

# 01 HTML, CSS とは

## 1. HTML, CSS

### 1-1. HTML とは

HTML とは Hyper Text Markup Language のそれぞれの単語の頭文字をとった呼称です。HTML という言語で作成されたファイルのことを HTML ファイルと呼び、ファイルの拡張子は「.html」となります。WEB ページの基本となるファイルで、表示させたい文章などはすべてこの中に記述します。また、HTML ファイルは他のドキュメントやファイルとリンクさせることができるのも特徴の一つです。

### 1-2. HTML の役割

単純な文章に意味や構造などを持たせることができるのが HTML です。「タグ」とよばれる記法を用いて、文章のそれそれが持つ「見出し」や「段落」といった意味をブラウザに伝えるのが HTML の役割です。HTML では基本、テキストはタグと呼ばれる印で囲みます。タグの種類によってテキストに「見出し」や「段落」といった様々な意味を与えます。これを要素と言います。タグを使うときは、開始タグと閉じタグで囲む必要があります。

### 1-3. CSS とは

CSS とは Cascading Style Sheet のそれぞれの単語の頭文字をとった呼称です。

CSS という言語で作成されたファイルのことを CSS ファイルと呼び、ファイルの拡張子は「.css」となります。

### 1-4. CSS の役割

WEB ページをデザインするための記述です。CSS によって文字の大きさや行間、文字や背景の色、配置などを自由にデザインすることができます。HTML ファイルの中に直接書き込む方法と、HTML とは別のファイルに記述し、それを HTML ファイルに読み込む方法があります。

### 1-5. 以下の文章をブラウザでみてみましょう

通常の文章（資料：html\_css/sec01/no01-05/nomal\_text/index.html）

```
これは見出します。  
これは本文です。これは本文です。これは本文です。これは本文です。これは本文です。これは本文です。  
箇条書き 1 つ目  
箇条書き 2 つ目  
箇条書き 3 つ目
```

html の文章（資料：html\_css/sec01/no01-05/html/index.html）

```
<h1> これは見出します。</h1>  
<p> これは本文です。これは本文です。これは本文です。これは本文です。これは本文です。これは本文です。</p>  
<ul>  
<li> 箇条書き 1 つ目 </li>  
<li> 箇条書き 2 つ目 </li>  
<li> 箇条書き 3 つ目 </li>  
</ul>
```

css で以下の記述を追加し見出しに赤色を指定した文章（資料：html\_css/sec01/no01-05/html\_css/index.html style.css）

```
h1 { color: red; }
```

