

HTML の基本

HTML とは

Web ページを記述するための[マークアップ言語](#)です。まずは簡単な HTML の見本を作成してみましょう。

HTML の基本(basic/index.html)

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML の基本</title>
</head>
<body>
  <p>こんにちは</p>
  <p>よろしくお願いします</p>
</body>
</html>
```

解説

[要素](#)は一部を除き開始タグと終了タグをセットで記述します。

[DOCTYPE 宣言](#)

```
<!DOCTYPE html>
```

ブラウザでの描画を完全標準準拠モードで適用させます。HTML5 ではこの記述になります。

[<html>](#)

```
<html lang="ja">
  HTML 文書の内容を記述
</html> 終了タグ
```

HTML 文書である事を表します。HTML 文書で基点(ルート)となる要素です。

必須であり省略できません。子要素には<head>と<body>要素があります。

すべてのタグに属性があり、それぞれに意味があります。

例えば「lang="ja"」は<html>の属性です。ブラウザが日本語のコンテンツとして解釈します。

<head>

文書 メタデータ を記述する要素です。ここに記述するのはタイトルや文字コードなどの情報です。画面上には直接表示されないものの、重要な情報を記載するための要素です。子要素として次の要素がある。

[<title>](#) (必須) [<meta>](#) [<link>](#) [<style>](#) [<script>](#) [<base>](#) [<noscript>](#) [<template>](#)

では、重要な要素から解説します。

<title>

```
<title>HTML の基本</title>
```

ブラウザのタイトルバーやページのタブに表示される文書の題名を定義できます。必須なので省略できません。

<meta>

```
<meta charset="UTF-8"> 文字コードを UTF=8 に設定している
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

HEAD 要素内に記述する文書のメタデータのことです。属性により様々な META 情報を設定できます。

<body>

```
<body>
  ここに本文
</body>
```

文書の本文要素。この部分がブラウザに表示される。区分化ルートと呼ばれる。

<p>

```
<p>こんにちは</p>
<p>よろしくお願いします</p>
```

テキストの段落を表します。初期設定で隣接するブロックと上下の空白ができます。

実際にエディタで作成してプレビューしてみましょう。
作成手順に慣れて反復して作成練習をすることで復習になります。

HTML 要素の分類

HTML の要素はたくさんありますが、要素ごとに分類されています。

まずは主要な要素を理解します。機能の特徴を理解しましょう。

(ア) メタデータコンテンツ

文書に対するメタデータ（文書構成に関する情報）を記述します。

<head>内に記述して文書に関する情報を記述します。これらはブラウザには表示されません。

代表的な要素

[`<link>`](#) [`<meta>`](#) [`<script>`](#) [`<style>`](#) [`<title>`](#)

(イ) 区分コンテンツ

文書の区分（セクション）を示します。これにより区分が定義され、ユーザーエージェントに目次のように解釈されます。

代表的な要素

[`<article>`](#) [`<aside>`](#) [`<nav>`](#) [`<section>`](#)

(ウ) 見出しコンテンツ

このコンテンツに属する要素は文書の見出しを定義します。

これにより区分（セクション）が定義されます。区分コンテンツと同じく、ユーザーエージェントに目次のように解釈されます。

代表的な要素

[`<h1>`](#) [`<h2>`](#) [`<h3>`](#) [`<h4>`](#) [`<h5>`](#) [`<h6>`](#)

(エ) 埋め込みコンテンツ

動画・画像・音声など文字ではないコンテンツが属します。

代表的な要素

[``](#) [`<video>`](#) [`<audio>`](#) [`<iframe>`](#)

区分要素

区分要素には次の 4 つの要素があります。

[`<nav>`](#)

現在の文書内の他の部分や他の文書へのナビゲーションリンクを提供するためのセクションを表します。
一般的な例としてメニュー、目次、索引などがあります。

[`<article>`](#)

文書などのコンテンツが自己完結しており、個別に配信や再利用を行うことが可能な構成物を表します。
例えば、フォーラムの投稿、雑誌や新聞の記事、ブログの記事、投稿したコメントなどがあります。

[`<section>`](#)

HTML 文書の中で単独のセクション（区間）を表します。`<article>`より明確ではないが見出しを付けてセクションを作成できる場合に使用します。`<article>`が適切であればそちらを優先します。

[`<aside>`](#)

文書のメインコンテンツと間接的な関係しか持っていない文書の部分を表現します。サイドバーやコールアウトボックスなどを表現するためによく使われます。

関連する要素たち

[`<header>`](#)

導入部やナビゲーション等のグループを表します。見出し要素、ロゴ、検索フォーム、作者名などの要素を含むことが可能です。

[`<footer>`](#)

直近の区分コンテンツまたは区分化ルート要素のフッターを表します。フッターには通常、そのセクションの著作者に関する情報、関連文書へのリンク、著作権情報等を含めます。

[`<main>`](#)

`<body>` の主要な内容を表します。主要な内容とは、文書の中心的な主題のことです。

[`<div>`](#)

