

Webサイトのトップページを作成しよう

【はじめに】

本日はプログラミングインターン初級編に参加していただき、ありがとうございます！

初級編は「プログラミングを学んだことがない人」「授業で学んだことを活かしてアウトプットしてみたい人」を対象にしたカリキュラムとなっております。

初めての人でも、このマニュアルを見ながら進めていただくと完成するようになっているので、ぜひ挑戦してみてください。

今回作成していただくのは**Webサイトのトップページ**です！

課題を進めているとわからないことが出てくると思います。

その場合は気軽に質問してくださいね♪

それではまず初めに、今回受講していただく「初級編のゴール」と「学習の目標」を確認していきましょう！

○初級編のゴール

- Webページ制作の楽しさを知ろう
- Webページ制作のイメージを掴もう

プログラミングと言えば、文法をたくさん覚えることや、ひたすらパソコンと向き合って取り組むイメージがあるかもしれません。

しかし、それは実際に何かを形にするために行なっていることです。

難しいこともあるかもしれませんが、Webページなど、目に見える形にすることでプログラミングの楽しさを知ってもらいたいです♪

また、どのようにWebページが出来上がっていくのかというイメージも掴んでいただければと思います！

○学習の目標

- HTMLとCSSのポイントを掴み、実際にWebページを作れるようになる
- Webページの構成を知り、コードで再現してみよう

HTMLとCSSの要点を抑えることで、今後どんなWebページを作る際にも活かすことができます。また、Webページがどのような構成なのかを知ることがWebプログラマとしての大事な一歩になるでしょう。

【プログラミングをする準備をしよう】

事前にコマンドプロンプト（あるいはターミナル）を利用してファイルやディレクトリを作成していただいたと思います。

今回はそれらを使ってプログラミングを行なっていきます。

実際にプログラミングを始める前に、プログラミングの環境を整えていきましょう！

①テキストエディタを開こう

テキストエディタとは、プログラミングなどのコード（文字の羅列）を書く際に使用するアプリケーションのことです。

プログラミングを記述する時は必ずと言っていいほど、テキストエディターを使います。

プログラミングは、実はパソコンにあるメモ帳でも書けるんです。

しかしテキストエディタを用いることで、どの部分がエラーになっているのかがわかったり、文字の色が要素ごとに変わったりするので、とても快適に使うことができます。

テキストエディタは様々な種類がありますが、今回は**Visual Studio Code**を使用します。

エディタを使ってプログラミングをする方法は、以下の通りです。

- ①Finder(Windowsの場合はエクスプローラー)で編集したいファイルを右クリックする。
- ②「アプリケーションを開く」など、ファイルをどのアプリケーションで開くか選択できる項目があるので、「Visual Studio Code」を選択する。

これでプログラムを記述できるようになります！

②ファイルをブラウザで表示しよう

次に、プログラミングしたものをパソコン上で表示できるようにしましょう。

パソコン上で表示するには、Webページを見に行くためのアプリケーション(**Webブラウザ**)を使用します。

Webブラウザの種類も沢山ありますが、今回は**Google Chrome**を使用します。

作ったファイルをWebブラウザで表示する方法は、以下の通りです。

- ①Google Chromeで新規ページを開く。
- ②Finder(Windowsの場合はエクスプローラー)で表示させたいファイルを選択し、Google Chromeの新規ページの方にドラッグ&ドロップする。

ファイルがWebブラウザで表示されたら成功です！

これでプログラミングをする環境が整いました！

これからはHTMLについて学んでいきましょう。

【HTMLについて】

○HTMLとは

HTMLはWebページの土台を作る言語です。

様々なタグを用いて、文章中の要素を明確に示すことができます。

○HTMLの書き方の基本

プログラミングの大前提として、**タグや記号は全て半角で書く**というルールがあります。

全角で書いた場合は反映されません。

また、空白を入れたい場合に全角スペースを入れた場合もエラーになってしまうので、

空白も半角スペースを入れることに注意しましょう。

以下でHTMLを書く際に覚えていただきたい書き方の基本を記載します。

①タグではさむ

開始タグと**終了タグ**を使います。2つのタグの間にテキストを入れることで、そのタグが持つ役割を適用することができます。

```
<タグ>テキスト</タグ>
```

