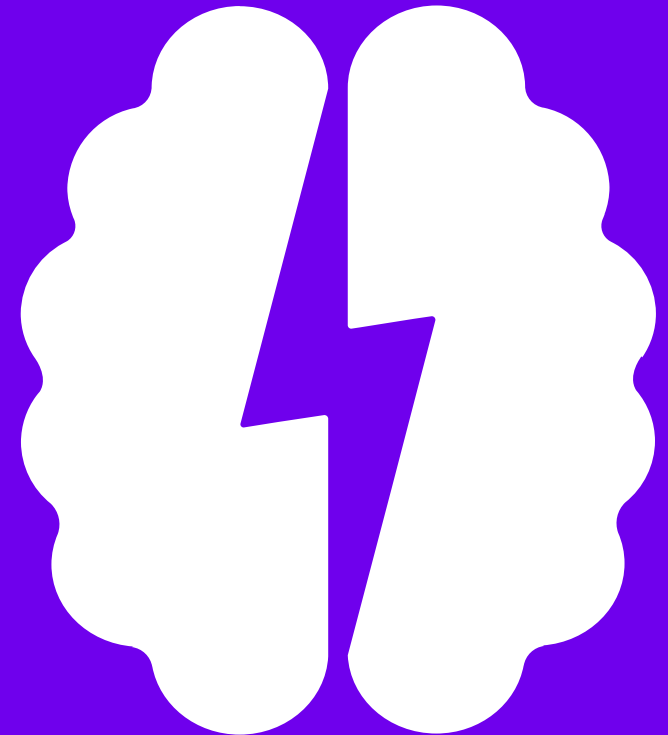


CURSO: DESARROLLO DE APLICACIONES WEB

Sesión 4: JavaScript - I

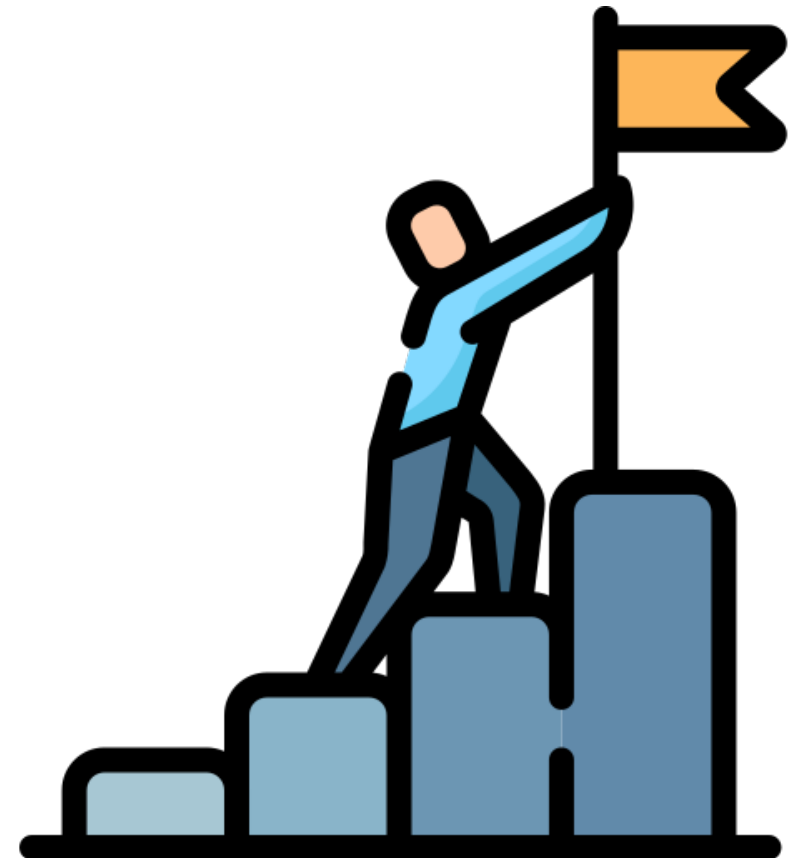


OBJETIVOS



Al finalizar la sesión, el alumno logrará:

- Aplicar JavaScript en el desarrollo de aplicaciones para agregar características interactivas básicas a los sitios web.



