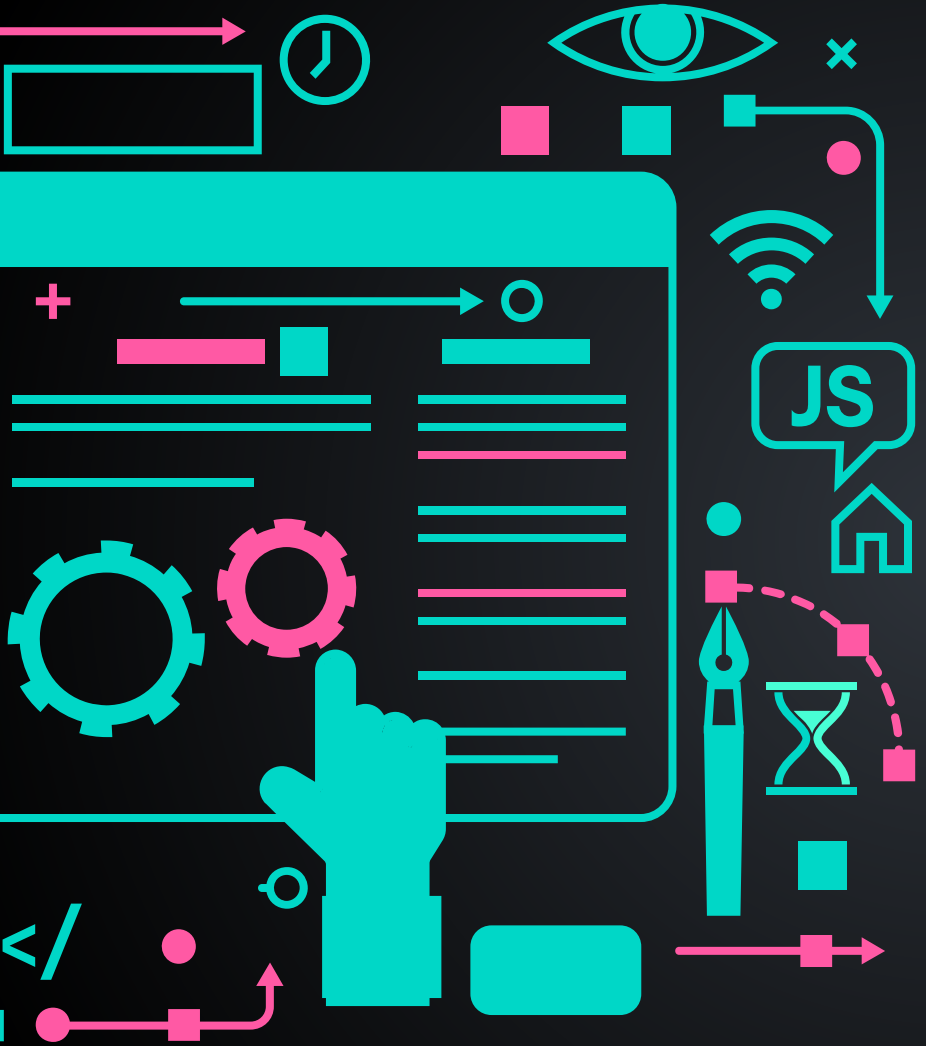


M1S5 - Fundamentos de JavaScript

AGENDA DEL DÍA

Mindset Digital ¿Cómo mantenernos motivados?	09:00 am	11:15 am	**Break** 20 minutos de descanso
Variables y Constantes Declaración y tipos de datos	09:35 am	11:35 am	**Ejercicio guiado** Programamos juntos
Coerción de datos Conversión de tipos de datos	09:55 am	12:05 am	**Ejercicio individual** Primer ejercicio individual
Operadores aritméticos y lógicos Operadores	10:15 am	12:35 pm	**Ejercicio Individual** Segundo ejercicio individual
Estructuras de control Controles de flujos	10:35 am	01:05 pm	**Ejercicio en equipo** Ejercicio en breakout rooms
Funciones Métodos en JS	10:55 am	01:45 pm	**Tareas y retos** Cierre y encuesta

Variables y Constantes

```
var ciudad = "Barcelona";
var animal = "Lobo";
var pais = "Argentina";
var edad = 35;
var apellidoPaterno = "Sánchez";
```

Nos permiten guardar información para hacer uso de su contenido.

