

# JS Functions



Las funciones nos permiten agrupar líneas de código con un nombre, para posteriormente, hacer referencia a ese nombre y realizar todo lo que se agrupe de esa función.

Para utilizar funciones hay que hacer 2 cosas:

- Declarar la función: es darle un nombre, definir los datos de entrada (opcional) y decirle las tareas (instrucciones) que realizará.
- Ejecutar la función: <<Llamar>> (Invocar) a la función, para que realice las tareas que contiene dentro. Se puede invocar la función la cantidad de veces necesaria en el programa principal.

```
> //Primer paso: Declarar la funcion
//En este caso, le damos el nombre -> SALUDAR

function saludar(){
  console.log("Bienvenidos al bootcamp 2024");
  console.log("Hoy vamos a ver: Funciones");
}

//Segundo paso: Llamar (invocar) a la funcion.
saludar();
```

---

Bienvenidos al bootcamp 2024

VM473:5

---

Hoy vamos a ver: Funciones

VM473:6

# Por qué declaramos funciones?



**Cuando un conjunto de instrucciones se va a usar muchas veces, nos permite reutilizar código, en vez de estar escribiendo muchas líneas iguales todo el tiempo.**

**Con las funciones, podemos separar el código en partes, para tener el programa organizado y en unidades más simples.**

**Al momento de trabajar en equipo, dividimos el trabajo en partes. Cada uno realizará una función y alguien o varios pueden utilizar tal función.**

**Al separar el código en fracciones, y con un nombre que describa qué hace esa función, el programa queda mucho más claro. Cuando ingrese un nuevo desarrollador, va a identificar fácilmente de qué se trata cada función.**



# El nombre de una función



El nombre de la función tiene que ser significativo de lo que va a hacer la función. También definiremos datos de entrada (si es necesario) y decirle que instrucciones va a tener dentro, qué es lo que va a hacer.

Algunos tips para el nombre de una función:

- Nombres simples, claros y descriptivos.
- Si es más de una palabra, utilizar nomenclatura camelCase.

# Repaso del alcance de las variables

**Var:** Alcance global

**Let:** Alcance de bloque

**Const:** Alcance de bloque

# Parámetros en las funciones

Las funciones se convierten mucho más flexibles al pasarle parámetros, que variables que existen solamente dentro de esa función. El valor de esas variables, es pasado desde el momento donde se ejecuta la función.

```
> //Se declara una funcion, que recibe un parametro llamado
  NOMBRE.
  function saludar(nombre){
    console.log("Hola " + nombre + "!");
  }

  //Cuando llamamos a la funcion, entre parentesis le pasamos el
  valor de ese parametro.
  saludar("Santiago");

  Hola Santiago!
```

# Más de un parámetro



Las funciones también pueden recibir más de un parámetro. En este caso debemos tener en cuenta que hay que respetar el orden en el que pasamos los valores, y en el que los usamos al llamarla.

```
function sumarDosNumeros(n1, n2){  
    console.log(n1+n2)  
}
```

```
sumarDosNumeros(5, 3)
```

```
//Imprime el numero 8 en consola
```



```
function presentacion(nombre, apellido, edad){  
  let frase = `Mi nombre es: ${nombre} ${apellido}, y tengo: ${edad} años`  
  console.log(frase)  
}  
  
presentacion("santiago", "piedrafita", 22)
```

# Return

Habitualmente buscamos es que esa función realice una tarea y nos devuelva la información al exterior de la función. Para así poder utilizarla por fuera de la misma, guardarla en una variable, etc.

Si yo tengo una función llamada “suma”, que recibe dos parametros (n1 y n2). Adentro, declaro la siguiente variable:

```
let suma = n1 + n2
```

Este resultado, no lo puedo usar por fuera de la función. Lo que hacemos en este caso es el uso de la palabra “*return*”.

Se utiliza la palabra clave **return**, que suele colocarse al final de la función. Cuando las instrucciones terminan, retornamos lo que necesitamos. También finalizamos la ejecución de la función, si es que existe código luego del return.

En los ejemplos anteriores, usábamos los resultados dentro de la misma función y solo para imprimirlos en consola. No podía salir al exterior.

```
function sumar(n1, n2){  
    return n1 + n2  
}  
  
let resultado = sumar(1, 2);  
console.log(resultado);  
//imprime el numero 3;
```



```
function nombreCompleto(nombre, apellido){  
    return nombre + " " + apellido  
}  
  
let resultado = nombreCompleto("santiago", "piedrafita");  
console.log(resultado);  
//imprime "santiago piedrafita";
```