

# HTMLのキホン

## 1. はじめに：Webページってどうやってできてるの？

みんなが普段見ているWebサイト、例えば学校のホームページや好きなアーティストの公式サイト、ニュースサイトなどは、どうやって作られているか知っていますか？ その基本的な設計図となるのが、これから学ぶ **HTML (HyperText Markup Language)** なんです。

HTMLは、Webページに表示したい文章や画像、リンクなどを「ここに見出しを置く」「ここに画像を貼る」「ここから別のページに飛ぶ」といった指示をコンピューターに伝えるための特別な言葉（マークアップ言語）です。

家を建てるときにまず骨組みを作るように、Webページを作るときもまずHTMLで「どこに何を表示するか」という **骨組み** を作ります。HTMLを学ぶことで、Webページがどのように構成されているかを理解し、自分で簡単なWebページを作れるようになります。

## 2. HTMLのキホン：タグと要素、属性

HTMLでは、**タグ** と呼ばれる特別な記号を使って、コンピューターに指示を出します。

### タグ (Tag)

タグは、`<` と `>` で囲まれた命令のことです。例えば、文章の段落を作るには `<p>` というタグを使います。

多くの場合、タグは **開始タグ** (`<p>`) と **終了タグ** (`</p>`) のペアで使われ、その間に内容を記述します。終了タグにはスラッシュ `/` が入るのが目印です。

```
<p>これは段落です。</p>
```

### 要素 (Element)

この **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。上の例では `<p>これは段落です。</p>` 全体が「p要素」です。Webページは、たくさんの要素を組み合わせて作られています。

### 属性 (Attribute)

タグには、追加情報（設定）を与えるための **属性** を指定できる場合があります。属性は、**開始タグの中** に **属性名="値"** の形で記述します。

例えば、リンクを作る `<a>` タグには、`href` という属性を使って、リンク先のURLを指定します。

```
<a href="https://www.nitac.jp/">NITAC公式サイトへ</a>
```

この例では、`<a>` がタグ、`href="https://www.nitac.jp/"` が属性、`NITAC公式サイトへ` が内容です。これで「NITAC公式サイトへ」という文字をクリックすると、指定したURLにジャンプするリンクになります。

## 空要素 (Empty Element)

タグの中には、内容を持たず、終了タグがないものもあります。これを **空要素** と呼びます。例えば、画像を表示する `<img>` タグや、改行する `<br>` タグがそうです。

```

<br>
```

## 3. HTML文書の骨組みを見てみよう

どんなWebページも、基本的には次のような決まった骨組み（構造）を持っています。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>ここにページのタイトルを書きます</title>
  <!-- ここにはWebページに直接表示されない情報が入ります -->
</head>
<body>
  <h1>こんにちは！</h1>
  <p>ここにWebページに表示したい内容を書きます。</p>
  <!-- 画像やリンクなどもここに入ります -->
</body>
</html>
```

それぞれの部分が何をしているか見てみましょう。

- `<!DOCTYPE html>`：「このファイルはHTML5という最新ルール of HTMLですよ！」と宣言するおまじないです。必ず一番最初に書きます。
- `<html>`：HTML文書全体の始まりと終わりを示します。すべての要素はこの `<html>` タグの中に書きます。
  - `lang="ja"`：これは `html` タグの属性で、「このページの主な言語は日本語ですよ」と示しています。
- `<head>`：Webページそのものに関する設定情報を書く部分です。ここに書いた内容は、通常Webページには直接表示されません。人間でいうと「頭脳」のような部分です。
  - `<meta charset="UTF-8">`：文字化けを防ぐためのおまじないです。「文字コードはUTF-8を使います」と指定しています。

- `<title>` : Webページのタイトルを指定します。ブラウザのタブや、お気に入りに登録したときに表示される名前になります。
- `<body>` : 実際にブラウザの画面に表示される内容（見出し、文章、画像、リンクなど）を書く部分です。人間でいうと「体」にあたる、Webページのメイン部分です。

このように、HTMLでは要素が別の要素の中に入る「入れ子構造」で全体の骨組みを作っていきます。

## 4. 様々な使ってみよう

`<body>` タグの中を書く、代表的なタグをいくつか紹介します。

ただし、**ここにあるものを全て覚える必要はありません**。実際に、Webページを作るときは、必要なものを調べながら使っていくことが多いです。まずは「こんなタグがあるんだな」と知っておいてください。

### 見出し（`<h1>` ～ `<h6>`）

文章のタイトルや小見出しを作るタグです。`<h1>` が一番大きな見出し（大見出し）で、`<h2>` , `<h3>` ... と数字が大きくなるにつれて、見出しのレベル（重要度）が下がっていきます。見出しを適切に使うことで、文章の構成が分かりやすくなります。