Variable

```
var nombre = "Pedro";
var edad = 35;
```

Constante

```
const fechaDeNacimiento = "12/Diciembre/1990";
const LugarDeNacimiento = "Monterrey";
```

Tipos de datos

- **STRING**
- **NUMBER**
- **BOOLEAN**
- **UNDEFINED**
- **NULL**

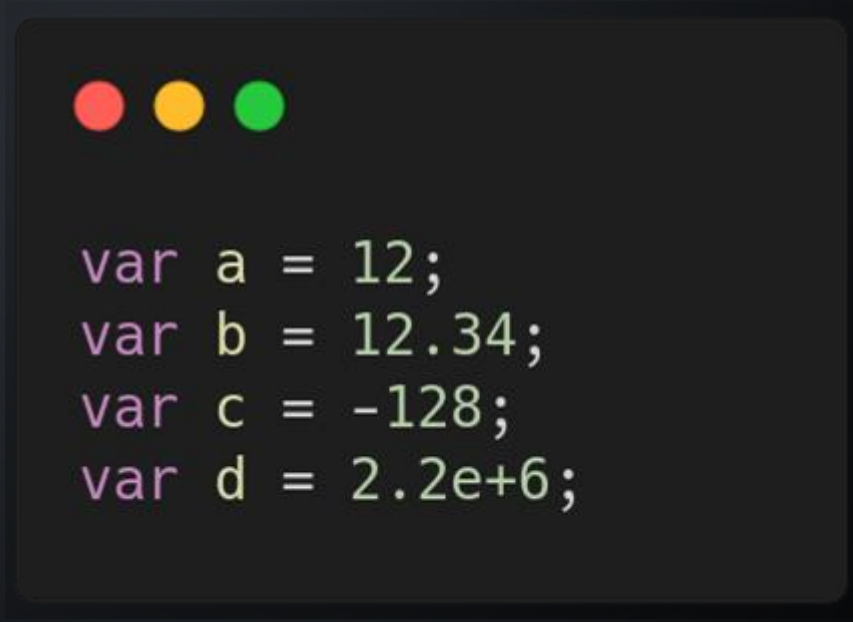
Tipo de dato utilizado para almacenar cadenas de texto.

```
var a = "Hello World!";
var b = 'Hello World';
var c = " I'm a student ";
var d = 'Yo escribí "Bienvenidos" ';
```

Tipos de datos

- **STRING**
- **NUMBER**
- **BOOLEAN**
- **UNDEFINED**
- **NULL**

Tipo de dato utilizado para almacenar números enteros, con punto decimal, positivos, negativos y con notación científica.



Tipos de datos

- STRING
- NUMBER
- BOOLEAN
- UNDEFINED
- NULL

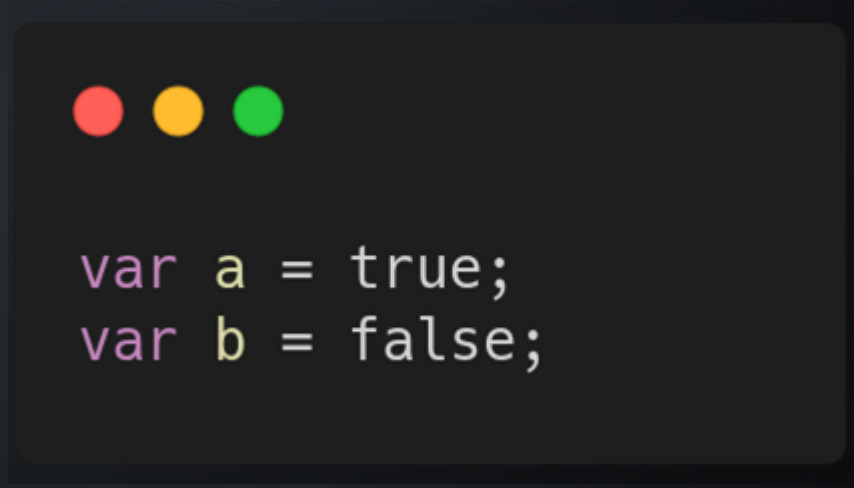
Casos especiales

```
var a = 7 / 0; //Infinity
var b = "Hola" / 4; //NaN
```

Tipos de datos

- STRING
- NUMBER
- BOOLEAN
- UNDEFINED
- NULL

Tipo de dato utilizado para almacenar valores booleanos.



