

10 コーディングを開始する準備

1. フォルダを作成する

1-1. 必要なフォルダを作成する

「base」という名前のフォルダを作成し、そのフォルダ内に「css」「images」「js」フォルダを作成します。

2. HTML ファイルを作成

2-1. HTML の基本構造を記述する

エディターを開き、ファイルを新規作成し、html の必須タグを記入します

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Sample site</title>
</head>
<body></body>
</html>
```

2-2. レスポンシブ対応のための記述

viewport を記述します。viewport（ビューポート）とは、「表示領域」を設定するために必要な記述です。

例えば、デスクトップ環境（PC など）のブラウザなどでは、ブラウザウィンドウの表示領域が viewport になります。PC 向けサイトをスマートフォンで表示すると、文字などが小さすぎて閲覧しづらくなります。そのような問題を解決し、スマホなどの小さな端末でも正しく表示するための設定です。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<title></title>
</head>
<body></body>
</html>
```

2-3. HTML ファイルとして保存

先ほど作成した「base」フォルダ内に「index.html」という名前で保存します。

2-4. meta タグ

基本的な meta タグを記述します。

description タグ

ページの説明・概要を指定する記述です。

検索結果にタイトルと一緒に表示される文章でもあります。タイトル同様、SEO 対策において重要な要素のひとつです。キーワードなどをしっかり含め、ページの内容を示す説明にしておきましょう。必須で設定しておきましょう。

```
<meta name="description" content="ここにページの概要が入ります。">
```

OGP

OGP とは、Open Graph Protocol の略で、Facebook や X (旧 Twitter) などの SNS でシェアした際に、Web ページのタイトルや概要、イメージ画像、URL 含めた詳細情報を正しく伝えるための HTML 要素のことです。

OGP の使用を宣言するために、まず head 要素に prefix 属性 (og: https://ogp.me/ns#) を指定します。

OGP 確認サイト参考: <https://rakko.tools/tools/9/>

```
<head prefix="og: https://ogp.me/ns#">
<meta property="og:url" content="ページの URL 絶対パス ">
<meta property="og:type" content="ページの種類 TOP ページであれば「website」、TOP 以外のページでは「article」を入力 ">
<meta property="og:title" content="ページのタイトル ">
<meta property="og:description" content="ページの説明文 ">
<meta property="og:site_name" content="サイト名 ">
<meta property="og:image" content="サムネイル画像の URL (画像 1200×630px) ">
```

favicon (ファビコン)

ファビコンを設定するための記述です。

ファビコンとは、ブラウザのタブやブックマークバーなどに表示されるアイコンのことです。(WP の場合 画像 512×512px)

ファビコン作成参考: <https://ao-system.net/favicon/>

```
<link rel="icon" href="favicon.ico">
```

apple-touch-icon

スマートフォンでホーム画面に追加したときや、ショートカットを作成した場合に表示されるアイコンを設定するための記述です。(画像 180×180px)

```
<link rel="apple-touch-icon" href="apple-touch-icon.png" sizes="180x180">
```

robots タグ

検索エンジンへのインデックスを避けるための記述です。

以下の記述をすることにより、検索結果として表示されなくなります。

```
<meta name="robots" content="noindex, nofollow">
```

3. CSS ファイルを作成

3-1. CSS の基本構造を記述する

文字コードの指定を記述します。

```
@charset "utf-8";
```

3-2. CSS ファイルとして保存

先ほど作成した「base」フォルダ内の「caa」フォルダ内に「style.css」という名前で保存します。

3-2. HTML ファイルに CSS ファイルを読み込む

index.html の head タグ内に <link href="css/style.css" rel="stylesheet"> を記述します。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content=" width=device-width,initial-scale=1">
<title></title>
<meta name="description" content=" ここにページの概要が入ります。 ">
<link href="css/style.css" rel="stylesheet">
</head>
<body></body>
</html>
```

4. リセット CSS

4-1. リセット CSS

Web ブラウザでは、HTML タグごとにブラウザ固有のスタイルが効いています。ブラウザ毎の差異をなくすために、リセット CSS を利用してデフォルトのスタイルをリセットしてコーディングを行うことが多いです。

参考： <https://github.com/nicolas-cusan/destyle.css/blob/master/destyle.css>

4-2. リセット CSS を作成

ここでは reset.css ファイルを新たに作成してリセット css を記述します。

先ほど作成した「base」フォルダ内の「caa」フォルダ内に「reset.css」という名前で保存します。

4-3. HTML ファイルに CSS ファイルを読み込む

index.html の head タグ内に <link href="css/reset.css" rel="stylesheet"> を記述します。

<link href="css/style.css" rel="stylesheet"> の前に記述します。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content=" width=device-width,initial-scale=1">
<title></title>
<meta name="description" content=" ここにページの概要が入ります。">
<link href="css/reset.css" rel="stylesheet">
<link href="css/style.css" rel="stylesheet">
</head>
<body></body>
</html>
```