

HTML 基礎マニュアル

制作：三上 絵利菜

○目次

- 1p. HTML とは？
HTML、CSS、Javascript の役割
拡張子
タグ
非推奨タグ
- 2p. 必須のタグ
整形とコメントアウト
ドキュメントタイプ宣言
終了タグのないタグ（XHTML）
属性
- 3p. サイトの中身を構成するタグ
ブロックレベル要素とインライン要素
文章、改行
- 4p. 見出し
リンク
- 5p. 画像
絶対パス、相対パス
- 6p. 箇条書きリスト
定義型リスト
- 7p. タグを識別するための属性
要素をまとめて整理する
- 8p. 【おまけ】ディレクトリ管理
【おまけ】命名規則

○ 必須のタグ [<html><head><title><body>]

HTML のタグはブラウザには表示されません。<h1> サンプル </h1> と書いた場合、「サンプル」のみ表示されます。また、HTML で記す情報は、Web ページに表示されるものだけではありません。ページ上表示されないけれど、サイトの情報として必要なタグ、また Web ページを作るうえで、最低限必要なタグを以下で説明します。

書き方例：

<html>	HTML を書き始めます
<head>	ヘッダ情報を書き始めます
<title> サンプルサイト </title>	サイトタイトルを示す
</head>	ヘッダ情報を書き終わります
<body>	文章の本体（表示される部分）を書き始めます
</body>	文章の本体（表示される部分）を書き終わります
</html>	HTML を書き終わります

○ 整形とコメントアウト

HTML ファイル内では、いくら改行や、（タグの外での）スペースを入れても、ブラウザの表示に変化はありません。ですので、適度に見やすいように改行をしたり、Tab（字下げ）を入れて、整形を行えます。また、タグで囲まれた部分以外には、テキストを書いてはいけませんが、コメントアウト [<!-- -->] を使うことで、制作上のメモなどを入れられます。

書き方例：

```
<!-- コメントです。表示されません。-->
```

○ ドキュメントタイプ宣言 [DOCTYPE]

HTML は、バージョンごとに使用できるタグや属性などが厳密に定義されています。これはドキュメントタイプ宣言と呼ばれるもので、「文書型の定義」を行うものです。大きく分けて HTML と XHTML があります。各ドキュメントタイプの違いは参考 URL をご覧ください。以下は、XHTML1.0 の Strict の記述です。

書き方例：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

参考 「DTD の違い - ホームページ作成入門」 <http://www.webword.jp/xhtml1/doctype/index1.html>

○ 終了タグのないタグ（XHTML）

タグの中にはテキストを囲う必要がなく、単体で使用する終了タグのないタグがあります。
 などがあげられますが、XHTML では最後に / を入れて、終了タグの代わりとしています。
 のように書きます。

○ 属性

タグによって、文章に意味を与えてあげられますが、タグに属性をつけることで、値を渡すことができます。属性 = " 値 " というように書きます。例えば、以下のソースは「ファイル名が 1.jpg の、横幅が 32 の、縦幅が 28 の、画像」という意味になります。

書き方例：

```

```

○ サイトの中身を構成するタグ

`<html>` や `<title>` など、Web ページに必須のタグを前項までで扱ってきましたが、いよいよブラウザで実際に表示される部分、サイトの中身を構成するタグを後項で紹介していきます。

`<body>` タグで囲まれた部分が、表示される部分、サイトの中身になります。

書き方例：

```
<body>

<h1> サンプルサイト </h1>
<p> サンプルのサイトです。 </p>

<h2> 項目 1 </h2>
<p> 項目 1 つめです。 </p>

</body>
```

○ ブロックレベル要素、インライン要素

ブロックレベル要素は見出し、段落など文章を構成する基本要素となるものです。ブロックレベル要素の内容には、別のブロック要素やインライン要素を含むことができますが、逆にインライン要素の中にブロックレベル要素を置くことはできません。

インライン要素は、ブロックレベル要素の内容として用いられるもので、文書の構造を構成するというより、ブロックレベル要素内の特定の部分になんらかの役割や機能を持たせる要素です。

参考 「ブロックレベル要素とインライン要素」 <http://www.kanzaki.com/docs/html/element-level.html>

○ 文章、改行 [`<p>
`]

文章（段落）を書くときに使用するタグです。`<p></p>` で囲って文章を書きます。途中で改行する場合は、`
` を書きます。

何でも 1 つの `<p>` 内に書かずに、段落が変わるときには違う `<p>` に書きましょう。

書き方例：

正しい例

```
<p>Web デザインとは、何でしょうか？ <br /> 単に見た目をかっこよくするものではありません。ユーザの求めている情報を、わかりやすく伝えるためのツールであると考えています。 </p>

<p> それでは、具体的にどうということなのか具体例を出しながら、考えていきましょう。 </p>
```

