

Webサイト制作はじめての一步

つくって学ぶ！ はじめてのWebサイトづくり 2DAYS

今日やること

1. Webページの仕組みを知る (HTML, CSS, JavaScriptって何?)
2. 部品を組み合わせてLP(ランディングページ)を作ってみる
3. 作ったページを自分流にカスタマイズする

ゴール： プログラミングの楽しさを体験し、「自分でも作れる！」を実感する！

Webページってどうやってできてるの？

Webページは、主に3つの言語の組み合わせでできています。

- HTML → 骨組み担当
- CSS → 見た目担当
- JavaScript → 動き担当

HTMLって何?

> ページの「骨組み」を作る言語

- HyperText Markup Languageの略
- 見出し、段落、画像などの「構造」を指定する
- 「タイトルはこれです」「本文の次に画像を置きます」といった指示書

```
<h1>こんにちは！</h1>  
<p>これは私のウェブサイトです。</p>  

```

HTML: 実際のコードを分解してみよう

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8">
    <title>私のウェブサイト</title>
  </head>
  <body>
    <h1>こんにちは！</h1>
    <p>これは私のウェブサイトです。</p>
  </body>
</html>
```

こんにちは！

これは私のウェブサイトです。

HTMLの基本構造

```
<タグ 属性="値">内容</タグ>  
<タグ 属性="値" />  
<!-- コメント -->
```

タグで囲って、中身を書く。属性(プロパティ)で細かい指定ができます。

中身がいないタグ(画像など)は、閉じタグを省略できることがあります。その場合、閉じカッコを `/>` にします。

```

```

HTMLの基本構造

```
<!DOCTYPE html> <!-- HTML5で書かれていますよ、という宣言 -->
<html lang="ja"> <!-- ページ全体を囲んで、言語などを指定 -->
  <head>
    <!-- ページのプロパティを設定。デザインファイルと紐づけたり、タイトルを指定したり。 -->
  </head>
  <body>
    <!-- ページの中身を指定 -->
  </body>
</html>
```

htmlでページを作るときは、このような構造で書くことになっています。

`<!DOCTYPE html>` は特殊なタグです。それ以外は基本的なタグのルールに従っています。

HTML: よく使うタグ

```
<h1>大見出し</h1>  
<h2>中見出し</h2>  
<p>段落。<span>一部だけ</span>  
  見た目を変えられます。</p>  
<div>  
    
  <p>divで複数のタグを囲めます</p>  
</div>
```

大見出し

中見出し

段落。一部だけ 見た目を変えられます。



150 × 150

divで複数のタグを囲めます

HTML: よく使うタグ

```
<h1>大見出し</h1>  
<h2>中見出し</h2>  
<h3>小見出し</h3>  
<h4>見出し4</h4>  
<h5>見出し5</h5>  
<h6>見出し6</h6>
```

htmlでは、`h1` から `h6` までの6段階の見出しタグがあります。数字が小さいほど大きな見出しになります。

適切なタグを使っておくと、Googleなどが内容を理解して検索順位を上げてくれたり、画面読み上げソフトが正しく読み上げてくれたりします。

大見出し

中見出し

小見出し

見出し4

見出し5

見出し6

HTML: よく使うタグ

```
<style>span{color: red;}</style>
```

```
<p>段落。<span>一部だけ</span>
```

```
見た目を変えられます。</p>
```

```
<p>二段落目。段落の中で<br>
```

```
無理やり改行することもできます。</p>
```

```
<p><strong>太字</strong> <em>斜体</em>
```

```
<mark>蛍光ペン</mark></p>
```

段落。一部だけ 見た目を変えられます。

二段落目。段落の中で
無理やり改行することもできます。

太字 斜体 蛍光ペン

長めの文章を書くときは、`<p>` タグで囲みます。paragraphのpです。

段落の中で一部だけ見た目を変えるためのタグはいくつかあります。`` は汎用的なタグで、それ単体ではスタイルはつきません。具体的なスタイルはCSSで指定します。書き方は後のスライドで。

`
` タグは段落の中に改行を入りたいときに使います。

HTML: よく使うタグ

```
<main>
  <h1>タイトル</h1>
  <div class="columns">
    <div class="section">
      <h2>セクション1</h2>
      <p>段落1</p>
    </div>