```
<h1>一番大きな見出し（ページのタイトルなど）</h1>
<h2>中くらいの見出し</h2>
<h3>小さな見出し</h3>
<p>見出しを使うと、どこに何が書いてあるか分かりやすくなりますね。</p>
```

### 段落（`<p>`）

文章のまとまり（段落）を作るタグです。文章は基本的にこの `<p>` タグで囲みます。ブラウザでは、`<p>` タグで囲まれた部分は前後に少しスペースが空いて表示されます。

```
<p>これは最初の段落です。HTMLはWebページの骨組みを作ります。</p>
<p>これは二番目の段落です。タグを使って要素を配置していきます。</p>
```

**注意:** HTMLでは、エディタで改行しても、ブラウザ表示では改行されません。改行したい場合は、次の `<br>` タグを使います。

### 改行（`<br>`）

文章の途中で強制的に改行を入れたいときに使う空要素です。`<p>` タグのように段落全体のスペースは空きません。

```
<p>ここで改行します。<br>次の行です。</p>
```

## リンク（<a>）

他のWebページや、同じページ内の別の場所にジャンプするためのリンクを作るタグです。 `href` 属性で行き先のURL（Webアドレス）やファイル名を指定します。

```
<p>詳しくは<a href="https://developer.mozilla.org/ja/docs/Web/HTML">MDNのHTML解説</a>をご覧ください。</p>
```

```
<!-- 同じフォルダにある別のHTMLファイルへのリンク -->
```

```
<a href="page2.html">次のページへ</a>
```

```
<!-- クリックすると新しいタブで開くリンク -->
```

```
<a href="https://example.com" target="_blank">新しいタブで開く</a>
```

`target="_blank"` という属性を追加すると、リンクを新しいタブ（またはウィンドウ）で開くことができます。

## 画像（<img>）

Webページに画像を表示するための空要素です。 `src` 属性で画像の場所（ファイル名やURL）を指定し、 `alt` 属性で画像が表示されなかった場合や、音声読み上げソフト用の代替テキストを指定します。

```
<!-- 同じフォルダにある画像を表示 -->
```

```

```

```
<!-- インターネット上の画像を表示 -->
```

```

```

```
<!-- サイズを指定する場合（CSSでの指定が推奨されます） -->
```

```

```

`width`（幅）や `height`（高さ）属性で画像のサイズを指定することもできますが、通常は後で学ぶCSSで行うのが一般的です。 `alt` 属性は必ず指定するようにしましょう。

## リスト（<ul>, <ol>, <li>）

項目をリスト形式で表示するタグです。

- `<ul>` (**Unordered List**): 順序のない箇条書きリストを作ります。通常、点（・）などで表示されます。
- `<ol>` (**Ordered List**): 順序のある番号付きリストを作ります。通常、1, 2, 3... と番号が付きます。
- `<li>` (**List Item**): 各リストの項目を表します。 `<ul>` または `<ol>` の中に書きます。

```
<h4>好きな果物（順不同）</h4>
```

```
<ul>
```

```
<li>りんご</li>
```

```
<li>バナナ</li>
<li>みかん</li>
</ul>

<h4>朝の準備（順番通り）</h4>
<ol>
  <li>起きる</li>
  <li>顔を洗う</li>
  <li>朝ごはんを食べる</li>
  <li>歯を磨く</li>
</ol>
```

## 区切り線（`<hr>`）

話題の区切りなどに、水平線を表示するための空要素です。

