



```
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

## 4.- Introducción TypeScript

## Angular | Introducción TypeScript

- Sintaxis tipado estático de datos

```
let variable: tipo;
```

El tipado es también gradual, es decir si TypeScript no conoce el tipo intentará inferir su tipo:

```
let variable = 12; // inferencia a tipo number
```

## Angular | Introducción TypeScript

- Tipos de datos en variables

string

number

boolean

tipo[] ó Array<tipo>

object ó { }

any

void

## Angular | Introducción TypeScript

- Tipos de datos de unión

let resultado: `string | number`;

## Angular | Introducción TypeScript

- Tipos de datos en funciones

```
function suma (a: number, b: number): number {  
    return a + b;  
}
```

Parámetros opcionales

```
... mensaje?: string...
```

## Angular | Introducción TypeScript

- Tipos de datos genéricos

```
function devResultado<T> (a:T): T {  
    return a;  
}
```

Definición del tipo en la ejecución

```
let b = devResultado<string>('Aprobado');
```

## Angular | Introducción TypeScript

- Interfaces

```
interface Jugador{  
    nombre: string;  
    apellidos: string;  
    edad?: number;  
}
```

```
let variable: Nombre;
```

## Angular | Introducción TypeScript

- Clases

```
class datosJugador {  
    public nombre: string;  
    public apellidos: string;  
  
    constructor(nombre: string, apellidos: string){  
        this.nombre = nombre;  
        this.apellidos = apellidos;  
    }  
  
    marcarGol(): void {  
        this.goles++;  
    }  
}
```



## Angular | Introducción TypeScript

- Constructor breve

```
class datosJugador {
public nombre: string;
public apellidos: string;

    constructor(public nombre: string, public apellidos: string){
this.nombre = nombre;
this.apellidos = apellidos;
    }

    marcarGol(): void {
        this.goles++;
    }
}
```