

Funciones de JS

newtab dice

Funciones!

Aceptar

Funciones Nativas

En JavaScript existen lo que se llaman funciones que si bien ya veremos cómo se crean, vamos a usar algunas que vienen armadas en JavaScript.

Una función son unos bloques de código que nos permiten ejecutar ciertas instrucciones, como por ejemplo “mostrar un mensaje”.

Si escribimos en la consola la función **alert("Mi primer función");** vamos a ver que aparece un mensaje en el navegador, mostrando el mensaje que escribimos adentro de los paréntesis.

```
> alert("Mi primer función");  
>
```



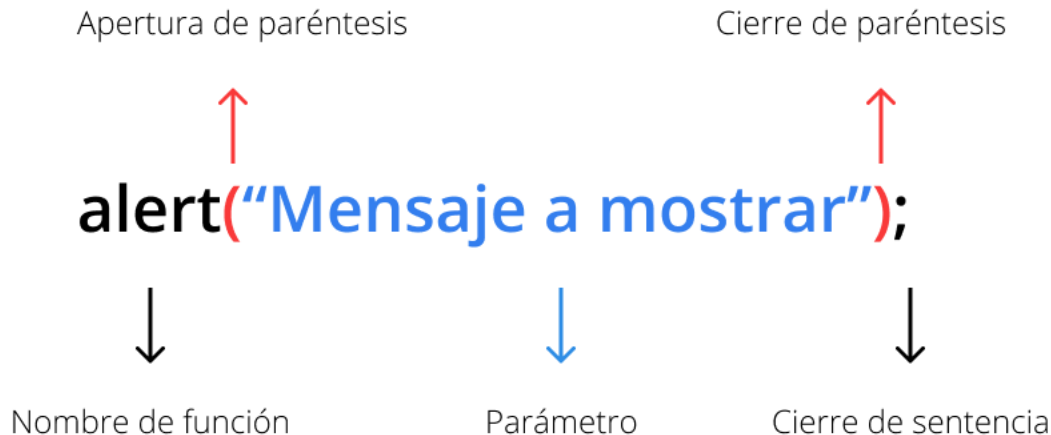
newtab dice
Mi primer función

Aceptar

Funciones Nativas

Vamos a decir que para ejecutar una función tenemos que escribir:

1. El nombre de la función
2. Abrir y cerrar paréntesis
3. Dentro de esos paréntesis un “parámetro” o algo que recibe la función.



Funciones Nativas

Así como el **alert**, existe otra función que se llama **console.log()** y lo que hace es mostrar un mensaje pero desde la consola:

```
console.log("Mensaje por consola");
```

> console.log("Mensaje por consola");
Mensaje por consola <u>VM111:1</u>
< undefined
>

Ejecuto la función

Muestra el mensaje

Lo que devuelve la función (aún no nos importa)

Ya vamos a ir viendo como estas funciones nos permiten interactuar con el usuario e incluso nos sirve a nosotros/as desarrolladores para entender qué está pasando con nuestro código.

Experimento 1

1. Crear una variable llamada **edad**
2. Agregarle como valor la edad que tengas
3. Crear una variable llamada **resultado**
4. Agregarle a la variable resultado, el valor de edad más 10 años más
5. Mostrar por consola ese resultado

¡Hacé el experimento y mirá el
paso a paso en la siguiente diapositiva!

Experimento 1

Resultado 1

```
> let edad;  
< undefined  
  
> edad = 28;  
< 28  
  
> let resultado;  
< undefined  
  
> resultado = edad + 10;  
< 38  
  
> console.log(resultado);  
38  
< undefined  
> |
```

Resultado 2

```
> let edad = 28;  
< undefined  
  
> let resultado = edad + 10;  
< undefined  
  
> console.log(resultado)  
38  
< undefined  
>
```

Resultado 3

A la hora de programar, pueden existir cientos de resultados distintos, lo importante es llegar a la solución y resolver el problema que se nos presenta, así que si lo resolviste distinto, ¡no hay problema!

Interacción con el usuario

Existe otra función que nos permite empezar a interactuar con el usuario y pedirle que nos de sus datos. Esta función se llama **prompt()**

La función prompt() le muestra un mensaje al usuario, y nos deja guardar la respuesta de un usuario en una variable. Podríamos pedirle al usuario que ingrese su nombre, y luego guardarlo en una variable.

Ingrese su nombre



Hola Juan!

```
> prompt("Ingrese su nombre")
```

Ejecutamos la función

newtab dice

Ingrese su nombre

Aceptar Cancelar

El usuario ve este mensaje

```
< "Juan"
```

Este es el resultado

Paso a Paso

Si bien en el ejercicio anterior le pudimos pedir información al usuario, no hicimos nada con esa información.

Vamos a crear un programa que le pida al usuario su nombre, y luego le mostraremos un cartel que lo salude.

1. Pedirle al usuario que ingrese su nombre usando un **prompt**.
2. Guardar la respuesta del nombre en una variable.
3. Mostrar un mensaje usando un **alert**, que salude al usuario.

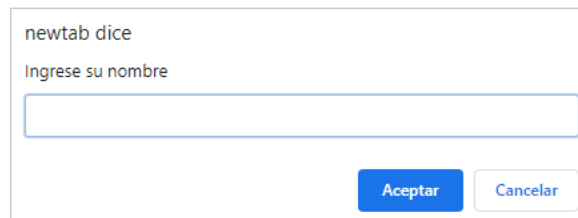
1

```
> let nombre = prompt("Ingrese su nombre");  
alert("Hola " + nombre);
```

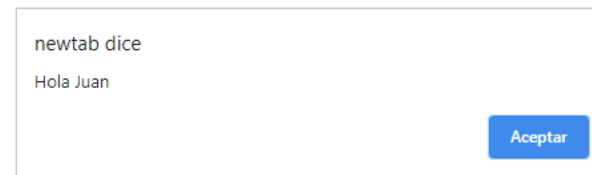
La respuesta de lo que ingresa el usuario, se guarda en la variable **nombre**

Hacemos un alert concatenando la palabra "Hola " y luego lo que hay en la variable de nombre.

2



3



Experimento II

1. Pedirle al usuario que ingrese su edad.
 2. Mostrar un cartel que muestre cuál va a ser su edad dentro de 20 años.
-

1. Pedirle al usuario que ingrese su edad.
2. Mostrar un cartel que muestre cuál es el doble de su edad.