```
<p>最初の話題はここまでです。</p>
<hr>
<p>ここから次の話題です。</p>
```

## 強調（`<strong>`，`<em>`）

文章の中で特に強調したい部分を示すためのタグです。

- `<strong>`：重要な部分を強調するためのタグです。通常、太字で表示されます。
- `<em>`：強調したい部分を斜体で表示するためのタグです。通常、イタリック体（斜め文字）で表示されます。

```
<p>この文章は<strong>重要な部分</strong>を強調しています。</p>
<p>この文章は<em>強調したい部分</em>を斜体で表示しています。</p>
```

## 下線（`<u>`）

文章の中で下線を引きたい部分を示すためのタグです。通常、下線が引かれます。

```
<p>この文章は<u>下線が引かれています</u>。</p>
```

## 引用（`<blockquote>`）

他の人の言葉や文章を引用するためのタグです。通常、インデント（左側に余白）が付き、引用符が表示されます。

```
<blockquote>
  <p>「HTMLはWebページの骨組みを作るための言語です。」</p>
```

```
<footer>— W3C</footer>
</blockquote>
```

## br ( <br> )

改行を入れたいときに使う空要素です。文章の途中で強制的に改行を入れます。

```
<p>ここで改行します。<br>次の行です。</p>
```

## ボタン ( <button> )

ボタンを作るためのタグです。通常、クリック可能なボタンが表示されます。

```
<button type="button">クリックしてね！</button>
```

## フォーム ( <form> )

ユーザーからの入力を受け取るためのフォームを作るためのタグです。通常、テキストボックスやラジオボタン、チェックボックスなどが含まれます。

```
<form action="/submit" method="post">
  <label for="name">名前:</label>
  <input type="text" id="name" name="name">
  <input type="submit" value="送信">
</form>
```

## グループ化 ( <div> , <span> )

これらのタグ自体は、見た目に直接影響を与えることは少ないですが、複数の要素をまとめたり、文章の一部をグループ化したりするために使います。後で学ぶCSSでデザインを適用する際に、非常に重要な役割を果たします。

- **<div>** : ブロックレベル要素（通常、前後に改行が入る）をまとめるための汎用的な箱です。関連する要素群（例えば、記事全体、サイドバーなど）を囲むのによく使われます。
- **<span>** : インライン要素（通常、前後に改行が入らない）をまとめるためのタグです。文章の一部分だけ（例えば、特定の単語）にスタイルを適用したい場合などに使います。

```
<div class="article">
  <h2>記事タイトル</h2>
  <p>これは記事の本文です。特に<span class="important">重要な部分</span>はここです。</p>
</div>

<div class="profile">
  
```

```
<p>自己紹介文</p>
</div>
```

`class="article"` や `class="important"` は属性ですが、これは主にCSSで特定の要素を指定するために使われます（今は「こういう使い方があるんだな」くらいでOKです）。

## コメント（`<!-- -->`）

`<!--` と `-->` で囲まれた部分はコメントとなり、ブラウザには表示されません。コードの説明を書いたり、一時的にコードの一部を無効にしたりするのに使います。

```
<!-- ここは見出しです -->
<h1>ページのタイトル</h1>

<p>本文です。</p>
<!-- <p>この行は一時的に表示しません。</p> -->
```

## その他

- `<table>`：表を作るためのタグです。行（`<tr>`）と列（`<td>`）を組み合わせ、データを整理して表示します。
- 'セマンティックタグ'：`<header>`、`<footer>`、`<nav>`、`<article>`、`<section>` など、意味を持ったタグが増えてきています。より意味のある構造を持つWebページを作るために使われます。

他にもたくさんのタグがありますが、まずはこれらの基本的なタグを覚えておくといいでしょう。

## 5. まとめ：HTMLでできること

お疲れ様でした！これでHTMLの基本的な骨組みと、よく使うタグについて学ぶことができました。

- HTMLはWebページの **骨組み（構造）** を作るための言語です。
- **タグ** を使って、見出し、段落、画像、リンクなどの要素を配置します。
- **属性** を使って、タグに追加情報を与えます。
- HTMLだけだと、見た目はとてもシンプルです。

HTMLはWeb制作の第一歩です。まずはこのHTMLでページの構造をしっかり作れるようになることが大切です。

次のステップでは、**CSS（Cascading Style Sheets）** という別の言語を学びます。CSSを使うと、HTMLで作った骨組みに色を付けたり、レイアウトを整えたりして、Webページの **見た目をデザイン** することができます。HTMLとCSSを組み合わせることで、本格的なWebページが作れるようになります。

## 補足：タグの表示タイプ

HTMLのタグには、表示されるとき、主に3つの「表示され方」のタイプがあります。今後学ぶCSSの `display` というプロパティで設定され、初期値はタグによって決まっています。

このプロパティによって、「改行されるか?」「幅や高さを変えられるか?」が変わります。特に**改行について**うまくいかないときにこの知識が必要になります。

## 1. ブロックタイプ ( `display: block` )

### 特徴

- 前後で**勝手に改行**
- 幅や高さを自分で決められる
- 特に幅を指定しない場合、左右のスペースが最大に広がる

### タグの例

- `<h1>` ~ `<h6>` (見出し)、`<p>` (段落)、`<div>` (グループ化の箱)
- `<ul>`, `<ol>`, `<li>` (リスト)、`<hr>` (線)
- `<header>`, `<footer>`, `<article>` などのセマンティックタグ

