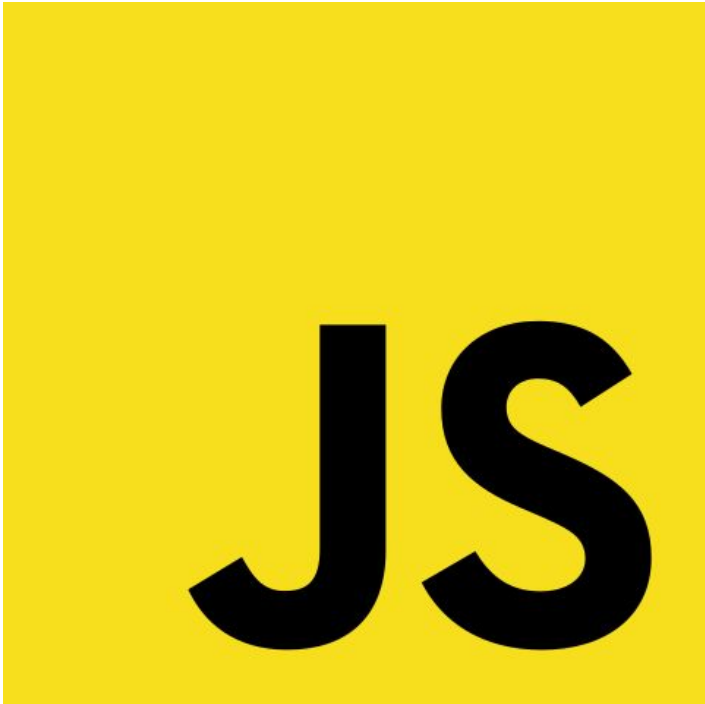


Introducción a Javascript



JS

¿Qué es Javascript?

Javascript es un lenguaje de programación basado en el estándar [ES\(Ecma script\)](#) que a la fecha se ha convertido en un lenguaje muy poderoso por sus características multiparadigma.

Pues si bien es el lenguaje que por excelencia corre del lado del cliente en los navegadores, también puede hacerlo en servidores, móviles etc.

Edwin Rozo Gómez -
edwin.rozo1@misena.edu.co

JS

Algunas características de Javascript



¿QUÉ ES JAVASCRIPT?

JavaScript nace de la necesidad de un lenguaje de programación que se ejecute en el navegador del usuario. En el año 1995, se lanza LiveScript por Brendan Eich, el cual, luego, por temas de marketing, es nombrado JavaScript.

Características principales

- Multi-paradigma
- Un lenguaje procedural tradicional
- Un lenguaje funcional
- Imperativo
- Dinámico en cuanto a tipos
- Orientado a eventos
- Orientado a objetos

Versátil

Multiplataforma

Windows, Linux, Apple

Sintaxis

Su sintaxis es similar a la usada en Java y C

C++, C

Se utiliza para:

- Hardware
- Plataformas de juegos
- Sistemas operativos
- Administración de sistemas
- Servidores de Internet
- Desarrollo Móvil
- Bases de datos

Síguenos en:

Iconos hechos por Icon de www.icon-icons.com

Khádev

JS

¿Cómo se ve Javascript?



javascript es un lenguaje amigable con una sintaxis entendible que permite al usuario intuir algunos elementos y más si por algún motivo viene de otros lenguajes de programación. Analizaremos algunos elementos que nos facilitarán la manipulación del **DOM con javascript a partir del clásico ejercicio de “Hello World”**

Estructura y semántica

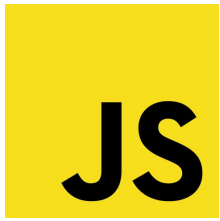
HTML

Estilos

CSS

Interacción

JAVASCRIPT



¿Cómo se ve Javascript?



Crear estructura básica en **HTML**, que se conectará con css y js.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <link rel="stylesheet" href="style.css">
8    <title>Mi first Hello World!</title>
9  </head>
10 <body>
11
12   <script src="/app.js"></script>
13 </body>
14 </html>
```

Conexión con css

Conexión con js

JS

¿Cómo se ve Javascript?



```
1  img {  
2    box-shadow: 0 0 100px rgb(255, 255, 255);  
3    margin-top: 2rem;  
4  }  
5  
6  .img-background {  
7    box-shadow: 0 0 100px rgb(255, 255, 255);  
8    margin-top: 2rem;  
9  }  
10  
11  #img-background {  
12    box-shadow: 0 0 100px rgb(255, 255, 255);  
13    margin-top: 2rem;  
14  }
```

Es el momento para recordar los selectores que se pueden manejar.

Etiqueta: ``

Clase: `.`

id: `#`

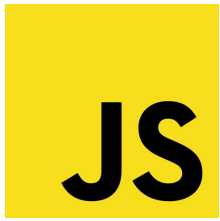
JS

¿Cómo se ve Javascript?



```
1  body {  
2    align-items: center;  
3    background-color: blue;  
4    display: flex;  
5    flex-direction: column;  
6    height: 100vh;  
7    margin: 0;  
8    margin-top: 1rem;  
9    padding: 0;  
10   width: 100vw;  
11 }
```

Luego de dar los estilos apropiados, todo está dado para generar la interacción.



¿Cómo se ve Javascript?



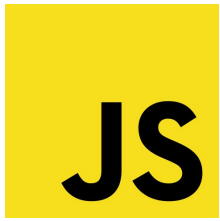
En principio, todo elemento del HTML(DOM) con el que interactuemos, deberá generar su referencia en javascript por medio de una variable, y utilizará los selectores: etiqueta, **id(recomendado)**, **clase**.

Ejemplo:

```
const numero1 = document.getElementById('input1');  
const numero2 = document.querySelector('input2');
```



```
1  const myTitle = document.querySelector('#h1-title');  
2  myTitle.textContent = 'Hello World';
```



¿Cómo se ve Javascript?



Posterior a la comunicación entre elementos, todo es cuestión de saber cómo y en qué momento ejecutar determinada acción. Para el ejemplo rellenaremos el texto del **h1** con el texto “**Hello World**”



```
1  const myTitle = document.querySelector('#h1-title');  
2  myTitle.textContent = 'Hello World';
```