間違い例

```
<p>Web デザインとは、何でしょうか？単に見た目をかっこよくするものではありません。ユーザの求めている情報を、わかりやすく伝えるためのツールであると考えています。

<br />
<br />
<br />
<br />
それでは、<br /> 具体的にどうということなのか具体例を出しながら、考えていきましょう。 </p>
```

→段落が変わる部分も、1 つの `<p>` で書いている。

→隙間をあける目的で、改行の `
` を使っている。

→文章の途中で、`
` を使っている。

○ 見出し [<h1> ~ <h6>]

文章内の見出しを書くときに使用するタグです。

見出しのレベル順に <h1> から <h6> まであって、<h1> が最も重要な見出しになります（1 つのページ内に 1 つが好ましい）。<h2> 以降はいくつでも使えますが、見出しとしての意味に合っているか考えて使いましょう。

書き方例：

正しい例：

```
<h1>Web デザインサイト </h1>
<h2>Web デザインとは？ </h2>
<p>Web デザインとは、何でしょうか？ <br />単に見た目をかっこよくするものではありません。ユーザの求めている情報を、わかりやすく伝えるためのツールであると考えています。</p>
<h2>Web デザイン関連の職種 </h2>
<h3>Web ディレクター </h3>
<p>サイト制作の進行管理を行う人です。</p>
<h3>Web コーダー </h3>
<p>HTML、CSS を担当して制作します。</p>
<h2>Web デザイン関連の学校 </h2>
<p>日本電子専門学校 </p>
```

間違い例：

```
<h2>Web デザインサイト </h2>
<h1>Web デザインとは？ </h1>
<h3>Web デザインとは、何でしょうか？単に見た目をかっこよくするものではありません。ユーザの求めている情報を、わかりやすく伝えるためのツールであると考えています。</h3>
<h1>Web デザイン関連の職種 </h1>
<h2>Web ディレクター </h2>
<p>サイト制作の進行管理を行う人です。</p>
<h2>Web コーダー </h2>
<p>HTML、CSS を担当して制作します。</p>
<h2>Web デザイン関連の学校 </h2>
<p>日本電子専門学校 </p>
```

→<h1> より先に <h2> がある。

→見出しではない部分に、<h3> が使われている。

→「Web コーダー」と、「Web デザイン関連の学校」並列の情報ではないのに、同じレベルの <h2> を使っている。

○ リンク [<a>]

リンクを貼るときに使うタグです。href 属性を使って、URL を指定できます。

URL は、絶対パスか、相対パスを使えます。パスについては後項で説明します。

<a> はインライン要素なので、他の <h1> や <p> といったブロックレベル要素で囲みましょう。

書き方例：

```
<p><a href="http://www.jec.ac.jp/">日本電子専門学校 </a></p>
```

○ 画像 []

画像を使用する時のタグです。必須の属性がいくつかあります。

終了タグを持たないので、/をつけましょう。

 はインライン要素なので、他の <h1> や <p> といったブロックレベル要素で囲みましょう。

src="URL"

画像のある場所を URL で指定します。絶対パスか、相対パスで指定できます。

width=" 数字 "、height=" 数字 "

画像の横幅、縦幅を指定します。画像サイズの単位は px ですが、px を記述する必要はありません。

alt=" 画像の説明 "

