7강. Directive

DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

07 강.

Directive

❑ 1) directive (지시자)

* directive

- directive는 컴포넌트가 관리하는 template내 위젯의 속성에 관여한다.
- directive는 위젯에 필요한 속성을 정의함으로써 디자인, 동작 방식, 유효성 검증등 다양한 방식으로 사용된다.

* directive 종류

가. Angular가 제공하는 시스템 directive (Structural directive)

https://angular.io/guide/structural-directives

- ngClass
- ngStyle
- *ngIf, ngSwitch
- *ngFor

나. @Directive 장식자를 이용한 사용자 directive

Directive

```
ng g directive My // Creates MyDirective
```

❑ 2) ngClass directive

가. ngClass

CSS 클래스 이름을 설정하여 스타일을 적용할 수 있는 속성 directive.

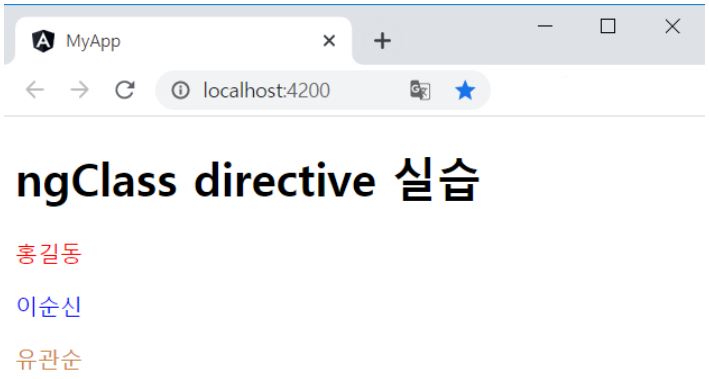
app.component.css

```
y-app > src > app > # app.compo
1   .red{
2       color: red;
3   }
4   .blue{
5       color: blue;
6   }
7   .my_color{
8       color: peru;
9   }
```

app.component.html

```
y-app > src > app > <> app.component.html > ...
1   <h1>{{title}}</h1>
2   <p ngClass="red">홍길동</p>
3   <p [ngClass]="color_blue">이순신</p>
4   <p [ngClass]="'my_color'">유관순</p>
```

```
export class AppComponent {
  title = 'ngClass directive 실습';

  color_blue = "blue";
}
```



❑ 3) ngStyle directive

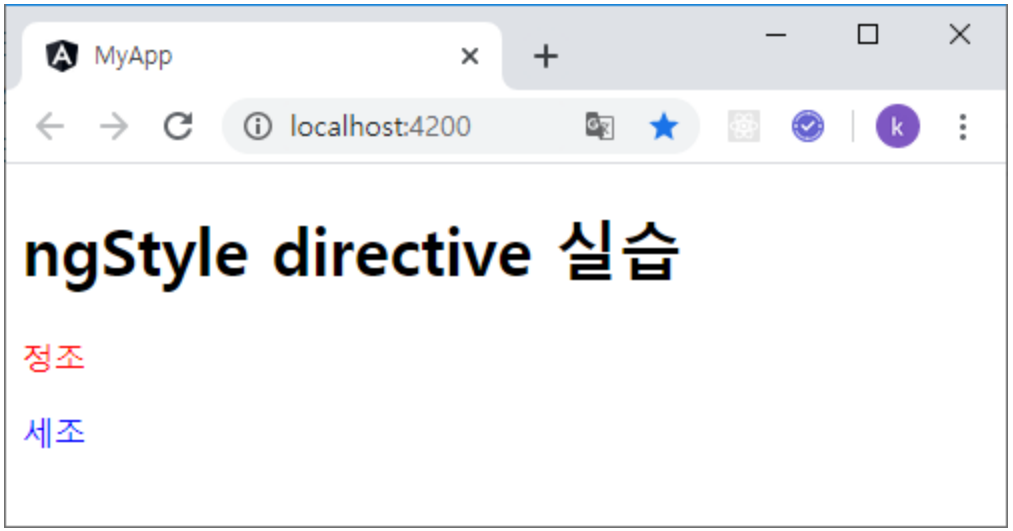
나. ngStyle

CSS 스타일을 직접 설정하여 스타일을 적용할 수 있는 속성 directive.

```
export class AppComponent {
  title = 'ngStyle directive 실습';

  my = "blue"
}
```

app.component.html

```
y-app > src > app > <> app.component.html > ...
1   <h1>{{title}}</h1>
2   <p [ngStyle]="{color:'red'}">정조</p>
3   <p [ngStyle]="{color:my}">세조</p>
```



