

A stylized logo for JavaScript. It features a bright yellow square with the letters 'J' and 'S' in a bold, dark grey font. The square is set against a black background that has a thick, L-shaped border on the left and bottom sides.

**JS**

# INTRODUCCIÓN A JAVASCRIPT

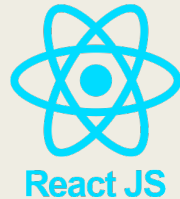
- Santiago Cano Herranz

# ¿Que es JavaScript?

- Un lenguaje de programación dinámico que se usa principalmente para crear webs “vivas” o dinámicas.
- Es un lenguaje de programación **Interpretado**, es decir no necesita ser compilado y es ejecutado por otro programa. En este caso un navegador.
- Es utilizado para generar interacciones desde el lado del cliente(Front-end) pero también puede utilizarse en el servidor(Back-end).
- Su principal objetivo es añadir interactividad a las páginas Web.
- Permite que las acciones ejecutadas por el usuario/cliente influyan en la información presentada en la página Web ya sea modificando o creando nuevos elementos. O lo que es lo mismo permite manipular el **DOM**
- Es utilizado como base de las principales librerías y frameworks mas utilizadas y demandadas por el mercado.

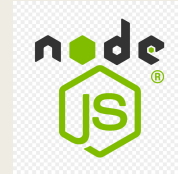
# Librerías y Frameworks basados en JS.

## React js



- Librería basada en js mantenida por Facebook.
- Permite crear interfaces de usuario de forma sencilla.
- Renderiza de forma optimizada la aplicación.
- Tiene frameworks derivados como [React Native](#) que nos permiten desarrollar app nativas utilizando una base de Js. Facebook e Instagram utilizan esta tecnología.

## Node js



- Librería de código abierto basada en js que permite el desarrollo de aplicaciones js del lado del servidor.
- Liviano eficiente y escalable.
- Utilizado por grandes compañías como Netflix, Ebay o PayPal entre otras.

## Angular js



- Framework de de código abierto basado en js mantenido por Google.
- Utilizado para desarrollar aplicaciones SPA Y PWA.
- Alto rendimiento y velocidad basado en la optimización.
- Utilizado por grandes compañías como Google , GitHub, Youtube o Nike entre otros

# Sintaxis básica Javascript

- Primitivos básicos y tipos de datos

```
1;
```

```
2.5;
```

```
// strings
```

```
"Hello world!";
```

```
'Javascript is Cool! 🤩';
```

```
// booleans
```

```
true;
```

```
false;
```

```
//null y undefined
```

```
null; //sin valor
```

```
undefined; //un valor que aun no está presente
```

# Sintaxis y estructura básicas

- Operaciones en números y cadenas

```
// operaciones aritméticas normales
```

```
1 + 1; // = 2
```

```
0.1 + 0.2; // = 0.30000000000000004
```

```
35 / 5; // = 7
```

```
5 / 2; // = 2.5
```

```
// la precedencia se aplica usando paréntesis ()
```

```
(1 + 3) * 2; // = 8
```

```
// operadores
```

```
10 % 2; // = 0
```

```
18.5 % 7; // = 4.5
```

```
// Las cadenas se concatenan con +
```

```
"Hello " + "world!"; // = "Hello world!"
```

```
"1, 2, " + 3; // = "1, 2, 3"
```

# Sintaxis y estructura básicas

- Operadores de comparación

```
// La igualdad es ===
1 === 1; // = true

// desigualdad es !==
1 !== 1; // = false

// Mas comparaciones
1 < 10; // = true
2 >= 2; // = true
"a" < "b"; // = true

//La coerción de tipo se realiza para comparaciones con dobles iguales...
"5" == 5; // = true
null == undefined; // = true

// ...A menos que uses ===
"5" === 5; // = false
null === undefined; // = false
```

# Sintaxis y estructura básicas

- Variables y asignaciones

```
let someVar = 5; // Declaración e inicialización de una variable
someOtherVar = 10; // Creado en alcance global
// declarando variable no inicializada
let someThirdVar; // = undefined
// Declarando más de 1 variable
let someFourthVar = 2, someFifthVar = 4;
// Algunas taquigrafías
someVar += 5; //equivale a someVar = someVar + 5; (= 10)
someVar *= 10; // ahora someVar is 100
// incrementación y decrementación
someVar++; // now someVar is 101
someVar--; // back to 100
```

# Sintaxis y estructura básicas

- Declaraciones de control

```
//if Declaraciones
let count = 1;
let (count == 3){
    //evaluamos si count tiene el valor 3
} else if (count == 4){
    //evaluamos si count tiene el valor 4
} else {
    //evaluamos si count no es 3 ni 4
}
// 'bucle While'
while (true){
    //bucle infinito!
}
//Do-while bucle
do {
    //haz alguna acción
} while (condición);
```



# Funciones en JavaScript

- Una función es simplemente un "trozo" de código que puedes usar una y otra vez, en lugar de escribirlo varias veces
- Las funciones permiten a los programadores desglosar o descomponer un problema en trozos más pequeños, cada uno de los cuales realiza una tarea en particular.
- Las funciones proporcionan una mejor modularidad para su aplicación y un alto grado de reutilización de código, lo que permite una mejor escala y ciclos de desarrollo más eficientes.

