

IIC2026 - Visualización de Información:

# JavaScript

*Introducción a los conceptos básicos*

Cristóbal Abarca Quiroga  
caabarca1@uc.cl


Hernán Valdivieso  
hfvaldivieso@uc.cl

# ¿Qué es?

*Un acercamiento a Javascript*

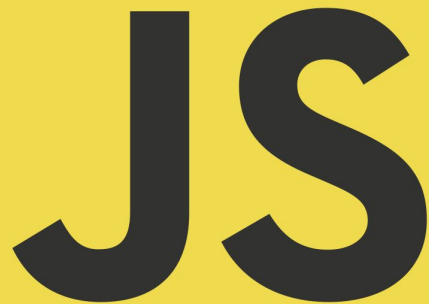
# ¿Javascript?

- Es un **lenguaje de programación** multiparadigma de alto nivel.

A large yellow square containing the letters 'JS' in a bold, dark grey, sans-serif font, representing the JavaScript logo.


# ¿Javascript?

- Es un **lenguaje de programación** multiparadigma de alto nivel.
- Es interpretado, al igual que Python y Ruby.

The image shows the letters 'JS' in a large, bold, dark grey font, centered on a solid yellow square background. This is a common logo for JavaScript.


# ¿Javascript?

- Es un **lenguaje de programación** multiparadigma de alto nivel.
- Es interpretado, al igual que Python y Ruby.
- Sirve para **manipular el DOM**

A large, bold, dark gray 'JS' logo is centered on a solid yellow square background.


# ¿Javascript?

- Es un **lenguaje de programación** multiparadigma de alto nivel.
- Es interpretado, al igual que Python y Ruby.
- Sirve para **manipular el DOM**
- Es **interpretado** por el navegador

A large, bold, dark gray 'JS' logo is centered on a solid yellow square background.

# ¿Javascript?

- Es un **lenguaje de programación** multiparadigma de alto nivel.
- Es interpretado, al igual que Python y Ruby.
- Sirve para **manipular el DOM**
- Es **interpretado** por el navegador

A large yellow square containing the letters 'JS' in a bold, dark blue, sans-serif font, representing the JavaScript logo.

# ¿Y para qué nos sirve modificar el DOM?

- Permite agregar **dinamismo** en la web





# **¿DOM?**

*Entiendo, pero ¿qué es?*

**¿DOM?**

**D**

Document

¿DOM?

D

Document

O

Object

¿DOM?

D

Document

O

Object

M

Model

**¿DOM?**

**D**

Document

**O**

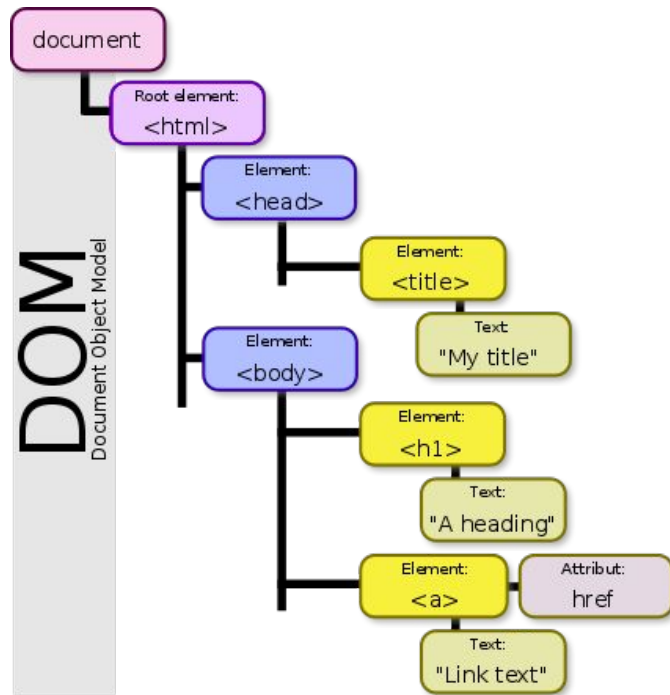
Object

**M**

Model

# ¿DOM?

- La interfaz que a través de una representación estructurada, permite acceder y manipular un documento **HTML**
- Cross-platform language-independent **API**
- Compatibilidad entre Browsers



Fuente:

[https://en.wikipedia.org/wiki/Document\\_Object\\_Model#/media/File:DOM-model.svg](https://en.wikipedia.org/wiki/Document_Object_Model#/media/File:DOM-model.svg)

# ¿Donde escribimos Javascript?

Ejemplos.html x JS ayudantia2.js

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset='utf-8'>
5      <title>Ejemplo Javascript + HTML</title>
6      <link href="style.css" rel="stylesheet" type="text/css">
7      <script src="ayudantia2.js"></script>
8    </head>
9
10   <body>
11     <div>
12       <h1>
13         Ejemplo 1: HTML + console.log()
14       </h1>
15       <p>
16         En este ejemplo tendremos 3 botones que al momento de hacer click, impriman en consola algún texto en específico.
17       </p>
18       <input type="button" id="boton1" value="Botoncito 1">
19       <input type="button" id="boton2" value="Botoncito 2">
20       <input type="button" id="boton3" value="Botoncito 3">
21     </div>
22
```

## Recommended



**4" speed flying | movr4r2 t1  
1407 3600kv 4x4.3x3 v1s**

quadmovr

1.3K views • 11 hours ago



**Visit the Island Run by  
Women**

Great Big Story ✓

243K views • 8 months ago



