

Lesson:

Introduction to Tailwind CSS



Topics to be covered

1. What is CSS Framework
2. Why CSS framework over CSS
3. What is Tailwind CSS
4. Advantages of Tailwind CSS
5. Disadvantages of Tailwind CSS
6. Understanding the utility first approach to CSS
7. Installing and setting up Tailwind CSS in a project

What is CSS Framework

A CSS framework is a pre-written collection of CSS files that provides a set of pre-built CSS classes and styles that you can use to style your HTML elements. Rather than starting every project from scratch, a CSS framework gives them tools to quickly create a user interface that they repeat and tweak during a project. They also enable the creation of more standards-compliant websites.

Why CSS framework over CSS

Consistency: The use of pre-built classes promotes consistency in your design, as all the styles are defined in one place and can be easily modified.

Efficiency: CSS frameworks can speed up the development process by providing pre-built styles and UI components that can be quickly and easily applied to HTML elements.

Responsive design: Many CSS frameworks include responsive design features that allow your website to adapt to different screen sizes and devices.

Overall, CSS frameworks are a useful tool for web developers and designers who want to create modern, responsive designs quickly and efficiently.

What is Tailwind CSS

Tailwind is a utility-first CSS framework for rapidly building custom user interfaces. It provides a set of pre-built CSS classes to help you quickly and easily style your HTML elements. It is designed to make building and styling websites faster and more efficient.

Tailwind CSS includes a comprehensive set of pre-built classes that cover a wide range of design and layout requirements. It also provides a customizable configuration file that allows you to add or remove classes, modify existing classes, and configure various aspects of the framework to suit your needs.

One of the key benefits of using Tailwind CSS is that it promotes consistency in your design, as all the styles are defined in one place and can be easily modified. It also helps you to avoid the pitfalls of writing custom CSS, such as specificity issues, which can lead to unexpected styling behaviour.

Advantages of Tailwind CSS:

- **Faster Development:** Tailwind CSS provides a set of pre-built CSS classes that can be easily applied to HTML elements, which speeds up the development process significantly.
- **Consistency:** By using Tailwind's predefined classes, developers can ensure a consistent look and feel across their entire project, without having to write custom CSS for each individual element.
- **Easy Maintenance:** With Tailwind CSS, all the styles are defined in one place, which makes it easier to maintain and update your project over time.
- **Responsive Design:** Tailwind CSS includes a range of responsive design classes that allow developers to create layouts that adapt to different screen sizes and devices.
- **Customizable:** Tailwind CSS provides a customizable configuration file that allows you to add or remove classes, modify existing classes, and configure various aspects of the framework to suit your needs.

Disadvantages of Tailwind CSS

- **Learning Curve:** Tailwind CSS has a steep learning curve, particularly for developers unfamiliar with its utility-first approach.
- **Large file size:** Tailwind CSS has a relatively large file size, which can impact page load times. This is because it includes a large number of utility classes that cover a wide range of styles and options. This can make it difficult to optimize web pages for speed and performance.
- **Overuse of Classes:** There is a risk of overusing Tailwind's predefined classes, which can lead to bloated HTML and CSS files.
- **Limited Flexibility:** Tailwind CSS's utility-first approach means that it may not be the best choice for more complex or unique design requirements.
- **Limited Browser Support:** Tailwind CSS relies heavily on CSS Grid and Flexbox, which may not be fully supported by older browsers.
- **Accessibility:** Developers must be careful to ensure that the pre-built classes they use do not compromise accessibility standards.
- **Maintenance:** Because Tailwind CSS relies heavily on utility classes, it can be challenging to maintain and update the codebase over time. If a class is removed or renamed, it can break the design and require extensive testing to fix.

Understanding the utility first approach to CSS

Let's begin with HTML & CSS and then compare

HTML

```
<div class="container">
  <h1 class="heading">Utility-first CSS?</h1>
</div>
```

CSS

```
.container {
  background-color: #6B7280;
  padding: 5rem;
  width: 24rem;
  margin: 5rem;
}

.heading {
  font-size: 1.25rem;
  margin: 0;
}
```

When we use the utility first approach, the above could be refactored into something like this

HTML

```
<div class="b-gray-600 p-20 w-96 m-20">
  <h1 class="text-xl m-0">Utility-first CSS?</h1>
</div>
```

Those tailwind classes will be converted to CSS something like this

```
bg-gray-600 : background-color: #6B7280;
p-20 : padding: 5rem;
m-20 : margin: 5rem;
w-96 : width: 24rem;
```

As you can see, it's simply replacing the high-level classes, such as container and heading, with the low-level classes, such as bg-gray-600 and p-20

These low-level classes are called utility classes. You can think of them as low-level design commodities with predictable names. That's why they're also called atomic classes.

Installing and setting up Tailwind CSS in a project

Use the Play CDN to try Tailwind right in the browser without any build setups. The Play CDN is designed for development purposes only and is not the best choice for production. But let's start learning with this and then gradually we will look into the better approach to set up the tailwind in a project which is the best choice for production as well.

Add the Play CDN script tag to the <head> of your HTML file, and start using Tailwind's utility classes to style your content.

```
<script src="https://cdn.tailwindcss.com"></script>
```

Example

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <!-- tailwind css cdn -->
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <!-- your code goes here -->
</body>
</html>
```