# 02 WEB サイトの基本構造

## 1. フォルダの構造

### 1-1. フォルダ

WEB ページは HTML ファイルや CSS ファイル、画像データなどがセットになって初めてその内容がブラウザに表示されます。

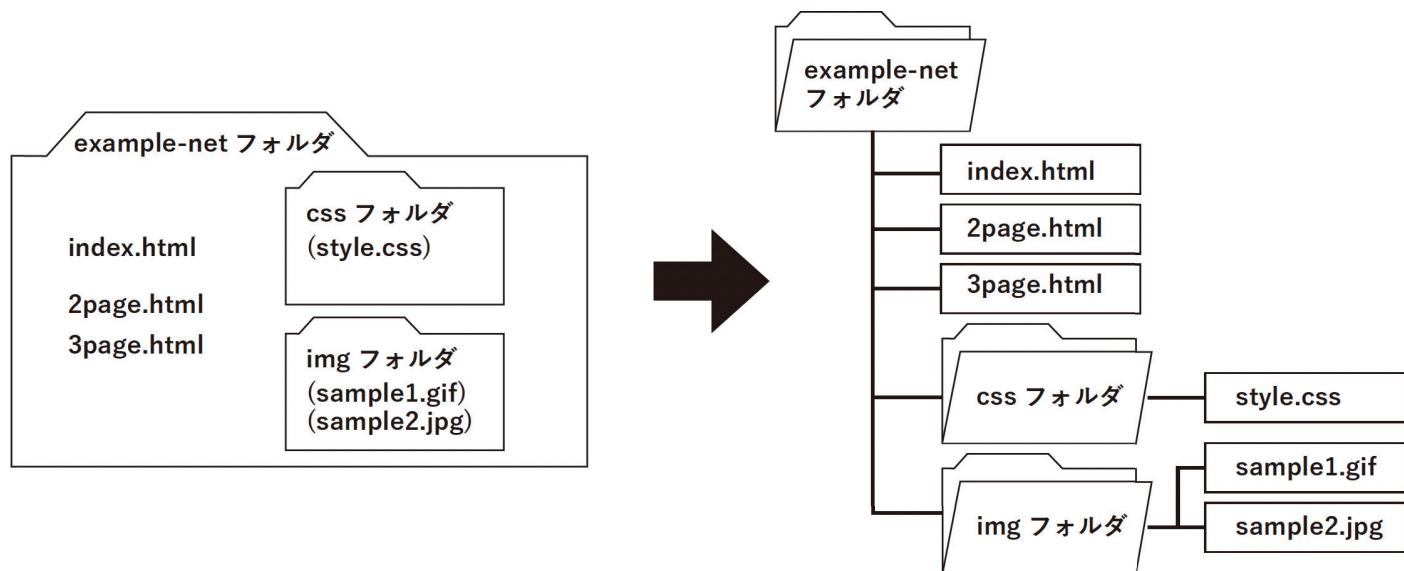
### 1-2. ファイルの命名

ファイルをアップロードするサーバは半角英数字にしか対応していないことが多いため、ファイル名も半角英数を使用します。

### 1-3. index.html

WEB ページとしてアップロードするフォルダ内には、index.html が必要です。

サーバーが HTML フォルダの中から最初に探して表示するもの、主に WEB ページのトップページにあたります。



## 2. リンクと階層

### 2-1. リンク

リンクとはハイパーテキストにおいて文書と文書や複数のファイル同士を結びつけることを言います。

リンクの種類	リンクの仕方
内部リンク	自分の WEB サイト内の各ページ間のリンク。
外部リンク	他の WEB サイトへのリンク。
ページ内リンク	同じページ内でのリンク。

## 2-2. パス

パスとは、ファイルの場所を指定する方法のことです。別のページ・ファイルにリンクしたり、あるいは自分のサイト内の別のページにリンクしたりする場合や、画像を表示したい時などに、それらのファイルがどこにあるのかをパスを用いて指定する必要があります。

## 2-3. 絶対パスと相対パス

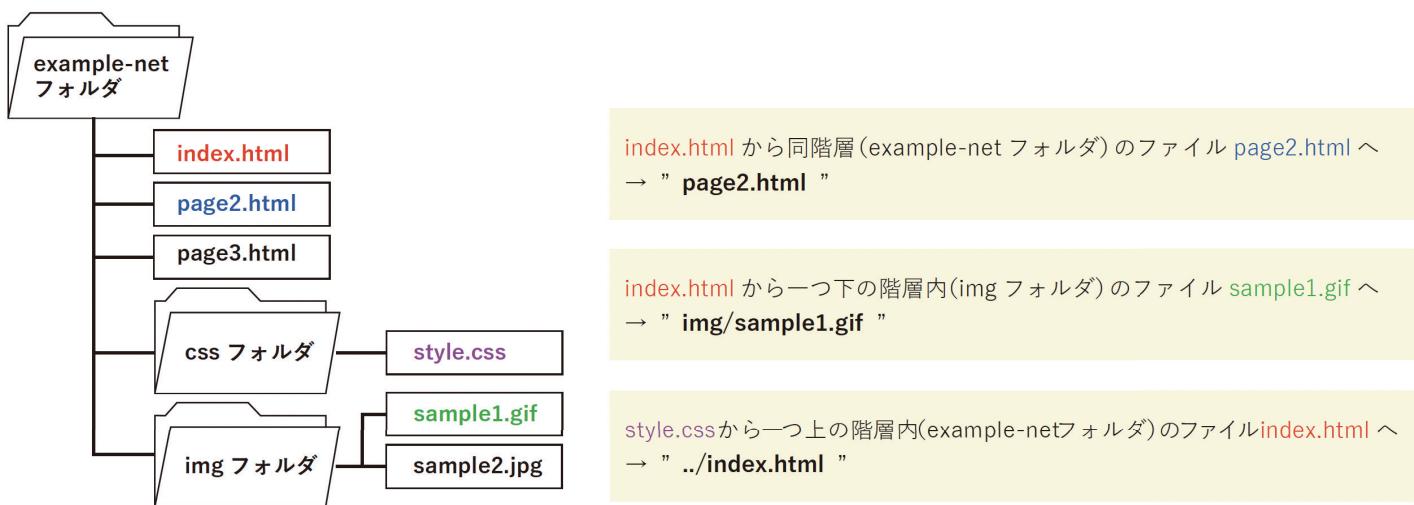
### 絶対パス

絶対パスとは、`http://` から始まるアドレス (URL) を使ってファイルの場所を指定することです。目的の場所そのものの位置を指定します。主に他のサイトにリンクしたりする場合などに使います。