```
>> console.log("Hello World")
Hello World
← undefined
```

debugger eval code:1:1



# Sintaxis

*Entendí lo anterior, ¿ahora como escribo Javascript?*

# Variables

```
variableUno = [12,3,4];  
numero_1 = 12;  
enteroRandom = 42;  
mejorLenguaje = "python"; // es una constante
```

} Python

---

```
variableSinInicializador = 12;  
var numero_1 = 3;  
let enteroRandom = 42; // block scope variable  
const mejorLenguaje = "python"; // es una constante
```

} Javascript

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let>

# Cuidado con el weak typing...

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
> "Esto es un string normal en Python"
=> 'Esto es un string normal en Python'
> "Pero esto no se puede hacer" + 3
Traceback (most recent call last):
  File "python", line 1, in <module>
TypeError: must be str, not int
> █
```

---

```
> "String normal en Javascript"
=> 'String normal en Javascript'
> "Tambien un string valido en Javascript" + 3
=> 'Tambien un string valido en Javascript3'
> █
```

**Cuidado con el weak typing...**

**Vamos a jugar un poco con el weak typing**

---

# Cuidado con el weak typing...

```
> '5' - 3
< 2
> '5' + 3
< "53"
> '5' - '4'
< 1
> '5' + + '5'
< "55"
> 'foo' + + 'bar'
< "fooNaN"
> '5' + - '2'
< "5-2"
> '5' + - + - - + - - + + - + - + - - - '2'
< "5-2"
> x = 3
< 3
> '5' + x - x
< 50
> '5' - x + x
< 5
```

# for loop

```
lista_de_pi=[3,1,4,1,5,9,2,6,5]
```

```
for numero in lista_de_pi:
```

```
    print(numero)
```

```
3
```

```
1
```

```
...
```

---

```
const listaDePi = [3, 1, 4, 1, 5, 9, 2, 6, 5];
```

```
for (var index = 0; index <= listaDePi.length; index++) {
```

```
    console.log(listaDePi[index]);
```

```
}
```

```
3
```

```
1
```

```
...
```

# for loop (ES6)

```
lista_de_pi=[3,1,4,1,5,9,2,6,5]
```

```
for numero in lista_de_pi:
```

```
    3
```

```
    1
```

```
    4. .
```

```
    print(numero)
```

---

```
const listaDePi = [3, 1, 4, 1, 5, 9, 2, 6, 5];
```

```
for (const numero of listaDePi) {
```

```
    console.log(numero);
```

```
}
```

```
3
```

```
1
```

```
4
```

```
...
```

# forEach

```
var javascript = ['j', 'a', 'v', 'a', 's', 'c', 'r', 'i', 'p', 't'];  
javascript.forEach(function(element) {  
  console.log(element);  
});  
j  
a  
v  
a  
.  
.
```



# if, else if , else

if verdad and falso:

```
print("No se imprimira esto")
```

elif verdad or falso:

```
print("Esto si se imprime")
```

else:

```
print("No se que imprimir aca")
```

Esto si se imprime

---

```
if (verdad && falso) {
```

```
    console.log("No se imprimira esto")
```

```
} else if (verdad || falso) {
```

```
    console.log("Esto si se imprime")
```

```
} else {
```

```
    console.log("No se que se imprimira aca")
```

```
}
```

Esto si se imprime

# functions

```
def multiplicador(a,b):  
    resultado = a * b  
    return resultado  
multiplicador(3, 6)  
18
```

---

```
function multiplicador(a, b) {  
    const resultado = a * b;  
    return resultado;  
}  
multiplicador(3,6);  
18
```

# arrow functions ( Solo Javascript \*\* )

```
const multiplicador = (a,b) => {  
  const resultado = a * b;  
  return resultado;  
};  
multiplicador(3,4);  
12
```

# arrow functions

```
const multiplicador = (a,b) => {  
  const resultado = a * b;  
  return resultado;  
};  
multiplicador(3,4);  
12
```

Pero si esto les parece demasiado verboso...

```
const multiplicador = (numero) => a * b;  
multiplicador(3,4);  
12
```

# Ejemplos

*Me va a costar un poco digerir lo anterior... ¿y ahora cómo lo ocupo con HTML?*

# Ejemplos

→ Click and log

→ Show and hide

→ Edit text

Ver **ayudantia02/Ejemplo.html** en el repositorio

# Práctico

*Ahora les toca a ustedes :)*

# **Javascript**

*Introducción a los conceptos básicos*

Agradecimientos a *Nebil Kawas* (@nakawas) por la información y a *Sofía Muñoz* por el template