# Funciones en JavaScript

- Cómo declarar y llamar a una función

//Una función JavaScript se define con la palabra clave function, seguida de un nombre, seguido de paréntesis ().

// Function Para calcular el producto de p1 y p2

```
function myFunction(a, b) {  
    return a * b;      // Function Devuelve el producto de a y b  
}
```

//Llamando a la función

```
let x = myFunction(4, 3); // Se llama a la función, el valor de retorno terminará en x
```

//Function Los parámetros se enumeran dentro de los paréntesis () en la definición de la función.

//Los argumentos de función son los valores recibidos por la función cuando se invoca.

//Dentro de la función, los argumentos (los parámetros) se comportan como variables locales.

# Funciones en JavaScript

- Ejemplos de uso de funciones integradas y creación de sus propias funciones

```
// Función JavaScript para convertir una cadena determinada en mayúsculas
```

```
function myFunction(thing){  
    return thing.toUpperCase();  
}
```

```
myFunction("hacer"); // = "HACER"
```

```
//Hacer una función para convertir grados celsius a fahrenheit
```

```
function celsiusToFahrenheit(celsius){  
    return celsius * 1.8 + 32;  
}
```

```
celsiusToFahrenheit(0); // = 32
```

# Definición y uso de objetos y arrays

- Tanto los objetos como las arrays se consideran "especiales" en JavaScript. Los objetos representan un tipo de datos especial que es mutable y se puede utilizar para almacenar una colección de datos (en lugar de un solo valor). Las matrices son un tipo especial de variable que también es mutable y también se puede utilizar para almacenar una lista de valores
- Los objetos son colecciones de pares clave-valor utilizados para almacenar y manipular datos que tienen propiedades y métodos, mientras que los arrays son colecciones ordenadas de valores utilizados para almacenar y manipular listas de datos.
- El propósito de los objetos y arrays es proporcionar una forma de organizar y trabajar con los datos de una manera estructurada, haciendo que sea más fácil acceder, manipular e iterar sobre los datos en los programas JavaScript

# Creación y manipulación de arrays en JavaScript.

// Arrays Son listas ordenadas de valores, de cualquier tipo.

```
var myArray = ["Hello", 45, true];
```

// Se puede acceder a sus miembros utilizando la sintaxis de subíndice de brackets cuadrados.

// Array Los índices comienzan en cero.

```
myArray[1]; // = 45
```

// Arrays Son mutables y de longitud variable.

```
myArray.push("World");
```

```
myArray.length; // = 4
```

// añadir/modificar un índice específico

```
myArray[3] = "Hello";
```

// Añadir y eliminar elementos de la parte delantera o trasera de una matriz

```
myArray.unshift(3); // Añadir como primer elemento
```

```
someVar = myArray.shift(); // Elimina el primer elemento y devuélvelo
```

```
myArray.push(3); // Añadir como último elemento
```

```
someVar = myArray.pop(); // Elimina el último elemento y devuélvelo
```

# Creación y manipulación de objetos en JavaScript.

// Los objetos de JavaScript son equivalentes a "diccionarios" o "mapas" en otros lenguajes: una colección inordenada de pares clave-valor.

```
var myObj = {key1: "Hello", key2: "World"};
```

// Las claves son cadenas, pero las comillas no son necesarias si son válidas

// Identificador de JavaScript. Los valores pueden ser de cualquier tipo.

```
var myObj = {myKey: "myValue", "my other key": 4};
```

// También se puede acceder a los atributos de objeto utilizando la sintaxis de subíndice,

```
myObj["my other key"]; // = 4
```

// ... O usando la sintaxis de puntos, siempre que la clave sea un identificador válido.

```
myObj.myKey; // = "myValue"
```

// Los objetos son mutables; se pueden cambiar los valores y se pueden añadir nuevas teclas.

```
myObj.myThirdKey = true;
```

