**Angular String Interpolation**

In Angular, String interpolation is used to display dynamic data on HTML template (at user end). It facilitates you to **make changes on component.ts** file and fetch data from there to HTML template (component.html file).

**For example:**

**component.ts file:**

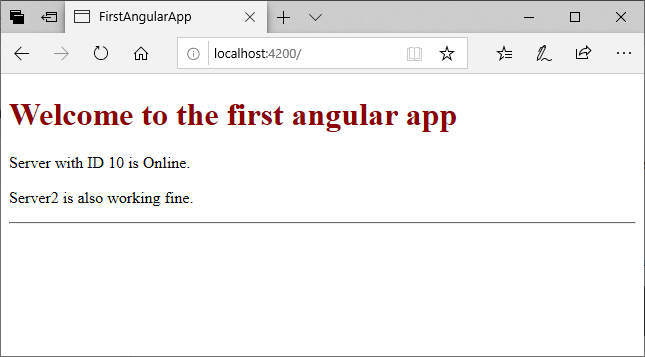
1. import {Component} from '@angular/core';
2. @Component(
3. {selector: 'app-server',
4. templateUrl: 'server.component.html'})
5. export class ServerComponent {
6. serverID: number = 10;
7. serverStatus: string = 'Online';
8. }

Here, we have specified serverID and serverStatus with some values. Let's use this in "component.html" file.

**component.html file:**

1. <p>Server with ID {{serverID}} is {{serverStatus}}. </p>

**Output:**



**String Interpolation vs Property Binding**

String Interpolation and Property binding both are used for same purpose i.e. one-way databinding. But the problem is how to know which one is best suited for your application.

Here, we compare both in the terms of Similarities, Difference, Security and the output you receive.

**Similarities b/w String Interpolation and Property Binding**

String Interpolation and Property Binding doth are about one-way data binding. They both flow a value in one direction from our components to HTML elements.

**String Interpolation**

1. import { Component } from '@angular/core';
2. @Component({
3. selector: 'my-app',
4. template: `
5. <h1>{{ fullName }}</h1>
6. `
7. })
8. export class AppComponent {
9. fullName: string = 'Robert Junior';
10. }

You can see in the above example, Angular takes value of the fullName property from the component and inserts it between the opening and closing <h1> element using curly braces used to specify interpolation.

**Property Binding**

1. import { Component } from '@angular/core';
2. @Component({
3. selector: 'my-app',
4. template: `
5. <h1 [innerHtml]='fullName'></h1>
6. `
7. })
8. export class AppComponent {
9. fullName: string = 'Robert Junior';
10. }

In Property binding, see how Angular pulls the value from fullName property from the component and inserts it using the html property innerHtml of <h1> element.

Both examples for string interpolation and property binding will provide the same result.

**Difference between String interpolation and Property Binding**

String Interpolation is a special syntax which is converted to property binding by Angular. It's a convenient alternative to property binding.

When you need to concatenate strings, you must use interpolation instead of property binding.

**Example:**

1. @Component({
2. selector: 'my-app',
3. template: `<div>
4. <h1>{{citedExample}}</h1>
5. </div>`
6. })
7. export class AppComponent {
8. citedExample: string = 'Interpolation foe string only';
10. }

**Property Binding is used when you have to set an element property to a non-string data value.**

**Example:**

In the following example, we disable a button by binding to the Boolean property isDisabled.

1. import { Component } from '@angular/core';
2. @Component({
3. selector: 'my-app',
4. template: `<div>
5. <button [disabled]='isDisabled'>Disable me</button>
6. </div>`
7. })
8. export class AppComponent {
9. isDisabled: boolean = true;
10. }

If you use interpolation instead of property binding, the button will always be disabled regardless isDisabled class property value is true or false.

1. import { Component } from '@angular/core';
2. @Component({
3. selector: 'my-app',
4. template: `<div>
5. <button disabled='{{isDisabled}}'>Disable Me</button>
6. </div>`
7. })
8. export class AppComponent {
9. isDisabled: boolean = true/false;
10. }