

自作 HP

「Mr.caffeine's Page」 について

目次

1. はじめに
2. 概要
 - 2.1 HTML とは
 - 2.2 HTML の構成
 - 2.3 CSS とは
3. ウェブページの解説とソースコード
 - 3.1 トップページ
 - 3.2 サブページ
4. おわりに
5. 参考文献

1. はじめに

本稿の目的は、ウェブサページ制作を報告することである。HTML や CSS によるウェブサイトの作り方や、仕組みを説明する。

2 節では概要を、3 節では、制作したページの説明をソースコードやキャプチャー画像とともに行う。

2019 年現在、世界各国にある企業の多くが自社ホームページを持っている。また、営利企業だけでなく、政府や地方公共団体といった組織のホームページも、日本ではほぼ 100% 存在する。そのようなホームページを使い、宣伝活動やお知らせの掲示、申し込みや各種申請・支払いができるようになってきている。つまり、インターネットを通じた情報発信が今とても重要になっているのである。

そうした現状の中、ウェブサイトを作成する言語である HTML や CSS を学習することで、世の中にあるウェブページがどのように作られているのかを知ることができるのである。ウェブページの裏側、つまりソースコードを理解することができるのであれば、適切な情報を正確に、かつ迅速に受け取ることができる能力を身につけることができる。また、作り方をすることで、自らがより高度な情報発信をすることができる。現在では Twitter や Facebook、ブログをはじめとした情報発信ツールを提供するサービスが多くあるが、いずれも文字数に制限があることや、発信できるメディアに制約があることが多い。自分でウェブサイトを作ることができれば、そのような制約なしに自由に多くの情報を発信することができるのである。Twitter や Facebook といったソーシャルメディアと自作ホームページを組み合わせることで相乗効果が生まれ、よりよい情報伝達経路が完成するのである。自分が作ったホームページでは、トップページへの Twitter のツイートを埋め込み、SNS 共有ボタンの実装、といった形でソーシャルメディアを活用した。

インターネット上には、詐欺などを目的とした悪質なウェブサイトやウェブ広告が数多く存在する。ウェブページの裏側である HTML や CSS のソースコードを閲覧し、理解することはできれば、危険なサイトにリンクしていないか、危険なスクリプトが埋め込まれていないか、などを知ることができる。結

果として、悪質なウェブサイトから自分の身を守ることができる。

以上のことが HTML・CSS を学ぶきっかけである。

2.概要

2.1 HTML とは

HTML とは HyperText Markup Language の略であり、ウェブページを作成するために開発された言語である。ハイパーテキスト(HyperText)とは、ハイパーリンクを埋め込むことができる高機能なテキストである。ハイパーリンクは下線のついたテキストなどをクリックすると別のページへ移動するリンクのことである。ハイパーテキストでは、ウェブページから別のウェブページにリンクを張ったり、ウェブページの中に画像や動画、音声といったデータを埋め込んだりすることが可能である。HTML とはこのようなハイパーテキストを、マークアップ（目印をつける）ことができる Language(言語)、という意味である。

(参照：HTML クイックリファレンス

<http://www.htmq.com/htmlkihon/001.shtml>)

現在存在する最新バージョンは HTML 5 であり、5 世代目にあたる。以前では JavaScript などといった他のプログラミング言語を使用しないと実現できなかった実装方法が、実現できるようになったものもある。詳しくは後で記述する。

2.2 HTML の構成

ここからは、どのようにハイパーテキストをマークアップするのかを説明する。

HTML の基本的な構成は以下のようにになっている。

```
<html>  
  <head>  
  </head>
```

```
<body>
</body>
</html>
```

<html>というタグは HTML 文書の開始を表しており、</html>は文書の終了を表している。また、それぞれを要素という。<head>要素には、文書のタイトルや概要、CSS への参照情報が記述する。<body>要素には、文章のメインの内容、つまり本文を記述する。文章だけでなく、画像や埋め込みスクリプト、他のウェブページへのリンクなどもここに記述する。

