

The background of the slide is a blurred image of a laptop screen displaying code. A solid yellow square is positioned on the left side, partially overlapping the code. The text 'JS' is written in large, bold, black letters on this yellow square.

JS

CURSO DE JAVASCRIPT

TecNM – Tlajomulco.
M.C. Jose Angel Torres Rangel

¿Qué es JavaScript?

JavaScript es un lenguaje de programación dinámico que tiene la capacidad de ser utilizado en muchos dispositivos diferentes. Puede usarse en ordenadores personales, servidores web y teléfonos inteligentes. Es un lenguaje interpretado, orientado a objetos o funcional, débilmente tipado.

JavaScript se emplea comúnmente para el desarrollo web front-end y más recientemente para algunos desarrollos back-end a través de frameworks como Node.js. o Next.js.

Nace con la necesidad de generar dinamismo en las páginas web y que a su vez los usuarios y las empresas pudieran interactuar unos con otros. Fue creado por Brendan Eich en 1995 y se convirtió en un estándar oficial del World Wide Web Consortium (W3C) en 1997.

JS

¿Por qué JavaScript?

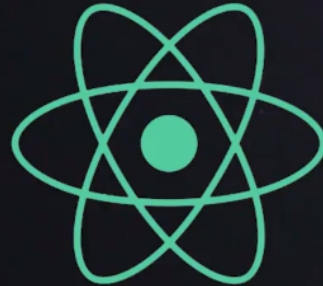
Desarrollo Web



Lit



Desarrollo de Apps



React Native

Desarrollo de Apps para Desktop



ELECTRON

Back-end / IOT



ECMAScript es una especificación de lenguaje de scripting en la que se basa JavaScript y Ecma International está a cargo de estandarizar este lenguaje.

5	2009
5.1	2011
6	2015
7	2016
8	2017
9	2018
10	2019
11	2020
12	2021

<https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>

1. Debilmente tipado, tipos de datos y variables
2. Funciones
3. Scope, Hoisting, Coercion, Truly y Falsy
4. Operadores
5. Condiciones
6. Arrays
7. Loops
8. Objects
9. Closures
10. Asincronismo
11. Fetch
12. DOM (Document Object Model)
13. Modules
14. jQuery
15. Proyecto

- Debilmente tipado

```
4 + "7"; // 47
```

```
4 * "7"; // 28
```

```
2 + true; // 3
```

```
false - 3; // -3
```

- Tipos de datos o valores (type of)

```
string
```

```
number
```

```
boolean
```

```
Object    {},[],null
```

```
undefined "type of algo"
```

- Variables y constantes

```
var nombre = "Juan"
```

```
let apellido = "Rodriguez"
```

```
const edad = 48
```

- Declarativas

```
function sumar(a, b){  
    return a+b  
}
```

- Expresion

```
let sumar = function sumar(a, b){  
    return a+b  
}
```

- Flecha

```
const saludar = (nombre) => {  
    console.log(`Hola ${nombre}`)  
}
```

JS Scope, Hoisting, Coercion, Truly y Falsy

- Scope (Global y Local)
- Hoisting (var y function)
- Coercion (implicita y explicita)
- Truly y Falsy

- Aritmeticos
 - +, -, *, /, -=, +=, *=, /=
- Unitarios
 - !, ++, --
- Comparacion
 - ==, ===, <, >, <=, >=, !=
- Logicos
 - &&, ||

- If

```
if (true){
```

```
}else{
```

```
}
```

- Operador ternario: Condition ? true: false

```
let numero=5
```

```
let res = numero === 5? "si soy numero":"No soy numero"
```

- switch

```
switch(variable){
```

```
  case 1:console.log('Soy uno')
```

```
  break
```

```
  case 2:console.log('Soy dos')
```

```
  break
```

```
  default:console.log('Soy otra cosa')
```

```
}
```

- Declaracion

```
let team1 = ['juan', 'pedro', 'luis']
```

```
let team2 = ['jose', 'rodri','lalo']
```

```
let teams = ['adrian', ...team1, ...team2]
```

- Propiedades

- Length

- Metodos

- push, unshift
- pop, shift
- indexOf
- foreach
- map
- filter
- reduce
- some
- find
- every
- join
- flat
- splice
- sort

- `for(inicializacion;condicion;incremento o decremento){}`
- `for(let item of arreglo){}`
- `while(condicion){}`
- `do{}while(condicion)`

