

# Sentencias de Control

## ▪ Funciones flecha

- Aparecen como novedades del ES6 (ECMAScript 2015) [1]
- Estructura menos redundante que la tradicional (funciones tradicionales)
- Sintáxis: `( param1, param2, ...rest ) => { expression; }`
  - Elimina la palabra reservada “function”
  - Se puede eliminar las llaves que delimitan el cuerpo, abriéndolo con una flecha
  - Se puede eliminar la palabra “return”

ANTES

```
function add ( a, b ) {
    return a + b;
}
```

AHORA

```
var add = ( a, b ) => a + b;
```

NOTA: Las funciones flecha son siempre anónimas

# Sentencias de Control

Se obvia los paréntesis si tenemos un solo parámetro

```
function duplicar(x){  
    return x * 2;  
}
```

```
var duplicar = x => x * 2;
```

Se indica paréntesis vacío si no tenemos parámetros

```
function saludar (){  
    return 'Hello World';}
```

```
var saludar = () => 'Hello World';
```

Si son varias instrucciones en el cuerpo, se debe incluir las llaves y el retorno

```
var calculoIVA = ( param1, param2 ) => {  
    var total, iva;  
    total = param1 + param2;  
    iva = total * 0.12;  
    result = total + iva;  
    return result;  
};
```

# Sentencias de Control

- Declaraciones

- var . - Declara una variable, opcionalmente inicializándola a un valor
- let.- Declara una variable local de ámbito de bloque, opcionalmente inicializándola a un valor.

- Condicionales

- If

```
if (condición1)
    sentencia1
else if (condición2)
    sentencia2
else if (condición3)
    sentencia3
...
else
    sentenciaN
```

# Sentencias de Control

- **for**

```
for ([expresion-inicial]; [condicion]; [expresion-final]) sentencia
```

- Ejemplo

```
for (var i = 0; i < 9; i++) {  
    n += i;  
    mifuncion(n);  
}
```

- **forEach**

- El método **forEach()** ejecuta la función indicada una vez por cada elemento del array.

```
const array1 = ['a', 'b', 'c'];  
  
array1.forEach(imprimir);  
function imprimir(item){  
    console.log(item)  
}
```

```
const array1 = ['a', 'b', 'c'];  
  
array1.forEach(element => console.log(element));  
  
// expected output: "a"  
// expected output: "b"  
// expected output: "c"
```

# Sentencias de Control

## ❖ Do...while [2]

```
let result = '';
let i = 0;

do {
    i = i + 1;
    result = result + i;
} while (i < 5);

console.log(result);
// expected result: "12345"
```

## ❖ Switch [2]

```
var foo = 0;
switch (foo) {
    case -1:
        console.log('1 negativo');
        break;
    case 0:
        console.log(0)
        break;
    case 1:
        console.log(1);
        break;
    case 2:
        console.log(2);
        break;
    default: console.log('default');
}
```