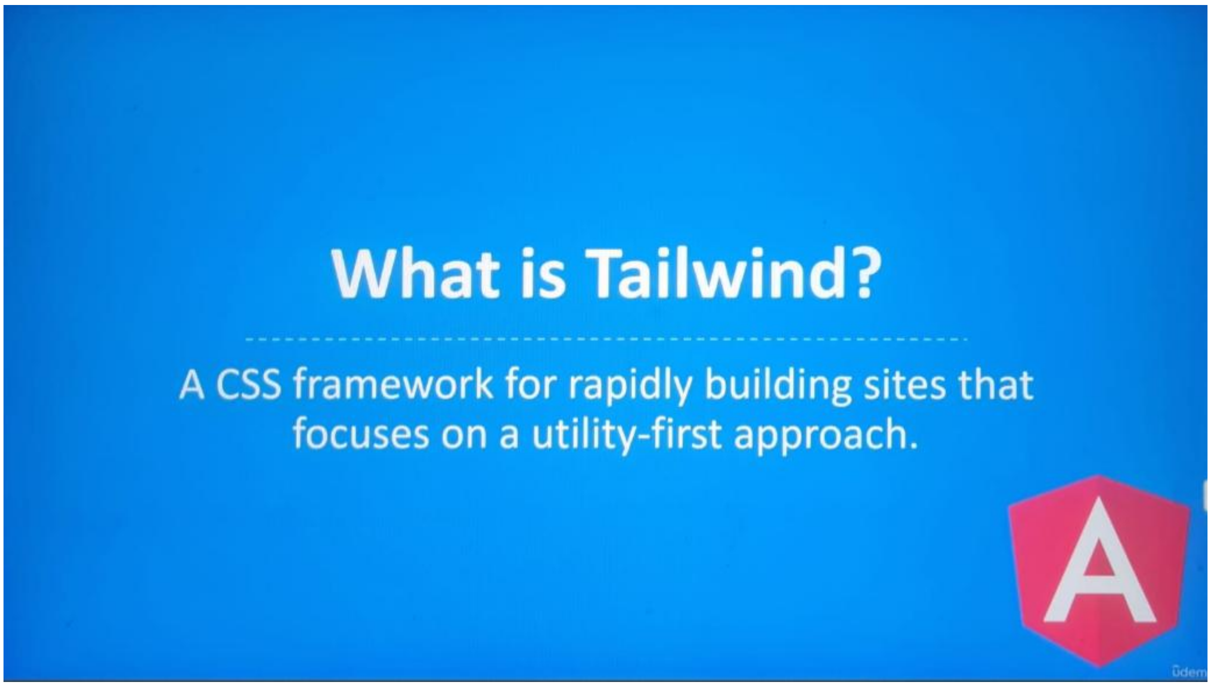
What is Tailwind?

In this lecture, we're going to talk about **Tailwind**.



What is Tailwind?

A CSS framework for rapidly building sites that focuses on a utility-first approach.

It's the new kid on the block.

In the past year, **Tailwinds** user base has grown exponentially. There's a good reason for it.

Tailwind makes developing sites so much easier and faster.

