



このコーナーは、HTML、CSS のことならなんでもおまかせの「マリー」と、うさぎ界 No.1 とも言われているデザイナー「ナムー」が、WEB ページ作成のコツを解説していきます。少し専門的な内容も含まれていますが、分からないことは積極的に調べて、1つ1つスキルアップをしていきましょう！



マリー

HTML、CSS のプロフェッショナル。うさぎだけど好きなブラウザは Fire Fox。



ナムー

うさぎ界を知り尽くした凄腕デザイナー。うさぎだけど好きな野菜はセロリ。



「競合ショップとの差別化するために、ボクのショップにオリジナル性を出したいんだけど、何かオススメの方法ないかな？」



「前にも教えたことのある HTML・CSS を使うと、文字を大きくしたり、色を付けたり、サイトのデザインを自由に変更できるよ！」



「なんだか難しそうだけど、初心者のボクでも使えるかな？」



「英語がたくさんあって難しそうに見えるけど、タグの意味とルールさえ覚えてしまえば簡単に書くことができるよ♪」



「じゃあ、今回は初～中級者向けの HTML・CSS に関する基本の基本を振り返ってみよう！」



「わーい♪マリーありがとう！ボク、一生懸命勉強して HTML・CSS マスターになれるように頑張るね！！」

★今さら聞けない HTML の基本編



「早速、タグの意味から振り返ってみるよ！」

・ **h1~6 [見出し 1~6]** : h1 が一番大きな (=最も強調される) 見出しです。その次が h2、h3…とつづきます。SEO 的にも重要なタグです。ページの見出しをつけることで、検索エンジンにページの内容を伝えます。

