

HTML/CSSとは

▼この章の目的

HTMLとは何か、CSSとは何かを理解することが目的です。

▼この章の概要

Webページがどうやって作られているか、その構成を理解して行きましょう。
HTMLとCSSはWebページを作るための言語になっています。

このサイトや、普段みているWebページもHTMLとCSSで作られています。

▼ShopifyにおけるHTML/CSSの役割

Shopifyでは、HTMLやCSSの知識がなくても、テンプレートを用いて、カスタマイズ画面からWebページを作ることが可能です。

しかし、テンプレートに用意されていないデザインのページは、作ることができません。
カスタマイズ画面から作成できるデザインは限られています。

そこで、既存テンプレートのコードのHTML/CSSを変更したり、書き加えたりします。
そうすることで、Shopifyの機能を利用しつつ、デザインも自由に変更することができるようになります。

Shopifyを用いてサイトを制作する際にも、思い通りのデザインにするためには、
HTML/CSSの知識は必要不可欠なので、しっかりと学習していきましょう。

▼HTMLとは

Webページの文章や画像など骨子を作っているのがHTMLです。HTMLとは Hyper Text Markup Language の略称で、Webページを構成するマークアップ言語がHTMLです。
マークアップとは、サイト内の文書構造をWebブラウザが正しく理解できるように、**意味づけしていく作業**を指します。

人間であれば、サイトを見たときに、どの部分が見出しで、どの部分が本文かを見ただけで判断することができますが、Webブラウザにはそれができません。

Webブラウザに文書構造を正しく理解させるためには、文書に**タグ**と呼ばれる目印をつける必要があります。

タグの例)

h1タグの場合：<h1>テックギーク - Shopifyマスター</h1>

<>で囲まれたものをタグと呼びます。

タグにはひとつひとつ意味が決められているので、タグをつけることによりWebブラウザは正しく文書構造を理解することができます。

文書にタグを用いて意味をつけることを、マークアップと呼びます。

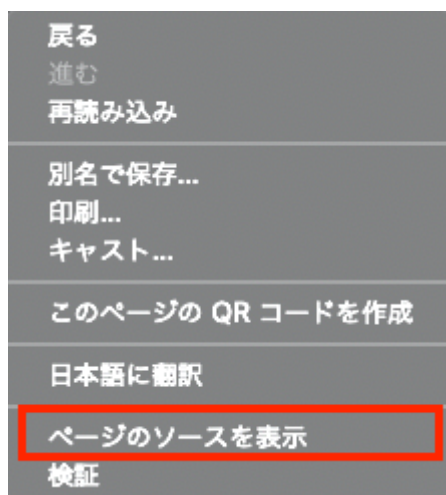
HTMLとはどのようなものか、実際に見てみましょう。

インターネットに公開されているウェブサイトは、どのサイトもどのようなHTMLで記述されているのか（ソースコードと呼びます）を見ることができます。
試しにテックギークのサイトのソースコードを見てみましょう。

[こちら](https://ryujimorimoto.github.io/techgeek-lp/)（<https://ryujimorimoto.github.io/techgeek-lp/>）からテックギークのサイトにアクセスしてください。