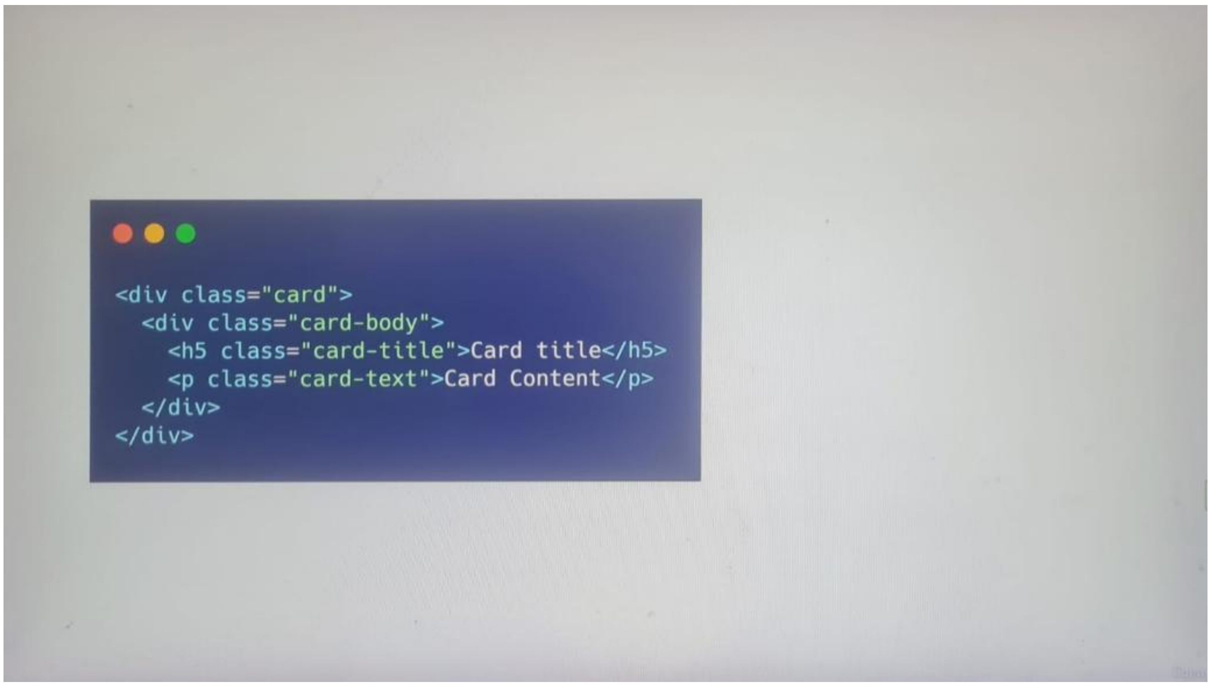
So, what is **Tailwind**? **Tailwind** is a **CSS framework**. It's similar to other frameworks like **Bootstrap** and **Bulma**. It stands out from the crowd because of the approach it takes with **CSS**.

Let's talk about how **Bootstrap** approaches **CSS**:

Bootstrap comes with classes for building UI elements.

For example, let's say we wanted to use the card components.

This code snippet shows how a card component can be created with **Bootstrap**:



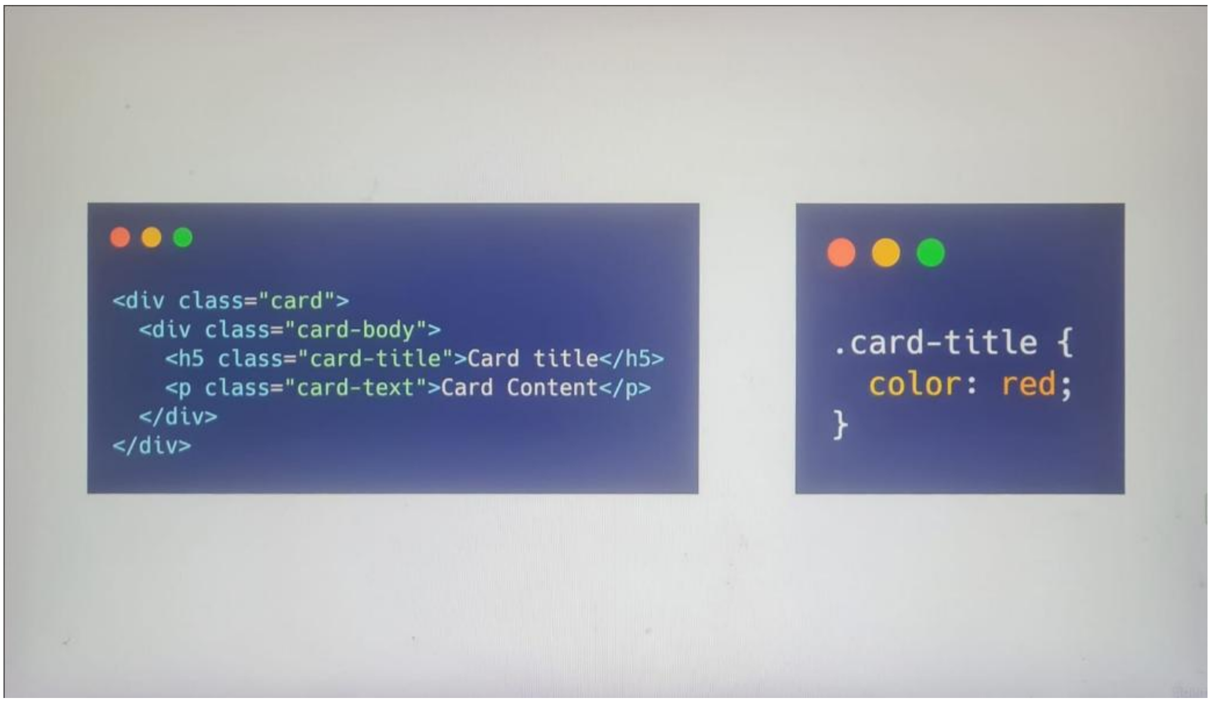
```html
<div class="card">
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Card Content</p>
  </div>
</div>
```