    <div class="section">
      <h2>セクション2</h2>
      <p>段落2</p>
      
    </div>
  </div>
</main>
```

タイトル

セクション1

段落1

セクション2

段落2

150 × 150

`<div>` タグは、複数のタグをまとめて1つのグループにしたいときに使います。

`<main>` や `<header>` などは見た目は同じですが、意味を付与できます。

CSSってなに？

> ページの「見た目」を飾る言語

- Cascading Style Sheetsの略
- 色、サイズ、フォント、レイアウトなどを指定する
- HTMLに「服を着せる」イメージ

```
h1 { /* ページ内のすべてのh1タグに適用 */  
  color: blue; /* 文字を青く */  
  font-size: 32px; /* 文字を大きく */  
}
```

CSS: 実際のコードで見てみよう

```
<h1>こんにちは！</h1>  
<p>これは私のウェブサイトです。</p>  

```

```
h1 {  
  color: blue;  
  font-size: 32px;  
}  
img {  
  display: block;  
  width: 50%;  
  margin: auto;  
}
```

こんにちは！

これは私のウェブサイトです。

600 × 400

CSSの基本構造

```
/* コメント */  
セクタ { /* どのタグに適用するか指定 */  
  プロパティ: 値;  
  プロパティ: 値;  
}
```

セクタでどのタグに適用するかを指定し、プロパティと値で見た目を指定します。
複数のセクタを指定することもできます。

```
h1, h2, h3 {  
  color: blue;  
}
```

CSSの基本構造: セレクタ

htmlのタグには、クラスやIDを付けることができます。同じクラスを複数のタグに付けられますが、同じIDは1つのタグにしか付けられません。

```
<p class="bold" id="paragraph1">太字の段落</p>
```

タグ名、クラス名、ID名の組み合わせを手がかりに、適用するタグを抽出してスタイルを指定します。

```
p { /* すべてのpタグに適用 */ }  
p.bold { /* class="bold"がついたpタグに適用 */ }  
.bold { /* class="bold"がついたすべてのタグに適用 */ }  
#paragraph1 { /* id="paragraph1"がついた唯一のタグに適用 */ }
```

CSSの基本構造: セレクタ

```
<h1>こんにちは！</h1>
<p>
  これは私のウェブサイトです。
  私の趣味は
  <span
    class="keyword programming"
  >プログラミング</span>と
  <span class="keyword">登山</span>です。
</p>
```

```
span.keyword {
  font-weight: bold;
}

span.programming {
  color: blue;
}
```

こんにちは！

これは私のウェブサイトです。 私の趣味
は **プログラミング**と **登山**です。

CSSの基本構造: プロパティ

```
div {  
  color: yellow; /* 文字色を黄色に */  
  font-weight: bold; /* 太字に */  
  font-size: 40px; /* 文字サイズを40pxに */  
  background-color: black; /* 背景色を黒に */  
  text-align: center; /* 中央寄せに */  
  margin: 30px; /* 外側の余白を30pxに */  
  padding: 20px; /* 内側の余白を20pxに */  
}
```

これはサンプルのdivです。

CSS: こんな書き方もできる

```
<body>
  <div>
    <h1>こんにちは！</h1>
    <p>これは私のウェブサイトです。</p>
  </div>
  
</body>
```

```
body {
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

こんにちは！

これは私のウェブサイトです。

150 × 150

JavaScript

> ページに「動き」や「機能」をつけるプログラミング言語

- ボタンをクリックしたらメニューが開く
- 入力フォームに文字を入れたらメッセージが出る
- ページの一部が自動で更新される
- 独自のhtmlタグを定義する

など、HTMLとCSSだけではできないことを実現します。

その分、新しい概念が多くて少し難しいので、今回はあまり詳しく説明しません。

JavaScript: 読み込み方法

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8" />
    <title>私のウェブサイト</title>

    <!-- 他のファイルを参照する -->
    <script src="js/script.js"></script>

    <!-- 直接コードを書く -->
    <script>
      const buttonClick = () => alert("ボタンがクリックされました!");
    </script>
  </head>
  <body>
    <h1>こんにちは！</h1>
    <button onclick="buttonClick()">クリックしてね</button>
  </body>
</html>
```

今日の作り方：魔法の部品（Web Components）

JavaScriptの技術を使うと、**自分だけのHTMLタグ**を作ることができます。

今回は、ファイルの中の `<template>` タグで定義された「**部品**」を自動的に読み込んで、**自分だけのHTMLタグ** として使えるようにするJavaScriptを用意しました。

`<template>` タグは、それ自体は画面に表示されないタグです。この中にHTMLやCSSを書けば、あとは好きな場所に部品を置くだけで、同じパーツを何度でも使えます。