CSS を適用させるための要素として使用される。（デザインフック）

区分要素を用いたテンプレートの見本(template/index.html)

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>区分要素を用いたテンプレート</title>
</head>
<body>
  <div class="container">
    <header>
      <h1>ページの見出し</h1>
      <nav>
        <ul>
          <li><a href="#">link</a></li>
          <li><a href="#">link</a></li>
          <li><a href="#">link</a></li>
          <li><a href="#">link</a></li>
        </ul>
      </nav>
    </header>
    <main>
      <section>
        <h2>セクションの見出し</h2>
        <article>
          <h3>記事の見出し</h3>
          <p>記事の内容</p>
        </article>
        <article>
          <h3>記事の見出し</h3>
          <p>記事の内容</p>
        </article>
      </section>
    </main>
    <aside>
      <ul>
        <li><a href="#">link</a></li>
        <li><a href="#">link</a></li>
        <li><a href="#">link</a></li>
        <li><a href="#">link</a></li>
      </ul>
    </aside>
    <footer>
      <p>copyright</p>
    </footer>
  </div>
</body>
</html>
```

解説

<div>によるデザインコンテナ (コンテンツ全体を包括)

```
<div class="container">  
  
</div>
```

この要素は区分要素ではないため、CSS を適用するための器 (コンテナ) として使用します。

<header>によるページ全体のヘッダー作成と<nav>によるグローバルナビゲーション

```
<header>  
  <h1>ページの見出し</h1>  
  <nav>  
    <ul>  
      <li><a href="#">link</a></li>  
      <li><a href="#">link</a></li>  
      <li><a href="#">link</a></li>  
      <li><a href="#">link</a></li>  
    </ul>  
  </nav>  
</header>
```

ページヘッダーを作成できます。

ここにはページの見出しを記述します。これはユーザーエージェントによって解釈されて、WEB ページの見出しになるために重要です。

併せてグローバルナビゲーション (主たるナビゲーション) を包括する場合があります。

<nav>を用いて箇条書きの要素を配置するのが一般的です。

箇条書き (リスト) 要素について

箇条書き (リスト) の要素には「順序なしリスト」と「順序付きリスト」

[](#)

順序なしリストを表します。行頭記号を伴うリストとして描画されます。

CSSにより行頭記号を変更できます。

[](#)

項目の順序付きリストを表します。番号付きのリストとして表示されます。

CSSにより番号の種類を変更できます。

[](#)

リストの項目を表すために用いられます。リストの各項目を表示します。

この要素に包括した要素がリストの表示項目になります。

[<a>](#)

他のページへのハイパーリンクを作成します。この要素に包含して、href 属性にリンク対象の URL を記述します。

例ではlinkと記述しています。link の文字にハイパーリンクが設定されます。

<main>による主たるコンテンツの記述・区分要素によるコンテンツの整理

```
<main>
  <section>
    <h2>セクションの見出し</h2>
    <article>
      <h3>記事の見出し</h3>
      <p>記事の内容</p>
    </article>
    <article>
      <h3>記事の見出し</h3>
      <p>記事の内容</p>
    </article>
  </section>
</main>
```

<main>は区分要素ではありませんが、<body>要素内の主たるコンテンツとして記述する部分に使用します。

<main>の子要素として<article>や<section>を用いて、内容を区分しつつ記述します。

これらの区分要素には内容に対する見出しを設定するのが一般的です。それとともに<p>で内容を記述します。

<aside>による関連要素

```
<aside>
  <ul>
    <li><a href="#">link</a></li>
    <li><a href="#">link</a></li>
    <li><a href="#">link</a></li>
    <li><a href="#">link</a></li>
  </ul>
</aside>
```

今回の例はサイドバーをイメージして記述しています。

主なコンテンツではなく関連した話題なので、外部サイトへのリンクをリストで設定しています。

サイドバーとメインのコンテンツを段組みして表示するにはCSSが必要になります。

<footer>によるページ全体のフッター作成

```
<footer>
  <p>copyright</p>
</footer>
```

ページ全体のフッターを作成しています。ここにはコピーライトやサイトマップや連絡先などが記載されます。

例には無いですが[<address>](#)のリンクも載せておきます。

見出し要素

見出し要素

見出し要素は<h1> <h2> <h3> <h4> <h5> <h6>の 6 種類がある。

<h1> が最上位で、<h6> が最下位です。デフォルトの CSS の効果で文字の大きさと太さが異なります。

使用上の注意

1. テキストの大きさを変えるために見出し要素を使用しないでください。
代わりに CSS の font-size プロパティを使用してください。
2. 見出しレベルを飛ばすことは避けてください。
常に <h1> から始め、次に <h2>、以下も同様にしてください。
3. 1 つのページで <h1> を 2 回以上使用することは避けてください。
複数の <h1> を使用してもエラーにはなりませんが、これは最善の方法とは見なされていません。
しかし読み上げソフトや SEO で有利になる場合があります。

では、実際に見出し要素がユーザーエージェントにどの様に解釈をされるのかを見てみましょう。

次のサンプルを作成します。

見出し要素の見本(heading/index.html)

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>見出し要素のサンプル</title>
</head>
<body>
  <h1>Heading-1</h1>
  <h2>Heading-2</h2>
  <h3>Heading-3</h3>
  <h4>Heading-4</h4>
  <h5>Heading-5</h5>
  <h6>Heading-6</h6>
  <h2>Heading-2</h2>
  <h3>Heading-3</h3>
  <h4>Heading-4</h4>
  <h4>Heading-4</h4>
  <h5>Heading-5</h5>
  <h6>Heading-6</h6>
  <h5>Heading-5</h5>
  <h3>Heading-3</h3>
  <h4>Heading-4</h4>
  <h4>Heading-4</h4>
</body>
</html>
```