タグの種類によっては、終了タグがないものもあります。
(例)imgタグ、brタグなど

②タグの中にタグを入れていく(=入れ子構造)

htmlタグの中にheadタグとbodyタグを入れ、headタグの中にtitleタグを入れるといったように、タグの中にタグを入れていくことで構成を組み立てていきます。

htmlタグという大きな箱の中にheadタグやbodyタグという箱を入れていくイメージです。

```
<!DOCTYPE html>
<html>
<head>
  <title>キャリアワークカフェ</title>
  <meta charset="UTF-8">
</head>
<body>
  <h1>プログラミングインターン</h1>
  <p>HTMLとCSSを使ってみよう!</p>
</body>
</html>
```

○HTMLファイルの基本構造

HTMLの書き方を知ってもらったところで、次にHTMLの構造について学んでいきましょう。

順番にタグの説明をしていきますので、実際に書いていきましょう。

まず初めに、このファイルはHTML文書だという宣言をします。

1番上の行に以下のコードを記述しましょう。このタグは終了タグはありません。

```
<!DOCTYPE html>
```

宣言した後は<html>タグを記述し、「ここからここまでがHTMLである」ことを示します。

<!DOCTYPE html>の下に記載します。

```
<!DOCTYPE html>
<html>
</html>
```

次は<html>タグの中に、Webページの様々な情報や設定ができる<head>タグを記述します。

ここからは入れ子構造にして記述していきます。

入れ子構造については、「HTMLの書き方の基本」の「②タグの中にタグを入れていく(=入れ子構造)」をおさらいしてみましょう。

```
<!DOCTYPE html>
<html>

<head>
</head>

</html>
```

次に、<head>タグの下に<body>タグを記述します。

今後Webの画面上に表示させたいものはタグ内に書いていきます。

```
<!DOCTYPE html>
<html>

<head>
</head>

<body>
</body>

</html>
```

これでHTMLを書くための基本構造が完成しました！

次からは、Webページの構成に合わせてタグを追加していきます。

○Webページの構成

次に、Webページの構成について見ていきましょう。

1枚のページの中で、主に3つのエリアに分かれています。

以下に記載される3つのエリアは<body>タグの中に書いていきます。

①ヘッダー

ページの上部を指し、主に企業やサービスのロゴやナビゲーションメニュー、電話番号などを配置します。

ナビゲーションメニューは、ページ間を移動するためのリンクが複数掲載されている要素のことです。

ヘッダーにリンクを貼ることで、ユーザーはどのページにいても別のページに飛べるようになります。

```
<header></header>
```

②コンテンツ

各ページに伝えたい情報や内容を記述していきます。

`<header>`タグの下に書きましょう。

```
<main></main>
```

③フッター

ページの下部を指し、主にコンテンツの著作権などを示す **コピーライト** を記載します。

ヘッダーにナビゲーションメニューを配置しない場合はフッターに配置することがあります。

`<main>`タグの下に以下を記述しましょう。

```
<footer></footer>
```

これで、HTMLファイルの構造とWebページの大枠が完成しました！次はヘッダー、コンテンツ、フッターそれぞれに他の要素を入れていきましょう！

○基本的なタグの説明

これからは、①ヘッダー、②コンテンツ、③フッターのそれぞれに必要な要素を追加していきます。

①ヘッダー

画像を挿入しよう

``タグで画像を挿入することができます。

`<header>`内に以下のコードを書き込み、キャリアワークカフェのロゴを入れてみましょう。

```

```

②コンテンツ

ページに見出しをつけよう

`<h1>`～`<h6>`タグを使用し、文書内に見出しをつけることができます。

`<h1>`が一番重要度が高く、`<h6>`が一番重要度が低いです。

今回は`<main>`タグの中で`<h1>`タグを使用し、Webページのタイトルをつけてみましょう。

```
<h1>LET'S GO ABOARD!</h1>
```