AGENDA



- Introducción a JS
- Sintaxis JavaScript
- Variables y Operadores
- Condicionales y estructuras
- HTML y CSS con JavaScript



i

inicio

d

desarrollo

a

aplicación

t

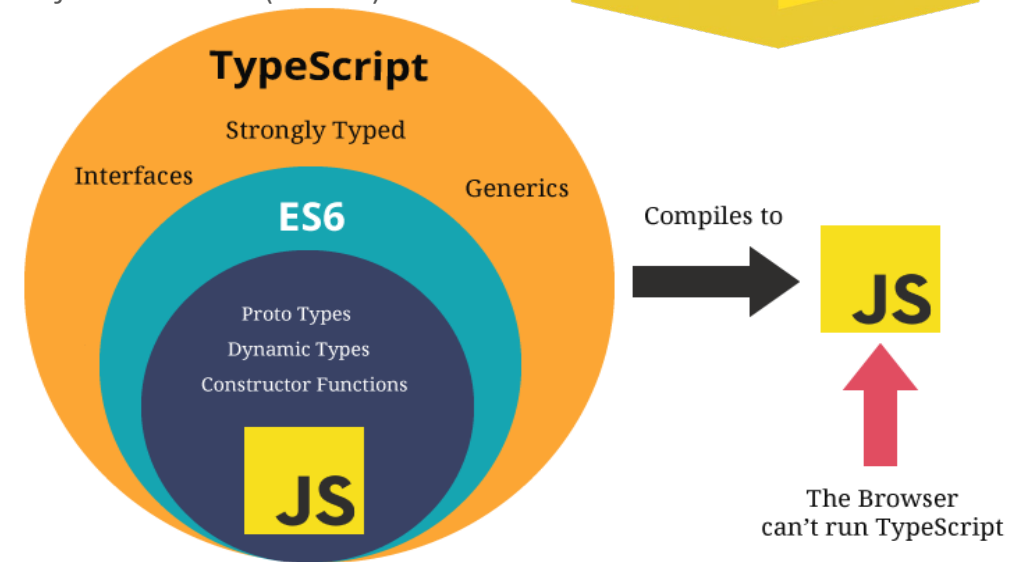
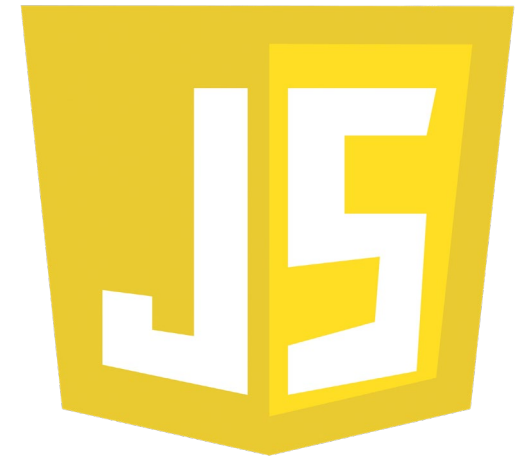
término



