

# Bootcamp de Full Stack

Bienvenidos a la clase N°11

- DOM
- Inserción de scripts
- Cuadros de diálogos
- Consola web
- Variables y constantes
- Tipos de datos
- Conversiones de datos
- Operadores: aritméticos, de comparación, lógicos, etc

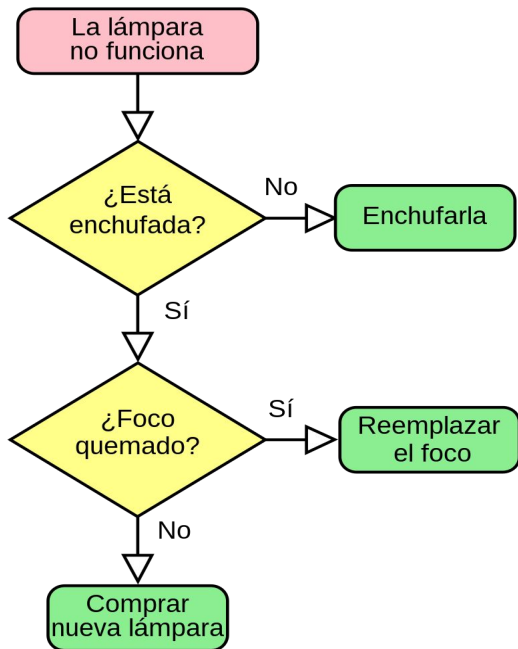
# Fundamentos

## Programación

### ¿Qué es un algoritmo?

En general, no existe ningún consenso definitivo en cuanto a la definición formal de algoritmo. Muchos autores los señalan como listas de instrucciones para resolver un cálculo o un problema abstracto, es decir, que un número finito de pasos convierten los datos de un problema (entrada) en una solución (salida)

Fuente: <https://es.wikipedia.org/wiki/Algoritmo>



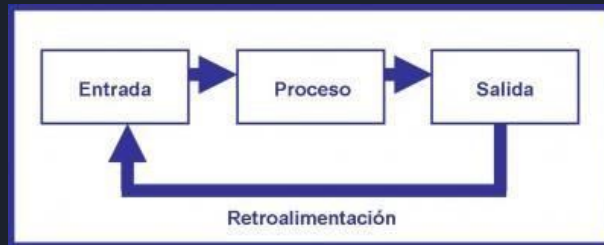
# Fundamentos

## Programación

### ¿Qué es un proceso?

Un proceso, en informática, puede entenderse informalmente como un programa en ejecución. Formalmente un proceso es "Una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados".

Fuente: [https://es.wikipedia.org/wiki/Proceso\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Proceso_(inform%C3%A1tica))



# Fundamentos

## Programación

### ¿Qué es un lenguaje de programación?

Un **lenguaje de programación** es un lenguaje formal (o artificial, es decir, un lenguaje con reglas gramaticales bien definidas) que proporciona a una persona, en este caso el programador, la capacidad y habilidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de un sistema informático, para que de esa manera se puedan obtener diversas clases de datos o ejecutar determinadas tareas.

Fuente: [https://es.wikipedia.org/wiki/Lenguaje\\_de\\_programaci%C3%B3n](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n)



# Fundamentos

## Programación

### ¿Qué es JavaScript?

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

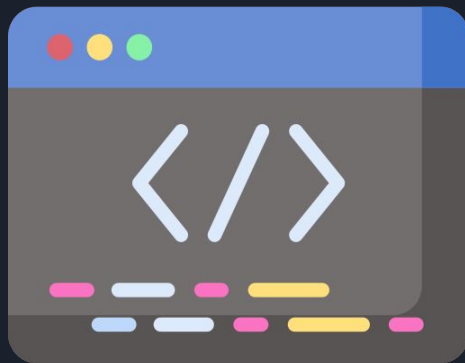
Fuente: <https://es.wikipedia.org/wiki/JavaScript>

**ECMAScript** es una especificación de lenguaje de programación publicada por **Ecma International** (es una organización internacional basada en membresías de estándares para la comunicación y la información). El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation.