```
<div style="border: 1px solid red;">ブロックタイプ。改行される</div>
<p style="border: 1px solid blue;">ブロックタイプ 幅も高さも変えられる!</p>
<!-- 自動的に縦に並んで、横幅いっぱいに広がろうとするのが特徴 -->
```

## 2. インラインタイプ ( `display: inline` )

### 特徴

- 前後で改行されず、**横に並ぶ**
- 幅や高さを指定しても**無視される** (中の文字や画像のサイズで決まる)
- 上下の `margin` (外側の余白) は効かない (左右は効く)
- `padding` (内側の余白) は上下左右に効くが、レイアウトに影響するのは左右のみ

### タグの例

- `<a>` (リンク)、`<span>` (グループ化)
- `<strong>`, `<em>`, `<u>` (文字装飾)
- `<input>`, `<button>`, `<label>` (フォーム関連の一部)

```
<span style="border: 1px solid green;">インラインタイプ。</span>
<a href="#" style="border: 1px solid purple;">横に並びます。</a>
<span style="border: 1px solid orange; width: 200px; height: 50px; margin-top: 20px; padding: 10px;">幅や高さを指定しても効かない。上下マージンも効かない。</span>
<!-- 文章の一部のように、行の中に埋め込まれるイメージ -->
```

## 3. インラインブロックタイプ ( `display: inline-block` )

ブロックとインラインのいいとこ取りをしたようなタイプです。

### 特徴

- 前後で改行されず、**横に並ぶ** (インラインの特徴)



- 幅や高さを自分で決められる（ブロックの特徴）

タグの例

- `<img>`（画像）
- `<button>` , `<input>`（フォーム関連の一部）

```
<span style="display: inline-block; border: 1px solid teal; width: 150px; height: 50px; margin: 10px;">インラインブロック。</span>
<button style="display: inline-block; border: 1px solid brown; width: 100px; height: 50px; margin: 10px;">幅も高さも指定できて、横に並ぶ！</button>
<!-- 横に並べたいけど、サイズも指定したいときに便利 -->
```

これらの表示タイプは、CSSで `display` プロパティを使って変更することもできます。例えば、`<a>` タグ（インライン）をボタンのように見せるために `display: inline-block;` や `display: block;` に変更することはよくあります。

## 表示タイプの比較まとめ

タイプ ( <code>display</code> )	改行される か？	幅・高さ指 定	横に並ぶ か？	タグの例（初期値）
<code>block</code>	する	できる	しない	<code>&lt;div&gt;</code> , <code>&lt;p&gt;</code> , <code>&lt;h1&gt;</code> , <code>&lt;ul&gt;</code> など
<code>inline</code>	しない	できない	する	<code>&lt;span&gt;</code> , <code>&lt;a&gt;</code> , <code>&lt;strong&gt;</code> , <code>&lt;em&gt;</code> など
<code>inline-block</code>	しない	できる	する	<code>&lt;img&gt;</code> , <code>&lt;button&gt;</code> , <code>&lt;input&gt;</code> など

最初は混乱するかもしれませんが、「なんか表示が思った通りにならないな？」と思ったときに、この表示タイプの違いを思い出してみてください。