サイト上で右クリックする(トラッキングパッド使用の場合は2本指で押す)と、下記のメニューが表示されます。

「ページのソースを表示」をクリックしてください。



下記のような画面が表示されたと思います。

行を折り返す ☐

```
1 <!DOCTYPE html>
2 <html lang="ja">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>テックギーク - shopify専門のオンラインスクール</title>
9   <meta name="description" content="">
10
11   <link rel="icon" href="images/favicon.png" type="image/png">
12   <link rel="stylesheet" href="css/style.css">
13   <script src="js/script.js" defer></script>
14
15
16 </head>
17
18 <body>
19   <header class="header">
20     <div class="modal" id="modal-gray" onclick="closeNav()"></div>
21     <div class="header__inner container">
22       <div class="header__logo">
23         
24       </div>
25       <a href="#" class="header__btn header__sp-btn pc-none">無料面談予約</a>
26     <nav class="nav">
27       <ul class="nav__list">
28         <li class="nav__item"><a href="#about" onclick="closeNav()">テックギークとは</a></li>
29         <li class="nav__item"><a href="#reason" onclick="closeNav()">なぜShopify?</a></li>
30         <li class="nav__item"><a href="#learning" onclick="closeNav()">学習の流れ</a></li>
31         <li class="nav__item"><a href="#services" onclick="closeNav()">サービス一覧</a></li>
32         <li class="nav__item"><a href="#fees" onclick="closeNav()">料金</a></li>
33         <li class="nav__item"><a href="#questions" onclick="closeNav()">よくあるご質問</a></li>
34         <li class="nav__item"><a href="#" class="header__btn tb-none" onclick="closeNav()">無料カウンセリングを予約</a></li>
35         <li class="nav__item"><a href="#" class="header__btn tb-only" onclick="closeNav()">無料予約</a></li>
36       </ul>
37     </nav>
```

今はまだどのようなことが書かれているのかわからないかと思いますが、これから一緒に学習していきましょう。

▼HTMLの基本的な書き方

HTMLは、基本的には下記のように、表示したい内容を**開始タグ**と**終了タグ**で囲んで記述します。

```
<p>ここに文章を書きます</p>
```

「ここに文章を書きます」の前に書かれている**<p>**を開始タグ、後に書かれている**</p>**を終了タグと呼びます。

開始タグは、**タグ名**を<>の記号で囲んで記述します。

終了タグは、開始タグと同じタグ名を</と>の間に記述し、**</ タグ名>**のようにします。

上記の例だと、「p」がタグの意味を表すタグ名で、段落を意味します。

つまり、「ここに文章を書きます」というテキストは、webサイト上で段落として表示されます。

タグには、たくさんの種類があり、それぞれに意味を持ちます。

また、開始タグの中には**属性**と**属性値**が入る場合もあります。属性は、「追加の情報」を与えるもので、タグに何か追加情報を与えたい時に利用します。

例えばリンクを表す<a>タグには、必ず属性と属性値が必要です。

```
<a href="#top">トップ</a>
```

hrefが属性、"#top"が属性値です。

<a>タグのhref属性はリンク先を指定する属性なので、属性値には、""の中にリンク先を記述します。

また、開始タグから終了タグまでの全体を**要素**と呼び、開始タグと終了タグの間（上記の<a>タグの例で言うと、「トップ」）を**要素の内容**または**コンテンツ**と呼びます。

後述するclass属性やid属性は非常によく使われるので使い方をしっかり覚えましょう。

▼終了タグのないタグ

タグの基本的な書き方は、開始タグと終了タグで囲んで記述すると説明しましたが、一部のタグには、終了タグがないものもあります。

画像を表示するタグや、入力フォームを作成する際に使用する<input> タグなどは終了タグがありません。

下記のように記述します。

```

```

```
<input type="text" name="name">
```

こういった終了タグのないタグのことを**空要素**と呼びます。

▼ 親要素、子要素

タグの中に別のタグを入れることもよくあります。
例えば下記のように記述します。

```
<div>
  <p>ここにテキストを入力します</p>
</div>
```

外側にある要素（この場合<div>タグ）を**親要素**、内側にある要素（この場合<p>タグ）を**子要素**と呼びます。

子要素は、必ず親要素の開始タグから終了タグの間に入っていないてはいけません。
下記のように、子要素の終了タグを親要素の外側を書くことはできません。

```
<div><p>ここにテキストを入力します</div></p>
```

▼ 兄弟要素

下記のように同じ階層にある要素（下記の場合<p>タグ同士）は、**兄弟要素**と呼びます。

```
<div>
  <p>ここにテキストを入力します</p>
  <p>ここにテキストを入力します</p>
</div>
```