Fuente: <https://es.wikipedia.org/wiki/ECMAScript>

# JAVASCRIPT

## Inserción de Script

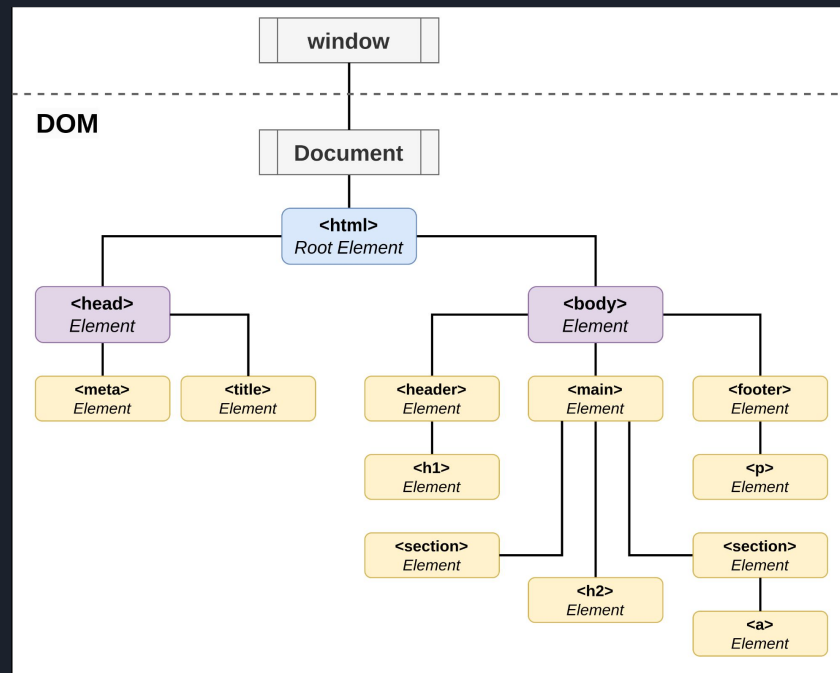


1. Insertar JS dentro de un atributo de evento, por ejemplo:  
`<button onclick="alert('Hola')">Saludar</button>`
2. Insertar JS en bloque por medio de `<script>`, por ejemplo:  
`<script> alert("Hola mundo"); </script>`
3. Insertar JS en archivo externo por medio de `<script>`, por ejemplo:  
`<script src="app.js"></script>`

# Fundamentos

## DOM

- El objeto **window** contiene al objeto **document**
- El **DOM** es justamente lo que refiere al objeto **document**



# JAVASCRIPT

## HTML

¿Qué es window?

Es un objeto representa la ventana que contiene un documento DOM, en cambio, la propiedad **document** apunta al **DOM** cargado en esa ventana.

### 1. Cuadro de diálogo en la ventana del HTML

- Mostrar y esperar a que el usuario cierre: `window.alert("Hola");`
- Mostrar y esperar a que el usuario confirme o cancele: `window.confirm("¿Desea continuar?");`
- Mostrar y esperar a que el usuario introduzca un valor, de ser necesario, se puede agregar un valor por defecto : `let respuesta = window.prompt("Ingresa tu edad", "18");`
- Mostrar URI actual: `alert(window.location);`
- Re-direccionar a una URI: `window.location = "https://www.educacionit.com/";`

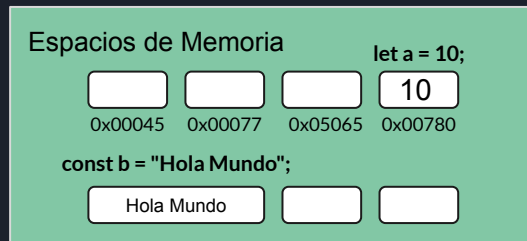
### 2. Escribir una cadena de texto en el HTML: `window.document.write("Hola");`



# JAVASCRIPT

## Variables y Constantes

1. Declaración de una variable: **let nombreDeLaVariable;**
2. Asignación del valor de una variable: **nombreDeLaVariable = "valor";**
3. Declaración y asignación de una constante: **const nombreDeLaConstante = "valor";**
4. Uso de comentario de línea **//** y comentario de bloque **/\* \*/**



# JAVASCRIPT

## Consola

1. Comandos de consola web:
  - a. Mostrar un mensaje genérico: `console.log();`
  - b. Mostrar un mensaje de información: `console.info();`
  - c. Mostrar un mensaje de error: `console.error();`
  - d. Mostrar el resultado negativo de una aserción: `console.assert( 1 > 2, "1 no es mayor 2");`
  - e. Mostrar el tiempo transcurrido en milisegundos para medir un proceso:
    - i. Punto inicial: `console.time();`
    - ii. Punto final: `console.timeEnd();`

# BREAK

Descansemos 10 minutos



# JAVASCRIPT

## Data Types

1. Tipo de datos primitivos: el tipo se puede preguntar con `typeof()`;
  - a. Valor de texto (caracteres alfanuméricos y especiales): **String**
  - b. Valor numérico (enteros y decimales): **Number**
  - c. Valor de verdad (verdadero/falso): **Boolean**
  - d. Valor indefinido: **Undefined**
  - e. Valor nulo: **Null**
  - f. Valor no numérico: **NaN**
2. Hacer un casting de datos
  - a. Transformar un string a boolean: **Boolean("true");**
  - b. Transformar un number a boolean: **Boolean(0);**
  - c. Transformar un string a number: **Number("12.52");**
  - d. Transformar un number a string: **String(500);**

# JAVASCRIPT

## Data Types

### 3. Hacer una conversión de datos

- a. Convertir un decimal (string) a entero (number): `Number.parseInt("15.77");`
- b. Convertir un decimal (number) a entero (number): `Number.parseInt(12.55);`
- c. Convertir un decimal (string) a decimal (number): `Number.parseFloat("20.21");`
- d. Convertir un entero (number) a decimal (number): `Number.parseFloat(50);`
- e. Convertir un entero (number) a entero (string): `let x = 50; x.toString();`
- f. Convertir un decimal (number) a decimal (string): `let z = 3.44; z.toString();`
- g. Convertir un boolean a string: `let y = true; y.toString();`

# JAVASCRIPT

## Operadores

1. Operador de concatenación de cadenas de texto
2. Operadores de incremento (++) y decremento (--)
3. Operadores aritméticos:
  - a. Suma + ó +=
  - b. Resta - ó -=
  - c. Multiplicación \* ó \*=
  - d. División / ó /=
  - e. Módulo % ó %=

# JAVASCRIPT

## Operadores

### 4. Operadores de comparación:

- a. Igual ==
- b. Igual estricto ===
- c. Diferente !=
- d. Diferente estricto !==
- e. Menor que <
- f. Menor o igual que <=
- g. Mayor que >
- h. Mayor o igual que >=

# JAVASCRIPT

## Operadores

### 5. Operadores lógicos:

- a. AND significa conjunción “y” &&
- b. OR significa disyuntiva “o” ||
- c. NOT significa negación !



# CIERRE DE CLASE

Continuaremos en la próxima clase

