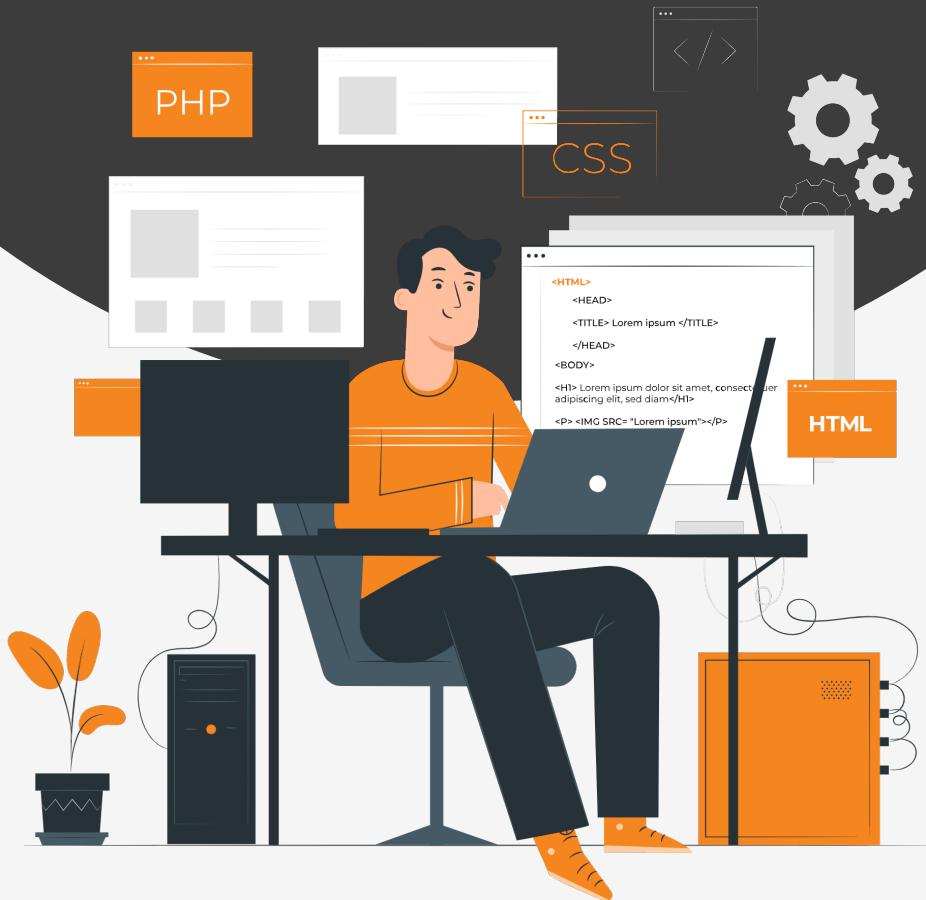


Lesson:

Class Naming Convention in CSS



Topics

- Introduction to a class naming convention
- Importance of consistent and meaningful class names
- Popular naming conventions like BEM

Introduction to a class naming convention

In CSS, A class naming convention in CSS is a set of guidelines used to create consistent and meaningful names for CSS classes applied to HTML elements. The primary goal of a naming convention is to enhance the organisation, maintainability, and readability of CSS code, especially in large projects with multiple developers.

By using a class naming convention, developers can easily understand the purpose and context of each CSS class, making it easier to maintain and extend the codebase. Different conventions exist, and each has its own unique approach to naming classes.

Importance of consistent and meaningful class names

The importance of consistent and meaningful class naming conventions includes the following points

- Understandability and Readability – Clear and proper class names make the code more readable and understandable for both the current developers and anyone who may work on the code in the future.
- Maintainability and Refactoring – Consistent and meaningful class names facilitate code maintenance and refactoring
- Code Reusability – A well-written class name is more likely to be reusable in different parts of the application or a project.
- Collaboration and Teamwork – In a collaborative programming or project environment multiple developers may work on the same project. Meaningful class names provide a common vocabulary to team members, making it easier to communicate and collaborate effectively
- Semantic Clarity – a class with meaningful names that align code with the problem domain, providing semantic clarity.
- Self-Explanatory Code and Documentation – well-chosen names act as documentation in themselves. It serves as comments that explain the purpose and functionality of the class, reducing the need for excessive code comments.
- Avoiding Naming Conflicts – Consistent class names help prevent naming conflicts with other classes or identifiers in the codebase.
- Professionalism and Code Quality – Using a consistent and meaningful class naming convention demonstrates professionalism and adherence to best practices. It reflects the developer's commitment to producing clean, organised, and high-quality code.

Popular naming conventions like BEM

In this modern age of web development, maintaining clean, scalable, and maintainable CSS code is crucial to building robust and efficient websites and web applications. A popular approach to achieving this goal is through the use of CSS naming conventions. Among various naming conventions, BEM (Block, Element, Modifier) stands out as one of the most widely adopted and effective methodologies. BEM provides a structured system for naming classes, offering numerous benefits that enhance code readability, organization, and collaboration among developers.

BEM which stands for **Block**, **Element** and **Modifier**, is a naming convention of CSS that was introduced by Yandex in order to better connect the HTML and CSS files and make developers be more descriptive with their class names.

- **Block** – The highest level of abstraction, representing a standalone component or entity on the page. Blocks are reusable and can be used independently. Block names are written in all lowercase and separated by hyphens. For example **.header**, **.menu**, **.button**
- **Element** – it represents a part of a block that has no standalone meaning. Elements depend on their parent block for context. Element names are written in all lowercase and are separated from the block name by a double underscore (___). For example **.header__logo**, **.menu__item**, **.button__icon**.
- **Modifier** – it is used to change the appearance or behaviour of a block or element. It represents variations or states of a block or element. Modifier names are written in all lowercase and are separated from the block or element name by a double dash (--). For example **.button--primary**, **.menu__item--active**

Syntax of BEM CSS naming convention

```
Unset
.block__element--modifier { /* ... CSS properties */}
```

Example of BEM CSS selector

```
Unset
<body>
  <div class="cards">
    <div class="card__image">
      
    </div>

    <div class="card__description">
      <h2 class="card__title--bold">TITLE HERE</h2>
      <p class="card__paragraph--italic">
        Lorem ipsum dolor sit amet, consectetur
        adipisicing elit. Recusandae
      </p>
    </div>
  </div>
</body>
```

The above example, shows an example of the BEM class naming convention used in CSS.

It's always recommended to use the BEM convention of the class naming convention in CSS as it follows consistency and meaningful naming of the class.