

第1回：HTMLとは何か

Webの構造を支える言語

今日の目的

- HTMLが**何のために存在するか**を理解する
- ブラウザがHTMLを**どう読み込み表示しているか**を理解する
- よく使う**HTMLタグ**を体系的に把握する
- 最近追加された**便利な要素**を知る

HTMLとは？

- HyperText Markup Language の略
- 「見た目」ではなく 構造 を定義する言語
- Webページの内容を意味的に記述するために存在
- ブラウザ、検索エンジン、機械が読み取れる形式

HTMLが存在する理由

- Webは「**情報共有**」のために設計された
- HTMLは文書を**構造化**するためのルール
- 検索・翻訳・スクレイピングなど、**機械が理解できる文書**を作るための仕組み

ブラウザがHTMLを読み込む流れ

1. HTMLを読み込む
2. DOMツリー（構造）を構築
3. CSSを読み込み → CSSOMツリーを構築
4. DOM + CSSOM → レンダーツリー生成
5. 描画（ペイント）



head / body の役割

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <title>サンプルページ</title>
  </head>
  <body>
    <h1>こんにちは</h1>
    <p>これはHTMLの例です。</p>
  </body>
</html>
```

- `<head>` : ブラウザや検索に必要なメタ情報
- `<body>` : 実際に画面に表示される内容

よく使うタグ（構造）

タグ	意味	よく使う場面
<header>	ページやセクションのヘッダー	ロゴ・ナビ
<nav>	ナビゲーション領域	メニュー・リンク
<main>	文書の主内容	ページの中心部分
<section>	トピックごとの区切り	機能・製品紹介など
<article>	独立したコンテンツ単位	ブログ・ニュースなど
<footer>	ページのフッター	コピーライト・リンクなど

よく使うタグ（テキスト）

タグ	役割
<code><h1></code> ~ <code><h6></code>	見出し構造を定義
<code><p></code>	段落
<code><a></code>	リンク
<code></code>	インライン要素をグループ化
<code></code> / <code></code>	重要 / 強調
<code>
</code> / <code><hr></code>	改行 / 区切り線

リスト・表関連タグ

タグ	概要
<code></code> / <code></code> / <code></code>	箇条書きリスト
<code><dl></code> / <code><dt></code> / <code><dd></code>	用語と説明のペア
<code><table></code>	表全体
<code><thead></code> / <code><tbody></code> / <code><tr></code> / <code><td></code>	表の構造化

メディア関連タグ

タグ	機能
	画像を表示
<video>	動画を埋め込み
<audio>	音声を埋め込み
<source>	メディアの複数形式を指定
<picture>	画面幅に応じて画像を切替

フォーム関連タグ

タグ	説明
<form>	入力フォームの全体
<input>	テキスト・チェックボックスなど
<textarea>	複数行テキスト
<select> / <option>	プルダウン
<button>	送信ボタン
<label>	入力項目の説明

最近の便利タグ

タグ	機能	例
<code><details></code> / <code><summary></code>	折りたたみセクション	
<code><dialog></code>	モーダルダイアログ	
<code><template></code>	非表示のHTMLテンプレート	
<code><slot></code>	Web Components内で使う差し込み箇所	

便利タグの例

```
<details open>
  <summary>もっと詳しく</summary>
  <p>この中に補足情報を入れられます。</p>
</details>
```

```
<dialog open>
  <p>確認しますか？</p>
  <button>OK</button>
</dialog>
```

Note: 上記のコードブロックは ````html` の言語指定により、hljsのクラスが付与され色分けされます。

HTMLを書くときの考え方

- 「見た目ではなく意味」を意識する
- CSSで装飾、JavaScriptで動きを追加
- HTMLの役割：構造を定義すること
- Reactでも最終的にHTML要素を生成している

演習

1. 自分の所属する会社・チームの紹介ページをHTMLで作ってみよう
2. ページ構造を意識して、`header` , `main` , `section` , `footer` を使う
3. `details` や `dialog` を1つ使ってみる

次回予告

👉 第2回：CSS基礎とデザインの考え方

HTMLで作った構造に、見た目とレイアウトを与えます。