INTRODUCCIÓN A JS



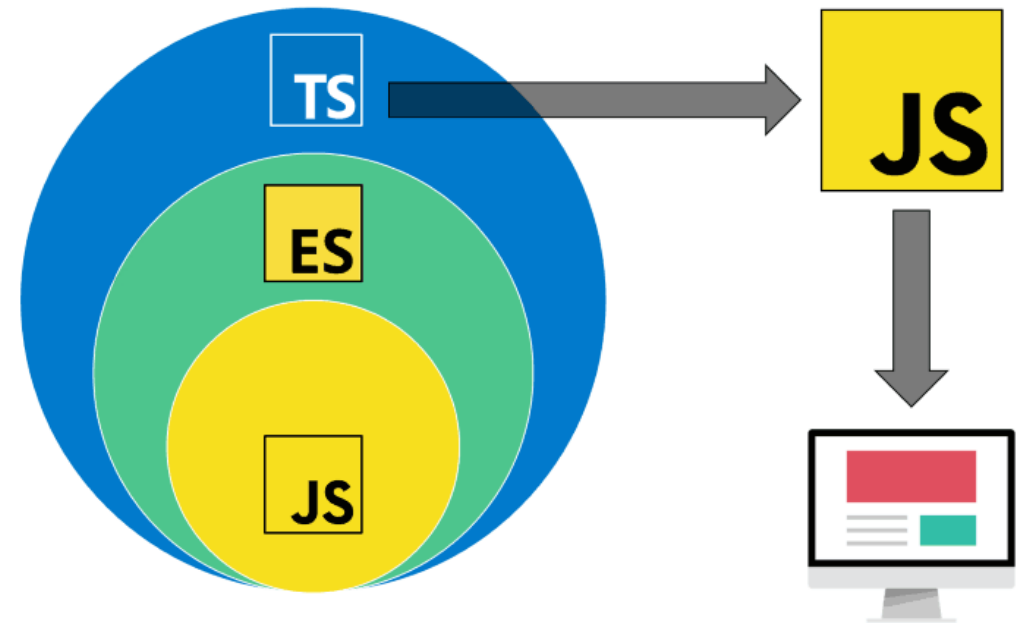
- Lenguaje de programación que se ejecuta en el navegador.
- Es uno de los lenguajes de programación más usados y populares del mundo.
- Nació en 1995 para darle interactividad a las páginas web y desde entonces ha evolucionado hasta convertirse en un lenguaje de programación de propósito general (se puede usar casi para cualquier cosa).
- JavaScript es una marca registrada de Oracle Corporation.
- Microsoft dio como nombre a su dialecto de JavaScript.
- World Wide Web Consortium diseñó el estándar Document Object Model (DOM).
- Permite manejar la interacción del usuario.
- Orientado a eventos y propiedades de objetos.
- Código fuente siempre visible.



INTRODUCCIÓN A JS



- ECMAScript es una especificación de lenguaje de programación publicada por Ecma International. El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation.
- Las características principales de Javascript se basan en el estándar ECMAScript, pero Javascript también tiene otras características adicionales que no están en las especificaciones del estándar.
- TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. La sintaxis es similar a la de JavaScript, pero añade tipos estáticos, objetos basados en clases y otras características más.
- Para que TypeScript funcione en una aplicación web necesita ser compilado y este al ser compilado genera JavaScript en alguna especificación de ECMAScript.



INTRODUCCIÓN A JS



(1) TypeScript vs JavaScript 🎭

```
1 let a: number = 10;
2 let b: number = 2;
3
4 function showProduct(first: number, second: number): void {
5   console.info("Mathematical! The result is " + first * second +
6   ".");
7 }
8 showProduct(a, b);
9
10 // Mathematical! The result is 20.
```

.ts

```
1 var a = 10;
2 var b = 2;
3
4 function showProduct(first, second) {
5   console.info("Mathematical! The result is " + first * second
6   + ".");
7 }
8 showProduct(a, b);
9
10 // Mathematical! The result is 20.
```

.js

SINTAXIS JAVASCRIPT



- Podemos incluir código Javascript dentro de una página HTML a través de la etiqueta SCRIPT.
- También se puede construir un archivo Javascript (.JS) y ser invocado en una página HTML.
- Por último, podemos adicionar Javascript dentro de un elemento HTML a través de su atributo ONCLICK.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>

<script type="text/javascript">
    alert("Un mensaje de prueba");
</script>

<script type="text/javascript" src="nombre.js"></script>

</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

<button onclick="alert('Un mensaje de prueba')">click aquí</button>

</body>
</html>
```


VARIABLES



- Una variable es un espacio que se guarda en memoria, para volver a usarlo después; también, puede cambiar dinámicamente.
- Dentro de JavaScript las variables pueden almacenar varios tipos de datos, los principales son:
 - String: Usar comillas dobles o simples indica al JavaScript que este valor es de tipo de dato String.
 - Number: En este caso no se usa comillas, porque es otro tipo de dato.
 - Bool: Los valores booleanos es solo 0 y 1, que traduciéndolo sería False o True, respectivamente.
- Las variables pueden ser declaradas, inicializadas y modificadas. Para declarar una variable no le tenemos que asignar un valor, al asignarle un valor la estamos inicializando.
- Una Variable en Javascript puede ser declarado de 3 formas: Var, Let y Const.

Diferencias entre Var , Let y Const

Var

```
Var Nombre = 'Frank';
```

El valor puede ser modificada desde cualquier bloque, sin restricción alguna, porque un **var** tiene scope global y local

Let

