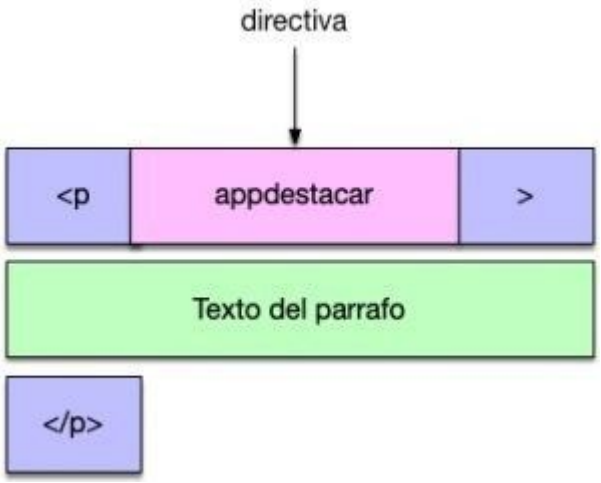
El concepto de Angular custom directive es uno de los conceptos que tenemos que conocer de Angular ya que nos permite extender la funcionalidad del framework de una forma muy sencilla. ¿Para que sirve un Angular custom directive? . Sirve como su nombre indica para construir una directiva personalizada es decir en vez de usar una directiva existente como ngClass o *ngIf podemos construir nuestras propias directivas y aumentar la extensibilidad de nuestro código.




Angular soporta dos tipos de custom directive  attribute directive y structure directive. Cada una de las cuales asume una responsabilidad diferente . Vamos a construir un ejemplo de cada una de ellas para poder entenderlas mejor.

# Angular Custom Directive (Attribute)

Lo primero que vamos a hacer es construir un custom directive de tipo atributo que nos ilumine el texto de cualquier etiqueta. El primer paso es usar el cliente de angular y hacer

ng generate directive destacar

Esto nos construira una nueva directiva con un selector por defecto “appDestacar” veamos el código:

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

@Directive({
  selector: '[appDestacar]'
})
export class DestacarDirective {

  constructor() { }

}
```

Acabamos de construir una directiva personalizada pero en estos momentos la directiva no hace nada . Es momento de modificar el constructor y añadir el código necesario para que la directiva se encarga de iluminar nuestro código.

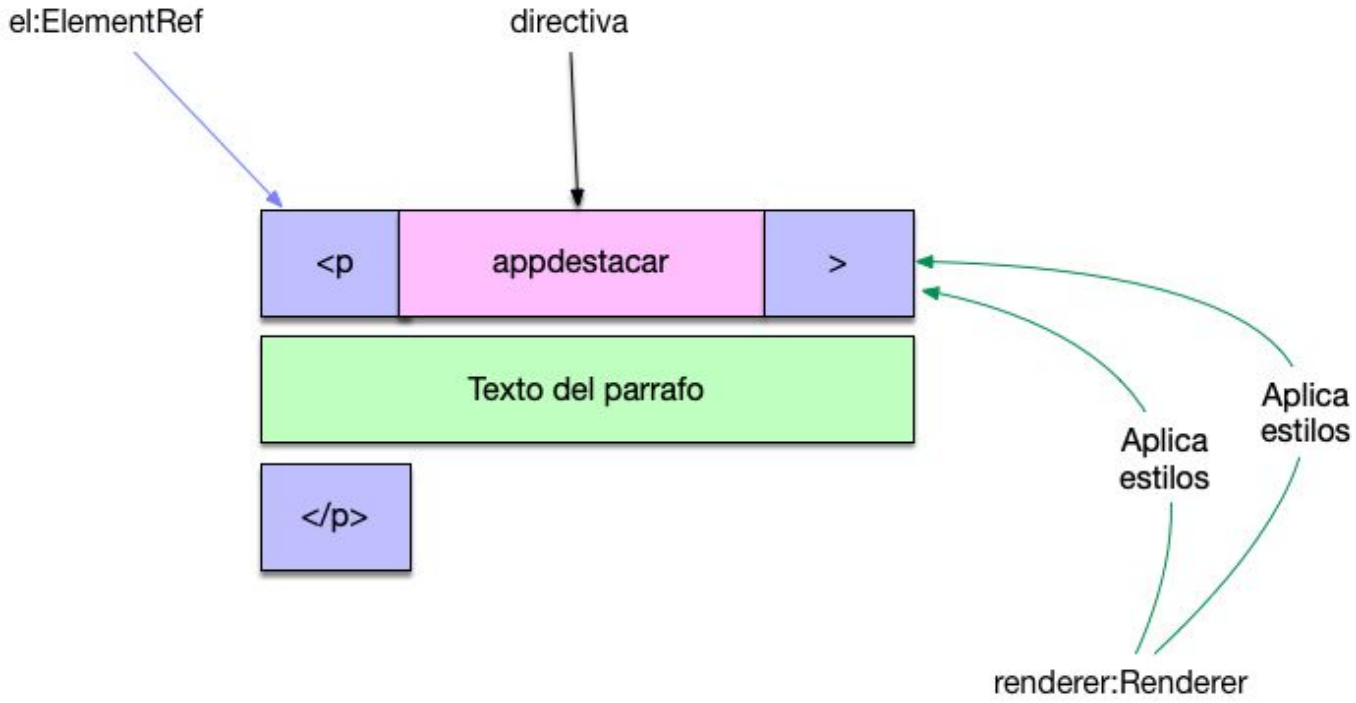
```
constructor(private el:ElementRef, private renderer:Renderer) {


    renderer.setElementStyle(el.nativeElement,'fontSize','50px');
    renderer.setElementStyle(el.nativeElement,'color','blue');
    renderer.setElementStyle(el.nativeElement,'border','2px solid
blue');

