

# HTMLの基本ルール

# 1. HTMLのインデント

- **目的:**

- 階層構造を明示して可読性を向上
- インデントがないと、構造が把握しにくくなります

- **基本ルール:**

- ネストした要素は子要素としてインデントを付ける
- スペース2～4個が一般的

例:

```
<div>  
  <p>これは段落です</p>  
  <ul>  
    <li>リスト項目1</li>  
    <li>リスト項目2</li>  
  </ul>  
</div>
```

## 2. ul/li のルール

- 基本:

- `<ul>`（無順序リスト）の中には**必ず** `<li>` 要素だけを記述する
- 他のタグやテキストを直接書くと、HTMLの構造が崩れる可能性があります

正しい例:

```
<ul>  
  <li>項目1</li>  
  <li>項目2</li>  
  <li>項目3</li>  
</ul>
```

※ 注意:

`<ul>` の中に直接 `<div>` やテキストを書かないようにしましょう

### 3.CSSはクラスを使おう

- メリット:

- 再利用性・保守性が向上
- 複数要素に共通のスタイルを適用できる

- 具体例:

HTML側ではクラス名を付与し、CSSファイルでスタイルを定義します。

## HTML:

```
<div class="card">
  <h2 class="card-title">タイトル</h2>
  <p class="card-content">ここに内容が入ります。</p>
</div>
```

## CSS:

```
.card {
  background-color: #f9f9f9;
  border: 1px solid #ccc;
  margin: 10px;
  padding: 20px;
}

.card-title {
  font-size: 1.5em;
  margin-bottom: 10px;
}
```



## 4. HTMLの親子要素と兄弟要素

- **親要素:**
  - 他の要素を内包する要素（例: `<div>`、`<ul>`）
- **子要素:**
  - 親要素内に含まれる要素
- **兄弟要素:**
  - 同じ親要素内に並ぶ要素

## 具体例:

```
<!-- 親要素と子要素の例 -->
<div class="container">
  <!-- 子要素 -->
  <header class="header">ヘッダー</header>
  <main class="content">
    <section class="section">
      <h1>セクションタイトル</h1>
      <p>セクションの内容</p>
    </section>
  </main>
  <footer class="footer">フッター</footer>
</div>
```

```
<!-- 同じ親を持つ兄弟要素の例 -->
<div class="sibling-container">
  <p class="sibling">段落1</p>
  <p class="sibling">段落2</p>
  <p class="sibling">段落3</p>
</div>
```

## 5.margin と padding の役割

- margin:
  - 要素同士（兄弟要素）の外側の余白を設定
  - 適切に使わないと、要素間のスペースが不均一になり、レイアウト調整が大変

- **padding:**

- 要素内部、つまりコンテンツとボーダー間の内側の余白を設定
- 親子要素間の余白としても機能する

例:

```
.box {  
  margin: 15px; /* 隣接する要素との間のスペース */  
  padding: 10px; /* コンテンツと境界線の間のスペース */  
}
```

## 6. position: absolute と position: fixed の使い分け

### position: absolute

- 特徴:
  - 最も近い**positionが指定された親要素**を基準に配置される
  - 通常のレイアウトフローから外れるため、重なり順（z-index）にも注意が必要
- 利用例:
  - 特定の領域にオーバーレイやポップアップを表示する際

例:

```
<div class="relative-container">  
  <div class="absolute-box">絶対配置されたボックス</div>  
</div>
```

# まとめ

- **インデント**: 読みやすいコードはインデントで階層が明確に
- **ul/li**: `<ul>` の中は **必ず** `<li>` を使う
- **CSSはクラス**: HTMLとCSSは分離し、クラスでスタイル管理
- **DOM構造**: 親子・兄弟の関係を正しく理解する
- **余白調整**: margin と padding を正しく使い分けることが大切
- **position**: absoluteは基準要素に依存、fixedは常に画面固定

