Components in Depth

- A component declares a reusable building block of an app
- A TypeScript class is used to define a component coupled with the `@component` decorator
- The `@component` decorator defines the following:

selector: `string` value defining the css selector inputs: `array of string` values defining the inputs to the component outputs: `array of string` values defining the output of the component properties: `array of string` values defining the properties events: `array of string` values defining the events host?: {[key: string]: string}, providers: `array of objects` defining the providers for the component exportAs: `string` value defining the exported value moduleId: `string` value defining the module id viewProviders: `array of objects` defining the providers for the view queries: {[key: string]: any}, changeDetection: `ChangeDetectionStrategy` object defining the strategy for detecting changes:

```
- `ChangeDetectionStrategy.Default`: sets detector mode to `CheckAlways`
- `ChangeDetectionStrategy.OnPush`: sets detector mode to `CheckOnce`
- `ChangeDetectionStrategy.Detached`: change detector sub tree is not a part of the main tree and should be skipped
- `ChangeDetectionStrategy.CheckAlways`: after calling detectChanges the mode of the change detector will remain `CheckAlways`
- `ChangeDetectionStrategy.Checked`: change detector should be skipped until its mode changes to `CheckOnce`
- `ChangeDetectionStrategy.CheckOnce`: after calling detectChanges the mode of the change detector will become `Checked`
```

templateUrl: `string` value for the url path to the template template: `string` value for the template styleUrls: `array of string` values defining url paths to css files styles: `array of string` values defining css styles:

```
-   styles: ['.myclass { color: #000;}'],
```

directives: `array` of directives used in the component pipes: `array` of pipes used in the component encapsulation: `ViewEncapsulation` value that defines template and style encapsulation options: - `ViewEncapsulation.None`: means do not provide any style encapsulation - `ViewEncapsulation.Emulated`: No Shadow DOM but style encapsulation emulation using extra attributes on the DOM (default method) - `ViewEncapsulation.Native`: means provide native shadow DOM encapsulation and styles appear in component’s template inside the shadow root.