

HTMLってなに?

HTMLは、HyperText Markup Language の略称です。日本語でわかりやすく言うと「ハイパーテキスト（リンク機能のある文書）をつくるための、テキストに印を付けるタイプの言語」という意味になります。

HTMLはWebページの構成要素を示す言語

HTMLは、Webページのコンテンツが含まれている、テキスト形式のファイルです。ワープロソフトなどの一般的な文書作成ソフトでは、フォントのサイズや配置位置といった表示方法を指定して、どれが見出しでどれが本文かがわかるようにします。しかし、HTMLでは表示の仕方は一切指定しません。「見出し」「段落」などの文書を構成している要素に印を付け、それぞれがなにかをわかるようにします。

実際のHTMLの印は英語を元にしてしますので少し違いますが、おおまかなイメージとしては次のようになります。

<見出し>名古屋の絶品グルメ <ここまで見出し>

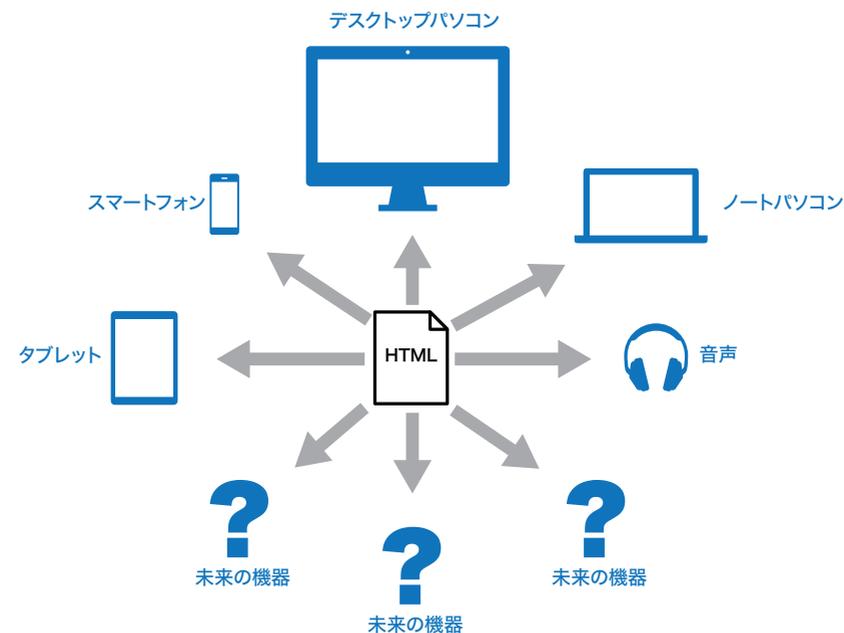
<段落> いわゆる「なごやめし」と呼ばれる名物料理の中には、美味しいものもあればちょっと首をかしげたくするようなものもあります。そして当然ですが、同じ料理でも店によって味はぜんぜん違ってきます。 <ここまで段落>

このような印(上の例では赤で示した部分)のことをHTMLでは「タグ」と言います。HTMLでは、文書を構成する要素をそれぞれタグで囲み、その部分がなにかを示します。タグで囲った部分をどのように表示するかは、後述のCSSと呼ばれる言語で指定します。

HTMLのタグのもっとも基本的な役割は、文書内の各構成要素の範囲と、その範囲がなにかを明確に示すことです。ただし、タグの中には画像や動画を文書内に組み込むものや、ほかのページにつながるリンクを作成するものなど、特別な機能を持つタグもあります。

HTMLで表示方法を指定しない理由

HTMLをテキスト形式にすることと、表示方法を指定しないことの利点は2つあります。1つは互換性です。たとえばWordの「.doc」で終わる拡張子のファイルなど、アプリケーション独自の形式を持つファイルの場合、現在は開くことが可能でも、10年後にその文書を開くことができる保証はありません。しかし、テキスト形式ならばたいていの文書作成ソフトで対応しているため、ほぼ確実に開くことができます。もう1つは、タグでその部分がなにかだけを示すなら、様々な環境で文書が利用可能になるということです。コンピュータ側でどれが見出しでどれが本文か、といった文書の構成要素が正しく認識できていれば、出力する機器に合わせて自動的にレイアウトもできます。環境や状況に応じた新しい使い方ができるようになるわけです。コンピュータが文書の構造をはっきりと認識できるようにしたHTMLは、これから登場するであろう未来の機器でも閲覧可能なファイル形式であると言えます。