文書内の項目を、自分自身がわかりやすくするために要素ごとに分けて記述する。また、ウェブページを表示するためのブラウザに構成を理解させるためにも分けて記述することが必要である。

HTML 文書の一番はじめに<!DOCTYPE html>を記述する。これはブラウザに記述されている HTML のバージョンを判別させるためのものである。

以下は自分のホームページにおける<head>要素内のソースコードである。

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="author" content="Mr.caffeine">
<meta name="description" content="ホームページ">
<link rel="stylesheet" type="text/css" href="index.css">
<link rel="shortcut icon" href="ico/favicon.ico">
<title>トップ | Mr.カフェインのページ</title>
```

一行目は、文字化けを防ぐために、文字コードを指定するという記述である。HTML 5 以降の場合<meta charset="UTF-8"> という形で短く記述することもできるが、今回は、アップデートされていない古いバージョンのままのウェブブラウザへの互換性を重視して、前記のような書き方を採用した。2行目は文書の著者を表し、3行目はページの概要を表している。href は hypertext reference の略であり、読み込ませたいリンク先を参照する際に使用する記述である。4行目により、CSS ファイルを HTML 文書に読み込ませる。ブラウザ

のタブに画像やタイトルを表示すると、内容が一目でわかりやすくなる。そこで、5・6行目のようなタグを挿入する。<title>~</title>はタブなどに表示されるタイトルであり、見出しではない点に注意したい。見出しをつけるタグは<h1>や<h2>などを使用する。

2.3 CSS とは

CSS とは Cascading Style Sheets (カスケーディング・スタイル・シート) の略であり、ウェブページのスタイルを指定するための言語である。CSS を記述することで HTML 文書にデザインをつけたり、レイアウトを変更したりすることが可能である。

CSS にもバージョンが存在しており、現在の最新バージョンは CSS 3 である。CSS 3 ではデザインに関する便利な仕様がいくつか追加された。今回作成したウェブページの中にも CSS 3 で新しく追加された要素を使用している。たとえば、文字や区切られた範囲などに影をつけるためには、以前であれば画像を駆使するしかなかったところを、text-shadow プロパティや box-shadow プロパティを使用することで簡単に実現可能になった。

3.ウェブページの解説とソースコード

3.1 トップページ

以下の画像は、2019年7月17日現在のトップページをキャプチャーしたものである。



(筆者作成)

はじめに、「ようこそ～」の部分を書いた。HTMLコードは以下の通りである。

<h2>ようこそ
Mr.カフェインのページへ！
</h2>

先程も少し触れたが、<h2>~</h2>は中見出しを表すタグである。<h1>はもっと大きく、逆に、<h3>は小さく表示される。本の構成で例えるなら、<h1>は章、<h2>は節、<h3>は項を表すようなものである。

途中の
強制改行を表す。HTML や CSS は基本的に改行やスペースはすべて無視して解釈される。そのため、意図的に改行をしたい場合は
を挿入する。

CSS コードは以下の通りである。

```
h2 {  
  color: rgb(76, 183, 199);  
  font-size: 185%;  
  text-align: center;  
}
```

<h2>要素を装飾するための記述である。color: rgb(76, 183, 199);は水色に近い色で表示するための記述である。カッコ内の数字は色の三原色それぞれの強さを表している。font-size プロパティはフォントの大きさを指定するためのプロパティである。今回は大きめに表示させたいので 185%を指定した。text-align: center;でテキストを中央に寄せることができる。

次に、ウェブサイトの紹介文（このサイトでは～）を書いた。

<p id="shokaibun">このサイトでは主に、研究報告や論文の公開をします。
ぜひご覧になってみてください！</p>

<p>タグは Paragraph の略であり、その名の通り、一つの段落であることを示すために使われる。id=""のように id 名をつけることにより、固有の部分を指定して CSS を適用することができる。