We need to create some div tags and slap some classes on them. Pretty simple, right?

The biggest criticism of **Bootstrap** is how similar a lot of sites look. It's because **Bootstrap** takes care of defining a predefined set of components. These components are ready to go out of the box. Customizing them is completely optional.

In some cases, you may want to change the appearance of some **Bootstrap** element.

For example, let's say we want the color of the title to be red:



```html
<div class="card">
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Card Content</p>
  </div>
</div>
```

```css
.card-title {
  color: red;
}
```

We can select the **card-title element** to apply the changes. This works, but it's going to affect other card elements with titles.

One solution to get around this is by using a better specifier, such as assigning an ID.

It works, but it doesn't scale very well.

This approach forces us to constantly switch between documents and style sheets.

You might also have to rewrite the same CSS. The more UI elements we have, the more CSS we'll have to write.

Another challenge is having to memorize what every class does.

The classes in **Bootstrap** describe the UI element. They don't describe what properties are applied to the element.

Tailwind resolve this issue by focusing on a different way to write templates.

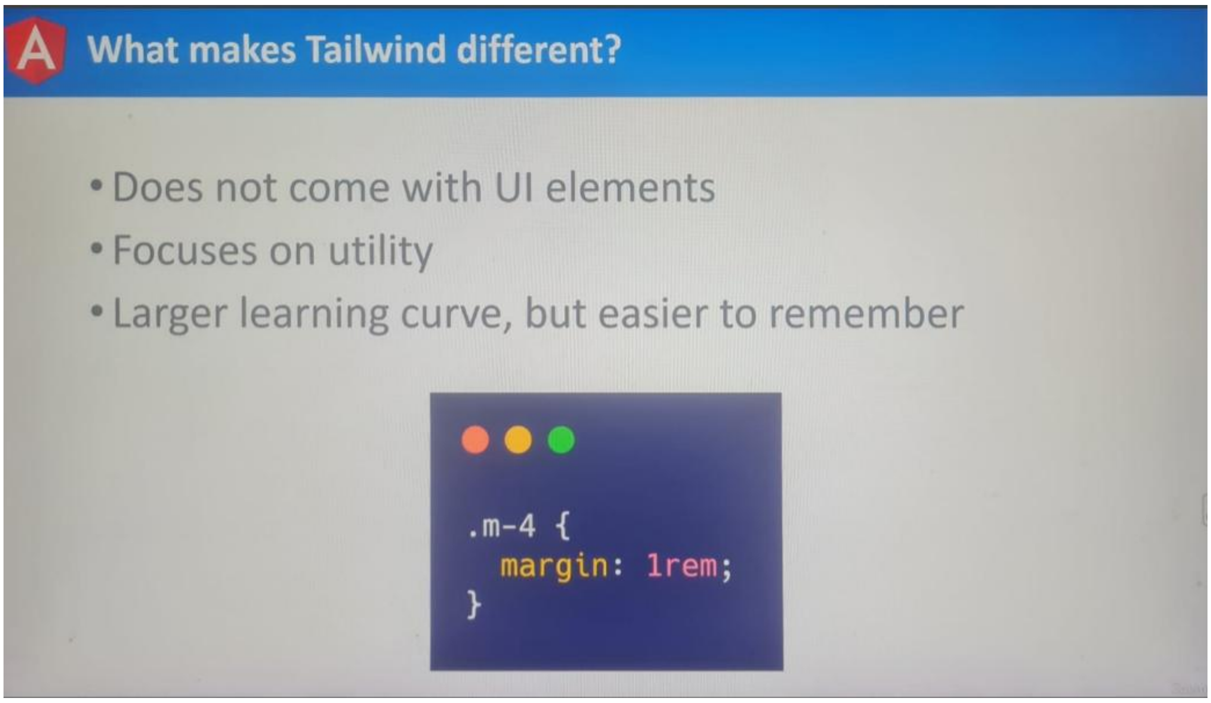
Tailwind does not focus on UI elements. If you were to browse the documentation, you won't encounter many UI elements. Instead, **Tailwind** focuses on **utility classes**.