グローバルメニューをつくる

<nav>

<nav>は、その部分が主要なメニュー（ナビゲーション）であることを示します。ページ内容を音声で読み上げる場合などに、メニューを読み飛ばしたり、逆にメニューに戻ったりできるようにすることを意図して用意されている要素です。

» 主要コンテンツを案内するグローバルメニュー

グローバルメニュー（グローバルナビゲーション）とは、サイト内にある主要コンテンツへのリンクをまとめたメニューのことです。一般に、すべてのページで共通して同じ場所に設置します。サイト全体ではなく特定の範囲や階層内において使用されるメニューのことは、ローカルメニュー（ローカルナビゲーション）と言います。

要素/プロパティ

HTML <nav>グローバルメニュー</nav>

nav要素は、ページ内の主要なナビゲーションのリンクを含むセクションを示すための要素です。一般的にはグローバルメニューをつくるために使用します。

グローバルメニューは多くの場合、同じ内容のものがページごとにメインコンテンツよりも前に配置されています。そのため、ページの内容を音声で読み上げさせると、毎回メインコンテンツの前に、それらの内容が読み上げられることになります。nav要素を使ってグローバルメニューであることをわかるようにしておくと、それらを読み飛ばしたり、逆に途中でグローバルメニューに戻ったりすることがかんたんにできるようになります。

nav要素はそのような用途を意図して用意されているため、ページ内のあちこちにnav要素があると、メニューに戻りたくても意図したものと違うメニューに戻ってしまうなどの混乱が生じる可能性があります。そのため、nav要素は主要なメニューにだけ使用することが推奨されています。

主要なナビゲーション全体の範囲はnav要素であらわしますが、ナビゲーション本体のマークアップには、リストをあらわすul要素とli要素が一般的に使用されます。また、ナビゲーションはリンクなので、a要素も使用します。

HTML

 Sec011

```
<nav>
  <ul>
    <li><a href="">ホーム</a></li>
    <li><a href="">会社案内</a></li>
    <li><a href="">製品情報</a></li>
    <li><a href="">お問い合わせ</a></li>
  </ul>
</nav>
```

CSS

 Sec011

```
nav ul {
  margin: 50px 0 0;
  padding: 0;
  text-align: center;
}
nav li {
  display: inline;
  margin: 0 0.5rem;
}
a {
  text-decoration: none;
}
```

▶ 実行結果(ブラウザ表示)

ホーム 会社案内 製品情報 お問い合わせ

行の文字揃えを指定する text-align

テキストの行揃えは、text-alignプロパティで設定します。値は、左揃えなら「left」、右揃えなら「right」、中央揃えなら「center」、両端揃えなら「justify」というようにキーワードで指定します。

行を揃える方向を指定

要素/プロパティ

CSS text-align: 行を揃える方向;

text-alignプロパティは、テキストの行揃えを設定するプロパティです。かつてはこのプロパティは単純に行揃えを指定するだけのプロパティでしたが、現在ではtext-align-allプロパティとtext-align-lastプロパティの両方の機能を併せ持ったショートハンドプロパティとなっています(ショートハンドプロパティとは、複数のプロパティの値をまとめて指定可能なプロパティのことです)。text-alignプロパティには、次の値が指定できます。

left	左揃えにする
right	右揃えにする
center	中央揃えにする
justify	両端揃えにする (最終行は除く)
justify-all	両端揃えにする (最終行を含む)
start	行の開始側に揃える (日本語の場合は左揃え)
end	行の終了側に揃える (日本語の場合は右揃え)
match-parent	親要素と同じにする

日本語の場合、値「start」は「left」と同じで、値「end」は「right」と同じになります。右側から左に書き進める言語もしくはそのように設定されている場合は「start」は「right」、「end」は「left」と同じになります。