```
#shokaibun {
  text-align: center;
}
```

この場合、紹介文のみが中央寄せで表示され、他の部分には影響しない。
次に、自己紹介欄と更新履歴欄を作成した。

```
<div id="intro">
  <h3><a href="4jikoshokai.html">自己紹介</a></h3>
  <ul>
    <li>名前：Mr.カフェイン</li>
    <li>職業：学生</li>
  </ul>
  <p><a class="mottomiru" href="4jikoshokai.html"> もっと見る
</a></p>
  <br>
</div>
<div id="new">
  <h3><a href="3koushinrireki.html">更新履歴</a></h3>
  <ul>
    <li>2019-07-16 ナビゲーションバーデザインの変更などの大
幅な更新</li>
    <li>2019-06-07 コンテンツページを公開</li>
    <li>2019-05-25 掲示板を公開 </li>
  </ul>
  <p><a class="mottomiru" href="3koushinrireki.html"> もっと
見る</a></p>
</div><br>
```

タグは Unordered List の略であり、順序のないリストを表示する際に使用する。タグの中にタグ(List Item)を入れることでリストを作成していく。自己紹介、更新履歴、は黒文字で表示されているが、それぞれの専用ページにリンクしている。CSS に以下の行を追加することで、下線を消し、黒色のままで表示させることができる。

```
h3 a {  
    text-decoration: none;  
    color: black;  
}
```

h3 a は子孫セレクタといい、<h3>要素の中の<a>要素に対してデザインを適用するためのものである。

工夫した点は「もっと見る」のボタンを作ったことである。

```
.mottomiru {  
    font-size: 70%;  
    border: solid rgba(66, 68, 69, 0.73) 2px;  
    background-color: rgba(90, 90, 92, 0.39);  
    display: inline-block;  
    text-decoration: none;  
    font-weight: bold;  
    color: black;  
    margin-left: 15%;  
}
```

適切な大きさや色になるように、何度も試行錯誤を繰り返した。その結果、上記のようなものが最適であると判断した。font-weight: bold;で文字を太字にしつつ、font-size: 70%;で小さめにすることで、見つけやすいが目立ちすぎないようにしたのである。display: inline-block;はボタンがあるかのように見せる

ために記述した。block として機能させつつ、inline のように扱うためのものである。インラインとは文章内という意味で、ブロックはかたまりのようなものである。つまり、block にすることでボタンのようなものを作りつつ、文章内の要素として容易に扱うことができる。

次にヘッダー部分（画面上部）について解説する。

```
<div id="logo">
  <a href="index.html"></a>
</div>
```

ロゴマークは画像で作成した。文字が浮き出て背景が見えるようにするために、JPG を PNG に変換し、透過画像にした。また、画像自体をリンクとして機能させるために href="index.html" を記述した。どこのページを見ている、ロゴマークをクリックまたはタップすれば、トップページ(index.html)に戻れるように実装したのである。このような実装方法は多くの個人や組織のウェブサイト採用されており、閲覧者が直感的にトップページに戻れるように意識した。

```
#logo {
  margin-left: 15px;
  margin-top: 8px;
}
```

また、ロゴマークが絶妙な位置に表示されるように上記のように CSS で指定した。

数多くの人に自分のウェブサイトを閲覧してもらうために、SNS を活用するという方法がある。今回、ツイッターを中心に SNS を活用した。そこで、ひとつめとして、SNS 共有ボタンをつけた。SNS アイコンをクリックすることで、各サイトにタイトルと URL が簡単に共有できるようになっている。

