



```
function(b, c) {  
  var d = a(c.form.querySelectorAll('input[type=checkbox][name="' + b.el.name + '"']));  
  if (0 === d.index(b.el)) {  
    var e = d.filter(":checked").length;  
    return e >= b.arg || g.minChecked.replace("{count}", b.arg)  
  }  
},  
maxSelected: function(a) {  
  return null !== a.val ? a.val  
},  
minSelected: function(a) {  
  return null !== a.val && a.val  
},  
radio: function(b) {  
  var c = a(this.form.querySelector  
  return 1 === c  
},  
custom: function(a, b) {  
  var c = b.options.custom[a.arg],  
  d = new RegExp(c.pattern);  
  return d.test(a.val) || c.errorMessage  
},  
remote: function(a) {  
  a.remote = a.arg  
}
```

Formación Desarrollo Aplicaciones Web con Angular 5

14.- Trabajando con directivas

Angular | Trabajando con directivas

- Directivas

Podemos definir las directivas como marcadores de los elementos del DOM que le proporcionan un cambio de apariencia o comportamiento.

Existen tres tipos:

- Componente (selector)
- Directivas estructurales `*ngFor`, `*ngIf`, `*ngSwitch`
- Directivas atributo `ngStyle`, `ngClass`

Angular | Trabajando con directivas

- ngFor

```
<elemento *ngFor= "let variable of array;  
                    let i = variable-local">
```

Proporciona las siguientes variables locales:

index: devuelve el número de iteración que se esté procesando.

last: devuelve un boolean indicando si es o no el último elemento de la iteración.

even: devuelve boolean indicando si index tiene valor par.

odd: devuelve boolean indicando si index tiene valor impar.

Angular | Trabajando con directivas

- ngIf

<elemento *ngIf="expresion">

<elemento *ngIf="expresion; else #identificador">

<ng-template id="identificador">

<!--contenido-->

</ng-template>

es la sintaxis de local template

Angular | Trabajando con directivas

- ngSwitch

<elemento [ngSwitch]="variable">

 <elemento *ngSwitchCase="valor 1">

 ...

 <elemento *ngSwitchCase="valor n">

</elemento>

Angular | Trabajando con directivas

- ngStyle

`<elemento [ngStyle]= '{propiedadCSS: expresion, ...}'>`

Angular | Trabajando con directivas

- ngClass

`<elemento [ngClass]= '{'claseCSS': expresion, ...}'>`