Angular

Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps.

Angular Components

Components are the main building block for Angular applications.

Each component consists of:

- An HTML template that declares what renders on the page
- A TypeScript class that defines behavior
- A CSS selector that defines how the component is used in a template
- Optionally, CSS styles applied to the template

Creating a component

- The best way to create a component is with the Angular CLI. You can also create a component manually.
- Creating a component using the Angular CLI
- To create a component using the Angular CLI:

- From a terminal window, navigate to the directory containing your application.
- Run the **ng generate component <component-name>** command, where <component-name> is the name of your new component.

Defining a component's template

- To define a template as an external file, add a templateUrl property to the @Component decorator.

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

```
@Component({
selector: 'app-component-overview’,
template: '<h1>Hello World!</h1>’,
})
```

```
@Component({
  selector: 'app-component-overview',
  template: `
    <h1>Hello World!</h1>
    <p>This template definition spans multiple lines.</p>
  `
})
```

Declaring a component's styles

Declare component styles used for its template in one of two ways:

By referencing an **external file**, or **directly within the component**.

```
@Component({
  selector: 'app-component-overview',
  templateUrl: './component-overview.component.html',
  styleUrls: ['./component-overview.component.css']
})
```

```
@Component({
  selector: 'app-component-overview',
  template: '<h1>Hello World!</h1>',
  styles: ['h1 { font-weight: normal; }']
})
```

Selector:

A CSS selector that tells Angular to create and insert an instance of this component wherever it finds the corresponding tag in template HTML.

For example,

if an application's HTML contains <app-hero-list></app-hero-list>, then Angular inserts an instance of the HeroListComponent view between those tags.

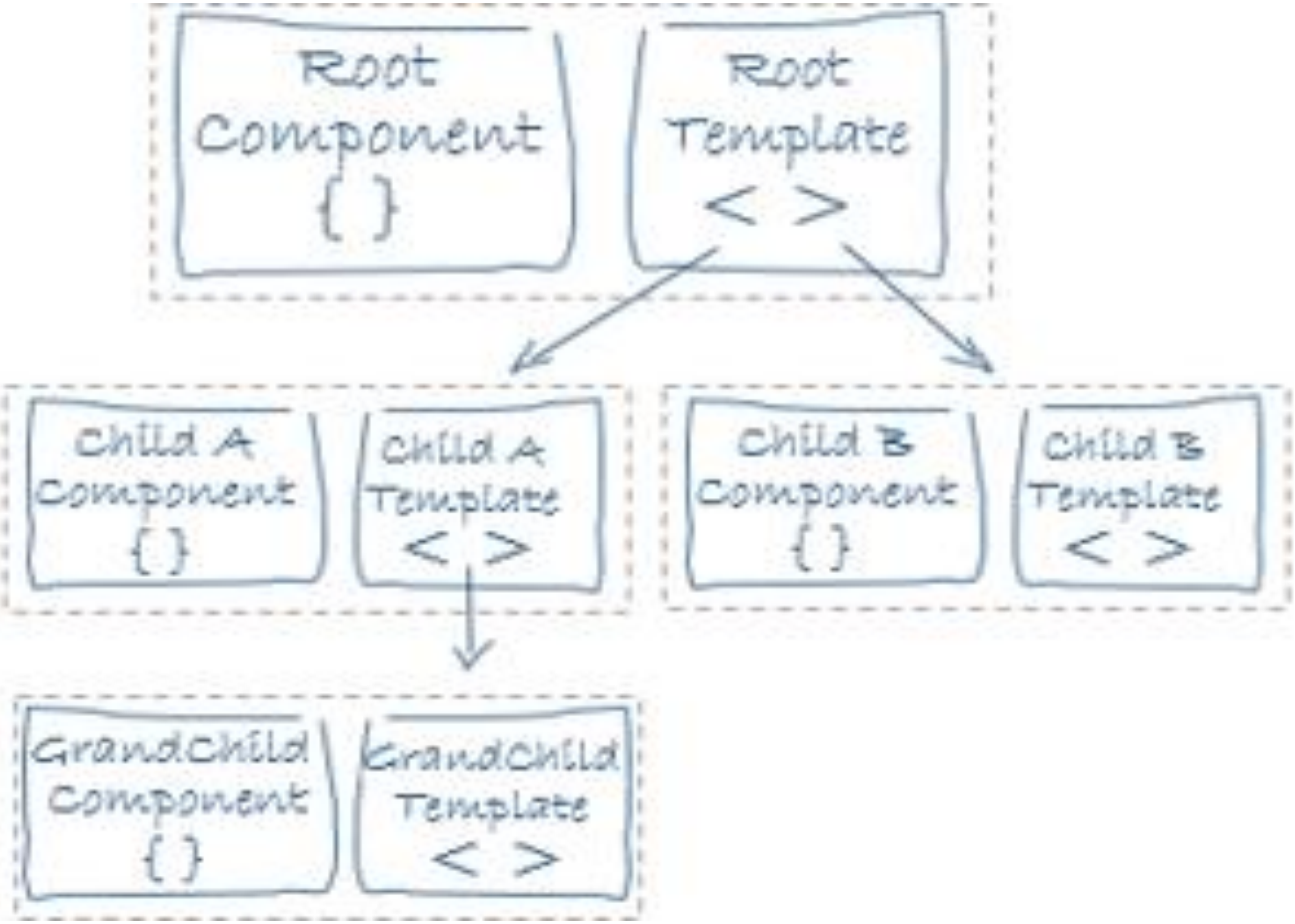
- **Template:** A template is a form of HTML that tells Angular how to render the component.
- **templateUrl**

The module-relative address of this component's HTML template. Alternatively, you can provide the HTML template inline, as the value of the template property.

This template defines the component's host view.

Template

- A template looks like regular HTML, except that it also contains Angular template syntax, which alters the HTML based on your application's logic and the state of application and DOM data.
- Your template can use *data binding* to coordinate the application and DOM data, *pipes* to transform data before it is displayed, and *directives* to apply application logic to what gets displayed.


Root Component
{ }
Root Template
< >
Child A Component
{ }
Child A Template
< >
Child B Component
{ }
Child B Template
< >
GrandChild Component
{ }
GrandChild Template
< >