完成したら[こちら](#)で確認しましょう。

次のように表示されているでしょう。

- 1. Heading-1
 - 1. Heading-2
 - 1. Heading-3
 - 1. Heading-4
 - 1. Heading-5
 - 1. Heading-6
- 2. Heading-2
 - 1. Heading-3
 - 1. Heading-4
 - 2. Heading-4
 - 1. Heading-5
 - 1. Heading-6
 - 2. Heading-5
- 2. Heading-3
 - 1. Heading-4
 - 2. Heading-4

このように文章の見出しとして章や節の役割をして、コンテンツの重要度をユーザーエージェントに伝えます。これを意識して作成することによって内容を整理してユーザーエージェントに解釈させると良いでしょう。

前節の区分要素を用いたテンプレートを見てみましょう。

- 1. ページの見出し
 - 1. *Untitled Section*
- 2. セクションの見出し
 - 1. 記事の見出し
 - 2. 記事の見出し
- 3. *Untitled Section*

Untitled Sectionと表示されているのは<nav>と<aside>の部分です。そこには見出し要素が無いからです。

置換要素

置換要素は画像や動画などの文字以外の情報を表示するための要素です。「置換」と呼称されるのは、それぞれの要素が独自の形式に置換られて表示されるからです。では、代表的な置換要素を見ていきましょう。

置換要素の見本(replaced/index.html)

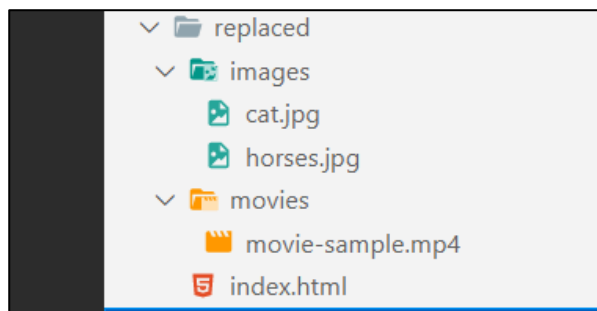
```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>置換要素の見本</title>
</head>
<body>
  <!-- 画像 -->
  
  

  <!-- 動画 -->
  <video src="./movies/movie-sample.mp4" controls>
    このブラウザでは再生できません。
  </video>

  <!-- GoogleMap など -->
  <iframe title="Inline Frame Example" width="300" height="200"
    src="https://www.openstreetmap.org/export/embed.html?bbox=-
0.004017949104309083%2C51.47612752641776%2C0.00030577182769775396%2C51.478569861898606&la
yer=mapnik">
  </iframe>

</body>
</html>
```

画像や動画の配置は以下の通りです。



画像の表示要素です。

```


```

src属性

画像の場所を指定します。必須属性で省略できません。この例は相対パス指定で記述されています。

alt 属性

読み上げソフトで読み上げられます。また、src 属性の画像がリンク切れの場合は代替え文字として表示されます。

<video>

動画の表示要素です。

```
<video src="./movies/movie-sample.mp4" controls>
  このブラウザでは再生できません。
</video>
```

src 属性

埋め込む映像コンテンツへの URL を指定します。この属性は省略可能です。

埋め込む映像の指定には、video 要素のブロック内で <source> を使用することもできます。

controls 属性

この属性が指定された場合、再生、音量、シーク、ポーズの各機能を制御するコントロールを表示します。

<iframe>

インラインフレームを用いて他のコンテンツを表示します。

```
<iframe
  title="Inline Frame Example"
  width="300"
  height="200"
  src="https://www.openstreetmap.org/export/embed.html?bbox=-
0.004017949104309083%2C51.47612752641776%2C0.00030577182769775396%2C51.478569861898606&la
yer=mapnik">
</iframe>
```

title 属性

読み上げソフトで読み上げられます。

src属性

動画の場所を指定は表示したい資源の URL を指定します。

width 属性

フレームの幅を CSS ピクセル数で示します。既定値は 300 です。

height 属性

フレームの高さを CSS ピクセル数で示します。既定値は 150 です。