HTMLとは

# なぜにHTMLを学ぶのか？

ズバリ、Webサイトを作成には必須のものになります。

これを理解しないとサイト作成ができません。

# HTMLってなに？

HTMLとは(HyperTextMarkupLanguage)ハイパーテキスト・マークアップ・ランゲージと呼ばれるます。

Webページを作るための最も基本的な言語です。

webページを裏側を見てよう。

どのページでもいいけど、ページのソースを見てみる！(ブラウザによって表示方法が違うので)

並んでいる文字がソースコードといいます。

このように、WebページはHTMLタグを使って構成されています。

これを読み込みブラウザで表示しているだけです。

# htmlファイルを作る方法は？

htmlファイルの作り方ですが、テキストエディタなどでhtmlタグを書き「.html」という拡張子をつけることで、htmlファイルであることをコンピュータが認識してくれます。(例 index.html)

## リンクを表示する

リンクを作成したい場合は、他のページに飛ばす場合に、リンクタグ<a>タグを使います。

<a href=”http://google.com”>googleへのリンク</a>

## 画像を表示する

画像を表示させるためには、<img>タグを使用します。

<img src=”画像パス” />

## 段落分けを行う

段落を分けるには<p>タグを使用します。

<p>段落の内容</p>

## HTMLの基本の構造

<!doctype html>

<html>

<head>

<meta charset="UTF-8">

<title>タイトル</title>

</head>

<body>

<h1>見出し1</h1>

<p>段落１</p>

</body>

</html>

## HTMLの基本の記述方法

実際にWebページに反映されるのは< body >の中なので、< body >～</body >の間に記述します。

タグ HTMLでは基本的にこのタグ<>で文字列を囲んで記述します。

開始タグ<>と終了タグ</>で囲う、そのひとつのかたまりを要素と呼びます。

※タグには全角文字を使うことはできませんので注意します。

## 基本タグ

<head>・・・文書のヘッダ情報を表す

<title>・・・文書にタイトルをつける

<base>・・・相対パスの基準URIを指定する

<link>・・・リンクする外部リソースを指定する

<meta>・・・その文書に関する情報（メタデータ）を指定する

<style>・・・スタイルシートを記述する

<script>・・・文書にJavaScriptなどのスクリプトを組み込む

<noscript>・・・スクリプトが動作しない環境用の表示内容を指定する

<body>・・・文書の本体を表す

<section>・・・1つのセクションであることを示す

<nav>・・・ナビゲーションであることを示す

<article>・・・記事であることを示す

<aside>・・・余談・補足情報のセクションであることを示す

<h1>～<h6>・・・見出しを付ける

<header>・・・ヘッダであることを示す

<footer>・・・フッタであることを示す

<address>・・・連絡先・問い合わせ先を表す

<p>・・・ひとつの段落(パラグラフ)であることを表す

<hr>・・・テーマや話題の区切りを表す

<pre>・・・半角スペースや改行をそのまま表示する

<blockquote>・・・引用・転載セクションであることを表す

<ol>・・・順序のあるリストを表示する

<ul>・・・順序のないリストを表示する

<li>・・・リストの項目を記述する

<dl>・・・定義・説明リストを表す

<dt>・・・定義・説明される言葉を表す

<dd>・・・定義用語や言葉の説明をする

<figure>・・・図表であることを示す

<figcaption>・・・図表のキャプションを示す

<div>・・・ひとかたまりの範囲として定義する

<main>・・・メインコンテンツであることを示す

<a>・・・ハイパーリンクを指定する

<em>・・・強勢する（アクセントを付ける）箇所を表す

<strong>・・・強い重要性を表す

<small>・・・免責・警告・著作権などの注釈や細目を表す

<s>・・・すでに正確ではなくなった内容を表す

<cite>・・・作品のタイトルを表す

<q>・・・引用句・引用文であることを表す

<dfn>・・・用語が使用されていることを示す

<abbr>・・・略語や頭字語であることを表す

<time>・・・日付や時刻を正確に示す

<code>・・・プログラムなどのコードであることを示す

<var>・・・変数であることを示す

<samp>・・・プログラムによる出力結果のサンプルであることを示す

<kbd>・・・ユーザーが入力する内容であることを示す

<sub>・・・下付き文字を表す

<sup>・・・上付き文字を表す

<i>・・・声や心の中で思ったことなど、他と区別したいテキストを表す

<b>・・・文書内のキーワードや製品名など、他と区別したいテキストを表す

<mark>・・・文書内の該当テキストを目立たせる

<ruby>・・・ルビをふる

<rt>・・・ルビのテキストを指定する

<rp>・・・ルビを囲む記号を指定する

<bdo>・・・文字表記の方向を指定する

<span>・・・ひとつの範囲として定義する

<br>・・・改行する

<wbr>・・・改行しても良い位置を示す

<ins>・・・追加された部分であることを示す

<del>・・・削除された部分であることを示す

<img>・・・画像を表示する

<iframe>・・・インラインフレームを作る

<embed>・・・プラグインデータを埋め込む

<object>・・・文書に外部リソースを埋め込む

<param>・・・プラグインのパラメータを指定する

<video>・・・動画を再生する

<audio>・・・音声を再生する

<source>・・・動画や音声などのURLや種類を指定する

<canvas>・・・図形を描く

<map>・・・イメージマップを作成する

<area>・・・イメージマップのハイパーリンク領域を設定する

<table>・・・テーブル（表）を作成する

<caption>・・・テーブル（表）にキャプションをつける

