angular: directivas y personalización

- componentes
  - directivas con plantilla asociada
  - son del tipo más común
- directivas de atributos
  - cambian la apariencia o el comportamiento de un elemento, componente u otra directiva
  - se usan como atributos de los elementos HTML (ngStyle)
- directivas estructurales
  - modifican la representación del DOM añadiendo o eliminando elementos
  - cambian la estructura de la vista (ngFor, ngIf)

directivas personalizadas

- para extender las capacidades del framework
- creamos nuestros propias directivas de atributo o directivas estructurales

directiva de atributo personalizada

- queremos crear algo así:

```
<div appDestacado>Oferta de la semana</div>
```

- usando el cli generamos el código genérico:

```
ng g[enerate] d[irective] directivas/destacado --spec=false
```

- obtendremos algo así y le añadiremos las líneas marcadas:

```
import { Directive } from '@angular/core';
@Directive({
    selector: '[appDestacado]'
})

export class DestacadoDirective {
    constructor( private el: ElementRef) {
         const nativeElement = el.nativeElement;
         nativeElement.style.fontSize = '28px';
         nativeElement.style.color = 'red';
         nativeElement.style.textAlign = 'center';
         nativeElement.style.border = '2px solid red';
    }
}
```

  - **ElementRef** nos da acceso al elemento al cual se ha de aplicar la directiva (angular lo inyecta mediante el mecanismo de inyección de dependencias)
- aceptando parámetros desde el componente .html

```
import {Directive, ElementRef, Input, OnInit} from '@angular/core';
@Directive({
    selector: '[appDestacado]'
})
```

```
export class DestacadoDirective implements OnInit {
    @Input('appDestacado') userColor: string;
    constructor(private el: ElementRef) { }
    ngOnInit(): void {
        const nativeElement = this.el.nativeElement;
        nativeElement.style.fontSize = '28px';
        nativeElement.style.color = this.userColor || 'red';
        nativeElement.style.textAlign = 'center';
        nativeElement.style.border = '2px solid ' + (this.userColor || 'red');
    }
}
```

- usamos el decorador **@Input** para recibir el parámetro que recibimos desde el html:

```
<div class="container main.container">
    <h1>Demo <small>directivas</small></h1>
    <hr>
    <app-clases></app-clases>
    <br><br>
    <div appDestacado>Oferta de la semana</div>
    <div [appDestacado]="'violet'">Oferta de la semana</div>
</div>
```

directiva estructural personalizada

```
ng g[enerate] d[irective] directivas/nult --skipTests
```

- en contraposición a las directivas de atributo, tenemos que acceder a través del constructor a la plantilla a la que la directive puede acceder, es decir, a la pantilla que posiblemente necesitemos modificar
- la directiva ha de tener un parámetro de entrada (a través de **@Input**)
- ejemplo:

```
<div class="container main.container">
    <h1>Demo <small>directivas</small></h1>
    <hr>
    <app-clases></app-clases>
    <br><br>
    <p *appMult="5">
        Saludos
    </p>
</div>
```

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

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

export class MultDirective {

    constructor(
        private templateRef: TemplateRef<any>,
```

```
        private viewContainer: ViewContainerRef) {

            @Input() set appMult(numero: number) {
                for (let i = 0; i < numero; i++) {
                    this.viewContainer.createEmbeddedView(this.templateRef);
                }
            }
        }
    }
```

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/development:angular:directivas

Last update: **09/02/2020 23:46**