

## Funciones definidas por el usuario

- Una función es un grupo de instrucciones que ejecuta una tarea o calcula un valor.
- Se llaman a través de eventos o desde nuestros scripts
- No se distinguen entre procedimientos o funciones, siempre son funciones.

### 1ª forma de declarar funciones

```
function nombreFuncion([param1, param2 ... ]){  
    // instrucciones  
}
```

- Si la función va a devolver algún valor, se usa **return**

```
function nombreFuncion([param1, param2 ... ]){  
    // instrucciones  
    return valor;  
}
```

- Las funciones se pueden invocar (llamar) de la siguiente forma:

```
nombreFuncion();           // llama y ejecuta a las instrucciones de la función
```

```
let variable = nombreFuncion(); // devuelve un valor y se asigna a variable
```

✓ Ejemplo:

```
function saludar(a,b){  
    alert(`Hola ${a} y ${b}`);  
}
```

- Funciones sin parámetros

✓ Ejemplo:

```
function saludo(){  
    alert("Hola, buenas tardes ");  
}
```

```
saludo();
```

- Funciones con parámetros:

✓ Ejemplo:

```
function saludo(mensaje){  
    console.log(mensaje);  
}  
  
saludo("Hasta la vista");
```

- A partir de 2015, los parámetros pueden tener por defecto algún valor que toma en caso de no pasarlo en la llamada:

```
function triple(n=0)  
{  
    return n*3;  
}
```

**Ejemplo:**

```
let x=6, y=4, z="hola";  
console.log(triple(9));  
console.log(triple(x));  
console.log(triple(x+y));  
console.log(triple(x)+ triple(y));  
console.log(triple(triple(9)));  
console.log(triple(z));    ¿Qué devuelve?  
console.log(triple());    ¿Qué devuelve?  
console.log(triple(2,3));  ¿Falla?
```

## 2ª forma de declarar funciones.

- Se llaman funciones anónimas. Se declara una función sobre una variable:

```
let triple = function(n=0){  
  
    return n*3;  
  
}
```

- Ojo: Primero declarar la función, luego invocar:
- `let valor = triple();`  
`console.log(valor);`

**3ª forma de declarar funciones: Funciones flechas**

- Fueron introducidas en la versión ES6, en el 2015.
- No lleva la palabra function
- Se coloca la flecha entre los parámetros y la llave de apertura
- Se asigna a una variable
- Ojo: Primero declarar la función, luego invocar.
- ✓ Ejemplo. Pasar de una función tradicional a flecha.

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

```
let suma= function(a,b){  
    return a+b;  
}
```

```
let vsuma= (a,b) => {  
    return a+b;  
}
```

**Ejercicios**

1. Crear una función que se llame cuadrado. Que reciba un parámetro y devuelva el cuadrado del número.
  1. Con funciones tradicionales
  2. Con función anónima
  3. Con función flecha.
  4. Invocar a las funciones con ejemplos.
2. Crear una función flecha que escriba por pantalla el factorial de un número. El programa debe solicitar al usuario un número mayor o igual que cero. Comprobar que el número es válido.

Ejemplo: factorial(5) devuelve 120.
3. Escribir en una página web 500 números aleatorios del 1 al 10.000 y que al lado diga si es par o no. Hacer una función flecha que dado un número devuelva “par” si es par o “impar” si es impar.
4. Crear una función flecha que escriba por pantalla el sumatorio de un número. El programa debe solicitar al usuario un número mayor que 0. Comprobar que el número es válido.

Ejemplo: sumatorio de 5 es: 5+4+3+2+1 = 15

5. Crear una función flecha que permita dibujar una tabla en una página web. Como parámetros tendrá el número de filas y de columnas. Por defecto 10 y 4. También tendrá un tercer parámetro que permite indicar el color del borde, que por defecto será negro.

La tabla se crea con un borde de 1 pixel entre cada celda. El borde exterior tendrá 3 pixeles y será del mismo color que el borde de las celdas.

La tabla ocupará toda la anchura de la página

- 1) Llamar a la función para crear la tabla con borde negro, de 10 filas y 4 columnas
- 2) Llamarla de nuevo para generar la table de 20 filas, 10 col con borde negro
- 3) Dibujar 10 tablas de 5 filas y 4 columnas con borde verde.

6. Crear una página con un botón ‘¿Es fin de semana?’

Al pulsar el botón se abrirá una ventana de 400x400 que mostrará una imagen con carita sonriente si es fin de semana o una carita no sonriente si no es fin de semana.

Utilizar funciones flechas en un fichero js independiente.

Descargar dos imágenes para poder lanzar el ejercicio.

7. Crear una página que muestre un lista de elementos (<ul><li>). Para ello, nada más cargarla, se abrirá otra subventana de 700x300 con un mensaje al usuario para que introduzca un número entre el 1 y el 20. No hace falta validarlo.

Después, se cerrará la subventana y se llamará a la función flecha que será la encargada de escribir en la página inicial la lista con tantas filas como número haya indicado de la forma: Si el usuario introduce el número 3, se mostrará:

- 1
- 2
- 3