Angular Directives

Angular directives are used to extend the power of the HTML by giving it new syntax.
Each directive has a name — either one from the Angular predefined like ng-repeat, or a custom one which can be called anything.

There are three kinds of directives in Angular :

1. **Components** : Directives with a template. Components are the most common of the three directives.

2. **Structural directives** : Change the DOM layout by adding and removing DOM elements.
Structural Directives change the structure of the view.NgFor, NgIf, NgSwitch are the examples of structural directive.

3. **Attribute directives** : Change the appearance or behavior of an element, component, or another directive.
Attribute directives are used as attributes of elements.
The built-in NgStyle directive is the example of Attribute directive.

We can create our own custom attribute directive by following below steps.

Use below command to create custom directive

```
ng generate directive Name_Of_Directive
```

After this command one .ts file gets created.

Example:

```
ng generate directive mydir
```

After this command new file named as my-dir.directive.ts gets created.

This file contains one class with directive decorator as

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

@Directive({
  selector: '[appMyDir]'
})

export class MyDirDirective
{
          constructor()
         {
         }
}
```

We can write our logic inside the class which gets executed by the Angular when we use our directive.

If we want to change colour of text when mouse cursor is on the text and again change the colour when mouse cursor is removed then we can write below code.

```
import { Directive } from ‘@angular/core';

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

```
@Directive({
  selector: '[appMyDir]'            // Name of our directive.
})

export class MyDirDirective
{
        // Use dependency injection of ElementRef service to access DOM element

        constructor(private ele:ElementRef)
        { }

        @HostListener('mouseenter') onmouseenter()
        {
                 this.ele.nativeElement.style.background='blue';
        }

        @HostListener('mouseleave') onmouseleave()
        {
                this.ele.nativeElement.style.background='yellow';
        }
}
```

We can use above created directive in .html file as

<h1 **appMyDir**>Marvellous Infosystems</h1>