❑ 4) *ngif directive

다. *ngIf

특정 요소를 보이거나 안보이게 설정하는 directive로서 조건이 true인 경우에 요소가 보이고 false인 경우에는 안보이게 된다.

app.component.html

```
y-app > src > app > app.component.html > ...
1   <h1>{{title}}</h1>
2   <h2>논리값 명시적 지정</h2>
3   <p *ngIf=true>Hello1</p>
4   <p *ngIf=false>Hello2</p>
5   <p *ngIf="true">Hello3</p>
6   <p *ngIf="false">Hello4</p>
7   <h2>속성 바인딩 지정</h2>
8   <p *ngIf="result1">world1</p>
9   <p *ngIf=result1>world2</p>
10  <p *ngIf="!result2">world3</p>
11  <p *ngIf="result3==10">world4</p>
```

```
export class AppComponent {
  title = 'ngIf directive 실습';

  result1 = true;
  result2 = 0;
  result3 = 10;
}
```

ngIf directive 실습

논리값 명시적 지정

Hello1

Hello3

속성 바인딩 지정

world1

world2

world3

world4

❑ 5) ngSwitch directive

라. ngSwitch, *ngSwitchCase, *ngSwitchDefault

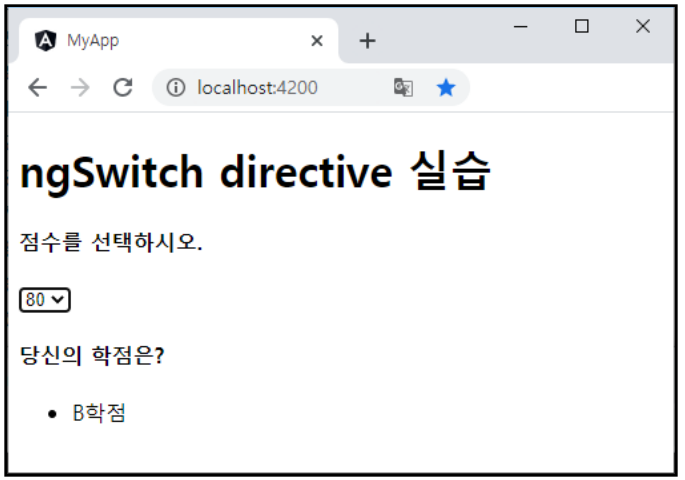
다중 비교시 사용되며 조건에 일치하는 요소만 선택하여 DOM에 포함시킨다.
(자바의 switch 문 기능과 동일)

app.component.html

y-app > src > app > app.component.html > ...

```
<h1>{{title}}</h1>
 <h4>점수를 선택하시오.</h4>
<select [(ngModel)]="grade">
    <option>90</option>
    <option>80</option>
    <option>70</option>
    <option>60</option>
</select>
<h4>당신의 학점은?</h4>
<ul [ngSwitch]="grade">
    <li *ngSwitchCase="90">A학점</li>
    <li *ngSwitchCase="80">B학점</li>
    <li *ngSwitchCase="70">C학점</li>
    <li *ngSwitchDefault>F학점</li>
</ul>
```

```
export class AppComponent {
  title = 'ngSwitch directive 실습';

  grade = 90
}
```



❑ ngModel의 사용

app.module.ts | app.component.ts | app.component.html

src > app > app.module.ts > ...

```ts
import { BrowserModule } from '@angular/platform-browse
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module'
import { AppComponent } from './app.component';
import {FormsModule} from '@angular/forms'; //import

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule //FormsModule추가
  ]
```

```ts

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'chul-app';
  grade=60;
}
```

```
</select>
학점은 {{grade}}<br><!--component의 grade사용-->
<h2>당신의 학점은?</h2>
<ul [ngSwitch]="grade"> <!--component의 grade사용-->
  <li *ngSwitchCase="90">A학점</li>
  <li *ngSwitchCase="80">B학점</li>
  <li *ngSwitchCase="70">C학점</li>
  <li *ngSwitchDefault>F학점</li>
</ul>
```

