Attribute Directives

An Attribute directive changes the appearance or behavior of a DOM element.

## Directives overview

There are three kinds of directives in Angular:

1. Components—directives with a template.
2. Structural directives—change the DOM layout by adding and removing DOM elements.
3. Attribute directives—change the appearance or behavior of an element, component, or another directive.

Components are the most common of the three directives. You saw a component for the first time in the Getting Started.

Structural Directives change the structure of the view. Two examples are NgFor and NgIf. Learn about them in the Structural Directivesguide.

Attribute directives are used as attributes of elements. The built-in NgStyle directive in the Template Syntax guide, for example, can change several element styles at the same time

Directive:

1. import { Directive, ElementRef, HostListener } from '@angular/core';
3. @Directive({
4. selector: '[appHighlight]'
5. })
6. export class HighlightDirective {
7. constructor(private el: ElementRef) { }
9. @HostListener('mouseenter') onMouseEnter() {
10. this.highlight('yellow');
11. }
13. @HostListener('mouseleave') onMouseLeave() {
14. this.highlight(null);
15. }
17. private highlight(color: string) {
18. this.el.nativeElement.style.backgroundColor = color;
19. }
20. }

<p appHighlight highlightColor="yellow">Highlighted in yellow</p> <p appHighlight [highlightColor]="'orange'">Highlighted in orange</p>

Directive.ts

import { Directive, ElementRef, HostListener, Input } from '@angular/core'; @Directive({ selector: '[appHighlight]' }) export class HighlightDirective { constructor(private el: ElementRef) { } @Input('appHighlight') highlightColor: string; @HostListener('mouseenter') onMouseEnter() { this.highlight(this.highlightColor || 'red'); } @HostListener('mouseleave') onMouseLeave() { this.highlight(null); } private highlight(color: string) { this.el.nativeElement.style.backgroundColor = color; } }

 app.component.html

<h1>My First Attribute Directive</h1>

<h4>Pick a highlight color</h4>

<div> <input type="radio" name="colors" (click)="color='lightgreen'">Green

<input type="radio" name="colors" (click)="color='yellow'">Yellow

<input type="radio" name="colors" (click)="color='cyan'">Cyan

</div>

<p [appHighlight]="color">Highlight me!</p>

app.component.ts

export class AppComponent { color: string; }

service.ts

export class Hero { constructor(public name: string, public state = 'inactive') { } toggleState() { this.state = this.state === 'active' ? 'inactive' : 'active'; } }

Animation:

import { Component } from '@angular/core';  
  
import {  
 trigger,  
 state,  
 style,  
 animate,  
 transition  
} from '@angular/animations';  
  
**@Component**({  
 selector: 'pop-over',  
 templateUrl: './pop-over.component.html',  
 styleUrls: ['./pop-over.component.scss'],  
 animations: [  
 trigger('popOverState', [  
 state('show', style({  
 opacity: 1  
 })),  
 state('hide', style({  
 opacity: 0  
 })),  
 transition('show => hide', animate('600ms ease-out')),  
 transition('hide => show', animate('1000ms ease-in'))  
 ])  
 ]  
})  
export class PopOverComponent {  
  
 show = false;  
  
 constructor() { }  
  
 get stateName() {  
 return this.show ? 'show' : 'hide'  
 }  
  
  
 toggle() {  
 this.show = !this.show;  
 }  
  
}

<div [@popOverState]="stateName">  
 <p>Hello! I'm a helpful message.</p>  
</div>  
<button (click)="toggle()">Toggle PopOver</button>

animations: [  
 trigger('photoState', [  
 state('move', style({  
 transform: 'translateX(-100%)',  
 })),  
 state('enlarge', style({  
 transform: 'scale(1.5)',  
 })),  
 state('spin', style({  
 transform: 'rotateY(180deg) rotateZ(90deg)',  
 })),  
 transition('\* => \*', animate('500ms ease')),  
 ])  
]

Local Storage

* if(localStorage){
* // Store data
* localStorage.setItem("first\_name", "Peter");
* // Retrieve data
* alert("Hi, " + localStorage.getItem("first\_name"));
* } else{
* alert("Sorry, your browser do not support local storage.");
* }

localStorage.removeItem(key).

localStorage.clear()

HttpClient

import { BrowserModule } from '@angular/platform-browser';  
import { NgModule } from '@angular/core';  
import { HttpClientModule } from '@angular/common/http';

import { AppComponent } from './app.component';

@NgModule({  
 declarations: [  
 AppComponent  
 ],  
 imports: [  
 BrowserModule,  
 HttpClientModule  
 ],  
 providers: [],  
 bootstrap: [AppComponent]  
})  
export class AppModule { }

import { Component, OnInit } from '@angular/core';  
import { HttpClient } from '@angular/common/http';

@Component({  
 selector: 'app-root',  
 templateUrl: './app.component.html',  
 styleUrls: ['./app.component.css']  
})  
export class AppComponent implements OnInit {  
 title = 'app';  
 results = '';

constructor(private http: HttpClient){

}

ngOnInit(): void {  
 this.http.get('https://api.github.com/users/seeschweiler').subscribe(data => {  
 console.log(data);  
 });  
 }

}

--

const req = this.http.post('http://jsonplaceholder.typicode.com/posts', {  
 title: 'foo',  
 body: 'bar',  
 userId: 1  
 })  
 .subscribe(  
 res => {  
 console.log(res);  
 },  
 err => {  
 console.log("Error occured");  
 }  
 );