

## Guía práctica funciones

Nombre de la declaración de la función.

parámetros

```
1 function ordinary1(a, b, c) {  
2     // implementación de la función  
3 }  
4  
5 const ordinary2 = function (a, b, c)  
6     return a + b + c;  
7 };
```

Las llaves ( { } ) y todo lo que hay entre ellas son el **cuerpo de la declaración de la función**.

instrucción devuelve explícitamente un valor de la función

## Rol desempeñado por las funciones

```
1 function add(x, y) {  
2     return x + y;  
3 }  
4 console.log(add(x: 2, y: 3));  
5  
6 const objeto = { metodo1 : add }  
7  
8 console.log(objeto.metodo1(x: 4, y: 5));
```

invocada a través de una llamada de la función.

almacenado en una propiedad

invocado a través de una llamada de método.

## Funciones de flecha

```
1 const f = function (x, y, z) { return x };  
2  
3 const f1 = (x, y, z) => { return x };  
4  
5 console.log(f(x: 1, y: 5, z: 8));  
6 console.log(f1(x: 1, y: 5, z: 8));  
7
```

La función de flecha (más o menos) equivalente se ve de la siguiente manera. Las funciones de flecha son expresiones.

Se elimina los {} y el return

```
1  const f = (x, y, z) => x;
2
3  const f1 = x => x;
4
5  console.log(f(x: 1, y: 5, z: 8));
6  console.log(f1(x: 2));
7
```

Si una función de flecha tiene un solo parámetro y ese parámetro es un identificador, puede omitir los paréntesis alrededor del parámetro.

Usamos los paréntesis para decirle a JavaScript que el cuerpo es una expresión (un objeto literal) y no un bloque de código.

```
1  const func1 = () => ({a: 1});
2
3  const func2 = () => {a: 1};
4
5  console.log(func1(1, 5, 8));
6  console.log(func2(2));
```

{ a: 1 }

undefined

### Manejo de parámetros

Los parámetros son parte de una definición de función.

```
1  const sum = (x, y) => x + y;
2
3  console.log(sum(x: 3, y: 6))
```

Los argumentos son parte de una llamada de función

Los valores predeterminados de los parámetros especifican el valor que se utilizará si no se ha proporcionado un parámetro.

```
1  function f(x, y : number = 5 ) {
2      return [x, y];
3  }
4
5  console.log(f(x: 1))
6
```

De dos parámetros solo proporcionamos 2 argumentos