Tipos de datos

- **STRING**
- **NUMBER**
- **BOOLEAN**
- **UNDEFINED**
- **NULL**

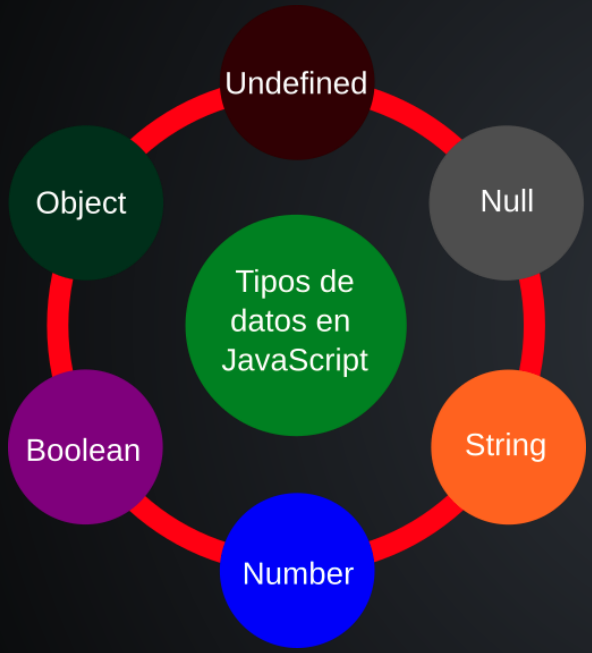
Cuando una variable es declarada sin asignarle un valor, por default su valor es *undefined*.

Tipos de datos

- STRING
- NUMBER
- BOOLEAN
- UNDEFINED
- NULL

Tipo de dato que significa que no tiene valor.

DUDAS



Coerción de datos

- Conversión de los tipos de datos.
- Variación de los tipos en tiempo de ejecución.

Coerción de datos

- STRING
- NUMBER
- BOOLEAN

```
123 + ''; //Implícito
String(123); //Explícito
String(3.14); // '3.14'
String(true); // 'true'
String(undefined); // 'undefined'
String(null); // 'null'
```

Coerción de datos

- STRING
- NUMBER
- BOOLEAN

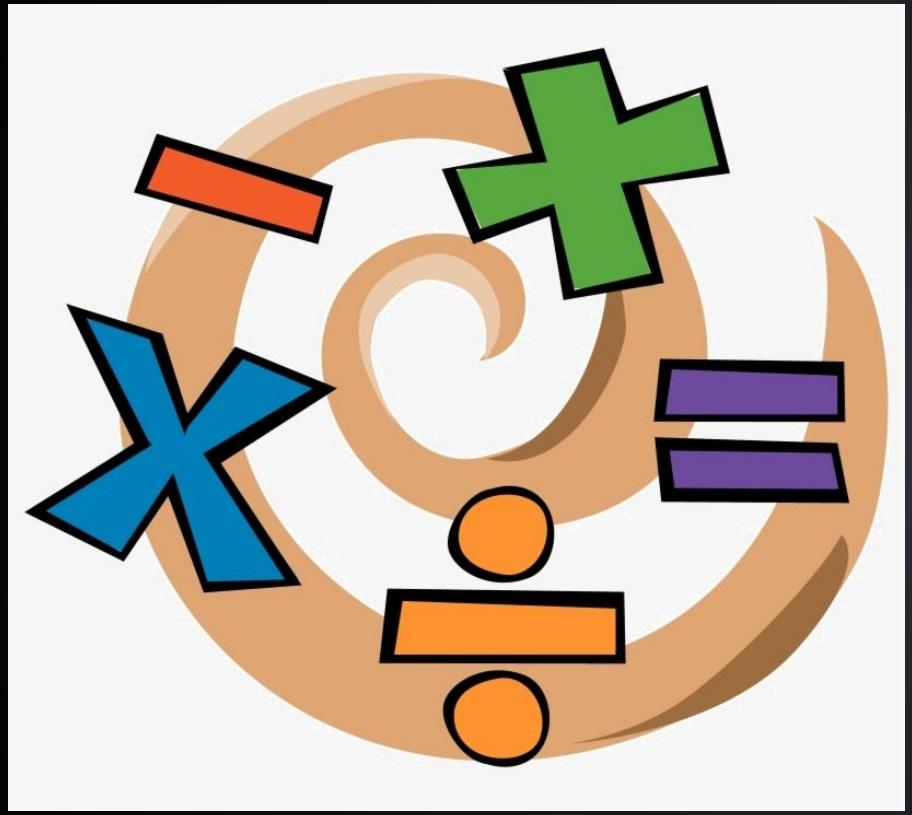
```
+ '123'; //Implícito
5 - '3'; //Implícito
3 * '3'; //Implícito
Number('   10   '); // 10
Number('-10'); // -10
Number('Hola'); // NaN
```