```
<template id="my-header">
  <style>
    header { background-color: #333; }
  </style>
  <header>
    <h1>私のウェブサイト</h1>
  </header>
</template>
```

今日の作り方：魔法の部品（Web Components）

皆さんは、あらかじめ用意された「部品」を組み合わせるだけで、LPが作れます。

- `<app-header>` → ヘッダー部品
- `<hero-section>` → メイン画像部品
- `<features-section>` → 特徴を紹介する部品

まるで**レゴブロック**のように、ページを組み立ててみましょう！

部品の中身を変えたり、色や画像を変えたりして、自分流にカスタマイズすることもできます。

もちろん、部品を自分で作ることもできますよ！ `<template>` のidが部品名になります。

さあ、作ってみよう！ (Let's Code!)

ここからは、続きのスライドを見ながら、メンターと一緒に作業を進めていきましょう！

このスライドは<https://events.tompeedia.jp/20251018/slide.marp.html>で見られます。

1. **STEP 1: 準備** (ファイルを開いて、部品を読み込む)
2. **STEP 2: 組み立て** (好きな部品を並べてみる)
3. **STEP 3: カスタマイズ** (文字や色、画像を変えてみる)
4. **STEP 4: オリジナル部品作成** (自分だけの部品を作る)

困ったときは？

- **調べるコツを覚えよう！**
 - geminiなどのAIにコードを渡して聞くと、かなり正確に答えてくれます。
 - [MDN Web Docs](#)には、HTML/CSS/JavaScriptのリファレンスが揃っています。[初心者向けの解説](#)もたくさんあります。
- **どんどんメンターに質問しよう！**
 - エラーが出た、思った通りに動かない、なんでも聞いてください。
- **どんどん先に進んでOK!**
 - 分かる人はAIなどの力も借りながら、どんどん進めてください。解説したのはあくまで取っ掛かりの部分です。好きなだけ先に進んでOKです。

楽しむことが一番大事！

それでは、コーディングを始めましょう！

LP作成チュートリアル（詳細版）

LP作成チュートリアル（詳細版）

こんにちは！このチュートリアルでは、10時間かけてじっくりと、HTMLとCSS、JavaScriptの基礎を使って、本格的なランディングページ（LP）を作成する方法を学びます。

一つ一つのステップを楽しみながら進めていきましょう！困ったときは、メンターや[MDN Web Docs](#)を頼ってください。

STEP 1: 準備

1. VSCodeのインストール

VSCodeをインストールします。授業などでもう入れてある人は、飛ばして次のステップに進んでください。

1. [VSCodeの公式サイト](#)にアクセスし、VSCodeをダウンロードしてインストールします。

2. 作業用フォルダを作る

1. PC内の好きな場所に、今回のイベント用のフォルダを作ります。

例: `tompedia-hackathon`

2. VSCodeを起動し、左上の「ファイル」メニューから「フォルダーを開く」を選択し、先ほど作成したフォルダを選択します。

STEP 2: 基本構造の理解

まずは、作業を始めるための準備をします。

1. VSCodeで `index.html` を作る

`index.html` という名前でファイルを作成してください。これが、あなたのWebページの土台になります。

2. 魔法のスク립トを読み込む

index.html の `<head>` タグの中に、以下の1行を書いてください。これが、部品（Webコンポーネント）を使うための魔法のスク립トです。

```
<script src="https://events.tompedia.jp/20251018/js/web-components.js" defer></script>
```


3. 部品カタログを読み込む

`components.html` に、LPで使える部品が `<template>` タグとしてたくさん用意されています。これを `index.html` で使えるようにするため、`components.html` の中身を全てコピーし、`index.html` の `<body>` タグのすぐ下に貼り付けてください。

STEP 3: ページの組み立てとカスタマイズ (2時間)

いよいよページを組み立てていきます。 `<template>` の `id` 名がそのまま部品のタグ名になります。

1. 主要な部品を配置する

index.html の `<body>` の下の方（貼り付けた `<template>` 群の後）に、以下のように好きな部品を配置してみましょう。