HTML

```
<h1>大見出し</h1>
<p>これは段落内の文章です。・・・</p>
```

Sec031

CSS

Sec031

```
h1 { text-align: center; }
p { text-align: right; }
```

実行結果(ブラウザ表示)

大見出し

これは段落内の文章です。これは段落内の文章です。これは段落内の文章です。

両端揃えの最後の行の扱い

text-alignの値「justify」と「justify-all」の違いは、行の途中までしかテキストがない行(行の途中で改行されている場合や、段落の末尾など)の表示方法です。「justify」を指定した場合は、行の途中までしかテキストがない行は両端揃えにはなりません。「justify-all」を指定すると、行の途中までしかテキストがない行も含め、すべての行が両端揃えになります。

ただし、現時点では「justify-all」に対応していないブラウザもあるので、使用するときは注意してください。

クリック中のリンクの色を変更する :active

擬似クラスの :active を使用すると、マウスボタンなどが押されている最中に適用するCSSを指定できます。このセレクタの指定は、:link や :visited、:hover よりもあとに配置します。なお、ここではa要素に設定していますが、ほかの要素でも適用可能です。

マウスボタンを押している状態のリンクにスタイルを適用

要素/プロパティ

CSS :active { ... }

:active は、その要素がアクティベートされている間(マウスボタンであれば、押してから放すまでの間)にだけスタイルを適用するセレクタです。

HTML

📄 Sec081

```
<p>
<a href="destination.html">リンクテキスト</a>
</p>
```

CSS

📄 Sec081

```
a:link { color: blue; }
a:visited { color: purple; }
a:hover { color: red; }
a:active { color: black; }
```

実行結果(ブラウザ表示)

リンクテキスト



マウスが重なったときのリンクの見た目を変更する :hover

擬似クラスの :hover を使用すると、マウスポインターが上に置かれている最中に適用するCSSを指定できます。このセレクタの指定は、:link や :visited よりもあと、:active よりも前に配置します。なお、ここではa要素に設定していますが、ほかの要素でも適用可能です。

マウスポインターが上に置かれているときにスタイルを適用

要素/プロパティ

CSS :hover { ... }

:hover は、マウスなどのポインティングデバイスを使って、ポインターを要素の上に重ねている間のみスタイルを適用するセレクタです。このセレクタの表示指定は、スマートフォンのようなポインティングデバイスを使用しない機器では無効になります。

HTML

📄 Sec082

```
<p>
<a href="destination.html">リンクテキスト</a>
</p>
```

CSS

📄 Sec082

```
a { text-decoration: none; }
a:link, a:visited { color: rgb(51,51,51); }
a:hover {
  color: rgba(51,51,51,0.5);
  text-decoration: underline;
}
a:active { color: red; }
```

実行結果(ブラウザ表示)

リンクテキスト



リストの違いと使い分け

 <dl>

HTMLには、大きく分けて3種類のリスト（項目を列挙する形式のテキスト）があります。は一般的な箇条書き、は連番付きの箇条書き、<dl>は各項目がペアになっている「用語解説」形式のリストです。

一般的な箇条書き形式のリスト

要素/プロパティ

HTML 一般的な箇条書きの各項目

一般的な箇条書き形式のリストを表示する場合は、タグを使用します。内容の各項目は、タグで囲って示します。

HTML

Sec084_1

実行結果(ブラウザ表示)

```
<ul>
  <li>先頭が記号のリスト項目 1</li>
  <li>先頭が記号のリスト項目 2</li>
</ul>
```

- 先頭が記号のリスト項目 1
- 先頭が記号のリスト項目 2

連番付きの箇条書き形式のリスト

要素/プロパティ

HTML 連番付きの箇条書きの各項目

連番付きの箇条書き形式のリストを表示する場合は、タグを使用します。内容の各項目は、と同じくタグで囲って示します。

HTML

Sec084_2

実行結果(ブラウザ表示)

```
<ol>
  <li>先頭が連番のリスト項目 1</li>
  <li>先頭が連番のリスト項目 2</li>
</ol>
```