What does that exactly mean?

Tailwind will define hundreds of classes. Each class will modify **one property**.

For example, there's a class called **m-4**. The **m-4** class will set the margins of an element to one **1rem**. There aren't any other properties under this class.

On the other hand, a **Bootstrap** class can **modify multiple properties** of an element.



What makes Tailwind different?

- Does not come with UI elements
- Focuses on utility
- Larger learning curve, but easier to remember

```
.m-4 {
  margin: 1rem;
}
```

For this reason, **Tailwind** has a larger learning curve. It's a framework that demands solid knowledge of **CSS**.

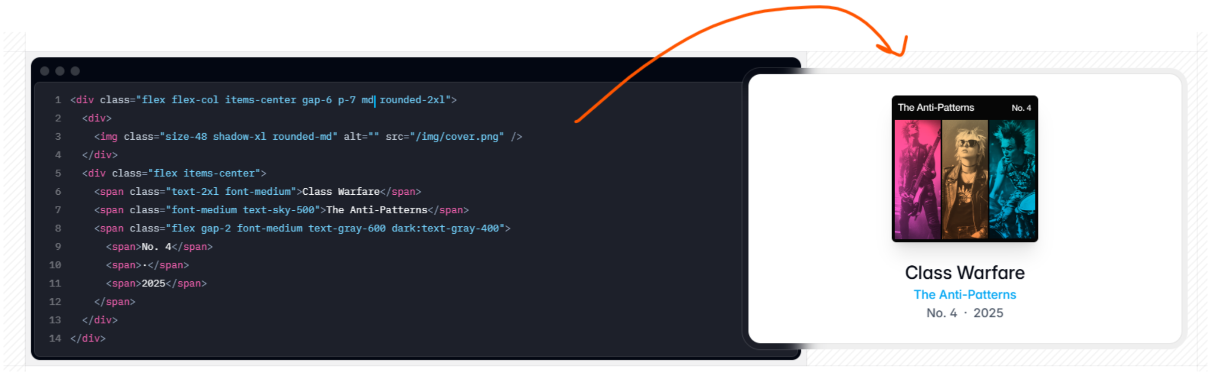
You won't get far if you don't know much about **CSS**. Luckily, **CSS** is a simple language.

Tailwind looks difficult to read, but it's not once you understand what each class does. By looking at the classes, you will immediately know what properties are being applied to the elements.

This saves you time from having to switch over to the stylesheet to understand the class.

In the resource section of this lecture, I provide a link to the official **Tailwind** site: https://tailwindcss.com/

On the home page, we'll be shown an example of the **type UI elements** we can create with this framework. On the right will find what our HTML will look like with **Tailwinds**:



```html
<div class="flex flex-col items-center gap-6 p-7 md rounded-2xl">
  <div>
    <img class="size-48 shadow-xl rounded-md" alt="" src="/img/cover.png" />
  </div>
  <div class="flex items-center">
    <span class="text-2xl font-medium">Class Warfare</span>
    <span class="font-medium text-sky-500">The Anti-Patterns</span>
    <span class="flex gap-2 font-medium text-gray-600 dark:text-gray-400">
      <span>No. 4</span>
      <span>·</span>
      <span>2025</span>
    </span>
  </div>
</div>
```