### 相対パス

相対パスとは、基準となるファイルから見て任意のファイルの場所がどこにあるのかを相対的に指定します。

階層	パスの記述方法
現在位置の同階層のファイルへ	ファイル名
現在位置の一つ下の階層 (フォルダ) 内のファイルへ	フォルダ名 / ファイル名
現在位置の二つ下の階層 (フォルダ) 内のファイルへ	フォルダ名 / フォルダ名 / ファイル名
現在位置の一つ上の階層 (フォルダ) 内のファイルへ	../ ファイル名
現在位置の二つ上の階層 (フォルダ) 内のファイルへ	../../ ファイル名



# 03 HTML の基本

## 1. HTML の基本

### 1-1. HTML の基本構造

<>で囲まれた文字列を「タグ」とよびます。

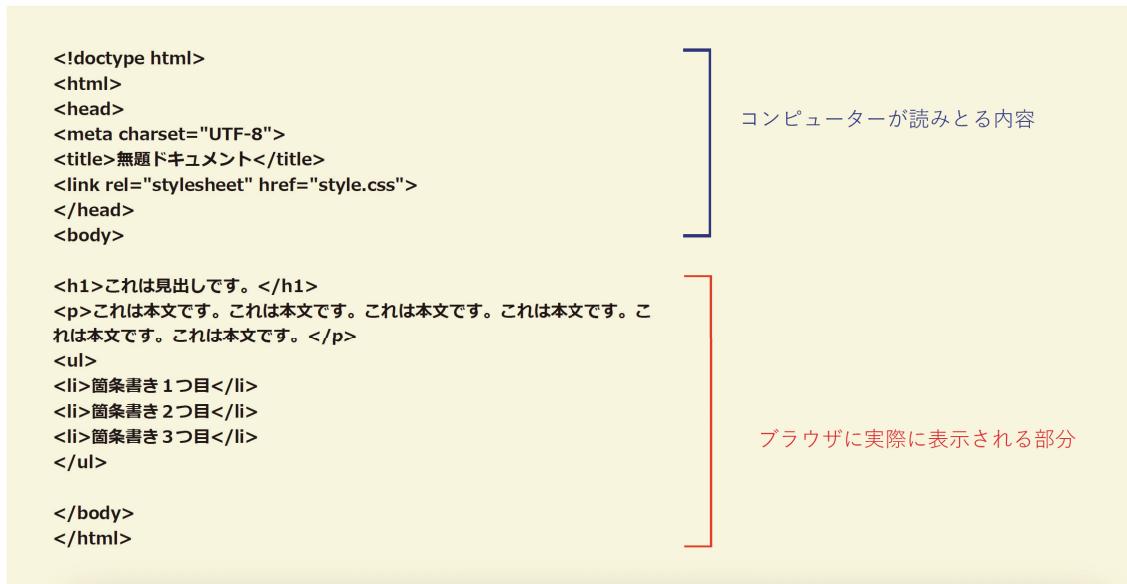
<>の中は基本的に半角英数小文字で入力し、タグはほとんどが <html>～</html> のように「開始タグ」と / (スラッシュ) で始まる「終了タグ」により構成されています。

開始タグを記述すると「ここから○○です」という宣言となり、終了タグを記述すると「○○はここまで」という宣言になります。

このようにして入力した HTML のことを「HTML ソース」や「コード」と呼びます。

HTML は <html> という開始タグで始まり、</html> という終了タグで終わる構造になっています。その中に「head 要素」と「body 要素」を記述するのが大枠となります。「head 要素」の中の内容は多くはコンピューターが読みとる内容です。

