Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

Class binding in angular 2

Suggested Videos
Part 9 - Property binding in Angular 2 | Text | Slides
Part 10 - html attribute vs dom property | Text | Slides
Part 11 - Angular attribute binding | Text | Slides

In this video we will discuss **CSS Class binding in Angular** with examples.

For the demos in this video, we will use same example we have been working with so far in this video series. In **styles.css** file include the following 3 CSS classes. If you recollect styles.css is already referenced in our host page - index.html.

```
.boldClass{
    font-weight:bold;
}

.italicsClass{
    font-style:italic;
}

.colorClass{
    color:red;
```


Complete Tutorials

JavaScript tutorial

Bootstrap tutorial

Angular tutorial for beginners

Angular 5 Tutorial for beginners

Important Videos

The Gift of Education

Web application for your business

```
}
```

In **app.component.ts**, include a button element as shown below. Notice we have set the class attribute of the button element to 'colorClass'.

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

@Component({
    selector: 'my-app',
    template: `
                <button class='colorClass'>My Button</button>
              `
})
export class AppComponent {
}
```

At this point, run the application and notice that the 'colorClass' is added to the button element as expected.

Replace all the existing css classes with one or more classes

Modify the code in app.component.ts as shown below.

1. We have introduced a property 'classesToApply' in **AppComponent** class
2. We have also specified class binding for the button element. The word 'class' is in a pair of square brackets and it is binded to the property 'classesToApply'
3. This will replace the existing css classes of the button with classes specified in the class binding

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

@Component({
    selector: 'my-app',
    template: `
                <button class='colorClass' [class]='classesToApply'>My Button</button>
              `
})
export class AppComponent {
    classesToApply: string = 'italicsClass boldClass';
```

How to become .NET developer

Resources available to help you

Dot Net Video Tutorials

ASP.NET Core Tutorial

Angular 6 Tutorial

Angular CRUD Tutorial

Angular CLI Tutorial

Angular 2 Tutorial

Design Patterns

SOLID Principles

ASP.NET Web API

Bootstrap

AngularJS Tutorial

jQuery Tutorial

JavaScript with ASP.NET Tutorial

JavaScript Tutorial

Charts Tutorial

LINQ

LINQ to SQL

LINQ to XML

Entity Framework

```
}
```

Run the application and notice 'colorClass' is removed and these classes (italicsClass & boldClass) are added.

Adding or removing a single class : To add or remove a single class, include the prefix 'class' in a pair of square brackets, followed by a DOT and then the name of the class that you want to add or remove. The following example adds boldClass to the button element. Notice it does not remove the existing colorClass already added using the class attribute. If you change applyBoldClass property to false or remove the property altogether from the AppComponent class, css class boldClass is not added to the button element.

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

@Component({
    selector: 'my-app',
    template: `
                <button class='colorClass' [class.boldClass]='applyBoldClass'>My
Button</button>
            `
})
export class AppComponent {
    applyBoldClass: boolean = true;
}
```

With class binding we can also use ! symbol. Notice in the example below applyBoldClass is set to false. Since we have used ! in the class binding the class is added as expected.

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

@Component({
    selector: 'my-app',
    template: `
                <button class='colorClass' [class.boldClass]='!applyBoldClass'>My
Button</button>
            `
})
export class AppComponent {
    applyBoldClass: boolean = false;
```

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Dot Net Interview Questions

Slides

Entity Framework

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

```
}
```

You can also removed an existing class that is already applied. Consider the following example. Notice we have 3 classes (colorClass, boldClass & italicsClass) added to the button element using the class attribute. The class binding removes the boldClass.

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

@Component({
    selector: 'my-app',
    template: `
                <button class='colorClass boldClass italicsClass'
                    [class.boldClass]='applyBoldClass'>My Button</button>
              `
})
export class AppComponent {
    applyBoldClass: boolean = false;
}
```

To add or remove multiple classes use ngClass directive as shown in the example below.

1. Notice the colorClass is added using the class attribute
2. ngClass is binded to addClasses() method of the AppComponent class
3. addClasses() method returns an object with 2 key/value pairs. The key is a CSS class name. The value can be true or false. True to add the class and false to remove the class.
4. Since both the keys (boldClass & italicsClass) are set to true, both classes will be added to the button element
5. let is a new type of variable declaration in JavaScript.
6. let is similar to var in some respects but allows us to avoid some of the common gotchas that we run into when using var.
7. The differences between let and var are beyond the scope of this video. For our example, var also works fine.
8. As TypeScript is a superset of JavaScript, it supports let

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

@Component({
```

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Java Video Tutorials

Part 1 : Video | Text | Slides

Part 2 : Video | Text | Slides

Part 3 : Video | Text | Slides

Interview Questions

C#

SQL Server

Written Test

```
    selector: 'my-app',
    template: `
            <button class='colorClass' [ngClass]='addClasses()'>My Button</button>
        `
})
export class AppComponent {
    applyBoldClass: boolean = true;
    applyItalicsClass: boolean = true;

    addClasses() {
        let classes = {
            boldClass: this.applyBoldClass,
            italicsClass: this.applyItalicsClass
        };

        return classes;
    }
}
```

We have included our css classes in a external stylesheet - **styles.css**. Please note we can also include these classes in the styles property instead of a separate stylesheet as shown below.

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

@Component({
    selector: 'my-app',
    template: `
            <button class='colorClass' [ngClass]='addClasses()'>My Button</button>
        `,
    styles: [`
            .boldClass{
                font-weight:bold;
            }

            .italicsClass{
                font-style:italic;
            }

            .colorClass{
                color:red;
            }
            `]
})
```

```
export class AppComponent {
    applyBoldClass: boolean = true;
    applyItalicsClass: boolean = true;

    addClasses() {
        let classes = {
            boldClass: this.applyBoldClass,
            italicsClass: this.applyItalicsClass
        };

        return classes;
    }
}
```

2 comments:

Ravi September 6, 2017 at 2:28 AM

I tried to add multiple classes using above example, I ended up with following error.

AppComponent.ngfactory.js:16 ERROR TypeError: _co.addClasses is not a function
at Object.View_AppComponent_0._co [as updateDirectives] (app.component.html:7)
at Object.debugUpdateDirectives [as updateDirectives] (core.umd.js:13107)

Reply

Replies

Ravi September 6, 2017 at 3:17 AM

Enclosing the classes in single quotes fixed this for me as below.
let classes = {
'boldClass': this.applyBoldClass,
'italicsClass': this.applyItalicsClass
};

Reply

Enter your comment...

Comment as: toilati123vn@g ▼

Publish Preview

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Links to this post

Create a Link

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)