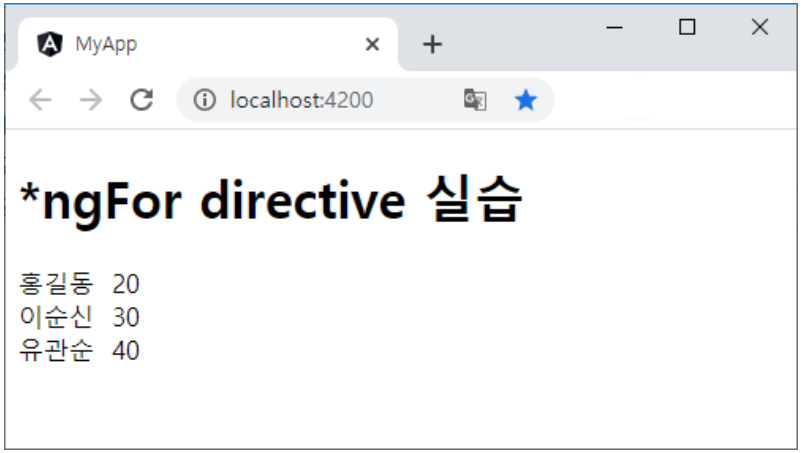
```
<h2>ngSwitch값을 동적으로 변경</h2>
<select [(ngModel)]="grade">
  <option>90</option>
  <option>80</option>
  <option>70</option>
  <option>60</option>
</select>
학점은 {{grade}}
```

❑ 6) *ngFor Directive

마. *ngFor

입력 받은 컬렉션 값을 반복하여 데이터 추출.
(자바의 for문과 동일 기능)

```
export class AppComponent {
  title = '*ngFor directive 실습';

  items:Object[]=[{name:"홍길동", age:20},
                  {name:"이순신", age:30},
                  {name:"유관순", age:40}]
}
```



component.ts app.component.html

app.component.html > ul > li

```
<h1>ngFor 디렉티브</h1>
<ul>
  <li *ngFor="let item of items; let idx= index">
    {{idx+1}}  {{item.name}}  {{item.age}}
  </li>
</ul>
```

❑ 6) *ngFor Directive

select 태그의 option값을 *ngFor 로 처리

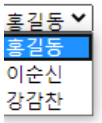
app.component.ts

```
ky > src > app > TS app.component.ts > ...
 1  import { Component } from '@angular/core';
 2
 3  @Component({
 4    selector: 'app-root',
 5    templateUrl: './app.component.html',
 6    styleUrls: ['./app.component.css']
 7  })
 8  export class AppComponent {
 9    title = 'inky';
10
11    names:string[]=["홍길동","이순신","강감찬"];
12  }
```