Coerción de datos

- STRING
- NUMBER
- BOOLEAN

```
!!2; //Implícito
Boolean(1); // Explícito
Boolean('Hola'); // true
Boolean(''); // false
!!1; // true
!!0; // false
```

DUDAS



Operadores aritméticos y lógicos

- Los operadores permiten realizar operaciones aritméticas o lógicas.

Operadores aritméticos

Operador	Descripción	Ejemplo
+	Adición	var res = 4 + 3
-	Substracción	var res = 4 - 3
*	Multiplicación	var res = 4 * 3
/	División	var res = 4 / 2
%	Residuo de división	var res = 4 % 2
++	Incremento	var res = 4++
--	Decremento	var res = 4--

Operadores lógicos

Operador	Descripción	Ejemplo
==	**Igualdad:** Devuelve *true* si ambos operandos son iguales.	3 == 3 3 == '3'
!=	**Desigualdad:** Devuelve *true* si ambos operandos no son iguales.	3 != 4
===	**Estrictamente igual:** Devuelve *true* si los operandos son iguales y del mismo tipo.	3 === 3
!==	**Estrictamente desigual:** Devuelve *true* si los operandos no diferentes y/o de diferente tipo.	3 !== '3'
>	**Mayor que:** Devuelve *true* si el operando de la izquierda es mayor que el de la derecha.	4 > 3

Operadores lógicos

Operador	Descripción	Ejemplo
>=	**Mayor o igual:** Devuelve *true* si el operando de la izquierda es mayor o igual que el de la derecha.	4 >= 4
<	**Menor:** Devuelve *true* si el operando de la derecha es menor que el de la izquierda.	12 < 15
<=	**Menor o igual:** Devuelve *true* si el operando de la izquierda es menor o igual que el de la derecha.	15 <= 15
&&	**And:** Devuelve *true* si ambas condiciones se cumplen.	2 > 1 && 0 < 1
||	**Or:** Devuelve *true* si al menos una de las condiciones se cumple.	2< 1 || 0 < 1

Truthy y Falsy

Valores Falsy		Valores Truthy	
Valor	**Descripción**	**Valor**	**Descripción**
false	Falso	true	Verdadero
0	Cero	1	Uno
'' ó ""	String vacío	'0' ó 'false'	String no vacío.
null	Nulo	[]	Arreglo vacío.
undefined	Indefinido	{}	Objeto vacío.
NaN	NaN	function (){}	Función vacía.