文章のタイトル、キーワード、ページ説明を記述したり、文字コードの種類を定義したりします。内容の多くはブラウザには表示されませんが WEB ページを形成する上で欠かせないものが格納されたエリアです。一方、「body 要素」の中の内容の大半はブラウザに実際に表示される部分です。



```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>無題ドキュメント</title>
<link rel="stylesheet" href="style.css">
</head>
<body>

<h1>これは見出します。</h1>
<p>これは本文です。これは本文です。これは本文です。これは本文です。これは本文です。</p>
<ul>
<li>箇条書き 1つ目</li>
<li>箇条書き 2つ目</li>
<li>箇条書き 3つ目</li>
</ul>

</body>
</html>
```

コンピューターが読みとる内容

ブラウザに実際に表示される部分

# 04 HTML の要素（タグ）と属性

## 1. タグ

### 1-1. タグと要素

<開始タグ> 内容 </終了タグ> のひとまとまりを「要素」と呼びます。

HTML のタグは入れ子構造にするという書き方の規則があります。<タグ 1><タグ 2><タグ 3>で始まったものは、</タグ 3></タグ 2></タグ 1>順で閉じる必要があります。



### 1-2. タグと属性

タグに性質を与えるのが「属性」です。

属性はタグに対して必ず付けなくてはならないものではなく、オプションのような扱いとしてタグの中に追記していくものです。基本的な書き方は<タグ 属性名 = “値”>となります。属性の手前は半角スペースを入れて区切れます。

例として、ハイパーリンクのタグは<a href=“リンク先 URL” target=“\_blank”>と記述しますが、この場合 a がタグ、href が属性、リンク先 URL が値ということになります。



# 05 HTML ファイルを作成する

## 1. HTML をマークアップする方法

### 1-1. エディタ

WEB ページはすべてテキストファイルでできています。そのため、制作にはどのようなテキストエディタを使用しても構いません。Windows 標準のワードパッドやメモ帳、Mac であればテキストエディットで制作することができます。その他、無料でダウンロード可能なテキストエディタも配布されているので、それらを利用することで作業効率を上げることができます。

## 2. 基本的なタグの性質を確認しながら実際にマークアップする

### 2-1. 見出し (<h1> ~ <h6>)

<h1>~</h1> で囲まれた文章が、最上位の見出し（大見出し）です。

タグの前後で改行され、上下に 1 行分のスペースがとられ、太字になります。

※<hn>~</hn> の n 部分には 1~6 の数字が入り、1 から順に文字のサイズが小さくなります。

とくに <h1> は Web ページの中で一番重要な部分に使う見出し要素なので、1 つのページに 1 個しか設置しないのが基本です。

### 2-2. 段落 (<p>), 改行 (<br>)

<p>~</p> で囲まれた部分が 1 つの段落として表示されます。<p> タグの前後に、1 行分のスペースがとられ、改行されます。

<br> は「改行してください」という宣言です。「ここを改行してください」という指示なので、開始タグ・終了タグに分けません。改行したいポイントに <br> をひとつ配置します。

### 2-3. 画像の表示 (<img>)

 は、画像を表示するタグです。src(source) は img タグの属性です。”” 内には画像の保存場所（ファイル名含む）を記述します。

### 2-4. ハイパーリンク (<a>)

リンクとは、HTML 上のテキストや画像をクリックすると、別のページにジャンプすることです。

別のページにジャンプするということは、「HTML ファイル」から「他の HTML ファイル」に移動する、ファイルとファイルを繋ぐ役割をします。このような文字や画像にリンクを施すことを「リンクを貼る」、ファイル同士を繋ぐことを、「リンクさせる」などと表現します。

リンクを設定するには以下のようなタグを記述します。

<a href=" リンクさせるファイルの URL "> リンクを設定する文字や画像 </a>

### 2-5. リスト (<ul>,<ol>,<li>)

リスト表示したい項目に <li> タグをつけて <ul> ~ </ul> または <ol> ~ </ol> の間に記述します。

<ul> は Unordered List の略で、リスト項目に順序を付けない場合に使用、デフォルトで項目の前にドットがつきます。

<ol> は Ordered List の略で、リスト項目に順序を付ける場合に使用します。デフォルトで項目の前に番号が振られます。

### 2-6. テーブル (<table>,<tr>,<th>,<td>)

表形式のデータ、つまり、行と列の組み合わせによるセルに含まれたデータによる二次元の表で表現される情報です。

<table></table> は表全体を定義、<tr></tr> は行を定義、<th></th> は見出しセル、<td></td> はデータを包含するセルを定義します。

