

Lección 19

# Funciones I

¿Qué es  
una función?



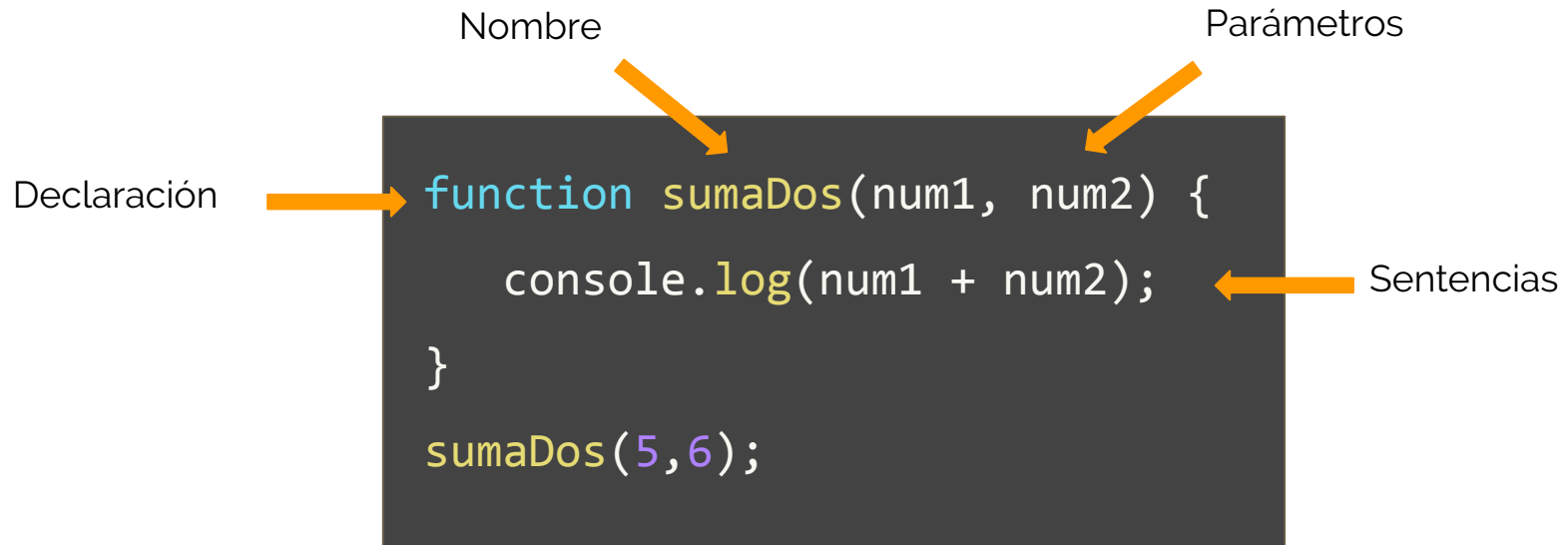
# Funciones

---

Una función es una serie de instrucciones definidas en un valor. Éstas nos permiten escribir el código una sola vez y luego reutilizarla múltiples veces. Ayudan a mantener el código DRY.

```
function saludo() {  
    console.log("Hola mundo!");  
}  
  
saludo();
```

# ¿Cómo es una función?



Una función es definida con la palabra reservada "function", a la que le llamamos declaración.

Los parámetros de una función se comportan como variables regulares dentro de la función, pero sus valores iniciales son dados al momento de llamar/invocar la función, no en la función por sí misma.

Ejecutar  
una función



## Ejecutar una función

---

También llamado *invocar*, *aplicar* o *llamar a una función*. así es como se le dice al momento donde deseamos utilizarla.

```
saludo();
```

La llamada a la función se realiza:

- ❑ Indicando el nombre de la función.
- ❑ Incluyendo los paréntesis del final.
- ❑ El carácter ; para terminar la instrucción.

# Argumentos



## Argumentos

---

Las funciones pueden o no necesitar valores para funcionar. Cuando una función necesita información adicional, es donde entran los argumentos.

Los argumentos son los valores que necesitan las funciones. Antes de utilizarlos, la función debe indicar cuántos argumentos necesita y cual es el nombre de cada argumento (a los que llamamos *parámetros*).

```
sumaDos(5,6);
```



## Argumentos

---

El orden de los argumentos es fundamental. EL primer dato de la llamada es el primer valor que espera la función, el segundo lo mismo y así sucesivamente.

```
function sumaDos(num1, num2) {  
    result = num1 + num2;  
    alert("El resultado es " + result);  
}  
sumaDos(5,6);
```

# Return



## Return

---

Las funciones no solamente puede recibir variables y datos, sino que también pueden devolver los valores que calculan. Para devolver valores dentro de una función se utiliza la palabra reservada **return**. Aunque las funciones pueden devolver valores de cualquier tipo, solamente pueden devolver un valor cada vez que se ejecutan.

```
function sumaDos(num1, num2) {  
    var result = num1 + num2;  
    return result;  
}  
var result = sumaDos(5,6);
```

# Scope en Funciones



## Scope

---

El scope (también llamado *alcance* o *ámbito*) es el espacio en el que las variables existen y al que pueden acceder enunciados u otras funciones. Si se define una variable dentro de una función, esa variable sólo existe para esa función, y otras funciones no pueden acceder a su valor (a menos que lo reciban como un argumento).

# Scope

---

Esto quiere decir que las variables creadas dentro de ellas, incluidos los parámetros, son “locales” a la función. Es decir, no se puede acceder a ellos fuera de la función.

Local



```
function sumaDos(num1, num2) {  
  var result = num1 + num2;  
  return result;  
}
```

```
var result = sumaDos(5,6);
```



Global