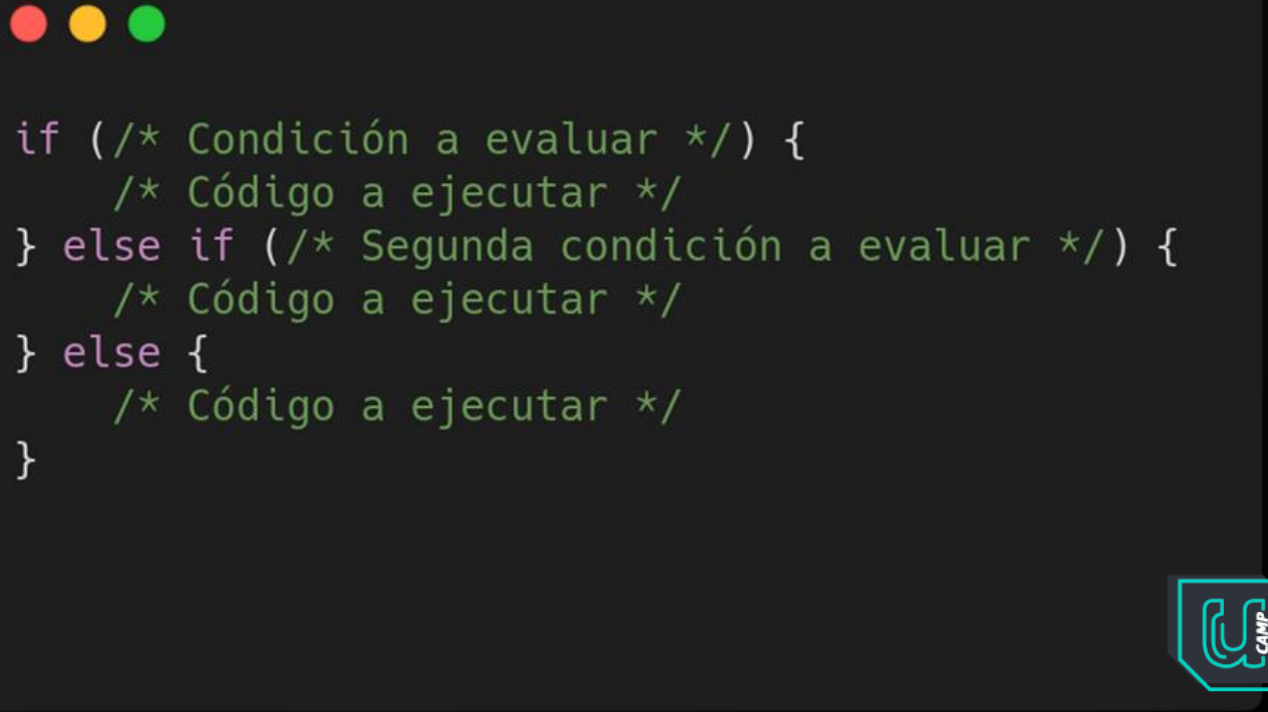
DUDAS

Estructuras de control

Nos permiten controlar el flujo de nuestros programas o scripts.

Condicionales

- IF/ELSE
- SWITCH



```
if (/* Condición a evaluar */) {
    /* Código a ejecutar */
} else if (/* Segunda condición a evaluar */) {
    /* Código a ejecutar */
} else {
    /* Código a ejecutar */
}
```

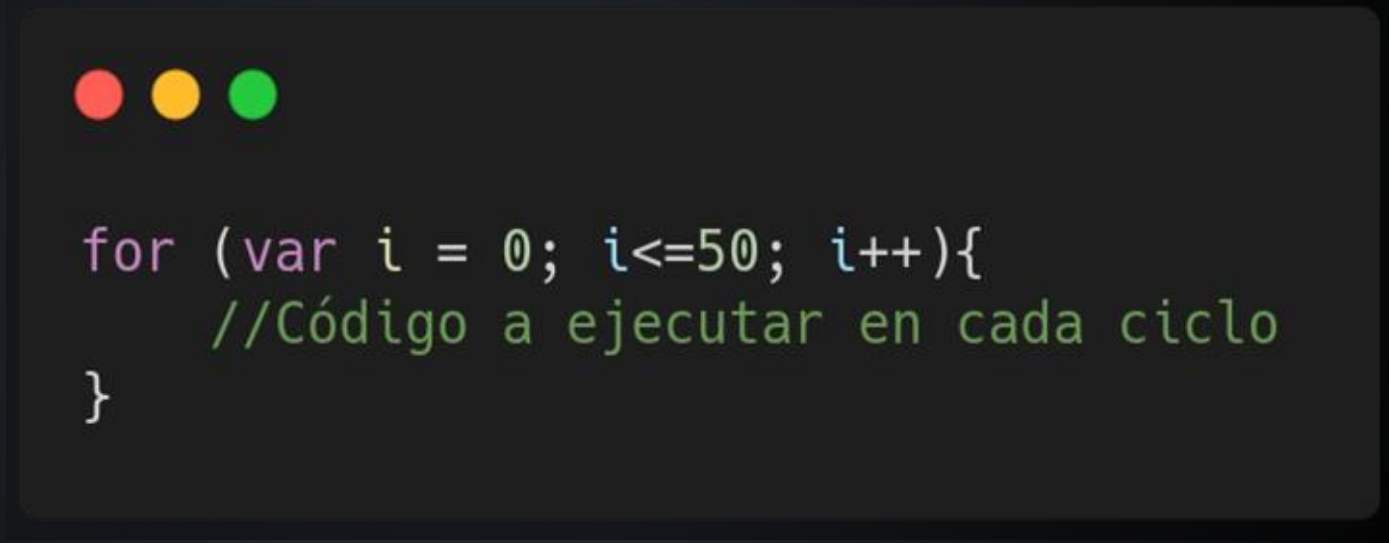
Condicionales

- IF/ELSE
- SWITCH

```
switch (/* Expresión a evaluar */) {
    case a:
        /* Código a ejecutar */
        break;
    case b:
        /* Código a ejecutar */
        break;
    default:
        /* Código a ejecutar */
}
```