```
Var num1 = 8;  
Function numero(){  
  let num1 = 10;  
  console.log (num1);  
}  
//El valor num1 no fue modificado
```

Introduce "**Block Scope**", lo que hace que la variable sea accesible solo dentro del bloque declarado

Const

```
Function numero(){  
  const num1 = 10;  
  console.log (num1);  
}
```

Mismo comportamiento que let, con la diferencia de que **no se puede re asignar el valor de una variable const**

VARIABLES



Una variable puede considerarse como un contenedor donde se pueden almacenar datos, una vez guardada podemos acudir a dicha variable a través del nombre que se le haya asignado.



DECLARACIÓN Y ASIGNACIÓN

- Palabra reservada de una variable, dependiendo el caso puede ser: (var, let, const)

- Nombre de la variable

```
var mi_variable = "Declarar variables";
```

- Operador de asignación
- Dato asignado a la variable

TIPOS DE DATOS

- **STRING:** `var name = "Juan";`
- **NUMBER:** `var x = 15;`
- **BOOLEAN:** `var pass = true;`
- **ARRAY:** `var animals = ["cat", "dog"];`
- **OBJECT:** `var person = {name: "jon"};`
- **EMPTY:** `var foo = null;`

BUENAS PRÁCTICAS

- 1 El nombre de la variable debe ser un nombre descriptivo que se identifique con el código
- 2 Las variables no se pueden iniciar con números
- 3 No se pueden usar caracteres especiales, ni es recomendable empezar una variable con mayúscula
- 4 Los tipos de escritura de una variable son: lowerCamelCase, UpperCamelCase, snake_case.
- 5 Tener en cuenta que hay nombres que no se pueden usar ya que se encuentran reservados para el lenguaje.

OPERADORES



Los operadores permiten manipular el valor de las variables, realizar cálculos complejos y tomar decisiones lógicas en función de comparaciones y otros tipos de condiciones.

ARITMÉTICOS

- +** Suma, Concatenar
- ++** Incremento
- Sustracción
- Decremento
- *** Multiplicación
- /** División
- %** Modulo

COMPARACIÓN

- >** Mayor que
- >=** Mayor o igual que
- <** Menor que
- <=** Menor o igual que
- ==** Igual a
- ===** Estrictamente igual
- !=** No igual a
- !==** Estrictamente no igual

ASIGNACIÓN

- =** Asignar valor a
- +=** Sumar y Asignar
- =** Restar y Asignar
- *=** Multiplicar y Asignar
- /=** Dividir y Asignar
- %=** Módulo y Asignar

LÓGICOS

- &** AND (y)
- ||** OR (o)
- !** NOT (no)

CONDICIONALES

- Var x = Condición**
- ?** Si es verdadero
- :** Si es falso

TERNARIO

CONDICIONALES Y ESTRUCTURAS



Repetitivas

while
do
for

Condicionales

if
else
switch

De Transferencia

break
continue

CONDICIONALES Y ESTRUCTURAS