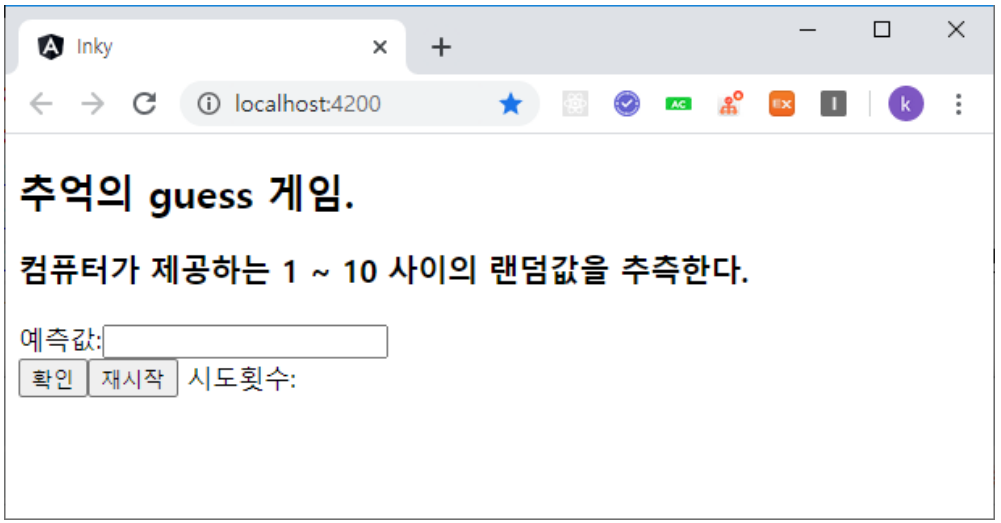
app.component.html

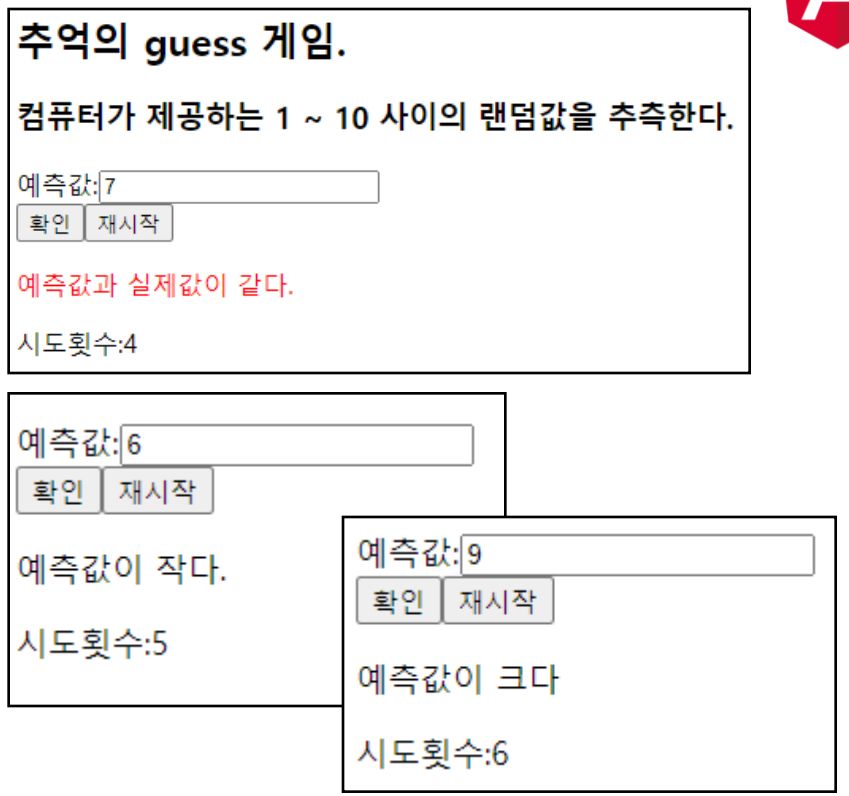
```
ky > src > app > <> app.component.html > ...
1  <h1>select의 option값을 *ngFor로 설정</h1>
2  <select>
3    <option *ngFor="let x of names">{{x}}</option>
4  </select>
```

select의 option값을 *ngFor로 설정



❑ 6] 실습 문제 1





실행순서

1. [재시작]버튼을 클릭한다. (초기화)
2. 예측값을 입력하고 [확인] 버튼을 클릭한다.

   가. 예측값 > 실제값 : “예측값이 크다” 메시지 출력
   나. 예측값 < 실제값 : “예측값이 작다” 메시지 출력
   다. 예측값 == 실제값 : “예측값과 실제값이 같다” 메시지 출력하고 글자색 red로 설정

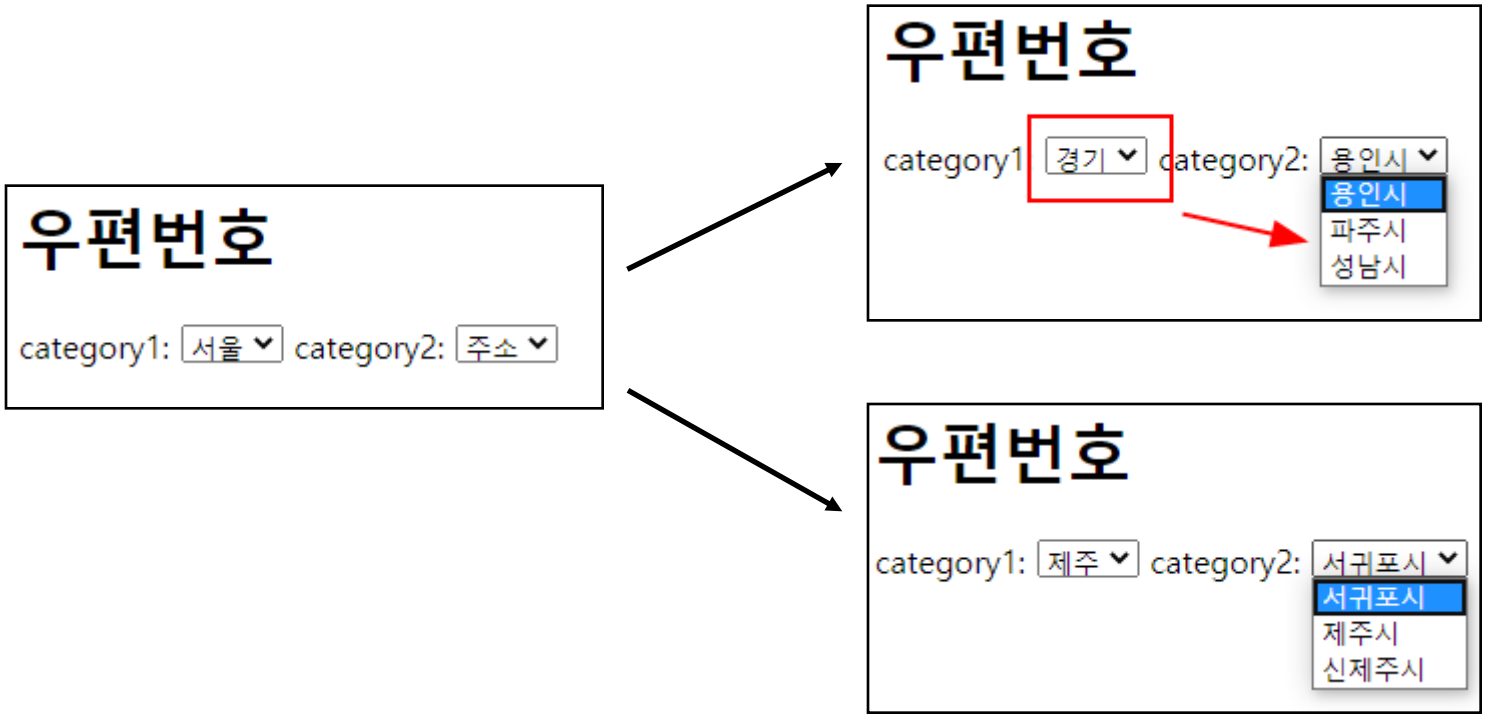
3. 시도횟수 출력한다.