・ **div [汎用的な区切り]**

・ **span [汎用的な範囲]** : この 2 つは何にでも使える入れ物です。

・ **p [段落]** : 文章のまとめりです。

・ **ul、li [リスト]** : 箇条書き等に使います。

・ **table [表]** : 表です。昔はページレイアウト目的で使われていましたが 本来の表としての意味が通らない場合は使わない方が良いでしょう。

■ 「div」と「p」ってどう違うの？

よく「div と p は使い方が似てるけど、どう違うの？」と聞かれます。

タグの意味を知るため、名前を翻訳してみましょう！

- ・ div = division = 区画
- ・ p = paragraph = (文章などの) 段落

div は「汎用的に使える入れ物」、p は「文章を入れる段落」となります。

画像や他のタグなどを入れる時は“div”、文章を入れる時は“p”を使うことが

タグ本来の意味になります。



「タグの意味を調べてみると、色々なことがわかるよ！」

■ ul と ol の違いはなに？

同じ様にみえて異なる2つのタグ。意味を調べてみましょう！

ul = unordered list = 順不同なリスト

ol = ordered list = 順序つきリスト

ということから、「順番の決まったリストは ol」「順番が決まってないリスト

は ul」を使うということがわかります。

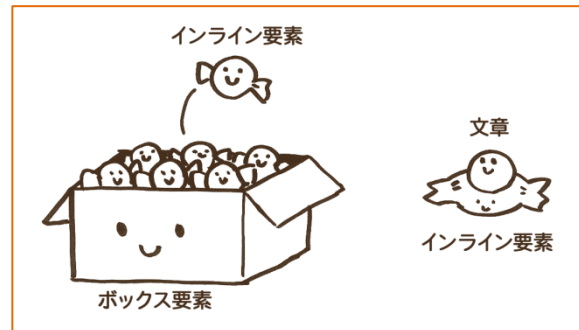
■ブロックレベル要素とインライン要素

タグには「ブロックレベル要素」と

「インライン要素」の2種類があり

ます。例えるならば、“**ブロック要素**

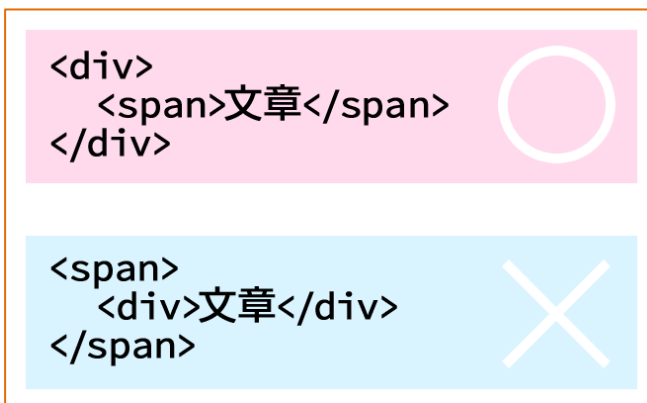
は箱” “**インライン要素**は箱の中に



ある**包み紙**”となり、文章はその中身に入ります。

ブロック要素は箱となるので、その中には同じ箱であるブロック要素、包み紙

であるインライン要素を入れることができます。しかしインライン要素の中に



ボックス要素を入れることはで

きません。なぜならば、小さい

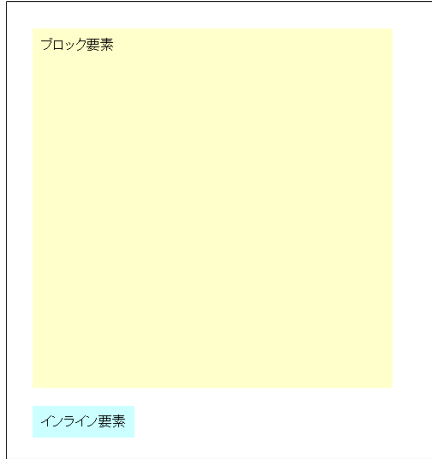
包み紙で大きな箱を包むことは

できないためです。

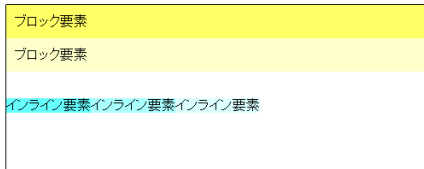
そしてブロック要素は「幅」・「高さ」の概念を持ちますが、インライン要素

は持ちません。「幅」・「高さ」の概念を持たないということは・・・

CSS で幅や高さをスタイル付けできない、ということです。縦方向の margin や padding（間隔や余白）もききません。

HTML	実際の表示
<pre><div>ブロック要素</div> インライン要素</pre>	
<pre>div { margin: 30px; padding: 10px; width: 400px; height: 400px; background-color: #ffc; } span { margin: 30px; padding: 10px; width: 400px; height: 400px; background-color: #cff; }</pre>	<p>インライン要素はブロック要素と違って margin や width などのスタイルが反映されない。</p>

ブロック要素の幅は親要素の幅いっぱいになり、要素の前後には改行が入ります。インライン要素は前後に改行が入りません。

HTML	実際の表示
<pre><div>ブロック要素</div> <div>ブロック要素</div> インライン要素 インライン要素 インライン要素</pre>	
	<p>ブロック要素は横幅いっぱいになり、インライン要素は並べて表示される。</p>



「ブロック要素・インライン要素は HTML の組立・CSS レイアウトにおいてとても重要な基礎だから、ちゃんと頭に入れておこうね！」

★今さら聞けない CSS の基本編



「次は CSS ね！読み込ませ方には数種類あるよ！」

・**外部参照** <head> タグ内で CSS ファイルを読みこませる指定をしたもの。

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

・**内部参照** <head> タグ内にスタイルを記述するもの。

```
<style type="text/css">
  p { color: #000; }
</style>
```

・**インライン指定** タグに style 属性を使って直接スタイルを記述するもの。

```
<p style="color: #000;">文章</p>
```

これらは、**インライン指定**>**内部参照**>**外部参照**の順番で優先されます。

外部参照の css で文字色を黒に指定していたとしても、インライン指定で文字色を赤に指定していれば実際の表示は赤になります。

HTML と CSS は分けた方が管理もしやすいので、外部参照をおすすめします。

■ CSS の優先順位

CSS には前ページでお伝えした以外にも、優先順位が決められています。

・後に読み込まれたほうが優先される

```
p { color: #black; } →文字色を黒にする  
  
p { color: #white; } →文字色を白に上書き
```

この例では、実際の文字色の表示は白になります。

・ポイント制度

「* (全称セレクトア) 」 0Point、「p,h1 などのタグ 」 1 Point

「.sample (class の場合) 」 10 Point、「#sample (ID の場合) 」 100 Point

```
p { color: #black; } →タグのみ  
  
p.class { color: #white; } →タグ+class  
  
p#id { color: #red; } →id のみ
```