```
if(/*Condicion*/ ){  
    //Codigo  
}else{  
    //Codigo  
}
```

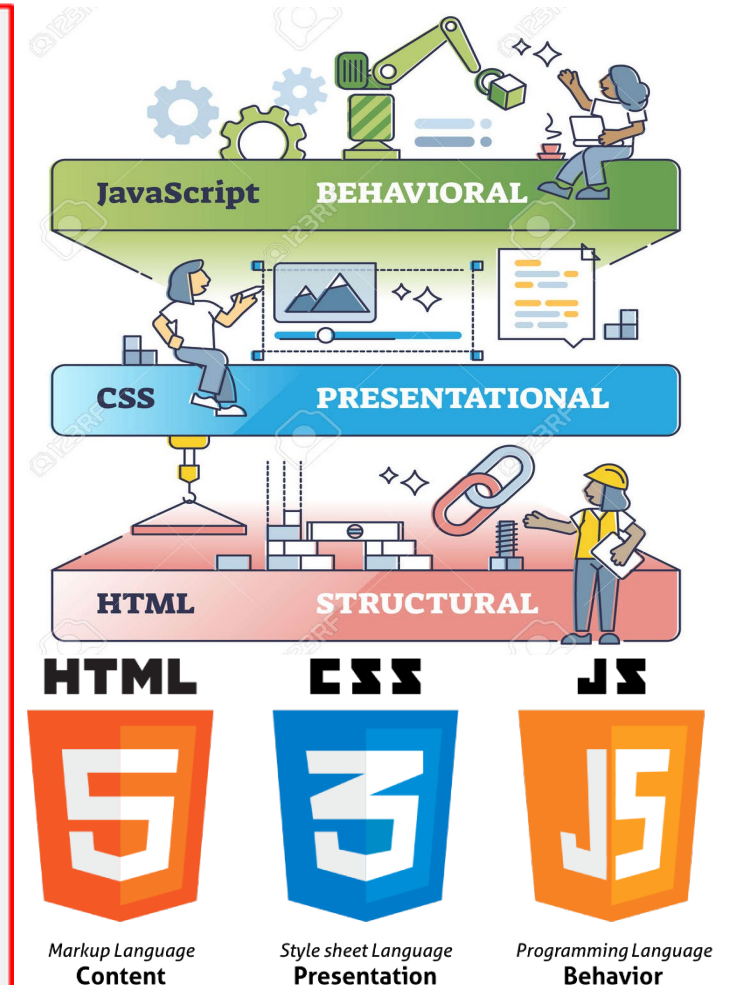
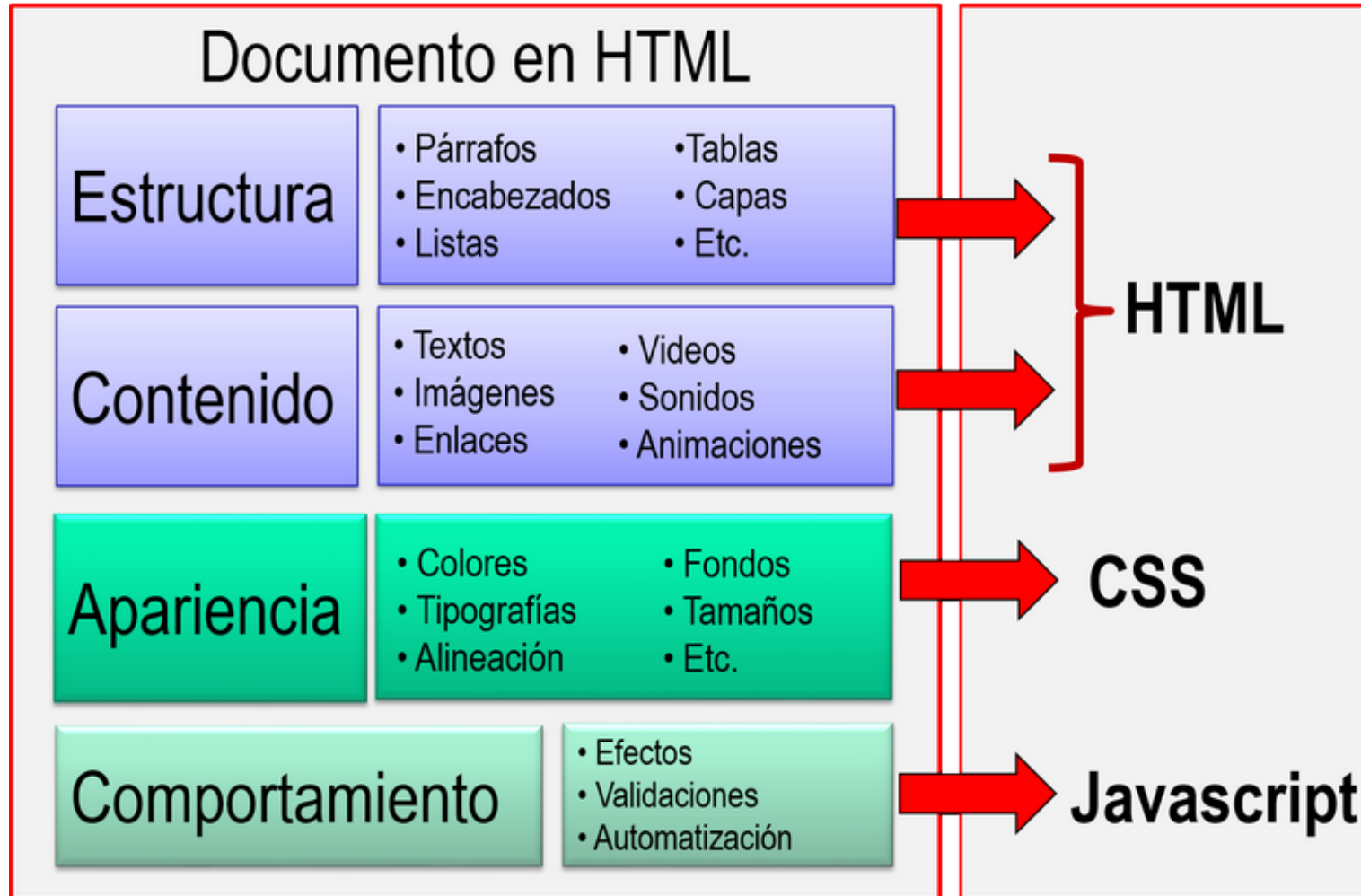
```
switch (Variable){  
    case(valor1):  
        //Codigo  
        break;  
    case(valor2):  
        //Codigo  
        break;  
    case(valor3):  
        //Codigo  
        break;  
    default:  
        /*enunciados a ejecutar cuando el valor no  
        coincida con ninguno de los anteriores*/  
        break;  
}
```

