

Component in Angular:

Components in Angular are TypeScript classes that are decorated with `@Component` metadata. They consist of:

Selector: It specifies how the component will be used in HTML.

TemplateUrl: It defines the HTML view of the component.

StyleUrls: It defines the styles associated with the component.

Class: It holds the data and logic of the component.

Example:

Suppose you have an Angular application and you want to create a simple component.

Create a new component:

Use the Angular CLI command to generate a new component.

```
//ng g c my-component
```

This command generates the necessary files for a new component, including TypeScript, HTML, CSS, and a spec file for testing.

My-component.component.ts:

```
import { Component } from '@angular/core';
```

```
@Component({
```

```
  selector: 'app-my-component',
```

```
  templateUrl: './my-component.component.html',
```

```
  styleUrls: ['./my-component.component.css']
```

```
  })

  export class MyComponent {

    title = 'My Component Title';

    // Other properties and methods can be defined here

  }
```

In `MyComponent` class, `title` is a property that holds the title of the component.

- The `@Component` decorator specifies the metadata for the component, including its selector, template URL, and style URLs.
- The HTML template (`my-component.component.html`) defines how the component looks in the view.
- The CSS file (`my-component.component.css`) provides styles specific to this component.

This component (`MyComponent`) can now be used within other components by adding `<app-my-component></app-my-component>` to their templates.

This example demonstrates the basic structure and usage of an Angular component. Components help in building modular and reusable UI elements in Angular applications.