Lecture 13: Angular Components

What are Angular Components?

- **Components** are the basic building blocks of Angular applications, controlling views (**UI**) and logic.
- Encapsulate UI, Logic, and styles for reuse and modular development.
- Think of components as Lego blocks that build complete structure.

Component Structure

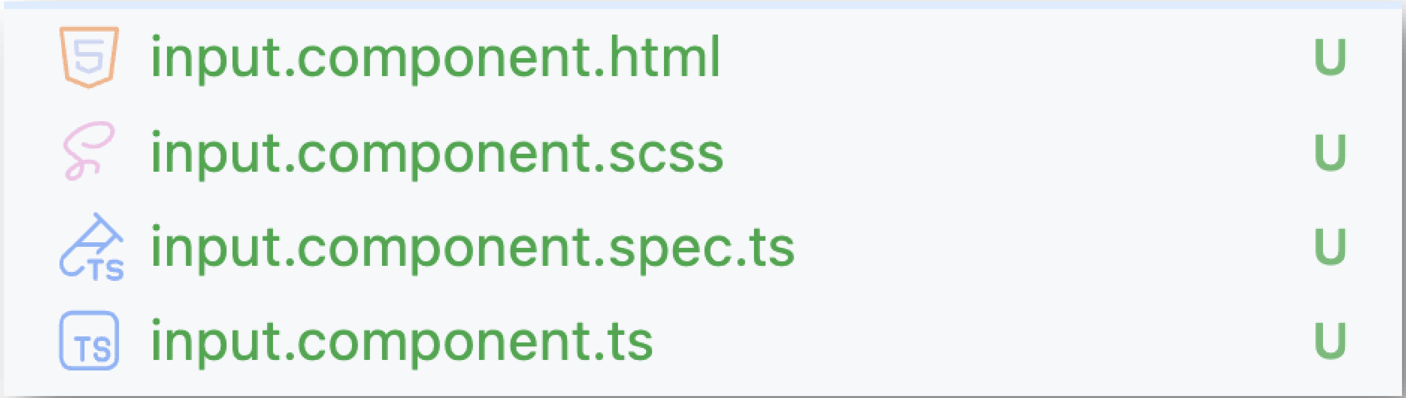
- **Template**: HTML for the view.
- **Class:** Typescript code for logic and data.
- **Styles:** CSS/SCSS for design.
- **Metadata:** Defines component's behavior.

How components work

- Represent piece of user interface.
- Interact using **Inputs** and **Outputs**.
- Parent-Child relationships.
- Communications between **Components**.

Creating Component

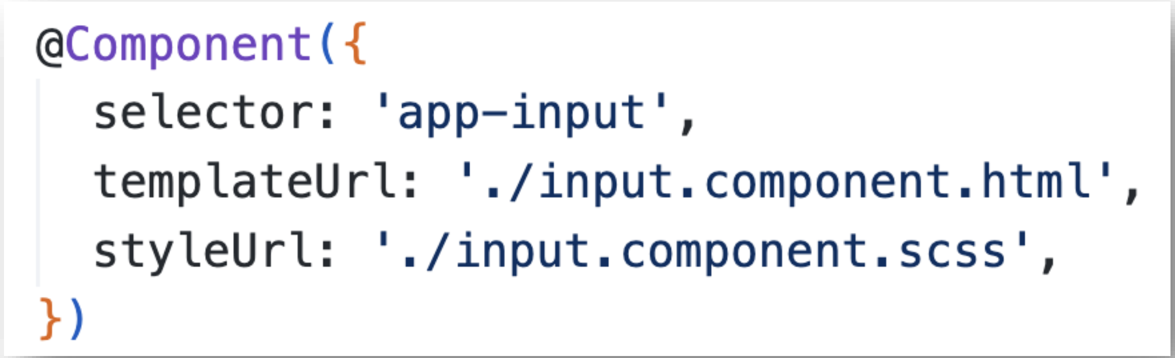
In Terminal: ng generate component component-name



```
input.component.html       U
input.component.scss       U
input.component.spec.ts    U
input.component.ts         U
```

Component Decorator

- **selector** Defines how to use the component in HTML
- **templateUrl** or **Template** points to HTML file or inline HTML
- **stylesUrls** or **styles** points to CSS/SCSS files or inline styles



```
@Component({
  selector: 'app-input',
  templateUrl: './input.component.html',
  styleUrl: './input.component.scss',
})
```

Templates and Data Binding

Templates:

- Define HTML stucture and use Angular syntax for dynamic content
- Use structural and attribute directives

Data Binding:

- **Interpolation { {   } }** – Embed dynamic data
- **Property Binding [property]** – Bind a property to an expression
- **Event Binding (event) -** Trigger actions from user events
- **Two-Way Binding [(ngModel)]** – Sync data between model and view

Directives:

- **Structural -   *ngIf   *ngFor -** Change DOM structure
- **Attribute -   ngClass   ngStyle -** Change DOM appearance or behavior

Lifecycle hooks

- **ngOnInit( )**: Initialization logic
- **ngOnChanges( )**: React to input changes
- **ngOnDestroy( )**: Cleanup