* Math.floor(Math.random()*10+1) 이용한다.

❑ 6] 실습 문제 2

우편번호를 찾는 어플리케이션을 구현 하시오.

- category1 과 category2는 컴포넌트의 배열에 저장된 데이터로서 Category1의 값에 따라서 category2 값이 동적으로 변경된다.



❑ 7) 사용자 정의 directive

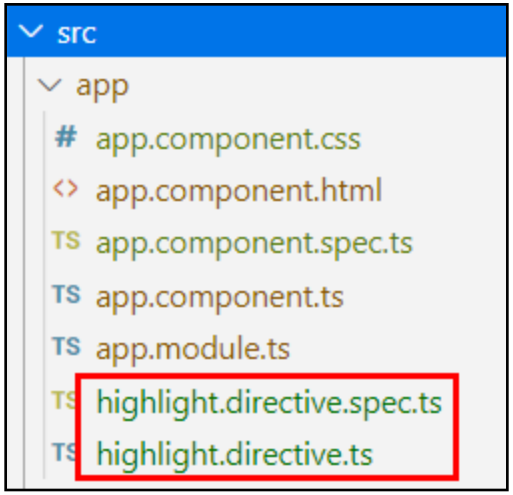
```
C:\angular_chul\chul-app>ng generate directive highLight
CREATE src/app/high-light.directive.spec.ts (237 bytes)
CREATE src/app/high-light.directive.ts (147 bytes)
UPDATE src/app/app.module.ts (478 bytes)
```

* 사용자 directive 생성 방법

```
PS C:\Angular_Study_chul_2\chul-app> ng g directive highlight
CREATE src/app/highlight.directive.spec.ts (236 bytes)
```

1. 사용자 directive를 생성한다.

https://angular.io/guide/attribute-directives



highlight.directive.ts

src > app > highlight.directive.ts > HighlightDirective > constructor

```ts
import { Directive, ElementRef, Renderer2 } from '@angular/core';

@Directive({
  selector: '[appHighlight]' //selector appHighlight을 html에서 사용
})
export class HighlightDirective {

  constructor(public eleRef:ElementRef, public render:Renderer2) {
    //ElementRef와 Renderer2를 사용
    render.setStyle(eleRef.nativeElement, "color", "red");
    render.setStyle(eleRef.nativeElement, "font-size", "20px");
  }

```

❑ 사용자 정의 directive app.module.ts에 등록

TS app.module.ts ×

src > app > TS app.module.ts > AppModule

```ts
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HighlightDirective } from './highlight.directive';//등록

@NgModule({
  declarations: [
    AppComponent,
    HighlightDirective//커스텀 디렉티브 등록
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
```