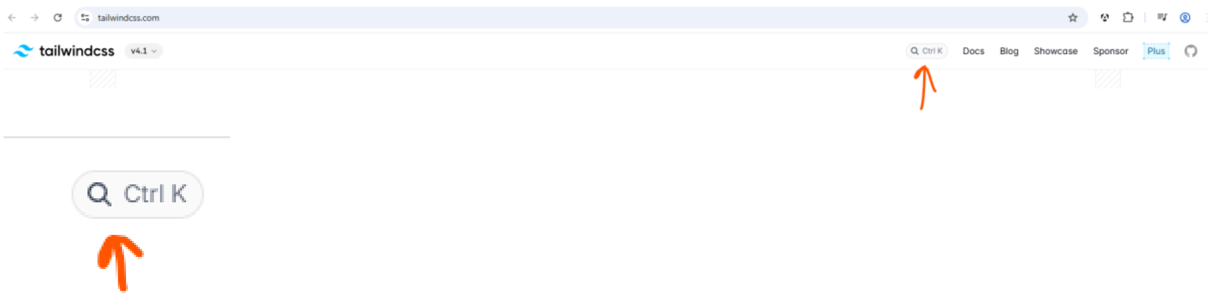
Unlike **Bootstrap**, multiple classes are being applied to the elements.

Each class will change one property.

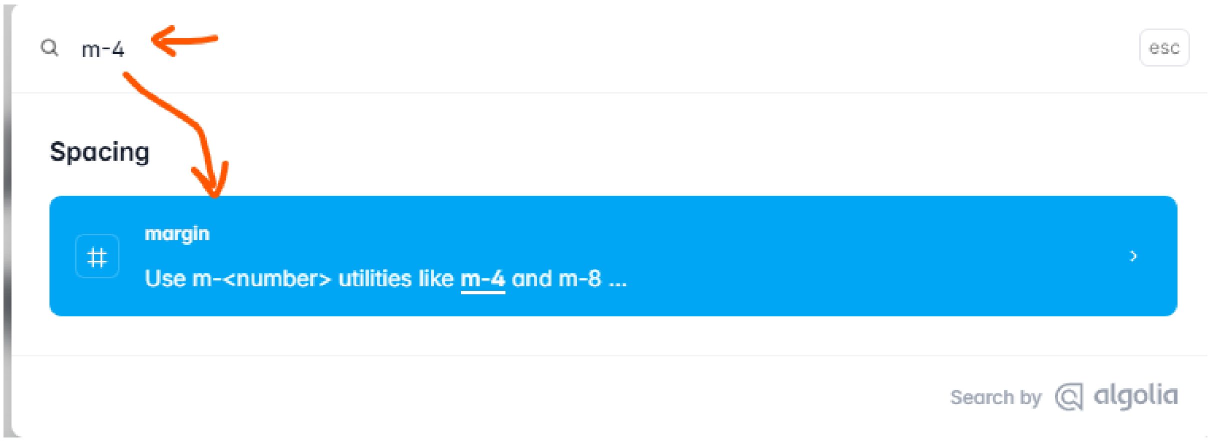
From this live example, we can see how each class affects the element as they're added in.

This leads us to the question How do we know which class to use?

Luckily, **Tailwind** has some of the best documentation around. Each class is well documented. At the top, we can search for any class.



Let's search for the **m-4** class we looked at earlier.



According to the documentation, this class will change the margins of an element.

It'll even tell us by how much.

We never have to look at the **CSS**. The documentation will always tell us what property and value a class will have.

We're not limited to a single class, either.

There are multiple classes for changing the margins of an element. We can add margins to one side or access. **Tailwind** seems great so far.

So, what's the catch?

Tailwind is much larger than most libraries. On a compressed, **Tailwind** is over 300 kilobytes large.

Luckily, **Tailwind** works hand in hand with a library called **PurgeCSS**.

In the resource section of this lecture, I provide a link to the **PurgeCSS** library: https://purgecss.com/

PurgeCSS will scan your HTML for classes that are being used.

It will then remove unused **CSS** from your style sheets.

PurgeCSS is commonly used with **Tailwind**. On average, most **Tailwind** sites use less than 10 kilobytes. This process is automatically done for us by installing the **Tailwind** module.

As you can see, **Tailwind** is rising in popularity for a reason.

If you're confused by any class we use, you can always consult the documentation.

It'll tell you right away what property gets applied to an element.

For this course, we're going to be using **Tailwind** for our Gaming Highlights app.

In the next lecture, we're going to install **Tailwind**.