   }
```

En este bloque de código la directiva usa a nivel de constructor dos variables el y renderer . La primera de ellas nos da acceso de forma directa al elemento en el cual la directiva se va a aplicar. El segundo parámetro es el objeto renderer que permite aplicar estilos al

elemento que manipulados.




En este caso nos queda aplicar la directiva a cualquier etiqueta

<p appDestacar>
componente hola
</p>
Esto automaticamente nos renderizará el párrafo de otro color y con otro tamaño destacándolo en el texto.

componente hola

Acabamos de construir nuestra primera directiva a nivel de atributo.

# Angular Custom Directive (Structural)

Las directivas estructurales funcionan de otra forma ya que son capaces de cambiar por completo el elemento en el cual se aplican. Ejemplos de directivas estructurales son *ngFor y *ngIf. Vamos a proceder a construir una directiva estructural. Para ello ejecutamos el comando:

ng generate directive multiplicar

Esto nos genera una nueva directiva vacía que nosotros deberemos rellenar en este caso la diferencia es que vamos a acceder a través del constructor a la plantilla a la que la directiva puede acceder .Es decir a la plantilla interna de la etiqueta y que nosotros podemos necesitar modificar:

```
import { Directive, TemplateRef, ViewContainerRef, Input } from
'@angular/core';

@Directive({
  selector: '[appMultiplicar]'
})
export class MultiplicarDirective {

  constructor(private templateRef: TemplateRef<any>,
    private viewContainer: ViewContainerRef) {
```

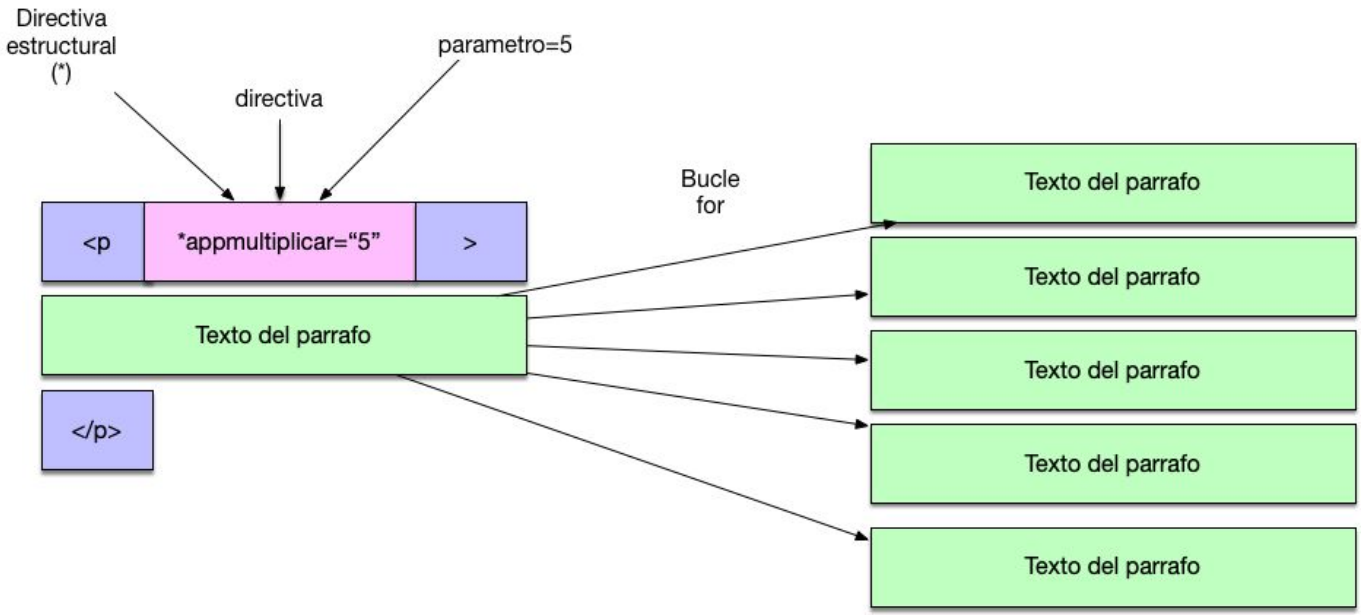
```
  }

  @Input() set appMultiplicar(numero: number) {
    for (var i = 0; i < numero; i++)
      // If condition is true add template to DOM
      this.viewContainer.createEmbeddedView(this.templateRef);

  }


}
```

En este caso lo que estamos haciendo son varias cosas en primer lugar permitir que la directiva tenga un parámetro de entrada utilizando @Input. Este parámetro de entrada permitirá que la directiva pueda variar su comportamiento de forma dinámica. En nuestro caso la directiva usa el parámetro para realizar un bucle for y apoyándose en la plantilla por defecto multiplicar su contenido n veces.




Vamos a ver como la directiva se puede aplicar en un componente:

<p *appMultiplicar=”5″>
componente hola
</p>
Como vemos es muy muy sencillo en cuanto este directiva se ponga en funcionamiento sobre el párrafo generará varias veces el mismo mensaje apoyándose en la plantilla que el párrafo incluye. Podemos ver el resultado en el navegador:

componente hola

componente hola

componente hola

componente hola

componente hola

El uso de Angular custom directives es de obligatorio conocimiento si queremos sacar todo el partido al framework.

1. Angular Services Singletons o no?
2. Angular ngIf, opciones y componentes
3. Angular router y su configuración
4. TypeScript Union Type y como usarlos
5. https://angular.io/