

DECORADORES

# DECORADORES

- En java lo llaman “**anotaciones**”
- Un **decorador** es una función que modifica en runtime:

- ☐ una **clase**
- ☐ una **propiedad**
- ☐ un **método**
- ☐ una **función**
- ☐ un **parámetro**

```
@decorador
class MiClase {
    @decorador x: number;
    @decorador m1() { ... }
}
```

```
@decorador
function f(@decorador x: number) {
    ...
}
```

- Sintaxis:

**@decorador**

donde decorador sería el nombre de esa función modificadora.

```
function decorador (target) { /* hacer algo con target */ }
```

# DECORADORES

- Se puede configurar con propiedades usando una factoría de decoradores:

```
function factoria (valor: string) {  
    return function(target) { /* hacer algo con target */ }  
}
```

- Sintaxis:

**@factoria ( parámetros)**

donde factoría sería el nombre de esa función factoría y los parámetros, serían los parámetros que lleve la factoría:

```
@factoria ('hola')  
class MiClase { ... }
```

# DECORADORES

- ¡ NO VEREMOS CÓMO CREARLOS !
- SÓLO DEBÉIS COMPRENDER CÓMO FUNCIONAN PARA USARLOS EN ANGULAR,
  - PUES LOS COMPONENTES Y MÓDULOS DE ANGULAR LOS USAN.

## Ejemplo de componente Angular:

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

@Component({
  selector: 'todo-list',
  templateUrl: 'todo-list.component.html',
  styleUrls: ['todo-list.component.css'],
  moduleId: module.id
})
export class TodoListComponent { ... }
```