

@layer Directive in Tailwind: Tailwind Tutorial #10

In this article, you will get a complete understanding of how to use the 'Layer' Directive in Tailwind CSS to enhance the developer experience. You can check out the process of setting up Tailwind CSS for production by clicking [here](#). Let's get started with the tutorial-

Creating Button - Using @apply directive

To understand the layer directive in detail, Let's use the button example as used in the previous video. You can easily create a button by copying the below piece of code-

```
<button class= "text-blue-700 bg-red-600 rounded-xl px-4 py-1">Click Me </button>
```

As we have discussed in the previous video, the '@apply' directive feature is used to apply a number of classes to an individual custom class. This means that after using the apply directive one can easily replace all the above utility classes with a custom class like 'btn' as shown below-

```
<button class= "btn">Click Me </button>
```

Layer Directive

You might know that there are three pre-existing layers in the tailwind CSS which are the base, components, and utilities. A very important point to be noted while creating a custom class using the 'Apply' directive is to maintain the order in which it is defined with respect to the base, component and utility layer. For example, In the below case, The 'btn' class will be rendered after the utility class-

```
# style.css > ...
1  @tailwind base;
2  @tailwind components;
3
4  @tailwind utilities;
5  .btn{
6    @apply text-white bg-red-600 rounded-xl px-2 py-1 text-sm;
7  }
```

→ The Three Layers in Tailwind CSS

→ Creating a 'btn' custom Class


→ Using the '@apply' directive

Figure 1.1: Layer Directive

Using Layer directive

Tailwind CSS provides a fascinating way of fixing this issue by providing the feature of Layer directive. The Layer directive allows you to create custom classes as a part of the three fundamental layers as shown below-

```
1  @tailwind base;
2  @tailwind components;
3  .componentsafter{
4      text-align: center;
5  }
6  @tailwind utilities;
7
8  @layer components{
9      .btn{
10         @apply text-white bg-red-600 rounded-xl px-2 py-1 text-sm;
11     }
12 }
```



Declaring another class

Using the '@layer' directive

Figure 1.2: Using Layer Directive

As a result, the 'btn' class will be added at the end of the component layer. To check this, You can create another custom class such as 'components after' and compare the position of the two classes with respect to the component class. You will find that the 'btn' class has been rendered before the 'components after' class. Thus, the layer directive is used to put the custom class before or after a specific layer.