Literales

```
let alumno={  
  nombre : 'Juan Perez',  
  edad : 23,  
  fechaDeRegistro : new Date(),  
  casado : true,  
  domicilio : {  
    cd : 'Gdl',  
    calle : 'Ramon Corona',  
    num : 89  
  },  
  telefonos : [  
    {casa:'123456'},  
    {movil:'654321'}  
  ],  
  misDatos : function(){  
    console.log(`mi nombre es ${this.nombre} y mi edad es ${this.edad}`)  
  }  
}
```

```
function auto(marca, modelo, annio){  
  this.marca = marca  
  this.modelo = modelo  
  this.annio = annio  
}
```

```
let autoNuevo1 = new auto("Tesla","Model 3",2020)
```

```
let autoNuevo2 = new auto("Tesla","Model X",2022)
```

```
class calculator {  
  constructor() {  
    this.valueA = 0  
    this.valueB = 0  
  }  
  sum(valueA, valueB) {  
    this.valueA = valueA  
    this.valueB = valueB  
    return this.valueA + this.valueB  
  }  
}
```

```
const calc = new calculator()  
console.log(calc.sum(2, 2))
```

```
class Estudiante {  
  constructor({nombre, edad, email})  
  {  
    this.nombre  
    this.email  
    this.edad  
    this.cursosAprobados = [ ]  
  }  
  aprobarCurso(nuevoCurso) {  
    this.cursosAprobados.push(nuevoCurso);  
  }  
}  
  
const estudiante1 = new Estudiante({  
  nombre: "Miguel",  
  email: "miguelito@platzi.com",  
  age: 28  
})
```



```
const alcancia = () => {  
  let ahorro = 0;  
  const agrega = (pesos) => {  
    ahorro += pesos;  
    console.log(ahorro);  
  }  
  return agrega;  
}
```

```
let miAlcancia = alcancia()  
miAlcancia(4)    //4  
miAlcancia(6)    //10  
miAlcancia(10)   //20
```

```
const miFuncion1 = () => {  
  setTimeout(()=>console.log("Hola a todos desde miFuncion1") ,5000)  
}
```

```
const miFuncion2 = () => {  
  setTimeout(()=>console.log("Hola a todos desde miFuncion2") ,2000)  
}
```

```
const miFuncion3 = () => {  
  miFuncion1()  
  miFuncion2()  
}
```

- Callbacks
- Promise
- Async – Await

```
const funcAsincrona = () => {  
  return new Promise((resolve, reject) => {  
    let numRandom = Math.floor(Math.random()*(100-0))+0  
    if (numRandom<90)  
      setTimeout(() => resolve('Exito!!'), 3000)  
    else  
      setTimeout(() => reject(new Error('Error!!')), 3000)  
  })  
}  
  
const otraFuncion = async () => {  
  const resul = await funcAsincrona();  
  console.log(resul);  
  console.log('Hello!');  
}
```

```
async function getDataFromFile(){
  const res = await fetch('http://localhost:8080/proyecto1JavaScript/alumnos.json',
    {method: 'GET', headers: {"Content-Type": "application/json"}})
  const datos = await res.json()
  datos.forEach((alumno)=>{
    console.log(alumno)
  })
}
```

- https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model/Introduction
- window y document
- Desde Html
`<script defer src="./archivo.js"></script>`
- archivo.js

- `document.querySelector("tag")`
- `document.querySelector("#image1")`
- `document.querySelector(".class")`
- `document.querySelectorAll("tag")`
- `document.getElementsByClassName("className")`
- `document.getElementById("id")`

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Events

```
let boton1 = document.getElementById("boton1")

boton1.addEventListener("click", btn1Click)

function btn1Click(event) {
  let input1 = document.getElementById("input1")
  alert(input1.value)
}
```

- `element.innerHTML`
- `element.value`
- `element.style`
- `element.classList`
 - `.add()`
 - `.remove()`
 - `.toggle()`
- `element.getAttribute('attribute')`
- `element.setAttribute('attribute', 'value')`

- `document.createElement('tag')`
- `elementParent.appendChild(elementChild)`

Es cuando se agregan archivos con mas codigo Javascript.

- En el archivo adicional

```
const hello = () => {  
  return 'hello'  
}
```

```
export default hello
```

- En el archivo principal

```
import {hello} from “./module.js”
```

- JQuery
 - <https://jquery.com/download/>
- Project
 - <https://github.com/jangeltr/cursoAyuntamientoTlajo.git>

Gracias

TecNM Tlajomulco

M.C. Jose Angel Torres Rangel

jose.tr@tlajomulco.tecnm.mx