Bucles

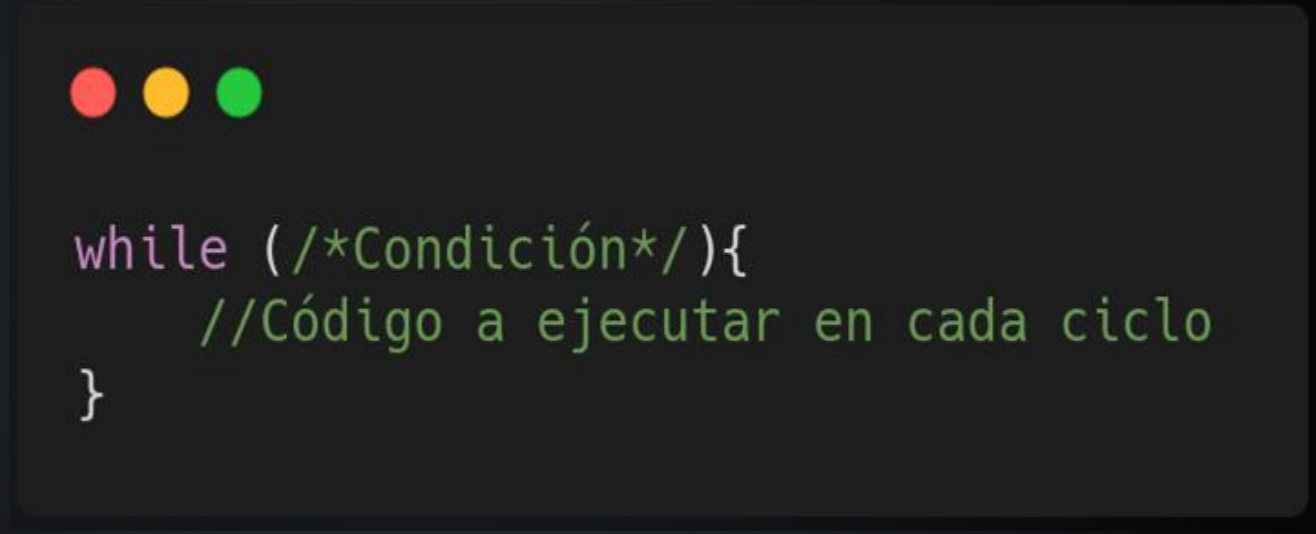
- FOR
- WHILE



```
for (var i = 0; i<=50; i++){
    //Código a ejecutar en cada ciclo
}
```

Bucles

- FOR
- WHILE



```
while (/*Condición*/){
    //Código a ejecutar en cada ciclo
}
```