<colgroup>・・・表の縦列をグループ化する

<col>・・・表の縦列の属性やスタイルを指定する

<tbody>・・・テーブル（表）のボディ部分を定義する

<thead>・・・テーブル（表）のヘッダ部分を定義する

<tfoot>・・・テーブル（表）のフッタ部分を定義する

<tr>・・・テーブル（表）の横一行を定義する

<td>・・・テーブル（表）のデータセルを作成する

<th>・・・テーブル（表）の見出しセルを作成する

<form> …… 入力・送信フォームを作る

<fieldset> …… フォームの入力項目をグループ化する

<legend> …… フォームの入力項目グループにキャプションを付ける

<label> …… フォーム部品と項目名（ラベル）を関連付ける

<input> …… フォームを構成する様々な入力部品を作成する

<button> …… ボタンを作成する

<select> …… セレクトボックスを作成する

<datalist> …… 入力候補となるデータリストを定義する

<optgroup> …… 選択肢をグループ化する

<option> …… セレクトボックスや入力候補リストの選択肢を指定する

<textarea> …… 複数行のテキスト入力欄を作成する

<keygen> …… フォーム送信時にキーを発行する

<output> …… 計算結果を示す

<progress> …… タスク完了までの進行状況を示す

<meter> …… 規定範囲内の測定値を表する

<details> …… 備考や操作手段などの詳細情報を示す

<summary> …… <details>の内容の要約を示す

<command> …… 操作メニューの各コマンドを指定する

<menu> …… 操作メニューを作成する

## <head>タグ

<head>タグは、文書のヘッダ情報を表す際に使用します。

ヘッダ情報とは、その文書に関する情報のことです。

head要素を正確に定義するなら、「そのHTML文書に関するメタデータを集めたもの」ということになります。

**サンプル・・・003.html**

## <title>タグ

<title>タグは、文書にタイトルをつける際に使用します。

<title>タグで文書のタイトルを指定する場合には、一つの文書に一つだけ、<head>～</head>の中に配置します。

一つの文書内で複数の<title>を指定することはできません。

**サンプル・・・004.html**

## <base>タグ

<base>タグは、相対パスの基準URIを指定する際に使用します。

<base>タグを使用する場合には、一つの文書に一つだけ<head>～</head>の中に配置します。

一つの文書内で複数の<base>を指定することはできません。

<base>タグを使用する場合には、href属性とtarget属性のどちらか、または、両方を指定する必要があります。

href属性には基準URIを、target属性にはリンク先ターゲットを指定します。

### 属性

**ref属性**

基準となるURIを指定（必須属性）

**target属性**

リンク先ターゲットを指定

**サンプル・・・005.html**

## <link>タグ

**<link>タグは、リンクする外部リソースを指定する際に使用します。**

**外部リソースとは、例えば、HTMLファイル、CSSファイル、RSSファイルなど、現在の文書ファイルに関連している外部ファイルのことです。**

**<link>タグは、<head>～</head>の中に記述します。**

**<link>タグを指定する場合には、href属性とrel属性が必須です。**

**href属性とrel属性が指定されていない場合には、<link>タグの指定は無効となります。**

**href属性には、リンクする外部ファイルのURLを指定します。**

### 属性

**href属性**

**リンクする外部リソースのURL(必須属性)**

**rel属性**

**現在のファイルとリンク先の外部ファイルとの関係性を表すキーワード(必須属性)**

**サンプル・・・006.html**

## <meta>タグ

**<meta>タグは、その文書に関する情報(メタデータ)を指定する際に使用します。**

**メタデータとは、“情報に関する情報”のことですが、HTMLの仕様では“その文書に関する様々な情報”を意味します。**

**<meta>タグは<head>～</head>の中に配置します。 meta要素を指定する際には、 name属性、http-equiv属性、charset属性のうち、少なくとも一つを指定する必要があります。 また、name属性、http-equiv属性を指定する際には、必ずcontent属性とセットで利用します。 セットで利用することでメタデータの定義（名前）とその内容が関連付けられます。**

**charset属性で文字エンコーディングを指定**

**<meta charset="UTF-8">**

**name属性でメタデータ名を定義、content属性でその内容を指定**

**<meta name="keywords" content="アドライン,システム開発">**

**<meta name="description" content="アドラインはシステム開発をメインにした企業です">**

**サンプル・・・007.html**

## **<style>タグ**

**<style>タグは、スタイルシートを記述する際に使用します。**

**通常は<head>～</head>の中に配置しますが、 head要素内に記述することで文書単位でスタイルを指定することができます。**

**サンプル・・・008.html**

## <script>タグ

**<script>タグは、文書にJavaScriptなどのスクリプトや、データブロックを組み込む際に使用します。**

**<script>タグは、<head>～</head>の中に配置することが多いですが、 <body>～</body>の中に配置しても問題ありません。**

**サンプル・・・009.html**

## <noscript>タグ

**<noscript>タグは、スクリプトが動作しない環境用の表示内容を指定する際に使用します。**

**サンプル・・・010.html**