```
for (var i = 0; i < /* Limite */; i++) {  
    /* Codigo a repetir */  
}
```

```
do{  
    //Codigo  
}while (/*Condicion*/);
```

```
while (/*Condicion*/){  
    /*Codigo*/  
}
```


HTML Y CSS CON JAVASCRIPT



Laboratorio N° 1: Crea una página con interacciones JS



- Elaborar una página HTML que nos permita interactuar dinámicamente con sus elementos haciendo uso JS.
- En este laboratorio, el alumno podrá identificar la sintaxis básica de JS.

Laboratorio N° 2: Crea una pirámide de números con JS



- Elaborar una pirámide usando los valores que están en el rango de 1 hasta un valor ingresado a un input luego de accionar un botón.
- En este laboratorio, el alumno podrá eventos y acciones a una página HTML haciendo uso de JS.

LINKS IMPORTANTES



- <https://www.campusmvp.es/recursos/post/typescript-contra-javascript-cual-deberias-utilizar.aspx>
- https://cynoteck.com/es/blog-post/typescript-vs-javascript-vs-ecmascript-know-the-difference/#Why_ECMA_Script
- <https://www.freecodecamp.org/news/what-is-typescript/>
- <https://github.com/castellanosfelipe/Class-examples-Javascript-and-typescript>
- <https://platzi.com/clases/1814-basico-javascript>
- <https://www.juanknet.com/aula/web/>
- <https://aprendejavascript.org/>
- <https://www.aprendejavascript.dev/>
- <https://developer.mozilla.org/es/docs/Learn/JavaScript>
- <https://lenguajejs.com/javascript/>
- <https://htmlcheatsheet.com/js/>

¿Qué te llevas de la sesión?





PREGUNTAS

!GRACIAS!



REFERENCIAS:

- Lenguaje Javascript (Manz.dev)
- Aprende Javascript (Jon Mircha)
- Aprende Javascript (Miguel Duran)
- Curso Básico de JavaScript (Diego De Granda)
- Dando apariencia a la web (Juan Sanabria)
- Capas html, css y javascript (Normaals)
- Diferencias entre var, let y const en Javascript (Sin Rueda Tecnológica)
- What is TypeScript? (Frances Coronel)
- JavaScript y Typescript (Fazt)
- Examples Javascript and typescript (Felipe Castellanos)
- TypeScript Vs JavaScript Vs ECMAScript (Pooja Sharma)
- TypeScript contra JavaScript (José Alarcón)
- JS CheatSheet (HTMLCheatSheet)



Trate de dar crédito a todos los autores de quienes sustraje parte o la totalidad de su contenido para la elaboración de esta presentación, pero si consideras que faltaste porque no te referencie o debo de modificar algo de tu propiedad, por favor, no dudes en hacérmelo saber, contactándome a: jperezgil@outlook.com