```
<div class="ninja_onebutton">
  <script type="text/javascript">
    //
      (function(d) {
        if
(window.NINJA_CO_JP_ONETAG_BUTTON_6b16c5656f2f5b362d4ee6763abca681) == 'undefined') {
          document.write("&lt;sc" + "ript type='text¥/javascript'
src='¥/¥/omt.shinobi.jp¥/b¥/6b16c5656f2f5b362d4ee6763abca681'&gt;&lt;¥/sc" +
"ript&gt;");
        } else {
window.NINJA_CO_JP_ONETAG_BUTTON_6b16c5656f2f5b362d4ee6763abca681.ONETAGButton_Load();
        }
      })(document);
    //]]&gt;
  &lt;/script&gt;&lt;span class="ninja_onebutton_hidden"
style="display:none;"&gt;&lt;/span&gt;&lt;span style="display:none;"
class="ninja_onebutton_hidden"&gt;&lt;/span&gt;
&lt;/div&gt;</pre></div><div data-bbox="138 820 847 891" data-label="Text"><p>Twitter 社の公式から共有アイコンを表示させるためのソースコードが公開されている。 <a href="https://publish.twitter.com/#">https://publish.twitter.com/#</a> にアクセスし、html コードをコピー &amp; ペーストすることで誰でも利用できる。</p></div><div data-bbox="485 898 511 915" data-label="Page-Footer"><p>11</p></div>
```

しかし、Twitter 以外の SNS への共有ボタンをそれぞれ容易しなくてはならないので、時間と手間がかかる。そこで、今使用中のウェブサーバーを提供する忍者ツールズを活用することにした。忍者おまとめボタンという名のサービスを利用することで各種 SNS 等の共有ボタンを一括で実装・管理することができるのである。忍者ツールズにログインし、忍者おまとめボタンを選択。その後、使いたい SNS やボタンのサイズ、デザインを選択肢すると、html コードが発行される。まとめてボタンを表示するためには JavaScript が必要であるが、忍者ツールズの管理下に置かれている JavaScript ファイルに直リンクするため、知識がなくても実装できるようになっているのである。

このような無料で使える外部スクリプトがインターネット上にはたくさん公開されているため、今回はそれらをフルに活用した。

次にサイト内検索バーを付けた。

```
<form method="get" action="http://www.google.co.jp/search"
target="_blank">
    <input type="text" name="q" size="31" maxlength="255"
value="">
    <input type="submit" name="btng" value="検索">
    <input type="hidden" name="hl" value="ja">
    <input type="hidden" name="sitesearch"
value="monster.nengu.jp">
</form>
```

<form>タグを使い入力フォームを作成、グーグル検索で自分のサイトのみの検索ができるように name="sitesearch" value="monster.nengu.jp" を記述した。

<input type="submit" name="btng" value="検索">は検索ボタンを配置するためのコードである。工夫した点は target="_blank" の部分である。検索をする際に新しいタブで開くようにしたほうが自然であると考えたため、このようにしたである。

次に、ロゴマークの真横に先程の SNS 共有ボタンと検索フォームを表示させるために、flex レイアウトを使用した。

```
#header {  
  display: flex;  
}
```

```
#logo {  
  width: 50%;  
}
```

```
#right1 {  
  flex: 1;  
}
```

通常、ブロック要素は縦並びで表示される。しかし、float プロパティや flex レイアウトなどを使えば横並びに表示させることが可能となるのである。今回はより直感的に配置ができる flex レイアウトを採用した。ただし、古いバージョンのブラウザでは flex レイアウトを解釈できないこともあるので、表示の互換性を確保したい場合には float プロパティを使用すべきである。

トップページには外部のスクリプトを使用したツールを以下のように4つほど利用した。

- ①SNS 共有ボタン
- ②Twitter 埋め込みツイート
- ③アクセスカウンター
- ④閲覧者カウンター

まず、Twitter 埋め込みツイートの表示方法を説明する。

ウェブサイトの更新情報などを発信するために、新しい Twitter アカウントを取得した。

<https://publish.twitter.com/#> にアクセスし、Embedded Timeline を選択し、自分の Twitter アカウントの URL を入力する。すると、以下のような HTML コードが取得できる。