```
<body>
  <!-- (ここにcomponents.htmlの中身を貼り付けた) -->

  <!-- ▼▼ ページのコンテンツを組み立てる場所 ▼▼ -->
  <app-header></app-header>
  <hero-section></hero-section>
  <features-section></features-section>
  <app-footer></app-footer>
  <!-- ▲▲ ここまで ▲▲ -->
</body>
```

2. ブラウザで確認

`index.html` をブラウザで開いて、どのように表示されるか確認しましょう。

- VSCodeのLive Preview拡張機能を使うと便利です。
- 左側の拡張機能アイコンをクリックし、「Live Preview」で検索してインストールします。
- `index.html` を右クリックして、「Show Preview」を選択すると、ブラウザでページが開きます。

3. コンテンツを自分だけのものに！

貼り付けた `<template>` の中身を直接編集して、文章や画像を変えてみましょう。

- **文字を変える：** `<h1>` や `<p>` タグに囲まれた文章を好きな言葉に変えてみましょう。
- **画像を変える：** `` タグの `src` 属性のURLを、好きな画像のURLに変えてみましょう。
- `https://source.unsplash.com/random/800x600` のようなランダムな画像URLで試すのも面白いです。
 - 自分で用意した画像を `src/demo/images` フォルダなどに入れて、相対パスで指定することもできます。(例: `src="images/my-photo.jpg"`)

STEP 4: CSSでデザインを自由自在に (4時間)

ここからがデザインの本番です！ `<template>` の中にある `<style>` タグを編集して、見た目を大きく変えていきましょう。

3.1. 色を変えてみよう

- `background-color` : 背景色を変えてみましょう。 `red` , `blue` のような名前でも、 `#ff6347` のようなカラーコードでも指定できます。
- `color` : 文字色を変えてみましょう。

3.2. 大きさと余白を調整しよう

- `font-size` : 文字の大きさを変えてみましょう。単位には `px` や `rem` を使います。
- `padding` : 要素の「内側」の余白です。窮屈な印象をなくするのに役立ちます。
- `margin` : 要素の「外側」の余白です。要素と要素の間の距離を調整します。
- `width` / `height` : 要素の幅や高さを指定します。

3.3. レイアウトを操る (Flexbox)

`features-section` では、`display: flex;` が使われています。このプロパティを色々試してみましょう。

- `justify-content` : `flex-start` (左揃え), `center` (中央揃え), `space-between` (均等配置)などを試してみましょう。
- `flex-direction` : `row` (横並び) を `column` (縦並び) に変えるとどうなるでしょうか？
- `gap` : 要素と要素の間の隙間を調整できます。

3.4. インタラクションを追加する (擬似クラス)

ユーザーがマウスを乗せたときに見た目が変わったらカッコいいですね。

- `.cta-button` のスタイルに、以下を追加してみましょう。ボタンにマウスを乗せると色が変わります。

```
.cta-button:hover {  
  background-color: #ff8a65; /* 少し明るい色に */  
  transform: scale(1.05); /* 少しだけ大きくする */  
  transition: all 0.2s ease; /* なめらかな変化のアニメーション */  
}
```

STEP 5: 新しい部品を作ってみよう (3時間)

既存の部品を改造するだけでは物足りない？ それなら、自分だけのオリジナル部品を作ってみましょう！

1. 新しい<template>を作る

index.html に貼り付けた部品群の一番下に、新しい <template> を追加します。例えば、「お客様の声」セクションを作ってみましょう。

```
<template id="testimonial-section">
  <style>
    /* ここに、この部品だけのスタイルを書く */
    .testimonial {
      padding: 2rem;
      background-color: #f0f8ff;
      text-align: center;
    }
    .quote {
      font-style: italic;
    }
    .author {
      margin-top: 1rem;
      font-weight: bold;
    }
  </style>
  <section class="testimonial">
    <h2>お客様の声</h2>
    <p class="quote">"このプロダクトのおかげで、私の人生は変わりました！"</p>
    <p class="author">満足したお客様</p>
```

2. 呼び出す

`<body>` の好きな場所に、`<testimonial-section></testimonial-section>` と書いて呼び出します。

3. スタイリング

あとはSTEP 3と同じように、自由にスタイルを調整してみましょう！

STEP 6: 仕上げと応用 (残りの時間)

- **全体のバランス調整:** 全ての部品を見ながら、色合いや余白のバランスを整えましょう。
- **応用課題に挑戦:** `exercise.md` に、さらにスキルアップするための課題を用意しました。ぜひ挑戦してみてください！
- **メンターに質問:** わからないこと、もっと知りたいことをメンターにどんどん質問しましょう！

このチュートリアルが終わる頃には、あなたはWebページの構造とデザインを自由に操る第一歩を踏み出しているはずです。楽しんでください！