この例では、実際の文字色の表示は一番ポイントが高い“赤”になります。

セレクタの親子関係やタグに class を複数つけたり、ポイントをうまく利用することでスマートな CSS が記述できます。

■リセット CSS

タグにはブラウザごとによってあらかじめ「デフォルトのスタイル」が設定されています。h1 が太字だったり、li の先頭に「・」がついているのはデフォルトスタイルによるものです。

デフォルトスタイルはブラウザごとに微妙に違ったりするので、ページレイアウトに影響が出てくることもあります。そんな時は**デフォルトスタイルをリセットする CSS ファイルを読みこませる**ことで、フラットな状態からスタイルづけをしていくことができます。

リセット CSS を配布しているサイトはいくつかありますのでそれを head 内に読み込めばすぐに使うことができます。

・ CSS Reset - YUI Library

<http://yuilibrary.com/yui/docs/cssreset/>

■ float の解除方法いろいろ

レイアウトで2つの要素を float を使って横並びにすることはよくあると思います。その場合、きちんと float を解除しないと並んだ要素を囲む親要素が潰れて表示され、レイアウト崩れにつながってしまいます。

float を解除するには、後続く要素に「clear: both;」を指定したり、後に要素がない場合は内容のない div をいれたりしますが、他にも方法があります。

・ clearfix

```
.clearfix {zoom:1;}
.clearfix:after {
  content: "";
  display: block;
  clear: both;
}
```

CSS にこの記述を加え、親要素に「clearfix」という class をつければ float を解除することができます。

・ 親要素に「overflow: hidden;」

clearfix を使わなくても、最初から親要素に「overflow: hidden;」を指定して

おくと、親要素が潰れることはありません。「overflow: hidden;」を使用する時は width の指定を忘れないようにしましょう。

■スタイルがきいてないときにまず確認すること

・文法ミス

タグ名のミスタイプや指定の末尾にあるセミコロンの抜けなど…。CSS の記述が長くなって「確認しきれない!」という時は、CSS の文法をチェックしてくれるバリデーションサービスがおすすめです。

※バリデーションサービスについては後の項目で解説します。

・class 名と ID 名のルール

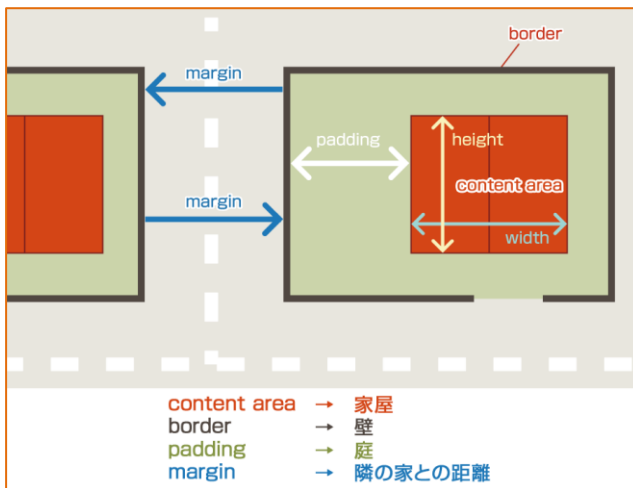
あまり知られていませんが、class 名と ID 名には、実はルールがあります。

- 使える文字は半角英数字、ハイフン (-)、アンダーバー (_)
- 先頭に数字とハイフン、アンダーバーは使えない

「class="001"」など、数字のみの class を指定してもスタイルは有効になりませんので、気をつけましょう。

■ ボックスモデル

CSS の margin や width などの関係を表す「ボックスモデル」について学びましょう。要素を“家”として考えるとわかりやすいです。



margin 同士は重なってしまうと相殺します。隣り合う要素と margin で距離をとる際は注意しましょう。

■ CSS ハック

CSS でレイアウトしていると、どうしてもブラウザによる表示の差が出てきてしまいます。そんな時は「CSS ハック」という“ある特定のブラウザだけが読み込むスタイル”の書き方があります。