```
<div class="twitter-timeline1">
  <a class="twitter-timeline" height="750px"
href="https://twitter.com/monster_nengu?ref_src=twsrc%5Etfw">Tweets by
monster_nengu</a>
  <script async src="https://platform.twitter.com/widgets.js"
charset="utf-8"></script>
</div>
```

このソースコードを任意の場所に貼り付けるだけで埋め込みツイートを表示することができる。

パソコンなどの横にスクリーンが長いデバイスで見た際に、埋め込みツイート欄と自己紹介・更新履歴欄が横並びで表示されると見やすい。そこで以下のよう記述した。

```
@media screen and (min-width:1280px) {
  #container {
    display: flex;
  }
  #main1 {
    flex: 1;
  }
  .twitter-timeline1 {
    width: 40%;
```

```
}  
}
```

@media screen and (min-width:1280px)は表示の幅が 1280px 以上のときのみ
{内のスタイルを適用するという意味である。}内に flex レイアウトを記述
することで 1280px 以上のデバイスで閲覧したときのみ、横並びで表示され
ようになる。



(筆者作成)

アクセスカウンターは <http://www.rays-counter.com/>を使用した。採用理由はデザインが豊富で、好みのデザインが見つかったからである。なお、忍者ツールズもアクセスカウンターを提供している。

```
<table border="0" cellspacing="0" cellpadding="0" id="countor1">
  <tr>
    <td align="center"><a href="http://www.rays-counter.com/"></a></td>
  </tr>
  <tr>
    <td align="center"></td>
  </tr>
</table>
```

<table>タグを使用し、それぞれのコマに画像を外部から直リンクすることで表示されている。

次に、閲覧者カウンターについて説明する。先程のアクセスカウンターとは違い、ある地点で何人の同時閲覧者がいるかを表示したものである。

```
<div id="visitor1">
  <script type="text/javascript"
src="http://monsternengu.vis1.shinobi.jp/js/" charset="utf-8"></script>
```

</div>

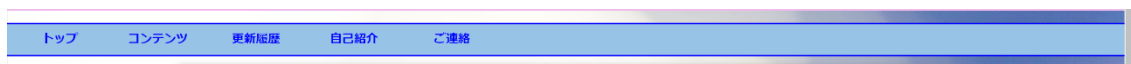
こちらは忍者ツールズが提供する、忍者ビジターを利用した。ログイン後に表示形式を選ぶと html コードが発行されるので、そのコードをコピーし、任意の場所に貼り付けるだけで簡単に導入することが可能である。

これらのように外部のツールを使用することにより、簡単に便利な機能を追加することができるのである。

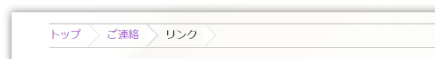
最後に、ナビゲーションバーについて解説する。

ナビゲーションバーは以下の2つの部分から構成されている。

①メニューバー



②パンくずリスト



どちらも素早く別のページに切り替えられるようにするためのものである。それでは、それぞれを分けて解説する。

①メニューバー

メニューバーは以下のような HTML で記述されている。

```
<nav>
```

```
  <ul class="ddmenu">
```

```
    <li class="top"><a href="index.html">トップ</a></li>
```

```
    <li class="top"><a href="2contents.html">コンテンツ</a>
```

```
  </ul>
```

```
    <li><a href="2-1ronbun.html">論文・報告書</a></li>
```

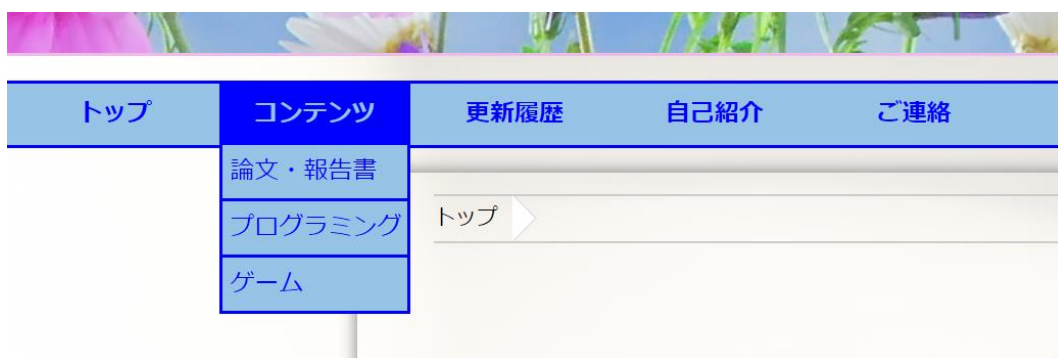
```
    <li><a href="2-2programing.html">プログラミング</a></li>
```