画像の説明を書きます。画像が見れない状況、視覚障害のあるユーザにとっての代替情報となります。

書き方例：

正しい例：

```
<h1></h1>
<p><a href="detail/index.html"></a></p>
```

間違い例：

```
<h1></h1>
<p><a href="detail/index.html"></a></p>
```

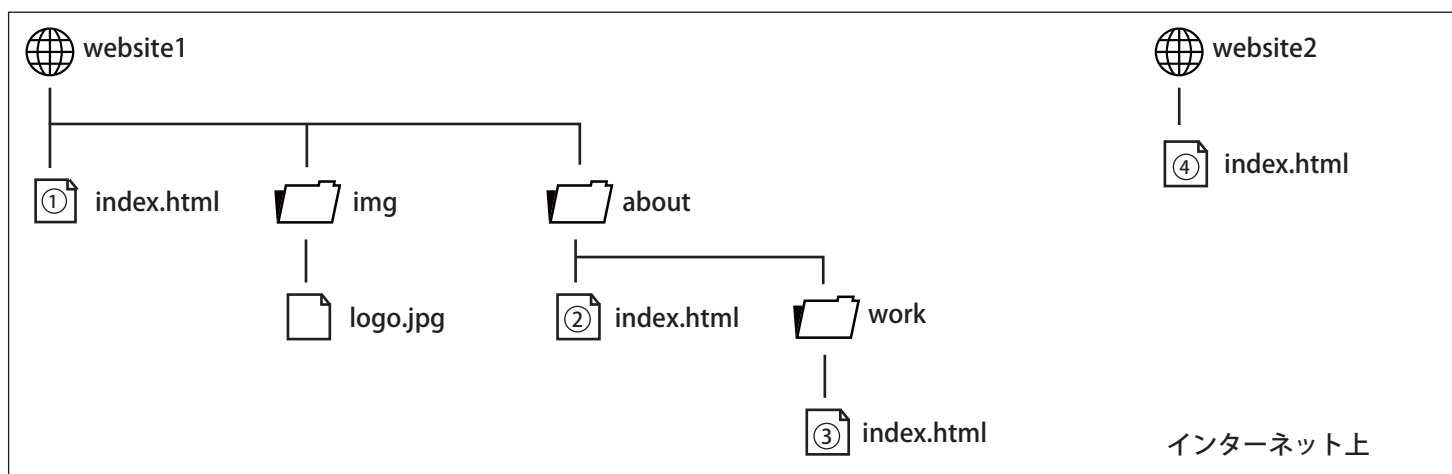
→width と height が抜けている。

→width と height に px が入っている。

→alt が不適切。画像が見えないような時にも、画像と同じ情報を伝えられる文章でなければダメ。

○ 絶対パス、相対パス [http:、../]

パスの書き方は、絶対パスと相対パスの 2 種類あります。絶対パスは「http:」から始まるインターネット上の URL、相対パスは現在のファイルから相対的に見たファイルの位置を表すもので「../（一階層出る）」「about/index.html（about フォルダの中の index.html）」といった書き方をします。



相対パス（同じサイト内でよく使う）

① から見た ③ → about に入って、work に入ったところの index.html → about/work/index.html
③ から見た logo.jpg → 一階層出て、一階層出て、img に入ったところの logo.jpg → ../../img/index.html

絶対パス（他のサイトを指定する場合よく使う）

④（どのファイルからでも） → インターネット上の website2 というサイトの index.html → http://www.website2/index.html

○ 箇条書きリスト []

並列の情報を並べる（リストを作る）時に、使用するタグです。

 で囲んだなかに を複数並べ、テキストを書きます。

書き方例：

正しい例：

```
<ul>
  <li><a href="index.html"> トップページ </a></li>
  <li><a href="blog/index.html"> ブログ </a></li>
</ul>
```

間違い例：

```
<ul></ul>
  <a href="index.html"><li> トップページ </li></a>
  <a href="blog/index.html"><li> ブログ </li></a>
```

→ で囲んだ範囲に が書かれていない。

→<a> が より外側にある。<a> のようなインライン要素は、内側に書く。

○ 定義型リスト [<dl><dt><dd>]

<dl> は辞書のような形のリストで、用語（<dt>）とその定義（<dd>）からなります。

<dt> は単独でも使えますが、<dd> は必ず対応する <dt> の後に書きましょう。1 つの <dt> に対して、複数の <dd> を使用することもできます。

書き方例：

正しい例：

```
<dl>
  <dt> はれ </dt>
  <dd> 天気がいい様子。 </dd>

  <dt> あめ </dt>
  <dd> 空から水が降ってくる現象 </dd>
  <dd> 甘いお菓子、キャンディー </dd>
</dl>
```

間違い例：

```
<dl>
  <dd> はれ </dd>
  <dd> 天気がいい様子。 </dd>
</dl>
<dl>
  <dt> あめ </dt>
  <dt> 空から水が降ってくる現象 </dt>
  <dt> 甘いお菓子、キャンディー </dt>
</dl>
```

→<dl> を1つずつ書いてる。「リスト」なので、ひとまとまりの <dl> でいい。

→<dd> の対応する <dt> がない。

→説明にあたる部分なのに <dt> になっている。

○ タグを識別するための属性 [id,class]

`<p>` が HTML ファイル内にいくつもあった時に 1 つだけ CSS で赤くしたい！、`<div>` で囲んだ範囲に news という名前をつけたい、などの時に識別する属性として、id と class を使います。この属性は、全てのタグにつけることができます。1 つの HTML ファイル内にて、同じ id 名は 1 つだけ、class はいくつでも使えます。ですが、なるべく必要な場合のみ書くようにしましょう。

書き方例：

正しい例：

```
<ul id="gloNav">
  <li id="top"><a href="index.html"> トップページ </a></li>
  <li class="other"><a href="blog/index.html"> ブログ </a></li>
  <li class="other"><a href="contact/index.html"> お問い合わせ </a></li>
</ul>
```

間違い例：

```
<ul class="gloNav">
  <li id="top"><a href="index.html"> トップページ </a></li>
  <li id="other"><a href="blog/index.html"> ブログ </a></li>
  <li id="other"><a href="contact/index.html"> お問い合わせ </a></li>
</ul>
```