▼ HTMLのコメント文

コードの中にコメントを残すこともできます。
コメントはブラウザ上には表示されません。
下記のように、<!--と-->の間にコメントを記入します。

```
<!-- ここにHTMLのコメントを記入します -->
```

行全体をコメントにする場合は、コメント記入後に入力カーソルがコメントにする行にあるのを確認してから、**command + /**を押すと、行全体に<!-- -->がつき、コメントになります。

▼ HTMLの基本構造

HTMLの基本構造は下記のようになっています。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
```

```
</head>
<body>

</body>
</html>
```

これらのタグは、HTMLを記述する際に必須のタグです。
どのような意味を持つのか見ていきましょう。

▼<!DOCTYPE html>

<!DOCTYPE html>は文書型宣言と呼ばれ、このドキュメントは、HTML5で書かれているということを宣言します。

HTMLドキュメントの最初に必ず記述する必要があります。

大文字で書いても、小文字でも書いても良いです。

▼<html lang="ja">

<html>タグは、このドキュメントがHTMLドキュメントであることを示すタグです。

必ず文書型宣言の後に記述し、すべてのHTMLタグは、<html>～</html>の間に記述します。

lang属性には言語を指定します。

ページの内容を日本語で書く場合には、"ja"を指定します。

▼<head>

<head>タグはドキュメント全体の情報（設定情報をメタデータと呼びます）を入れるタグです。

<head>の中に、ページのタイトルや説明、SEO対策のためのタグなどを記述します。

<head>タグ内に書かれた内容は、ページには直接表示されません。

▼<meta charset="UTF-8">

<meta charset="UTF-8">は、文字コードを「UTF-8」にするという指定です。

この指定をしないと、文字化けをしてうまく表示されない場合があるので、必ず記述する必要があります。

"UTF-8"は"utf-8"のように小文字で書いても良いです。

この<meta charset="UTF-8">は<meta>タグの一種です。

<meta>タグとは、ページの情報を検索エンジンやブラウザに伝えるために使用されるタグです。

<meta>タグを用いて記述する情報は、このほかにもたくさんあります。

shopifyのテーマDawnでもいくつかの<meta>タグが使用されています。

他の<meta>タグについては、また後のカリキュラムで説明します。

▼<title>

<title>は、ページのタイトルを記述するタグです。

<title>〜</title>の間に書かれたテキストは、ブラウザのタブ、ブックマークされた際や検索画面のページタイトルとして表示されます。

(テックギークのサイトの画像を挿入)

▼<body>

ページに表示したいコンテンツは、全てこの<body>タグの中に記述します。Shopifyでのコーディングの大半はこのbodyタグの中に記述していきます。

▼CSSとは

主にHTMLで構成した文字、画像などをデザインする役割を持つのがCSSです。

CSSとは、Cascading Style Sheets の略称で、Webページのスタイルを指定するための言語です。

HTMLには、表示を制御する機能がないので、文字の色や背景色・配置などを変更することができません。

HTMLだけでは、Webページのデザインを整えることができないため、CSSを用いてデザインを整えます。

CSSを用いれば、文字の色や背景色・配置などの他にも、かなり複雑なデザインを表現することができます。

ほとんどのWebページは、**HTMLでページの内容を記述し、CSSで見た目を整える**といった方法で作成されています。

▼CSSの基本的な書き方

▼CSSの基本書式

CSSの基本的な書き方は下記の通りです。

```
セレクトー {  
  プロパティ: 値;  
}
```

HTMLの「どの部分の（**セレクトー**）、何を（**プロパティ**）、どのように変えるか（**値**）」を指定します。

具体的には、下記のように記述します。

