



```
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

Formularios

Angular | Formularios

- Template driven

En el módulo

```
import { FormsModule } from '@angular/forms';
```

En la plantilla

```
<form #nombreForm="ngForm"  
  (ngSubmit)="metodo(nombreForm.value)">  
  <input .... name="propiedad" ngModel>
```

En la clase del componente

```
metodo(objetoForm) {  
  this.objeto = {  
    propiedad: nombreForm.controls.propiedad.value,  
  };  
}
```

Angular | Formularios

- Validación en la plantilla

Estados del formulario/campo

- pristine | dirty
- untouched | touched
- invalid | valid

#propiedad="ngModel" propiedad.estado //boolean

Angular | Formularios

- Validación CSS
 - ng-pristine | ng-dirty
 - ng-untouched | ng-touched
 - ng-invalid | ng-valid

```
ng-estado {  
    reglas css  
}
```

Angular | Formularios

- Model driven/Reactive forms

En el módulo

```
import {ReactiveFormsModule} from '@angular/forms';
```

En la plantilla

```
<form [formGroup]="nombreForm"  
      (ngSubmit)="metodo()">  
  <input .... formControlName="propiedad">
```

Angular | Formularios

Importaciones

```
import { FormGroup, FormControl, Validators } from '@angular/forms';
```

En la clase del componente

```
nombreForm: FormGroup;
```

```
ngOnInit() {  
    this.nombreForm = new FormGroup({  
        propiedad: new FormControl(valordeinicializacion),  
    });  
}
```

```
metodo() {  
    this.objeto = {  
        propiedad: this.nombreForm.get(propiedad).value,  
    };  
}
```

Angular | Formularios

- Validators

En la clase del componente

```
propiedad: new FormControl("", [Validators.validador]),
```

En la plantilla

```
... *ngIf="nombreForm.get('propiedad').estado..."...
```

```
... *ngIf="nombreForm.get('propiedad').hasError('validador')..."...
```

Angular | Formularios

- valueChanges

En la clase del componente

```
this.nombreForm.valueChanges.subscribe(objetoForm => {  
    // código actualización  
    setValue(valor, {emitEvent: false})  
});
```

En la plantilla

```
...value="{{nombreForm.value.propiedad}}"...
```