

INTERFACES

INTERFACES

[enlace a la documentación](#)

Fundamentos

- Determina la “pinta” que tiene algo: con **interface**

```
interface Shape {  
  color: number;  
  name?: string;  
  // opcional  
}
```

```
interface Circle extends  
  Shape {  
    r: number;  
    center: Point;  
  }
```

- Una clase implementa una interfaz con **implements**

```
class CircleImpl implements Circle {  
  constructor(public color: number, public r: number, public center: Point) { }  
}  
let cir: Circle = new CircleImpl(222, 20, new Point(0, 0));
```

- Cualquier cosa **que se parezca** implementa la interfaz

```
let cir: Circle = { color: 222, name: 'hola', r: 20, center: new Point(0,0) };
```

INTERFACES

[enlace a la documentación](#)

- Se suelen usar para describir requerimientos a la hora de pasar objetos a una función. Indicaría **los requerimientos mínimos**:

Ejemplo:

```
interface LabelledValue {  
    label: string;  
}  
  
function printLabel (labelledObj: LabelledValue) {  
    console.log(labelledObj.label);  
}  
  
let myObj = {size: 10, label: "Size 10 Object"};  
printLabel(myObj);
```

Este ejemplo funciona:

aunque myObj tiene más campos que “label”, pero al menos cumple los requisitos de “LabelledValue” (eso es lo que indicamos, pues lleva el campo “label”).