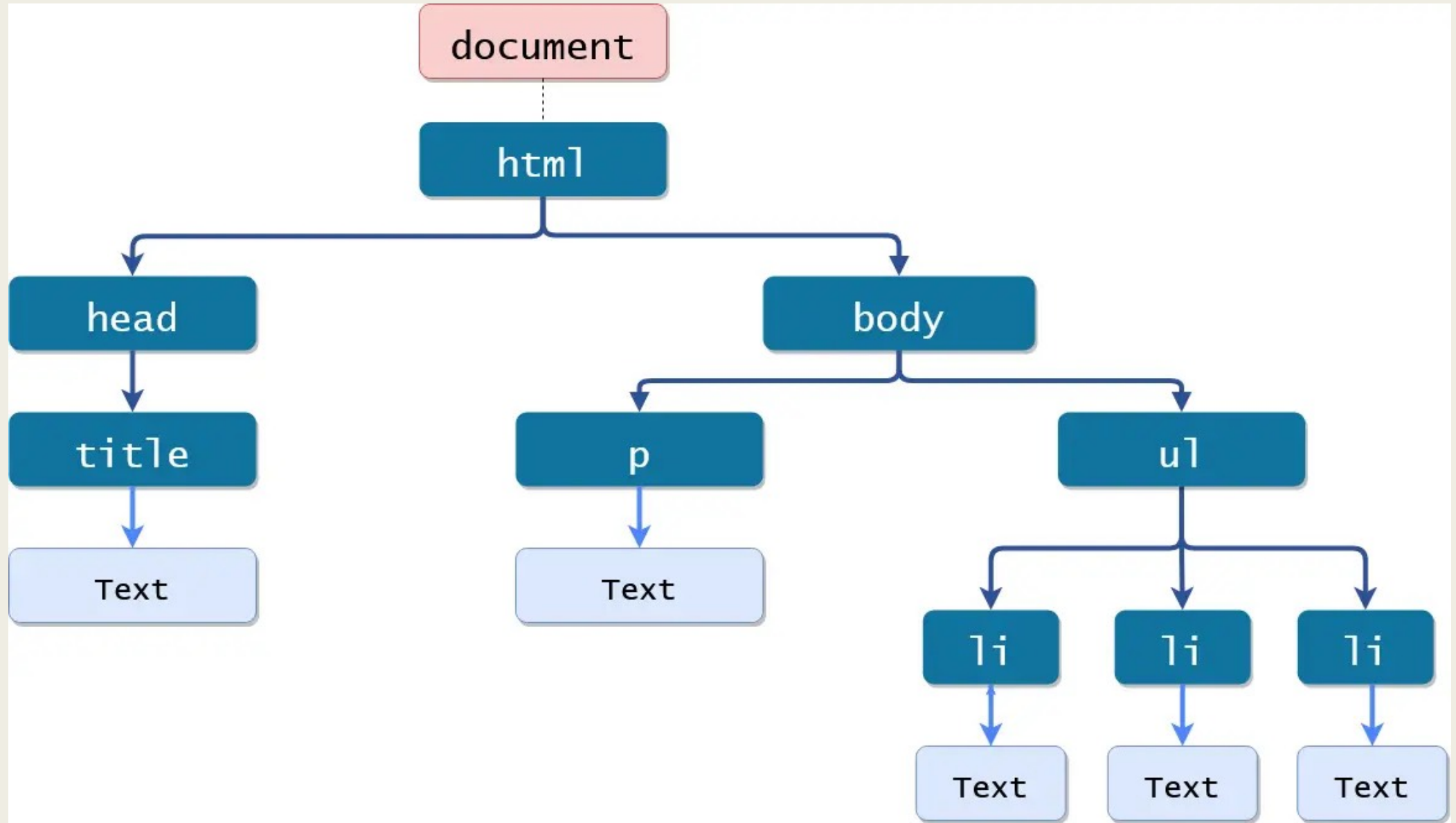
// Si intentas acceder a un valor que aún no está establecido, devolverá undefined

```
myObj.myFourthKey; // = undefined
```

# Manipulación del Dom con Javascript

- El Document Object Model (DOM) es una interfaz de programación que permite a los desarrolladores web manipular el contenido y la estructura de la página web de forma dinámica utilizando lenguajes de programación como JavaScript.
- DOM es una interfaz independiente de la plataforma e independiente del idioma, que permite a los desarrolladores utilizar cualquier lenguaje de programación que admita el estándar web para interactuar con el documento.
- DOM representa la página web como una estructura jerárquica similar a un árbol de nodos y objetos. Cada nodo representa un elemento, atributo o contenido de texto del documento.
- Los nodos pueden tener relaciones entre padres, hijos y hermanos entre sí, formando una estructura en forma de árbol.
- El DOM proporciona métodos y propiedades que permiten a los desarrolladores acceder y manipular el contenido y la estructura del documento, incluida la creación, modificación y eliminación de nodos y elementos.

# Manipulación del Dom con Javascript





# Usar JavaScript para acceder y manipular elementos en una web

- JavaScript se puede utilizar de muchas maneras para lograrlo.
  - *Cambiar el contenido del texto de un elemento:*

```
// HTML: <p id="example">Hello, World!</p>  
let element = document.getElementById("example");  
element.textContent = "Goodbye, World!";
```

- *Cambiar el atributo de un elemento:*

```
// HTML:   
let element = document.getElementById("example");  
element.setAttribute("src", "image2.jpg");
```

# Enlaces y Guías de Interés

- **Tutorial Básico js-html-css** : <https://www.youtube.com/watch?v=Q9fwkpxr3Dw>
- **Tutorial Básico js creando una app de lista de tareas** : <https://www.youtube.com/watch?v=9N7iuyYnqpg>
- **Curso completo gratuito JS** : <https://www.youtube.com/watch?v=ivdTnPI1ND0&t=25020s>

Gracias.