```
h1 {  
  color: red;  
}
```

HTMLの**どの部分を変更するか**を選択するのが（上記の例では、h1）**セレクトー**です。

上記の例では、タグをセレクターとして使用しましたが、セレクターには様々なパターンがあります。

かなりたくさんのパターンがあるので、実際に使いながら少しずつ覚えていきましょう。セレクターで指定した部分の**何を変えるのか**を指定するのが（上記の例では、color）、**プロパティ**です。

colorは、テキストの色を変えるという指定です。

他にも、background-color（背景色の指定）や、font-size（文字サイズの指定）など様々なプロパティがあります。

こちらも実際に使いながら少しずつ覚えていきましょう。

値には、セレクターで指定した部分のプロパティを、どのような値に変更するかを指定します。

波括弧（{}）やコロン（:）、セミコロン（;）は、お決まりの文法なので、覚えるまでは真似をしながら記述してください。

▼CSSの適用方法

HTMLで作成したページにCSSを適用させるには、大きく分けて下記の3つの方法があります。

- CSSファイルを用意して、HTMLファイルから読み込む
- HTMLファイルの<head>内に<style>タグを追加して指定する
- HTMLタグの中にstyle属性を追加して指定する

一つ一つ見ていきましょう。

▼CSSファイルを用意して、HTMLファイルから読み込む

最も一般的なのが、この方法です。

まず、「.css」の拡張子がついたファイルを用意します。

次にHTMLの<head>内に下記のコードを追加し、ファイルを読み込みます。

href属性には、ファイル名を指定します。

```
<link rel="stylesheet" href="style.css">
```

実際に<head>タグ内に書くと下記のようになります。

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="style.css">
</head>
```

これで、style.cssにcssを記述すれば、スタイルが適用されます。

▼HTMLファイルの<head>内に<style>タグを追加して指定する

HTMLファイルの<head>内に<style>タグを追加して、その中にcssを記述する方法です。
具体的には、下記のように記述します。

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    h1 {
      color: red;
    }
  </style>
</head>
```

この方法だと、記述したHTMLファイルでのみ適用され、複数ページに共通するCSSの場合は、HTMLファイルごとに記述しなければなりません。
管理も難しくなるため、実際のサイトで使用されることは少ないですが、特定のページのみデザインを変えたい時などには使えます。

▼HTMLタグの中にstyle属性を追加して指定する

HTMLタグに直接CSSを記述する方法です。
下記のように記述します。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1 style="color: red;">タイトル</h1>
</body>
</html>
```

この方法だと、スタイルは直接書かれたタグにのみ適用されます。
一つ一つのタグに書かなくてはいけないので手間がかかりますが、他の方法で指定するよりもスタイルの優先順位が高いため、スタイルを上書きしたい場合や、特定の要素のみスタイルを適用したい場合に使えます。

▼CSSのコメント文

HTMLと同様に、CSSファイルの中にもコメントを残すことができます。
ページ数が増えてくると、かなりたくさんのCSSを記述する必要があります。

そういった場合に、どこに適用しているCSSかなど、CSSファイルの中にコメントを残しておくと、管理もしやすくなります。
下記のように記述します。

```
// ここにCSSの一行のコメントを記入します。

/* ここにCSSの複数行のコメントを記入します。
   ここにCSSの複数行のコメントを記入します。
*/
```

HTMLと同様に、行全体をコメントにする場合は、コメント記入後に入力カーソルがコメントにする行にあるのを確認してから、**command + /**を押すと、コメント文にすることができます。

