Creating Custom Directive in Angular 2

In this exercise, we will implement a custom directive with events. To implement the custom directive we need following objects

- Directive
  - The Directive object for executing the Custom Directive Decorator.
- ElementRef
  - The element reference in the DOM for which custom directive is used.
- HostListener
  - The event on the directive which will action an action on DOM.
- Input
  - To accept the input value for the property in the directive.
- Renderer
  - To render the element with the directive.

Step 1: In the Angular 2 application, add a new file of name custom.directive.ts in the **app** folder with the following code in it

```
import { Directive, ElementRef, HostListener, Input,Renderer } from
'@angular/core';

@Directive({
  selector: '[textColor]'
})
export class TextColorDirective {
  //1. The Element reference of Target
  constructor(private el: ElementRef, public renderer: Renderer) { }
  //2. Default Color
  @Input() defaultColor: string;
  //3. The TextColor input
  @Input('textColor') setHighlightedColor: string;

  //4. The MouseEnter Event for the Element
  //using Directive
  @HostListener('mouseenter') onMouseEnter() {
    this.setcolor(this.setHighlightedColor || this.defaultColor || 'red');
  }
  //4. The MouseEnter Event for the Element
  //using Directive
  @HostListener('mouseleave') onMouseLeave() {
    this.setcolor(null);
  }

  //5.Settig color of the element
  private setcolor(color: string) {
    //this.el.nativeElement.style.backgroundColor = color;
     this.renderer.setElementStyle(this.el.nativeElement,
       'backgroundColor', color);
  }
}
```

Read comments on the above code which will explain every step implemented in the code.

Step 2: In the **app** folder add a new file of name custom.directive.component.ts with the following code in it

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

@Component({
    selector: 'my-app',
    templateUrl: './app/democustomdirective.html'
})
export class ColorDirectiveComponent implements OnInit {
    color:string;
    constructor() { }

    ngOnInit() { }
}
```

The above is the Angular 2 component which declares the **color** property, this will be exposed to view.

Step 3: In the **app** folder add a new html file of name democustomdirective.html with the following markup in it

```
<h1>Custom Directive</h1>

<h4>Select the color</h4>
<div>
  <input type="radio" name="rdcolor" (click)="color='yellow'">Yellow
  <input type="radio" name="rdcolor" (click)="color='red'">Red
  <input type="radio" name="rdcolor" (click)="color='magenta'">Magenta
</div>
<p [textColor]="color">Set My Color!</p>

<p [textColor]="color" defaultColor="violet">
  Set My Color too!
</p>

<hr>
<p><i>Mouse over the following lines to see fixed colos</i></p>

<p [textColor]="'yellow'">Highlighted in yellow</p>
<p textColor="orange">Highlighted in orange</p>
```

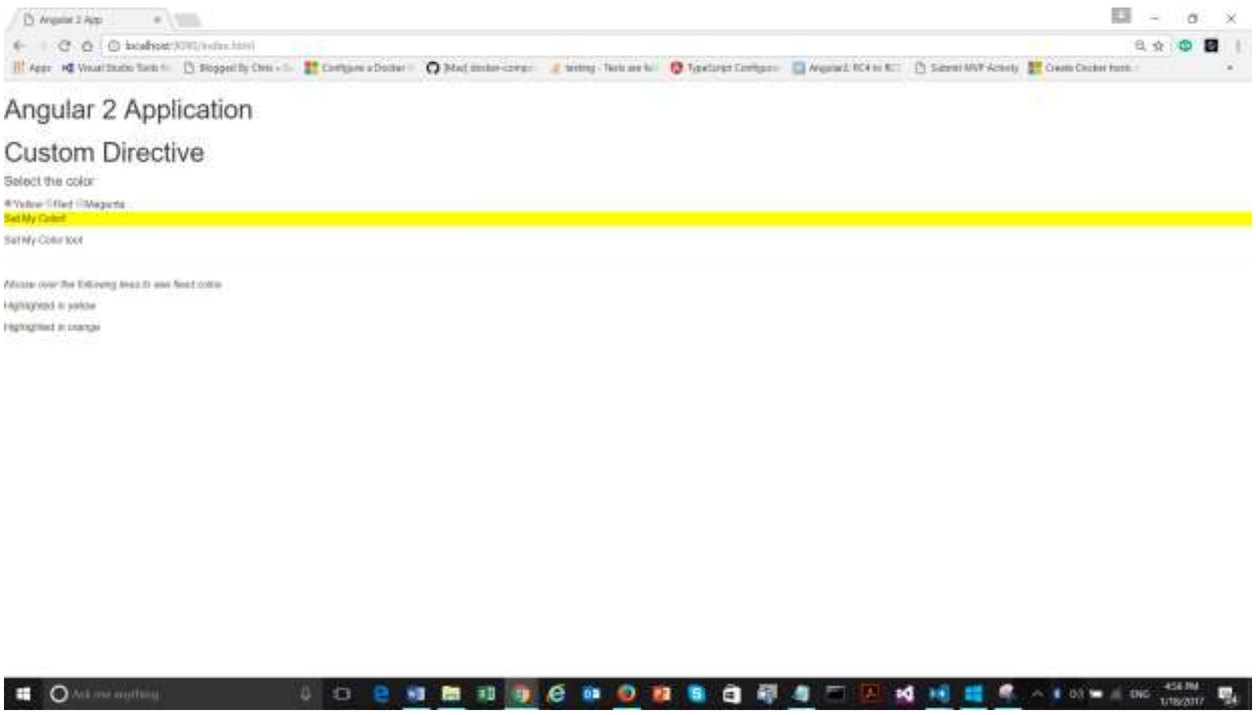
The above code contains markup where the **textColor** directive is applied

Step 4: Modify the main.ts to import the Directive and component defined in step 1 and 2

```
 import { NgModule } from '@angular/core';
import {FormsModule} from '@angular/forms';

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { BrowserModule } from '@angular/platform-browser';
import {TextColorDirective} from './custom.directive';
import {ColorDirectiveComponent} from './custom.directive.component';
//Regisyration of imported classes in bootstrap
@NgModule({
   imports: [BrowserModule,FormsModule],
   declarations: [TextColorDirective, ColorDirectiveComponent],
   bootstrap:[ColorDirectiveComponent]
})
export class AppModule { }
//The Call to Bootstrapper for all declararions/providers
platformBrowserDynamic().bootstrapModule(AppModule);
```

Run the application the result will be shown as in the following image



Angular 2 Application

Custom Directive

Select the color:

● Yellow ○ Red ○ Magenta

Set My Color!

Set My Color too!

Mouse over the following lines to see fixed colors

Highlighted in yellow

Highlighted in orange

Click on the Radio button with color name and move mouse cursor from the text, the text will be set with selected color as shown in the above image.