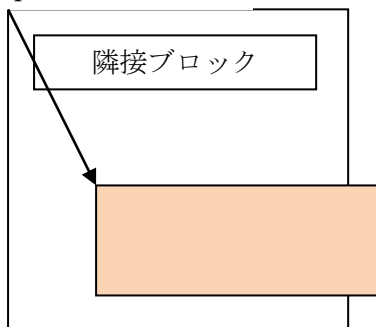
通常の配置

position:fixed



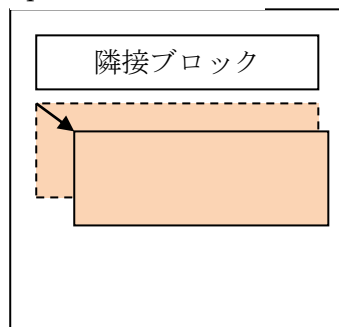
画面の左上を基準とした座標で指定する。

position:absolute



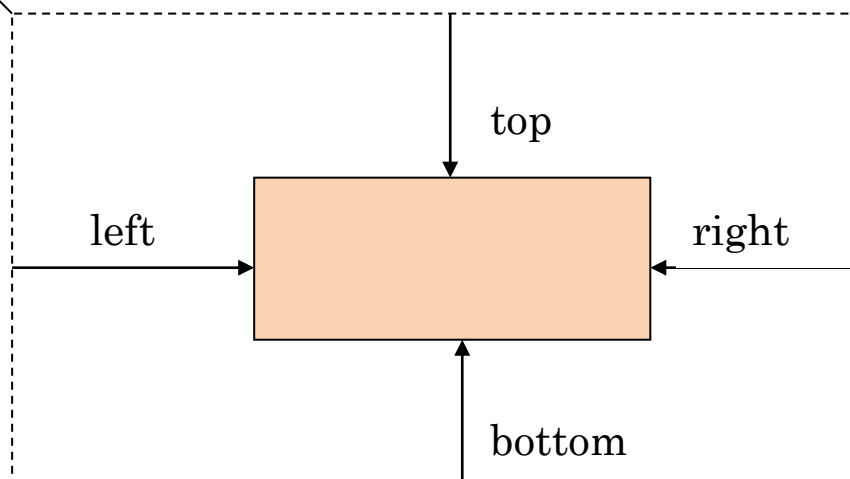
親要素の左上を基準とした座標で指定する

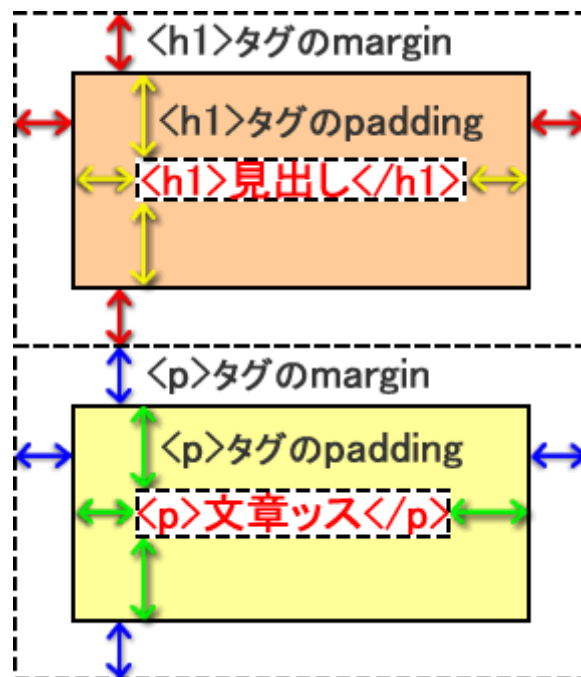
position:relative



本来の配置される場所を基準とした座標で指定する

座標の基準





margin: 値1 値2 値3 値4

値が1個 値1=上下左右

値が2個 値1=上下 値2=左右

値が3個 値1=上 値2=左右 値3=下

値が4個 値1=上 値2=右 値3=下 値4=左

CSS の display プロパティ

display: block

display: inline

display: inline-block

ブロックレベル要素

文書の骨組みとなる要素です。見出し要素 (h1, h2, h3...h6)、p、ul、ol、li、div、table など。要素の前後に改行が入り、カチッとしたブロックを積んでいくような表示になります。

- 縦に積まれていく
- 幅 width と高さ height が指定できる
- 上下左右に margin を指定できる
- 上下左右に padding を指定できる
- text-align は要素の中身に適応される。
- vertical-align は指定できない。

## インライン要素

ブロックレベル要素の中身として使われる要素です。a、span、strong など。

テキストの一部として扱われるので、要素の前後には改行は入らず、テキスト状に横に横にずーっと流れていくような表示になります。

こんなかんじ。

- 横に並んでいく
- 幅 width と高さ height は指定できない。幅や高さは文字列の長さや文字の大きさなど、内容物のサイズ。
- 左右だけ margin を指定できる
- 左右に padding を指定できる。(実は上下も指定できるけど、前後の行や要素にかぶってしまうので、あまり効果はわからない)
- text-align を親ブロックに付けることで指定できる。
- vertical-align を指定できる。

ソース (sample.css)

```
img.sample1 {display: block;}
.sample2 {display: inline;}
```

ソース (sample.html)

```
<html>
<head>
<link rel="stylesheet" href="sample.css" type="text/css">
</head>
<body>
<p>
画像をブロックレベルで表示
します。
</p>
<h3 class="sample2">見出し</h3>
<p class="sample2">見出しと段落をインラインで表示します。</p>
</body>
</html>
```

