



Argentina
programa
4.0

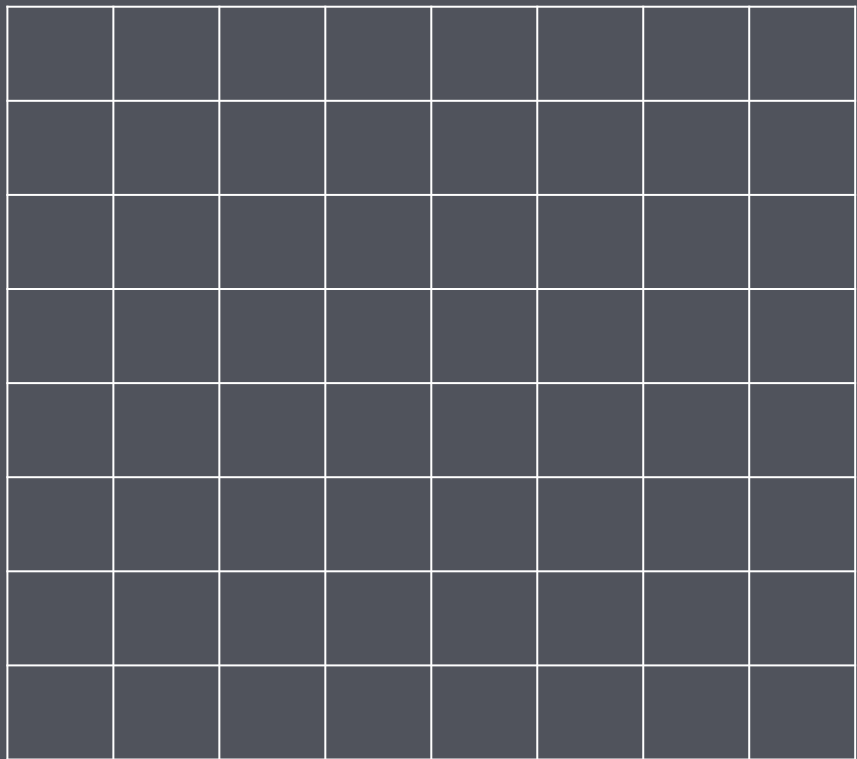
Introducción a Javascript

Programa “Introducción a la Programación Web”



Introducción a Javascript

- Alcance de las variables
- Estructuras de control
- Funciones



Javascript - Variables

Existen tres sentencias básicas que debemos conocer:

- **var**: declara variables con alcance local o global. Puede ser inicializada con algún valor.
- **let**: declara variables de alcance local, con ámbito de bloque. Puede ser inicializada con algún valor.
- **const**: los valores asignados a las constantes no pueden cambiarse, ni tampoco se pueden re-declarar. Tienen ámbito de bloque.

Javascript - Var vs. Let

Let te permite declarar variables limitando su alcance (scope) al bloque, declaración o expresión donde se esté usando.

Var define una variable global o local en una función sin importar el ámbito del bloque.

Javascript - Var vs. Let

```
var unNumero = 5;
var otroNumero = 10;
if (unNumero === 5) {
    let unNumero = 4; // El alcance es dentro del bloque if
    var otroNumero = 1; // El alcance es global
    console.log(unNumero ); // 4
    console.log(otroNumero ); // 1
}
console.log(unNumero ); // 5
console.log(otroNumero ); // 1
```

Javascript - Variables y tipos de datos

JavaScript es un lenguaje débilmente tipado y dinámico.

Las variables en JavaScript no están asociadas directamente con ningún tipo de valor en particular, y a cualquier variable se le puede asignar (y reasignar) valores de todos los tipos.

Ejemplo:

```
var miVariable = 42;      // miVariable ahora es un número  
  
miVariable      = 'bar';  // miVariable ahora es un string  
  
miVariable      = true;   // miVariable ahora es un booleano
```

Estructuras de Control

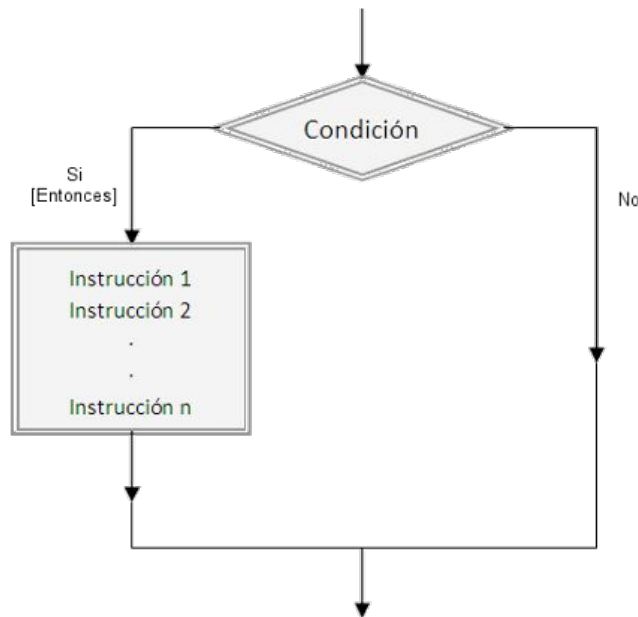
¿Qué son? ¿Para que se utilizan?

Estructuras de Control

Decisión Simple

Es la estructura de control más común. En esta, se obliga a evaluar una condición, que corresponde a expresiones lógicas.

Si la condición es **verdadera**, se ejecuta un conjunto de instrucciones. Si la condición es **falsa**, se ignoran y se continúa el programa después de la estructura.