DUDAS

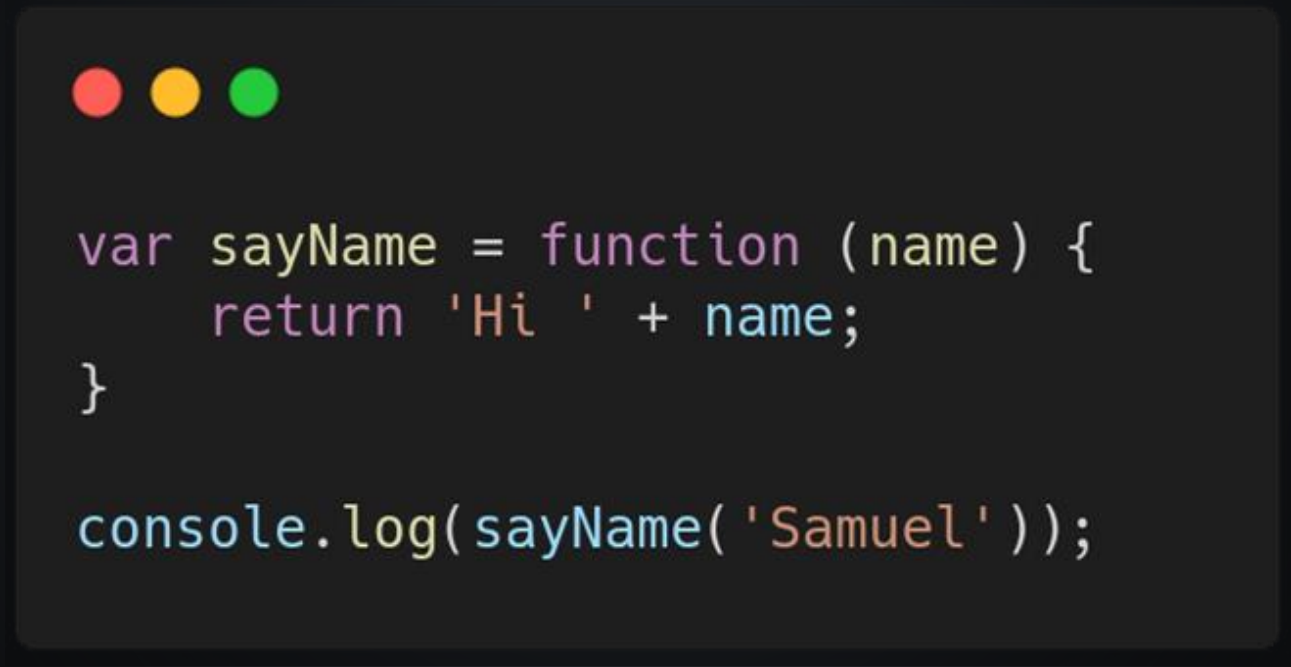
Funciones

Una función es un conjunto de sentencias que realizan una tarea.

Declaración de función

```
function myFunction(parameter1, parameter2) {
    //Código a ejecutar
}
```

Expresión de función



```
var sayName = function (name) {
    return 'Hi ' + name;
}

console.log(sayName('Samuel'));
```

Expresión ejecutada inmediatamente

```
(function (){
    var name = 'Samuel';
    console.log(name);
})();
```

DUDAS

EJERCICIO GUIADO

1. Crear una función de JS que imprima los primeros “n” números pares

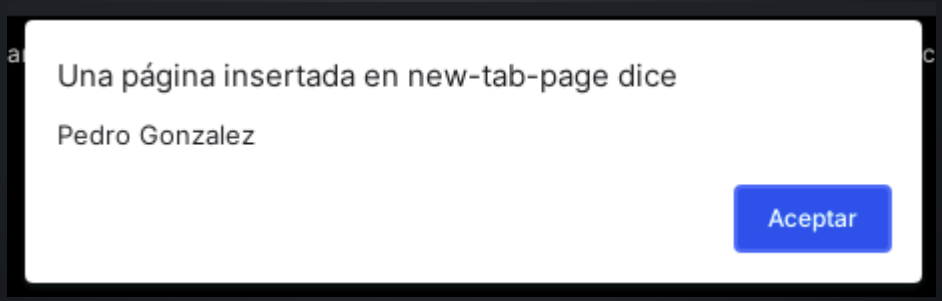
```
pares(6) //2, 4, 6, 8, 10, 12
pares(3) //2, 4, 6
pares(10) //2, 4, 6, 8, 10, 12, 14, 16, 18, 20
```

DUDAS

EJERCICIO INDIVIDUAL

1. Crear un método que reciba nombre y apellido y en un alert regrese la concatenación del nombre completo.

   nombreCompleto(“Pedro”, “Gonzalez”)



DUDAS

EJERCICIO INDIVIDUAL

1. Crear un método que reciba tu nombre y tu edad y determine si eres mayor de edad para solicitar un permiso de conducir.

permiso(“Roberto”, 15) // “Roberto, no puedes solicitar el permiso”
permiso(“Mario”, 36) // “Mario, puedes solicitar el permiso”

DUDAS

EJERCICIO EN GRUPOS

1. Crear un método que reciba 5 calificaciones y retorne el promedio. Y si es mayor a 70 imprima que aprobo/ si es menor, que no aprobó

promedio(70, 80, 80, 90, 60) // Aprobado: 76.
promedio(70, 50, 75, 70, 60) // No Aprobado: 65.

DUDAS

RETOS

1. **Practicar la parte de algoritmos de js con:**
   Hacker Rank:
   https://www.hackerrank.com/domains/algorithms
   Code Wars:
   https://www.codewars.com/