### 2-7. 汎用コンテナー (<div>,<span>)

単体では特に意味を持たないタグですが、

<div></div> で囲った部分をブロックレベル要素としてグループ化することができる汎用的なタグです。ブロックレベル要素としてグループ化を行いますので、前後に改行が入ります。

<span></span> で囲った部分をインライン要素としてグループ化することができるタグです。インライン要素で文章の一部として利用されますので前後に改行は入りません。

それぞれのタグでグループ化することで、指定した範囲にスタイルシートを適用したりすることができます。

### 3. HTML をマークアップする

#### 3-1.

プレーンテキスト (資料 : html\_css/sec05/no03-01/nomal\_text/index.html )

それぞれのライフスタイルにあったビジネスの拠点〇〇コワーキング 大見出し  
 長野県〇〇市のコワーキングスペースです。 段落  
 フリーランスの方はもちろん、短い時間に集中して仕事をしたいビジネスマンや、学生さんにもおすすめです。  
 設備 中見出し  
 以下の設備は無料で使用することができます。 段落  
 • wifi / 電源 リスト  
 • コピー機  
 • スタンドライト  
 利用料金 中見出し  
 ドロップイン 500 円 /2 時間 テーブル  
 会員 10,000 円 /1 ヶ月  
 会議室 1,000 円 /1 時間  
 お問い合わせはこちらから リンク

html をマークアップ (資料 : html\_css/sec05/no03-01/html/index.html )

```

<h1> それぞれのライフスタイルにあったビジネスの拠点〇〇コワーキング </h1>
<div>
  <p> 長野県〇〇市のコワーキングスペースです。<br> フリーランスの方はもちろん、短い時間に集中して仕事をしたいビジネスマンや、学生さんにもおすすめです。</p>
</div>
<h2> 設備 </h2>
<p> 以下の設備は <span> 無料 </span> で使用することができます。</p>
<ul>
  <li> wifi / 電源 </li>
  <li> コピー機 </li>
  <li> スタンドライト </li>
</ul>
<h2> 利用料金 </h2>
<table>
  <tr>
    <th> ドロップイン </th>
    <td> 500 円 /2 時間 </td>
  </tr>
  <tr>
    <th> 会員 </th>
    <td> 10,000 円 /1 ヶ月 </td>
  </tr>
  <tr>
    <th> 会議室 </th>
    <td> 1,000 円 /1 時間 </td>
  </tr>
</table>
<a href="#" > お問い合わせはこちらから </a>

```

# 06 CSS の基本構造

## 1. CSS の基本構造

### 1-1. セレクタとは

CSS を使って文章にスタイルを適用するには、「セレクタ」と呼ばれるものを使用します。セレクタとは、「CSS のスタイルを適用する対象を選択するためのもの」です。様々な要素（タグ）によって意味付けをなされた HTML の文章を装飾するため、CSS は HTML 上の文章そのものではなく、タグを目印にしてスタイルを適用していきます。

### 1-2. セレクタの基本構造

セレクタの構造は、「セレクタ」「プロパティ」「値」の三つの組み合わせが基本的な形です。

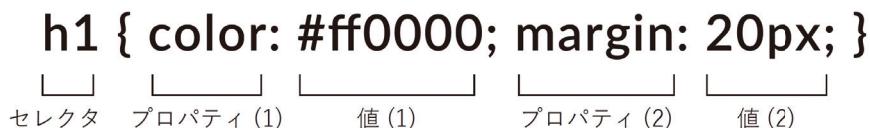
記述順もこれと同様で、見た目の順番どおり「○○の ×× を△△にする」と読み取ることができます。これらの一連の内容を「規則集合（ルールセット）」といいます。

セレクタ以降、左中括弧（{）で始まり、右中括弧（}）で終わる部分を「宣言ブロック」といいます。さらに中括弧の内側に、コロン（:）で区切った一帯の部分を「宣言」といいます。レイアウトや装飾の指定はこの宣言の部分によって表現されます。



セレクタ 1 つに対して複数のプロパティを加えていくには、値の後を必ずセミコロンで区切り、同じルールで記述していきます。これをくりかえしていくことで、HTML に様々な装飾を適応していきます。

宣言が 1 つの場合や、最後にくる宣言にはセミコロンは不要ですが、後から宣言を追加する場合にセミコロンの記述忘れの原因になることがあるので、常に記述するようにすると良いでしょう。