→1 度しか使わないのに、class が使われている。class でも問題はないが、1 度しか使わないものは id が好ましい。
→id="other" が 2 つある。同じ id 名は 1 つの HTML ファイル内で 1 度しか使えない。複数の場合は、class。

○ 要素をまとめて、整理する [<div>、]

`<div>` はそれ自体には意味を持たないタグです。`<div></div>` で囲んだ範囲をひとかたまりとして整理したり、CSS でレイアウトの指定を行う時に利用します。

同じ意味の `` タグがありますが、`<div>` はブロックレベル要素、`` はインライン要素です。

書き方例：

正しい例：

```
<div id="about">
  <p><span>Web デザインとは、何でしょうか？ </span><br /> 単に見た目をかっこよくするものではありません。ユーザの求めている情報を、わかりやすく伝えるためのツールであると考えています。</p>
  <p> それでは、具体的にどうということなのか具体例を出しながら、考えていきましょう。</p>
</div>
```

間違い例：

```
<div id="about">
  <span><p>Web デザインとは、何でしょうか？ <br /> 単に見た目をかっこよくするものではありません。ユーザの求めている情報を、わかりやすく伝えるためのツールであると考えています。</p></span>
</div>
<div id="about2">
  <p> それでは、具体的にどうということなのか具体例を出しながら、考えていきましょう。</p>
</div>
```

→`` が `<p>` より外側にある。`` はインライン要素なので、内側に書く。

→ひとかたまりとして意味のない範囲で囲んでいる。`<p>` 自体が文章というブロックであるため、それ以上まとめて整理する必要がなければ、`<div>` を使わない。

○【おまけ】 サイトのディレクトリ管理

Web サイトを作るうえで、ディレクトリ管理も重要です。更新しやすさ、アクセスしやすさに関わります。

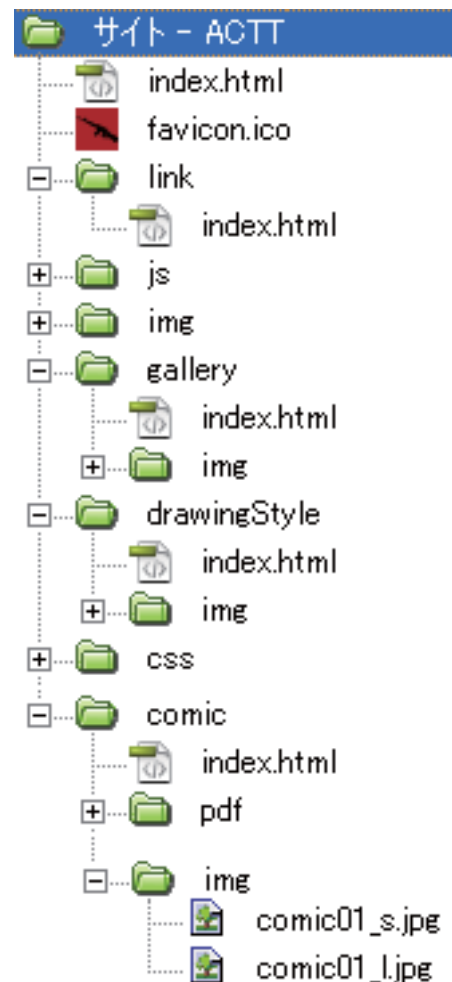
例えば about というページを作りたい場合は、about.html とせずに、「about/index.html」とします。すると、ブラウザで <http://www.website/about/> にアクセスすれば、index.html まで打たなくても繋がります。

index には、そういった意味もあるので、全ての html ファイルが index.html となるようにフォルダでの階層分けをしっかりとしましょう。

○【おまけ】 命名規則

管理のしやすさといったところで、画像ファイルなどの命名規則は定義しておくとう便利です。

comic の 01 枚目の L サイズの画像だから「comic01_l.jpg」にしよう。など、一定の規則性のあるファイル名にしておくと、編集しやすいです。



○【おまけ】 テキストエディタ

HTML はメモ帳やワードパッドでも制作できます。

しかし、HTML の制作に特化したテキストエディタだと、カラーリングや予測変換機能が着いているので、より便利に HTML の作成が出来ます。以下は、無料でダウンロードできるおすすめのテキストエディタです。

「Crescent Eve」

<http://www.kashim.com/eve/>

「サクラエディタ」

http://sakura_editor.at.infoseek.co.jp/

「TeraPad」

<http://www5f.biglobe.ne.jp/~t-susumu/library/tpad.html>

「秀丸エディタ」

<http://hide.maruo.co.jp/software/hidemaru.html>