文章を入力しよう

`<p>`タグを使用し、文章を入力します。

文章を入力する際は主にpタグを使用することが多くなります。

改行を行う場合は、行末に`
`をつけます。

エディタ上では改行できていても、`
`をつけていない場合は、ブラウザ上では改行されないため注意しましょう。

```
<p>
  これは、Career Worq Cafeのプログラミングインターンで作成するページです。<br>
  HTMLとCSSの基礎を勉強します。<br>
  このページを作成することで、Webページの見た目の作り方が学べます。<br>
  完成を目指して頑張りましょう！
</p>
```

ボタンを作成しよう

`<button>`タグを使用してボタンを設置します。

`onclick`属性でURLを設定することで、別ページにアクセスすることができます。

今回はキャリアワークカフェのWebページにアクセスするボタンを設置してみましょう。

```
<button type="button" onclick="location.href='https://careerworq-
cafe.com/'">CLICK!!</button>
```

③フッター

コピーライトを記述しよう

`<small>`タグを使って、コンテンツの著作権情報を表す「コピーライト」を記述しましょう。

`<small>`要素を使うことで、本文の内容に比べて重要ではない情報を小さく表示し、テキストの意味合いを弱めることができます。

```
<small>&copy;2019 Career Worq Cafe</small>
```

これで、HTMLの枠組みが完成しました！

次はCSSを使って装飾していきましょう！

【CSSについて】

○CSSとは

CSSとは、HTMLで組み立てた枠組みに、より具体的な設定ができる言語のことです。

(例) ボタンの色を変える、写真の角を削って丸くする etc.

○CSSの書き方の基本

CSSを記述する時は、**セレクトア**、**プロパティ**、**値**の3つを使います。

```
セレクトア{  
  プロパティ: 値  
}
```

セレクトアはスタイルを変更したい要素のことです。

装飾したいタグ(<p>,<h1>,など)を指定することで装飾を施すことができます。

プロパティはスタイルの種類、**値**はスタイルをどのように(どれぐらい)適用させるかを指します。

また、プロパティの右横に「:」(コロン)、値の右横に「;」(セミコロン)が必要です。

これらが無い場合はせっかくコードを書いても反映されません。

特にセミコロンを忘れることが多いので、忘れずに書きましょう！

例えば<h1>のテキストの色を青くしたい場合は以下のように記述します。

```
h1 {  
  color: blue;  
}
```

上のCSSについて、各用語と合わせて説明すると以下の通りです。

- **セレクトア**(スタイルを変更したい要素)・・・**h1**タグ(一番大きい見出しのこと)
- **プロパティ**(スタイルの種類)・・・**color**(文字の色を変えるプロパティ)
- **値**(スタイルをどのように適用させるか)・・・**blue**(青色を適用する)

○実際に装飾してみよう

まずは、CSSを適用するためにHTMLファイルの<head>内に<link>タグを設定します。

以下のように<link>タグを設定することで、htmlファイルとcssファイルが関連づけられます(リンクされます)。

関連づけることでhtmlファイルのタグに、cssファイルでスタイルの変更を適用することができます。

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

①ベースになるスタイルを設定しよう

これからはWebページの構成である「①ヘッダー」、「②コンテンツ」、「③フッター」に合わせて、「④ベースになるスタイル(①～③に共通する要素)」の4つに分けて装飾していきます。

まず初めに「①ベースになるスタイル」です。

文字の大きさや背景の色など、例外がない限り同じ設定をしたい場合は<body>タグに対して装飾を指定します。

今回は以下の要素に装飾を加えます。

- 文字の色
- 基本的な文字の大きさ
- 行間
- 背景の画像の配置方法

初めてプログラミングを体験する方は「へえ、こんな感じで変わるんだ」と、参考程度に見ていただけると幸いです！

プログラミングを勉強したことがある方は、これから出てくるプロパティはWebページを作る際によく出てくるものなので、

覚えておいても損はないです！

ぜひコードを書いてみてどのように変化するのかを見てみてください！

```
body{
  color:#ffffff;
  font-size:15px;
  line-height:2;

  background-image: url(../img/bg-index.jpg);
  background-position: center center;
  background-attachment: fixed;
  background-size: cover;
}
```