太字で強調した箇所の記述を足すことで特定のブラウザにスタイルを指定することができます。

- ・ IE6 のみ適用させる

```
* html #sample { color: #000; }
```

- ・ IE7 のみ適用させる

```
*:first-child + html #sample{ color: #000; }
```

- ・ IE8 のみ適用させる

```
#sample{ color: #000/9; }
```

- ・ Firefox のみ適用させる

```
#sample, x:-moz-broken, x:last-of-type { color:#000; }
```

- ・ Chrome、Safari に適用させる

```
@media screen and (-webkit-min-device-pixel-ratio:0) {
```

```
  #sample{ color:#000; }
```

```
}
```

★バグチェックの方法



「教えてもらった通り HTML と CSS を作ったけど、思い通りの表示にならないよお・・・でも、どこが間違ってるかわからないんだ！」



「そんな時のために便利なツールがあるよ！これから教えるツールを使ってソースのバグを見つけてみてね！」

■バリデーションサービス

オンラインで **HTML** や **CSS** の文法を**チェック**してくれるサービスです。

タグのミスタイプやとじ忘れなどのミスが一目でわかります。ファイルをアッ

プロードしたり、直接入力することでチェックできます。

- ・ **HTML バリデータ** (英語)

<http://validator.w3.org/>

- ・ **CSS バリデータ** (日本語)

<http://jigsaw.w3.org/css-validator/validator.html.ja>

■ Firebug

<https://addons.mozilla.org/ja/firefox/addon/firebug/>

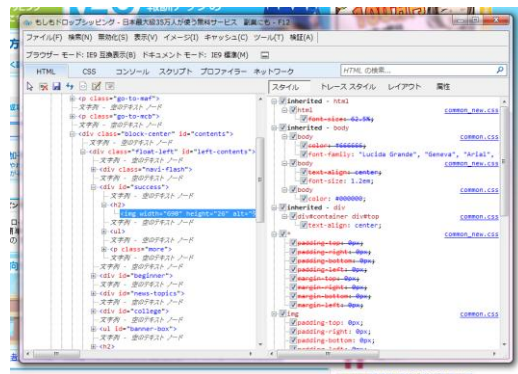
以前もこの連載で紹介したことのある、ページ作成に非常に役立つ Firefox のアドオンです。調べたい箇所をクリックすると、その要素に適用されているスタイルが一目でわかります。

■デベロッパツール

Firebug に似た機能はその他のブラウザにも標準でついています。名前は違いますが機能は一緒なので、ぜひ使ってみましょう。

・ Internet Explorer

“F12”をクリックすると開発者ツールが起動します。



・ Chrome

“Ctrl + Shift + I” でデベロッパツールが起動します。



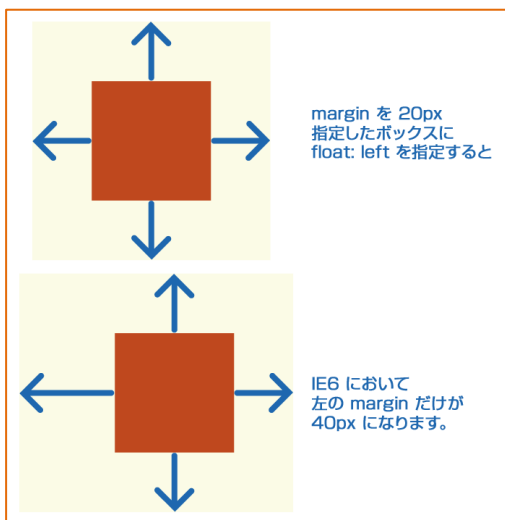
- Safari

Ctrl + Alt + I でデベロッパツールが
起動します。



■注意したい margin 倍加バグ

「IE6 において、float した方向に margin を指定すると margin が2倍になる」というバグです。回避方法としては、float した方向に margin を指定し



ないか CSS ハックで IE6 のみ margin の指定を半分の値にすることです。

古い IE のシェアは大きいので、先ほどの CSS ハックと合わせて、全てのブラウザで表示が同じになるようにしましょう。



「基礎を踏まえて HTML を使えば、どんな OS やブラウザでも同じように表示されて安心感のあるショップサイトを作れるよ！」

★ おしまい ★