- 先頭が連番のリスト項目 1
- 先頭が連番のリスト項目 2

「用語解説」形式のリスト

要素/プロパティ

HTML <dl>「用語解説」形式の各項目</dl>

用語とその解説といったように、各項目がペアになっている「用語解説」形式のリストを表示する場合は、<dl>タグを使用します。内容の各項目は、<dt>で囲ったテキストと<dd>で囲ったテキストで示します。

HTML

Sec084_3

```
<dl>
  <dt>用語 1</dt>
  <dd>用語 1 に関する説明の文章 1 です。</dd>
  <dt>用語 2</dt>
  <dd>用語 2 に関する説明の文章 2 です。</dd>
</dl>
```

実行結果(ブラウザ表示)

用語 1
用語 1 に関する説明の文章 1 です。

用語 2
用語 2 に関する説明の文章 2 です。

画像をトリミングして表示する object-fit

画像や動画は、指定した幅と高さ単純に伸び縮みするだけでなく、指定サイズの領域内で縦横比を変えずに全体を表示させたり、隙間が生じない状態にして一部をトリミングして表示させたりもできます。

≫ 大きさを自動で調整

要素/プロパティ

CSS `object-fit: cover;`

width プロパティと height プロパティで元の縦横比とは異なるサイズを指定すると、表示される画像の縦横比も変わってしまい、いずれかの方向に引き伸ばされたような表示になってしまいます。画像に対して「object-fit: cover;」を指定すると、縦横比は変更せずに、指定された領域を隙間なく埋める大きさに自動調整して、余った部分はトリミングした状態で表示させることができます。次ページの例では、次のような縦長と横長の画像を順に表示させています。



CSS では縦横 200 ピクセルにして正方形で表示させていますが、「object-fit: cover;」が指定されているので、隙間がない状態になるまで拡大または縮小され、余った部分はトリミングされた状態で表示されています。

HTML

Sec108_1

```


```

CSS

Sec108_1

```
img {
  width: 200px;
  height: 200px;
  object-fit: cover;
}
```

▶ 実行結果(ブラウザ表示)



≫ どのパターンでフィットさせるのかを指定

要素/プロパティ

CSS `object-fit: キーワード;`

object-fit プロパティは、指定されたサイズの領域の中で、画像や動画をどのパターンでフィットさせて表示するのかを設定します。次の5種類のキーワードのうち、どれか1つが指定できます。縦横比が変わるのは初期値の「fill」だけです。

fill	画像を引き伸ばしたり縮めたりして指定された大きさにする。縦横比は維持されない
cover	縦横比を維持したまま、指定されたサイズの領域全体を隙間なく埋める(最小の)大きさにする
contain	縦横比を維持したまま、指定されたサイズの領域内で画像全体が表示される(最大の)大きさにする
none	画像は引き伸ばしたり縮めたりせずに元の大きさのまま指定されたサイズ内に表示する
scale-down	値「contain」と「none」のうちの小さくなる方で表示する

114

CSS

背景画像を表示する
background-image

ボックスの背景として画像を表示させるには、background-imageプロパティを使用します。画像のパスは、url() という関数形式の書式で指定します。背景には複数の画像を表示させることも可能です。

» 背景画像のパスを指定

要素/プロパティ

CSS background-image: url(画像のパス);

CSS background-image: url("画像のパス");

background-imageプロパティは、ボックスに背景画像を表示させるプロパティです。任意の要素に背景画像を表示できます。CSSでは、画像のパスはurl() という関数形式の書式で指定します。画像のパスは引用符で囲っても問題ありません。

下の例では、次の画像を背景画像として使用しています。背景画像が縦横に繰り返し表示されていますが、これはbackground-repeatプロパティで制御できます。

**CSS**

Sec114

```
body { background-image: url(dot.png); }
```

▶ 実行結果(ブラウザ表示)



なお、背景画像はカンマで区切って複数指定できます。先(左側)に指定している画像ほど、上に重なって表示されます。

115

CSS

背景画像の位置を固定する
background-attachment

背景画像は、初期状態だとコンテンツとともにスクロールします。background-attachmentプロパティを使用することで、背景画像だけを表示領域に固定して動かないようにすることができます。