```

    <li><a href="2-3game.html">ゲーム</a></li>
  </ul>
</li>
<li class="top"><a href="3koushinrireki.html">更新履歴</a></li>
<li class="top"><a href="4jikoshokai.html">自己紹介</a></li>
<li class="top"><a href="5gorenraku.html">ご連絡</a>
  <ul>
    <li><a href="5-1link.html">リンク</a></li>
    <li><a href="5-2keijiban.html">掲示板</a></li>
    <li><a href="5-3otoiawase.html">お問い合わせ</a></li>
  </ul>
</li>
</ul>
</nav>

```

つまり、でリストをつくっただけの簡単な構造である。しかし、初期状態ではの中にあるが表示されていないことがわかる。そこで、“コンテンツ”にマウスカーソルを置くと以下の画像のようなサブメニューが表示される。



(筆者作成)

これを実現するためには JavaScript を使用するのが一般的だが、CSS 3 ならば JavaScript なしで実装可能である。以下は CSS コードである。

```
ul.ddmenu {
  background-color: rgb(151, 195, 231);
  margin-left: -9px;
  margin-right: -9px;
  padding-left: 9px;
  padding-right: 9px;
  border-top: solid blue 2px;
  border-bottom: solid blue 2px;
}
```

この部分では、背景色やボーダーラインを設定している。border-top: solid blue 2px;、border-bottom: solid blue 2px;は上下に2ピクセルの青線を引くという意味である。マージンとパディングをそれぞれプラスとマイナスの値に設定することで、横幅いっぱいに表示されるようになる。

```
ul.ddmenu li {
  width: 125px;
  display: inline-block;
  list-style-type: none;
  position: relative;
}
```

上記はメインメニューの装飾をしている。

display: inline-block;によりリストを横並びにしている。list-style-type: none;はリストの先頭の・を非表示にし、position: relative;はサブメニューの表示の基準位置を決めるための記述である。

```
ul.ddmenu a {
  background-color: rgb(151, 195, 231);
  color: blue;
```

```
line-height: 40px;
text-align: center;
text-decoration: none;
font-weight: bold;
display: block;
transition: 0.1s;
}
```

上記は、メニューの中の<a>、つまりリンクに関する記述である。display: block;によって、ブロック化し、文字だけでなくその周りもリンクとして機能するようになる。Transitionを0.1秒にすることで、マウスが他の項目に移動してから0.1秒間だけ、選択していた項目の背景色を青のままに表示することになり、より自然な体験ができるようになっている。

```
ul.ddmenu a:hover {
  background-color: blue;
  color: rgb(151, 195, 231);
}
```

上記は、疑似クラスを利用しており、マウスカーソルを置いたときに背景色を文字色が反転するようにデザインした。

```
ul.ddmenu ul {
  display: none;
}
```

サブメニューは初期状態で表示されないように、display: none;を指定する。

```
ul.ddmenu ul {
  display: none;
}
```

```
margin: 0px;
padding: 0px;
position: absolute;
}
```

```
ul.ddmenu li:hover ul {
  display: block;
}
```

上記も擬似クラスを利用しており、マウスカーソルが乗ったときだけサブメニューが開くようになっている。

```
ul.ddmenu ul li {
  width: 121px;
  border-bottom: 2px solid blue;
  border-left: 2px solid blue;
  border-right: 2px solid blue;
  z-index: 2;
}
```

メインメニューの一つの項目の幅は 125px である。サブメニューの幅を 121px にすることで、左右の実線と合わせて 125px になり、サブメニューがメインメニューのぴったり真下で合うようになる。z-index は重なり順を指定するプロパティであり、数字が大きいほど全面に表示されるようになる。z-index を指定した理由は、次に説明するパンくずリスト内のリンクを被った際に、全面的に表示されるようにしたかったからである。