❑ 7) 사용자 정의 directive

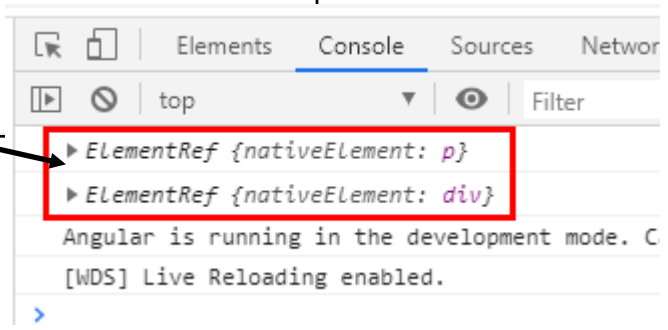
2. 생성된 directive를 실제 template 태그에 적용 시킨다.

app.component.html

```
<h1>{{title}}</h1>
<h3>다음 p태그에 사용자 directive 적용</h3>

<p appHighlight>홍길동</p>
<div appHighlight>이순신</div>
```

3. ElementRef와 Renderer2 객체를 사용하여 기능 구현 (생성자 주입)

```
export class HighlightDirective {

  constructor(public eleRef:ElementRef, public render:Renderer2){
    console.log(eleRef);
  }
}
```



❑ 사용자정의 directive highlight.directive.ts

TS highlight.directive.ts × | TS app.module.ts | <> app.component.html

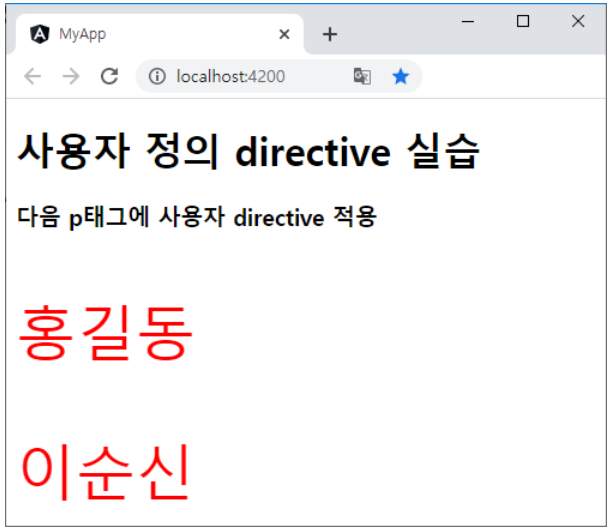
src > app > TS highlight.directive.ts > HighlightDirective > constructor

```
1   import { Directive, ElementRef, Renderer2 } from '@angular/core';
2
3   @Directive({
4     selector: '[appHighlight]'
5   })
6   export class HighlightDirective {
7   //element:태그의 의미
8     constructor( public eleRef:ElementRef, public render:Renderer2) {
9       console.log(eleRef.nativeElement, render);
10      render.setStyle(eleRef.nativeElement, "color", "red");//그리기
11      render.setStyle(eleRef.nativeElement, "font-size", "50px");
12    }
13
14  }
15
```

❑ 7) 사용자 정의 directive

4. Renderer2 객체의 setStyle 메서드 이용하여 CSS스타일 적용

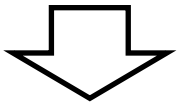
```
export class HighlightDirective {

  constructor(public eleRef:ElementRef, public render:Renderer2){
    render.setStyle(eleRef.nativeElement, "color","red");
    render.setStyle(eleRef.nativeElement, "font-size","50px");
  }
}
```



❑ 8] 사용자 정의 directive 이벤트

* 사용자 정의 directive 이벤트 처리 방법

```
@HostListener('이벤트명') 메서드명(){  }
```



```
@HostListener('mouseenter') onMouseEnter() {

}

@HostListener('mouseleave') onMouseLeave() {

}
```