宣言には、半角スペースや改行、インデントなどを自由に入れることができます。Web サイトを一人で作成している分には、自分のルールで好きなように記述してもよいのですが、Web 制作の現場では、複数の人が CSS を作成し、管理するケースが多くあるため、ある程度決まったルールのもと記述していくと良いでしょう。

一般的な記述の方法として…

- セレクタのあと、中括弧の内側、コロンのあとに半角スペースを入れます。
- プロパティの前はタブキーでインデントし、最後の右中括弧は、改行して記述します。

h1 { color: #ff0000; }

h1 {  
color: #ff0000;  
margin: 20px;  
}

# 07 CSS のセレクタ

---

## 1. よく使用されるセレクタ

### 1-1. 基本的なセレクタの種類

CSS を使って文章にスタイルを適用するには、「セレクタ」と呼ばれるものを使用します。セレクタとは、「CSS のスタイルを適用する対象を選択するためのもの」です。様々な要素（タグ）によって意味付けをなされた HTML の文章を装飾するため、CSS は HTML 上の文章そのものではなく、タグを目印にしてスタイルを適用していきます。

### 1-2. 要素セレクタ

要素セレクタは、要素（タグ）名で対象を指定してスタイルを適応する最も基本的なセレクタです。

例えば、`h1 { color: #ff0000; }` と記述した場合、ファイル内に存在する `h1` 要素全てに `color: #ff0000;` が適応されます。

### 1-3. ID セレクタ、class セレクタ

#### ID セレクタ

HTML で特定の要素だけにつけられた ID 属性を目印にしてスタイルを適用するものです。

適用させたい要素に、HTML 側で ID 属性を使用して任意の ID 名（ここでは `example` とします）を付与します。（例：`<div id="example">`）ID 名は固有で、そのページ（HTML）で一度しか使用できません。

そして、CSS 側でその ID 名をセレクタとしてスタイルを記述します。ID であることを示すために ID 名の前に「シャープ (#)」を付けて記述します。（例：`#example { color: #ff0000; }`）

#### class セレクタ

HTML で特定の要素につけられた class 属性を目印にしてスタイルを適用するものです。

HTML 側では、スタイルを適用したい要素に class 属性を使用して任意の class 名（ここでは `content` とします）を付与します。（例：`<div class="content">`）class は同じ HTML の中に同一名を複数回使うことができます。

CSS 側ではその class 名をセレクタとしてスタイルを記述します。class 名であることを示すために class 名の前に「ドット(.)」を付けて記述します。（例：`.content { color: #ff0000; }`）

### 1-4. 複数セレクタ、子孫セレクタ

複数のセレクタに対して同じスタイルを適用させたい場合は、セレクタを「カンマ (,)」で区切って記述します。（例：`h1, p { color: #ff0000; }`）

ある要素の中にある特定の要素にスタイルを適用させるのが子孫セレクタです。例えば、クラス名 `content` の要素（`<div class="content">`）の中にある `p` 要素にだけスタイルを適応したいとすると、親となる要素（この場合 `.content` 要素）と子となる要素（この場合 `span` 要素）の順にセレクタを記述します。この際にセレクタとセレクタの間は半角スペースが必要です。（例：`.content p { color: #ff0000; }`）

# 08 CSS の特性

## 1. CSS のルールや特性

### 1-1. CSS が適用される優先順位 (資料 : [html\\_css/sec08/no01-01/about\\_css/index.html](http://html_css/sec08/no01-01/about_css/index.html) [style\\_1.css](http://style_1.css) [style\\_2.css](http://style_2.css))

CSS ではセレクタから始まる規則集合を 1 括りとして、縦方向に書き連ねていきます。

上に書いてあるものから順に読み込まれ、あとから（下に）書いたものが優先して適用されるのが基本ルールです。

また、もう一つの特性として「上書き」があります。同じ要素を示すセレクタが複数あった場合も、基本ルール同様に下に記述したものが優先的に適応されます。複数の同名セレクタで、同じプロパティにそれぞれ値が違う記述があったとすると、下に書いたものが上に書かれたスタイルに上書きされます。複数の同名セレクタで、それぞれ異なるプロパティが記述されている場合は「上のスタイル+下のスタイル」といったように両方のスタイルを合わせて適応されます。

