**To create a custom directive in Angular, you can follow these steps:**

**Generate the Directive:**

Use the Angular CLI to generate a new directive. Open your terminal or command prompt and run the following command:

**ng generate directive directive-name**

Replace directive-name with the desired name of your custom directive.

**Implement the Directive:**

After generating the directive, navigate to the generated .ts file in the src/app directory (or the directory specified when generating the directive). The file should have a format like **directive-name.directive.ts.**

**Define the Directive:**

In the directive file, define the directive class and import necessary Angular modules and classes.

The directive class should implement the Directive decorator.

**import { Directive, ElementRef, Input } from '@angular/core';**

**@Directive({**

**selector: '[appCustomDirective]' // Selector for using the directive**

**})**

**export class CustomDirective {**

**// Add your directive logic here**

**}**

**Add Logic to the Directive:**

Inside the CustomDirective class, you can add logic that defines the behavior of your directive. You can interact with the DOM using the **ElementRef** and modify the element's properties, listen for events, and more.

**import { Directive, ElementRef, Input } from '@angular/core';**

**@Directive({**

**selector: '[appCustomDirective]'**

**})**

**export class CustomDirective {**

**constructor(private el: ElementRef) {**

**this.el.nativeElement.style.backgroundColor = 'yellow';**

**}**

**}**

**Use the Directive in a Template:**

You can use your custom directive in an Angular component's template by applying the directive's selector as an attribute.

**<div appCustomDirective>**

**This content will have the custom directive applied to it.**

**</div>**

In this example, the appCustomDirective attribute applies your custom directive to the div element.

**Add the Directive to a Module:**

To use your custom directive in your application, you need to add it to an Angular module's declarations array. Open the module file (e.g., app.module.ts) and import your directive class.

**import { CustomDirective } from './path-to-custom-directive.directive';**

**@NgModule({**

**declarations: [**

**// ...**

**CustomDirective // Add your custom directive here**

**],**

**// ...**

**})**

**export class AppModule {}**

By following these steps, you'll be able to create and use your custom directive in Angular applications. Custom directives are a powerful way to encapsulate reusable behavior and modify elements in the DOM.