» 背景画像を表示領域に固定する

要素/プロパティ

CSS background-attachment: fixed;

background-attachmentプロパティの値として「fixed」を指定すると、その背景画像は表示領域(ビューポート)に対して固定され、コンテンツがスクロールしても動かなくなります。

HTML

Sec115

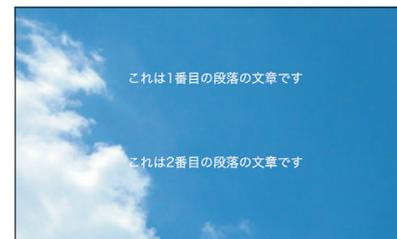
```
<p>これは1番目の段落の文章です</p>
~中略~
<p>これは9番目の段落の文章です</p>
```

CSS

Sec115

```
body {
  background-image: url(cloud.jpg);
  background-attachment: fixed;
}
```

▶ 実行結果(ブラウザ表示)



表のセルを横や縦に結合する colspan rowspan

colspanとrowspanは、わかりやすく言うとセルを結合させる属性です。正確には、colspanは指定された列数分だけセルを横に拡張させる属性で、rowspanは指定された行数分だけセルを縦に拡張させる属性です。

≫ 列または行のいくつか分引き伸ばすか

要素/プロパティ

HTML `<td colspan="何列分の幅にするか"> ~ </td>`

HTML `<td rowspan="何行分の高さにするか"> ~ </td>`

colspan属性を指定されたセルは、そこから指定された列数分の幅に拡張されます。rowspan属性を指定されたセルは、そこから指定された行数分の高さに拡張されます。なお、拡張された範囲に元々あるセルは不要となるため、HTMLからは削除しておく必要があります。これらの属性は<th>でも使用できます。

HTML

[Sec127](#)

```
<table border="1">
  <tr><td colspan="2">01</td><td>03</td></tr>
  <tr><td colspan="3">04</td></tr>
  <tr><td>07</td><td>08</td><td rowspan="2">09</td></tr>
  <tr><td>10</td><td>11</td></tr>
</table>
```

▶ 実行結果(ブラウザ表示)

01	03	
04		
07	08	09
10	11	

表のヘッダーとフッターをつくる <thead> <tfoot>

<thead>は表のヘッダー部分、<tfoot>は表のフッター部分をあらわすタグです。要素内容には0個以上のtr要素が入られます。複数のヘッダー行をグループ化したり、CSSを適用しやすくしたい場合などに活用できます。

≫ 表のヘッダーとフッター

要素/プロパティ

HTML `<thead>表のヘッダー行</thead>`

HTML `<tfoot>表のフッター行</tfoot>`

HTMLの表には、表のヘッダー部分をあらわすthead要素とフッター部分をあらわすtfoot要素が用意されています。両者とも要素内容としては0個以上のtr要素を入れることができますが、thead要素もtfoot要素も1つの表内に1つずつしか配置できない点に注意してください。なお、これらの要素を追加しても、CSSを指定しない限りは表示上の変化はありません。

HTML

[Sec128](#)

```
<table border="1">
  <thead>
    <tr><th>品目</th><th>個数</th></tr>
  </thead>
  <tr><td>りんご</td><td>20</td></tr>
  <tr><td>みかん</td><td>35</td></tr>
  <tfoot>
    <tr><th>計</th><th>55</th></tr>
  </tfoot>
</table>
```

140

HTML

文字入力欄をつくる
<input type="text">

<input>で1行の文字入力欄をつくるには、type属性の値に「text」を指定します。「text」はtype属性の初期値なので、この属性を省略した場合も文字入力欄になります。複数行の文字入力欄をつくる場合は<textarea>を使用します。

» 文字入力欄

要素/プロパティ

HTML <input type="text">

input要素のtype属性に「text」を指定すると、1行の文字入力欄になります。初期状態では20文字分の幅で表示されますが、幅はsize属性で変更できます。値には文字数を整数で指定します。ただし、この属性はあくまで入力した文字が問題なく見えるようにするために使うものです。正確な幅を指定するのであればCSSを使用してください。