❑ 사용자 정의 이벤트 처리

```
C:\angular_chul\chul-app>ng generate directive highLight
CREATE src/app/high-light.directive.spec.ts (237 bytes)
CREATE src/app/high-light.directive.ts (147 bytes)
UPDATE src/app/app.module.ts (478 bytes)
```

```
import { HighlightDirective } from './highlight.directive';

@NgModule({
  declarations: [
    AppComponent,
    HighlightDirective//등록
  ],
  imports: [
```

high-light.directive.ts

src > app > high-light.directive.ts > HighLightDirective > onMouseLeave

```
1   import { Directive, ElementRef, HostListener, Renderer2 } from '@angular/core';
2
3   @Directive({
4     selector: '[appHighLight]'
5   })
6   export class HighLightDirective {
7   //서비스에서 배웠던 의존성 주입
8   //element:태그의 의미
9     constructor(public eleRef:ElementRef, public render:Renderer2) { }
10
11    @HostListener("mouseover") onMouseOver(){
12      this.render.setStyle(this.eleRef.nativeElement, "color", "red");
13      this.render.setStyle(this.eleRef.nativeElement, "font-size", "50px");
14    }
15    @HostListener("mouseleave") onMouseLeave(){
16      this.render.removeStyle(this.eleRef.nativeElement,"color");
17      this.render.removeStyle(this.eleRef.nativeElement,"font-size");
18    }
```

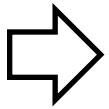
❑ 8] 사용자 정의 directive 이벤트

```
export class HighlightDirective {

  constructor(public eleRef:ElementRef, public render:Renderer2){

  }
  //이벤트 처리
  @HostListener('mouseover') onMouseOver(){
    this.render.setStyle(this.eleRef.nativeElement, "color","red");
    this.render.setStyle(this.eleRef.nativeElement, "font-size","50px");
  }
  @HostListener('mouseleave') onMouseLeave(){
    this.render.removeStyle(this.eleRef.nativeElement,"color");
    this.render.removeStyle(this.eleRef.nativeElement,"font-size");
  }
}
```

사용자 정의 directive 이벤트 실습

다음 p태그에 사용자 directive 적용

홍길동



사용자 정의 directive 이벤트 실습

다음 p태그에 사용자 directive 적용

홍길동

❑ 9] 사용자 정의 directive - @Input

* 사용자 정의 directive에서 @Input 사용

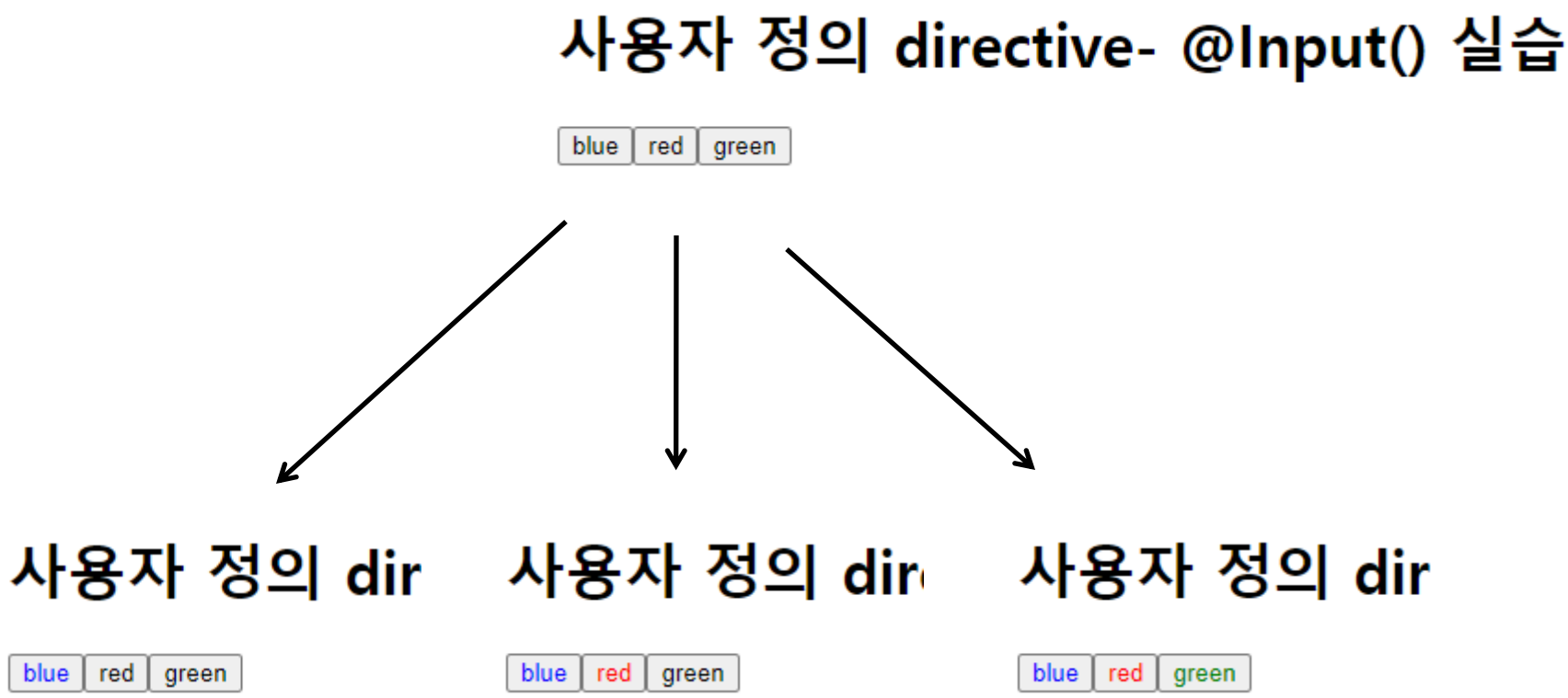
app.component.html

```
y-app > src > app > app.component.html > ...
1   <h1>{{title}}</h1>
2   <button appHighlight="blue">blue</button>
3   <button appHighlight="red">red</button>
4   <button appHighlight="green">green</button>
```

```
C:\angular_chul\chul-app>ng generate directive highLight
CREATE src/app/high-light.directive.spec.ts (237 bytes)
CREATE src/app/high-light.directive.ts (147 bytes)
UPDATE src/app/app.module.ts (478 bytes)
```

```
export class HighlightDirective {

  @Input("appHighlight") colorName:string;

  constructor(public eleRef:ElementRef, public render:Renderer2){}

  @HostListener('click') changeColor(){
    this.render.setStyle(this.eleRef.nativeElement, "color", this.colorName);
  }
}
```

