**Lesson05 Styles in Angular 4**

**Notes:-**

**\*There are 3 different ways to apply styles in Angular Project**

**Option 1:**Specify the following <table> and <td> styles in external stylesheet - **styles.css** 

table {

    color: #369;

    font-family: Arial, Helvetica, sans-serif;

    font-size: large;

    border-collapse: collapse;

}

td {

    border: 1px solid black;

}

**Advantages:**

**1-Visual Studio editor features (Intellisense, Code completion & formatting) are available.**

**2-Application maintenance is also easy as we only have to change the styles in one place if we need to change them for any reason.**

**Disadvantages:**

**1-The Style sheet that contains the styles must be referenced for the component to be reused.**

**2-Since styles.css is referenced in index.html page, these styles may affect the table and td elements in other components, and you may or may not want this behavior.**

**Option 2 : Specify the styles inline in the component HTML file as shown below.**

**<table style="color: #369;font-family: Arial, Helvetica, sans-serif;**

**font-size:large;border-collapse: collapse;">**

**<tr>**

**<td style="border: 1px solid black;">First Name</td>**

**<td style="border: 1px solid black;">{{firstName}}</td>**

**</tr>**

**<tr>**

**<td style="border: 1px solid black;">Last Name</td>**

**<td style="border: 1px solid black;">{{lastName}}</td>**

**</tr>**

**<tr>**

**<td style="border: 1px solid black;">Gender</td>**

**<td style="border: 1px solid black;">{{gender}}</td>**

**</tr>**

**<tr>**

**<td style="border: 1px solid black;">Age</td>**

**<td style="border: 1px solid black;">{{age}}</td>**

**</tr>**

**</table>**

**Advantages :**

**1-Visual Studio editor features (Intellisense, Code completion & formatting) are available.**

**2-Component can be easily reused as the styles are defined inline**

**3-Styles specified using this approach are local to the component and don't collide with styles used elsewhere in the application.**

**Disadvantages :**

**1-Application maintenance is difficult. For example, if we want to change the <td> border colour to red we have to change it in several places**.

**Option 3 : Specify the styles in the component html file using <style> tag as shown below**

**<style>**

**table {**

**color: #369;**

**font-family: Arial, Helvetica, sans-serif;**

**font-size: large;**

**border-collapse: collapse;**

**}**

**td {**

**border: 1px solid black;**

**}**

**</style>**

**<table>**

**<tr>**

**<td>First Name</td>**

**<td>{{firstName}}</td>**

**</tr>**

**<tr>**

**<td>Last Name</td>**

**<td>{{lastName}}</td>**

**</tr>**

**<tr>**

**<td>Gender</td>**

**<td>{{gender}}</td>**

**</tr>**

**<tr>**

**<td>Age</td>**

**<td>{{age}}</td>**

**</tr>**

**</table>**

**Advantages:**

**1-Component can be easily reused as the styles are defined inline with in the component itself**

**2-Application maintenance is also easy as we only have to change the styles in one place**

**3-Visual Studio editor features (Intellisense, Code completion & formatting) are available**

**Option 4 : Specify the styles in the component TypeScript file using the @component decorator styles property as shown below. Notice the styles property takes an array of strings containing your styles.**

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

**@Component({**

**selector: 'my-employee',**

**templateUrl: 'app/employee/employee.component.html',**

**styles: ['table { color: #369; font-family: Arial, Helvetica, sans-serif; font-size: large; border-collapse: collapse;}', 'td {border: 1px solid black; }']**

**})**

**export class EmployeeComponent {**

**firstName: string = 'Tom';**

**lastName: string = 'Hopkins';**

**gender: string = 'Male';**

**age: number = 20;**

**}**

**Advantages :**

**1-Component can be easily reused as the styles are defined inline with in the component itself**

**2-Application maintenance is also easy as we only have to change the styles in one place for this component if we need to change them for any reason.**

**3-Styles specified using this approach are local to the component and don't collide with styles used elsewhere in the application.**

**Disadvantages :**

**1-Visual Studio editor features (Intellisense, Code completion & formatting) are not available.**

**Option 5 : Specify the styles using the @component decorator styleUrls property. The styleUrls property is an array of strings containing stylesheet URLs.   
  
Step 1 : Right click on the "employee" folder and add a new StyleSheet. Name it employee.component.css  
  
Step 2 : Copy and paste the following styles for <table> and <td> elements in employee.component.css**

**table {**

**color: #369;**

**font-family: Arial, Helvetica, sans-serif;**

**font-size: large;**

**border-collapse: collapse;**

**}**

**td {**

**border: 1px solid black;**

**}  
Step 3 : In employee.component.ts file reference employee.component.css stylesheet using styleUrls property as shown below. Please note, the stylesheet path is relative to index.html file.**

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

**@Component({**

**selector: 'my-employee',**

**templateUrl: 'app/employee/employee.component.html',**

**styleUrls: ['app/employee/employee.component.css']**

**})**

**export class EmployeeComponent {**

**firstName: string = 'Tom';**

**lastName: string = 'Hopkins';**

**gender: string = 'Male';**

**age: number = 20;}**

**Advantages :**

**Component can be easily reused as both the stylesheet itself and it's path are included with in the component**

**Application maintenance is also easy as we only have to change the styles in one place**

**1-visual Studio editor features (Intellisense, Code completion & formatting) are available**

**2-Styles specified using this approach are local to the component and don't collide with styles used elsewhere in the application**