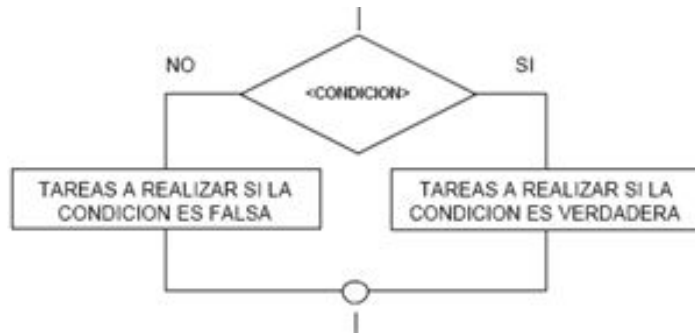


Estructuras de Control

Decisión Doble

Las estructuras condicionales dobles permiten elegir entre dos opciones o alternativas posibles en función del cumplimiento o no de una determinada condición.

Se utiliza la declaración **“else”**, que indicará el código que debe ejecutarse en caso de que la condición no se cumpla. La declaración **“else”** no es obligatoria.



Estructuras de Decisión

IF

```
if (condición) {  
    // Código a ejecutar si la condición se cumple  
}
```

ELSE

```
if (condición) {  
    // Código a ejecutar si la condición se cumple  
} else {  
    // Código a ejecutar si la condición no se cumple  
}
```



Operadores Lógicos

OPERADORES LÓGICOS Y RELACIONALES	DESCRIPCIÓN	EJEMPLO
==	Es igual	a == b
===	Es estrictamente igual	a === b
!=	Es distinto	a != b
!==	Es estrictamente distinto	a !== b
<, <=, >, >=	Menor, menor o igual, mayor, mayor o igual	a <=b
&&	Operador and (y)	a && b
	Operador or (o)	a b
!	Operador not (no)	!a

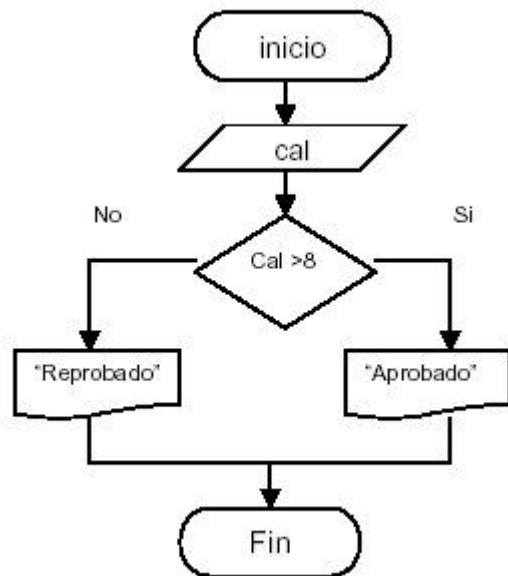
Estructuras de decisión

SWITCH

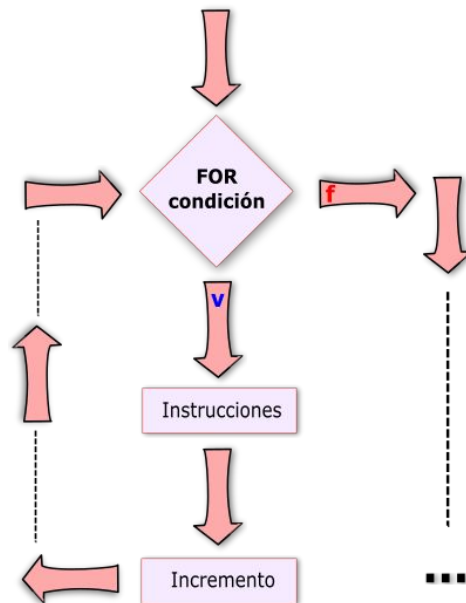
```
switch (expresión) {  
    case valor1:  
        //Declaraciones ejecutadas cuando el resultado de expresión coincide con el valor1  
        [break;]  
    case valor2:  
        //Declaraciones ejecutadas cuando el resultado de expresión coincide con el valor2  
        [break;]  
    ...  
    case valorN:  
        //Declaraciones ejecutadas cuando el resultado de expresión coincide con valorN  
        [break;]  
    default:  
        //Declaraciones ejecutadas cuando ninguno de los valores coincide con el valor de la expresión  
        [break;]  
}
```

Estructuras

Estructuras de decisión



Estructuras de repetición



Estructuras de repetición

Las estructuras de decisión que conocemos:

- FOR
- WHILE
- DO WHILE

¡Vamos al código!

Funciones

¿Qué son las funciones?

Funciones

- Una función es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor.
- Las funciones pueden tomar parámetros que modifiquen su funcionamiento.
- Las funciones son utilizadas para descomponer grandes problemas en tareas simples y para implementar operaciones que son comúnmente utilizadas durante un programa y de esta manera reducir la cantidad de código.

¿Qué diferencia hay entre “función” y “procedimiento”?

¿Cómo declaramos una función?

Una función es un bloque de código definido para realizar una acción en específica.

```
function nombre(parametro1, parametro2, parametro3) {  
    // código a ejecutar.  
}
```

- Los **parámetros** de una función son listados dentro de los paréntesis ()
- Los **argumentos** de una función son los **valores** recibidos cuando la función es invocada.
- Dentro de la función, los **argumentos** (parámetros) actúan como variables locales.

Referencias

- [Variables - Javascript](#)
- [Funciones - Javascript](#)
- [Ciclos de Repetición - Javascript](#)
- [Estructuras de Control - Javascript](#)

¿Preguntas?



**Argentina
programa
4.0**

Gracias!