### 1-2. セレクタの種類による適用の優先順位 (資料 : [html\\_css/sec08/no01-02/about\\_css/index.html](http://html_css/sec08/no01-02/about_css/index.html) [style.css](http://style.css))

1-2、3 で紹介した「要素セレクタ」、「ID セレクタ」、「class セレクタ」にも適用される優先順位があります。

優先順位としては、一番高いのが ID セレクタ、続いて class セレクタ、要素セレクタの順となります。これらは点数で表すことができ、子孫セレクタを詳細に記述した場合に合計点数の高いものが優先されます。

セレクタの種類ごとの点数

ID セレクタ  
(例 : #example)

100 点

class セレクタ  
(例 : .content)

10 点

要素セレクタ  
(例 : h1, p, div など)

1 点

子孫セレクタの点数換算例

`div p a { color: #ff0000; }`  
要素セレクタ (1 点) + 要素セレクタ (1 点) + 要素セレクタ (1 点) = 3 点

`#example a { color: #ff0000; }`  
ID セレクタ (100 点) + 要素セレクタ (1 点) = 101 点 ←最優先

`.content a { color: #ff0000; }`  
class セレクタ (10 点) + 要素セレクタ (1 点) = 11 点

## 2. ブロックレベルとインライン

### 2-1.1. ブロックレベル要素, インライン要素

HTML で定義されている要素のうちの多くは、ブロックレベル要素か、インライン要素に分類されます。

これらは CSS でインライン要素をブロックレベルにしたり、その逆を行ったりすることができます。

### 2-2. ブロックレベル要素

ブロックレベル要素は、見出し・段落・表など、文書を構成する基本となる要素で、一つのブロック（かたまり）として認識されます。CSS で高さやマージンなどの設定が可能です。また、一般的なブラウザでは前後に改行が入ります。

<div>,<h1> ~ <h6>,<p>,<table>,<ul> など

### 2-3. インライン要素

インライン要素は、主にブロックレベル要素の内容として用いられる要素で、テキストの一部として扱われます。一般的なブラウザでは前後に改行が入りません。

<br>,<span>,<strong> など

### 2-4. 置換インライン要素

インライン要素の中で画像やフォーム部品など、ほかのものに置き換わる要素のことを置換インライン要素と言います。

または、インライン要素の中で幅と高さがあり、CSS でそれらの設定が可能なものをさします。

<img>,<input>,<iframe> など

# 09 擬似要素・擬似クラス

## 1. 擬似要素と擬似クラス

### 1-1.

#### 擬似要素

擬似要素とは文章の前後や、1文字目、1行目など、要素の一部に対して影響を与える事ができるものです。

::before (要素の前に指定した内容を追加する)  
 ::after (要素の後に指定した内容を追加する)  
 ::first-letter (要素の最初の1文字目に指定した内容を追加する)  
 ::first-line (要素の最初の一行目に指定した内容を追加する) など

例：::first-letter (資料：html\_css/sec09/no01-01/pseudo/index.html style.css)

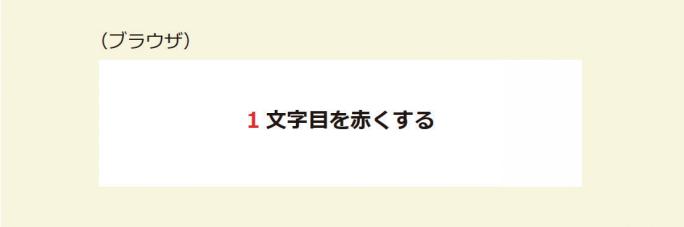
(HTML)

```
<h1>1 文字目を赤くする </h1>
```

(CSS)

```
h1::first-letter { color: #ff0000; }
```

(ブラウザ)



#### 擬似クラス

擬似クラスは指定したもの全体に特定の条件下で影響を与えるものです。

:link (a要素が未訪問を表している場合を指定)  
 :visited (a要素が訪問済を表している場合を指定)  
 :active (要素が選択されている場合を指定)  
 :hover (要素をマウスオーバーしている場合を指定)  
 :nth-child(n) (要素内の隣接している要素のn番目を指定) など

例：:hover (資料：html\_css/sec09/no01-01/pseudo/index.html style.css)

(HTML)

```
<a> マウスオーバー時赤くなる </a>
```

(CSS)

```
a:hover { color: #ff0000; }
```

(ブラウザ)