①ヘッダーのスタイルを変更しよう

次に、<header>要素に装飾を加えましょう。

ここでは以下の要素を変更します。

- ヘッダーの高さ
- ヘッダーの中央揃えにする

```
header{
  height: 200px;
  margin: 0 auto;
}
```

次に、ヘッダーのロゴの位置を変えましょう。

marginとpaddingは位置を変えるときによく使われます。

```
img{
  margin-left: 100px;
```



```
margin-top: 20px;
}
```

②コンテンツのスタイルを変更しよう

<main>要素全体に装飾を加えましょう。

ここでは以下の要素を変更します。

- コンテンツエリアの横幅
- コンテンツエリアの中央揃え
- テキストの中央揃え

```
main{
  width: 1020px;
  margin: 0 auto;
  text-align: center;
}
```

以下でそれぞれの要素に装飾を加えていきましょう。

大見出し

今回は大見出しに関して、以下の項目を変更します。

- フォントの種類を変更
- フォントの大きさ
- 行間
- 大見出しと<p>のテキストとの間隔

先ほど<body>全体にフォントの大きさのスタイルを変更しましたが、今回は大見出しとして文字を強調させたいので、

例外として文字の大きさを変更します。

```
h1{
  font-family: 'Rubik Mono One', sans-serif;
  font-size: 80px;
  line-height: 1;
  margin-bottom: 100px;
}
```

また、フォントの種類を変更する際は、HTMLファイルの内に貼る必要があります。

<head>タグの中に以下の<link>タグを貼ることで、Web上でGoogleが提供しているWebフォントと関連づけられ(リンクされ)、

フォントをよりオシャレに変更することができます。

Webフォントは「Google Fonts」(<https://fonts.google.com/>)で探すことができます。

ぜひお気に入りのフォントを見つけてみてください！

```
<link href="https://fonts.googleapis.com/css?
family=Rubik+Mono+One&display=swap" rel="stylesheet">
```

テキスト

テキストに関して、以下の項目を変更します。

- フォントの大きさ
- テキストの色
- テキストとボタンとの間隔

```
p{
  font-size:25px;
  color:#ffffff;
  margin-bottom: 100px;
}
```

ボタン

ボタンに関して、以下の項目を変更します。

- ボタンの色
- テキストの色
- テキストの大きさ
- ボタンの横幅
- テキストの中央揃え
- 行間
- ボタンの中央揃え
- ボタンの四隅を丸くする
- 枠線の太さ、色

```
button{
  background-color: #009cd3;
  color: #ffffff;
  font-size: 30px;
  width: 250px;
  text-align: center;
  line-height: 50px;
  margin: 0 auto;
  border-radius: 5px;
  border: 3px solid #009cd3;
}
```

また、ボタンにカーソルを合わせた時にスタイルを変更する設定ができます。
今回はカーソルを合わせた時に以下の項目が変わるように設定してみましょう。

- ボタンの色
- テキストの色

```
button:hover{
  background-color: #ffffff;
  color: #009cd3;
}
```

③フッターのスタイルを変更しよう

最後に、<header>要素全体に装飾を加えましょう。

ここでは以下の要素を変更します。

- フッターの横幅
- フッターの縦幅
- テキストの中央揃え

```
footer{
  width: 1020px;
  margin: 100px auto 0 auto;
  text-align: center;
}
```

以上でWebサイトのトップページは完成です！お疲れ様でした！

プログラミングを書いてみていかがでしたか？

「なかなかうまくできない」「完成したけどこの部分の意味がよくわからない」などあれば、気軽に質問してくださいね！

1つでも多くのことを吸収して、プログラミングを楽しんでいただければ幸いです☆