❑ 9] 사용자 정의 directive - @Input



❑ 6] 실습 문제 3

* 마우스 오버(over)시 글자크기와 border 색상 변경하도록 구현 하시오.

커스텀 디렉티브 실습

- 

위험한 식탁
- 

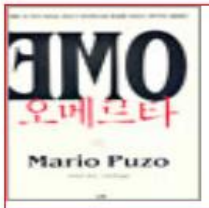
공부의 비결
- 

오메르타
- 
행복한 여행

커스텀 디렉티브 실습

- 

위험한 식탁
- 

공부의 비결
- 

오메르타
- 
행복한 여행

❑ 실습

```
C:\angular_chul\chul-app>ng generate directive red-border
CREATE src/app/red-border.directive.spec.ts (237 bytes)
CREATE src/app/red-border.directive.ts (147 bytes)
UPDATE src/app/app.module.ts (478 bytes)
```

TS red-border.directive.ts × | TS app.component.ts | <> app.component.html

src > app > TS red-border.directive.ts > RedBorderDirective

```ts
import { Directive, ElementRef, HostListener, Renderer2 } from '@angular/core';

@Directive({
  selector: '[appRedBorder]'
})
export class RedBorderDirective {

  constructor(public eleRef:ElementRef, public render:Renderer2) { }
  @HostListener("mouseover") onMouseOver(){
    this.render.setStyle(this.eleRef.nativeElement, "border", "2px solid red");
    this.render.setStyle(this.eleRef.nativeElement, "font-size", "20px");
  }
  @HostListener("mouseleave") onMouseLeave(){
    this.render.removeStyle(this.eleRef.nativeElement, "border");
    this.render.removeStyle(this.eleRef.nativeElement, "font-size");
  }
```

❑ 실습

-border.directive.ts | TS app.component.ts × | app.component.html

app > TS app.component.ts > AppComponent > books

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'chul-app';
  books=[
    {id:"p01", name:"위험한 식탁", price:2000, date:'20170202', img:'a.jpg'},
    {id:"p02", name:"공부의 비결", price:3000, date:'20170202', img:'b.jpg'},
    {id:"p03", name:"오메르타", price:2400, date:'20201202', img:'c.jpg'},
    {id:"p04", name:"행복한 여행", price:4000, date:'20180202', img:'d.jpg'},
    {id:"p05", name:"해커스 토익", price:1000, date:'20130202', img:'e.jpg'},
    {id:"p06", name:"도로 안내서 ", price:2000, date:'20150202', img:'f.jpg'}
  ]
```

rder.directive.ts | TS app.component.ts | app.component.html ●

> app.component.html > ul > li

```html
<ul>
  <li *ngFor="let book of books" appRedBorder>
    <img src="../../assets/image/{{book.img}}" width="100" height="100">{{book.name}}
  </li>
</ul>
```

수고하셨습니다.