▼class属性、id属性

ページ内で同じタグを複数使いたい、デザインを変えたいという場合はたくさんあります。

そんな時はclass属性とid属性が使えます。

class属性・id属性を用いると要素に名前をつけることができるので、そのつけた名前に対してCSSを適用することができます。

class属性・id属性ともにタグの中に記述します。
どちらもすべてのタグに指定することができます。

▼class属性

タグにclass属性を指定する場合は、下記のように記述します。

```
<p class="class-name"></p>
```

cssで指定する際は、「.」の後にclass名を記述します。

下記のように記述します。

```
.class-name {
  color: red;
}
```

▼id属性

タグにid属性を指定する場合は、下記のように記述します。

```
<p id="id-name"></p>
```

cssで指定する際は、「#」の後にid名を記述します。

下記のように記述します。

```
#id-name {  
  color: red;  
}
```

▼ class属性とid属性の違い

class属性とid属性には主に下記のような違いがあります。

- 1つのHTMLドキュメントで使える回数
- 1つのタグに付けられる個数
- CSSの優先順位
- id属性はページ内リンクのリンク先に指定できる

▼ 1つのHTMLドキュメントで使える回数

class名は、1つのHTMLドキュメント内で何度でも同じclass名を使用できます。
一方、id名は1つのHTMLドキュメント内で1度しか使用できません。

例えば下記のように同じclass名を複数回使うことはできますが、

```
<body>  
  ○<p class="class-name"></p>  
  ○<p class="class-name"></p>  
</body>
```

下記のように同じid名を複数回使うことはできません。

```
<body>  
  ×<p id="id-name"></p>  
  ×<p id="id-name"></p>  
</body>
```

下記のようにタグが違ってても、同じid名は使えません。

```
<body>  
  ×<p id="id-name"></p>  
  ×<div id="id-name"></div>  
</body>
```

もちろん下記のように、id名が違えば使用可能です。

```
<body>  
  ○<p id="id-name1"></p>  
  ○<p id="id-name2"></p>  
</body>
```

▼1つのタグに付けられる個数

class名は、半角スペースで区切ることで1つのタグに対して、複数のclass名をつけることができます。

下記のように記述すれば、<p>タグに対して「class-name1」と「class-name2」の2つのclass名をつけることができます。

```
<body>
  ○<p class="class-name1 class-name2"></p>
</body>
```

この後も半角スペースで区切れば、3個以上のclass名をつけることもできます。

一方、id名は1つしかつけられません。

下記のようなidの付け方はできません。

```
<body>
  ×<p id="id-name1 id-name2"></p>
</body>
```

1つのタグに対して、id名とclass名を両方つけることはできます。

その場合もclass名は複数個つけることが可能です。

```
<body>
  ○<p id="id-name" class="class-name1 class-name2"></p>
</body>
```

▼CSSの優先順位に注意

CSSの指定方法がいくつかありますが、同じ場所に異なるデザインを指定した場合はどうなるのでしょうか。CSSには、何を優先的に反映させるか優先順位があり、その優先順位に乗っ取りデザインが反映されます。

例えば、class名を指定したCSSと、id名を指定したCSSでは、id名を指定したCSSの方がスタイルの優先順位が高くなります。

CSSが反映されない！と思ったら、他の場所でCSSを指定されていないか確認しましょう。

▼id属性はページ内リンクのリンク先に指定できる

1つのid名は、HTMLドキュメント内で一度しか使用できないため、id属性をページ内リンクのリンク先に指定することができます。

例えば下記のように記述すると、「id-name」というidのついた要素へ、リンクを作成することができます。

```
<body>
```

```
<a href="#id-name">リンク</a>
<div id="id-name">id-name</div>
</body>
```

▼ class名とid名の命名ルール

class名とid名は自分で決めることができますが、いくつかルールがあります。このルールを守って命名しないとCSSが反映されないなので、注意が必要です。

- 半角英数字、「-」（ハイフン）、「_」（アンダースコア）のみを使用する
- 必ず英字で始める
- 全角スペース、半角スペースを入れない

class名で半角スペースを入れた場合は、複数のclass名になります。

▼ HTMLとCSSの調査方法

HTMLとCSSには、たくさんのタグやプロパティがあります。すべて覚えることは不可能なので、必要なことはその都度調べましょう。

その際、タグの使い方などが載っているリファレンスサイトなども活用すると良いでしょう。

下記のようなサイトがあります。

<https://developer.mozilla.org/ja/docs/Web/HTML/Element>

<http://www.htmq.com/>

ブラウザによって対応していないプロパティなどもあるので、主要なブラウザで使えるかどうかもしらべで確認します。