文字入力欄に最初から文字が入力されている状態にしておくには、その文字をvalue属性の値として指定してください。

HTML

Sec140

```
<form action="userinfo.cgi" method="post">
  <label>
    名前:<input type="text" name="user_name">
  </label>
  <input type="submit">
</form>
```

▶ 実行結果(ブラウザ表示)

名前:

141

HTML

チェックボックスをつくる
<input type="checkbox">

<input>でチェックボックスの項目をつくるには、type属性の値に「checkbox」を指定します。チェックボックスは、複数の選択肢の中から個数を限定せずに選択できるようにする場合に使用します。

» チェックボックス

要素/プロパティ

HTML <input type="checkbox">

input要素のtype属性に「checkbox」を指定すると、チェックボックスになります。チェックボックスが1つの項目に対する複数の選択肢となっている場合、それらのチェックボックスすべてにname属性で同じ名前を付ける必要があります。データ送信時は、選択されたチェックボックスのvalue属性の値が送信されます。checked属性は、その項目が最初から選択されている状態にしたいときに使用します。

HTML

Sec141

```
<p>
<label><input type="checkbox" name="place" value="hokkaido">北海道</label>
<label><input type="checkbox" name="place" value="tokyo" checked>東京</label>
<label><input type="checkbox" name="place" value="okinawa">沖縄</label>
</p>
```

▶ 実行結果(ブラウザ表示)

北海道 東京 沖縄

2カラムレイアウトの位置を左右逆にする

このセクションでは、SECTION 219～221で示した「2カラムレイアウトの例」の、2カラムになっている部分のボックスを入れ替え、左右を逆にする方法を紹介합니다。HTMLについてはSECTION 219～221と同じものを使い、CSSだけを変更します。

≫ floatによる2カラム部分を入れ替える

floatの場合は、main要素とaside要素に指定しているfloatプロパティの値「left」と「right」を入れ替えると、2カラムのボックスが左右逆になります。

なお、この例のHTMLはSECTION 219と同一のため省略しています。HTMLコードの詳細は、SECTION 219(P.380)を参照してください。

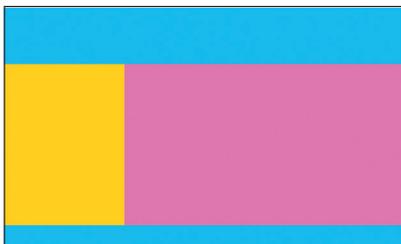
CSS

Sec222_1

```
body {
  margin: 0;
}
header {
  width: 100%;
  height: 70px;
  background: deepskyblue;
}
main {
  float: right;
  width: calc(100% - 150px);
  height: 200px;
  background: hotpink;
}
aside {
  float: left;
  width: 150px;
  height: 200px;
  background: gold;
}
```

```
footer {
  clear: both;
  width: 100%;
  height: 30px;
  background: deepskyblue;
}
```

▶ 実行結果(ブラウザ表示)



≫ Flexboxによる2カラム部分を入れ替える

Flexboxの場合は、orderプロパティを使用します。値には順番を示す整数を指定します。orderプロパティの初期値は0です。

aside要素に1、main要素に2と指定することでaside要素が左側、main要素が右側になります。ただし、これだけだとフッターの順番が0のままなので、フッターを最後に表示させるために、「order: 3;」をフッターに指定します。

なお、この例のHTMLはSECTION 220と同一のため省略しています。HTMLコードの詳細は、SECTION 220(P.382)を参照してください。

CSS

Sec222_2

```
body {
  display: flex;
  flex-wrap: wrap;
  margin: 0;
}
header {
  width: 100%;
  height: 70px;
  background: deepskyblue;
}
main {
  order: 2;
  width: calc(100% - 150px);
  height: 200px;
  background: hotpink;
}
aside {
  order: 1;
  width: 150px;
  height: 200px;
  background: gold;
}
```

```
footer {
  order: 3;
  width: 100%;
  height: 30px;
  background: deepskyblue;
}
```

▶ 実行結果(ブラウザ表示)