```
ul.ddmenu ul li a {
  line-height: 35px;
  text-align: left;
```

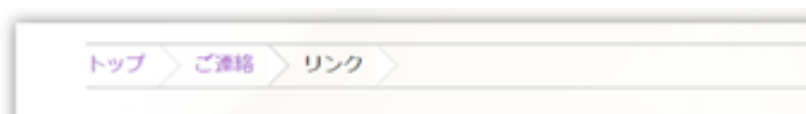
```
padding-left: 5px;
font-weight: normal;
}
```

メインメニューでは太字にしたが、サブメニューでは通常の太さの文字にするために、font-weight: normal;の一行を追加した。

```
ul.ddmenu ul li a:hover {
    background-color: blue;
    color: rgb(151, 195, 231);
}
```

②パンくずリスト

パンくずリストとは画像のように表示のことである。



童話『ヘンゼルとグレーテル』で、主人公が森で迷子にならないように通り道にパンくずを置いていった、というエピソードに由来する。(引用:IT用語辞典) 閲覧者が、ウェブサイトのどこの階層を閲覧しているのかわかりやすくするために、実装した。こちらも多くウェブサイト採用されていて、見慣れたデザインである。

```
<div class="breadcrumbs">
  <nav>
    <ul>
      <li><a href="index.html">トップ</a></li>
      <li class="active"><a href="5gorenraku.html">ご連絡</a></li>
    </ul>
  </nav>
</div>
```

こちらにもとを使った単純なリストで構成されている。それぞれの文字がリンクになっており、クリックするとすぐに上位の階層に戻れるようになっている。

```
.breadcrumbs {  
  border: solid 1px #ccc;  
  border-width: 1px 0;  
}
```

```
.breadcrumbs ul,  
.breadcrumbs li {  
  margin: 0;  
  padding: 0;  
}
```

```
.breadcrumbs a {  
  z-index: 1;  
  color: #9b4dca;  
  text-decoration: none;  
}
```

```
.breadcrumbs a:hover {  
  color: #606c76;  
}
```

```
.breadcrumbs ul {  
  display: flex;  
  width: 100%;  
  margin: auto;
```



```
}
```

```
.breadcrumbs li {  
  list-style: none;  
  font-size: 14px;  
  height: 30px;  
  line-height: 30px;  
  margin-right: 35px;  
  position: relative;  
}
```

```
.breadcrumbs li:before,  
.breadcrumbs li:after {  
  content: "";  
  display: block;  
  position: absolute;  
  top: 0;  
  border-top: 15px solid transparent;  
  border-right: 15px solid transparent;  
  border-bottom: 15px solid transparent;  
}
```

```
.breadcrumbs li:before {  
  right: -40px;  
  border-left: 15px solid #ccc;  
}
```

```
.breadcrumbs li:after {  
  right: -39px;  
  border-left: 15px solid #fff;
```

```
}
```

```
.active a {  
  color: #000;  
}
```

CSS コードは割と長めになっているが、いずれもデザインに関するものであり、今までに説明した内容によるものである。

3.2 サブページ

トップページ以外はすべてトップページをコピーして作られており、内容のみ変更して作成した。

4. おわりに

本稿ではウェブページを作成することにより、ひと通りの HTML・CSS の構成、実装方法を習得することができた。これからも、さらに見やすく、わかりやすいサイトを目指して更新をしていきたい。また、JavaScript や PHP といった他のウェブ言語についても勉強したい。

5. 参考文献

© HTMQ 「HTML クイックリファレンス」 (2019-07-15 最終アクセス)
<http://www.htmq.com/htmlkihon/001.shtml>

千貫りこ 「これからはじめる HTML&CSS の本」 技術評論社