

¿Que es JavaScript?

- JavaScript es un lenguaje de programación que se utiliza principalmente para aportar dinamismo a sitios y aplicaciones web. Funciona en complemento con los lenguajes web HTML y CSS, permitiendo crear nuevas funcionalidades e interacciones avanzadas con los usuarios, así como interactuar con otras aplicaciones o servicios backend.

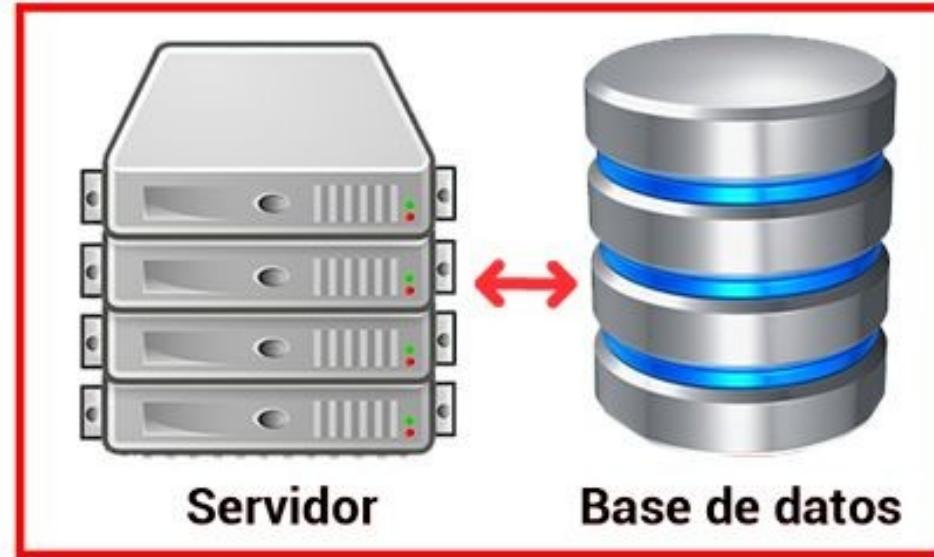
Lenguaje Interpretado

- A diferencia de un lenguaje de programación compilado (como Python), Javascript es un lenguaje interpretado, lo que significa que se ejecuta por medio de un programa interprete.
- Los navegadores son los programas que se encargan de interpretar y ejecutar el código Javascript. Cuando se accede a alguna pagina atraves del navegador, el se encarga de leer y ejecutar todos los archivos que conforman la aplicación (HTML, CSS, JS).

Frontend



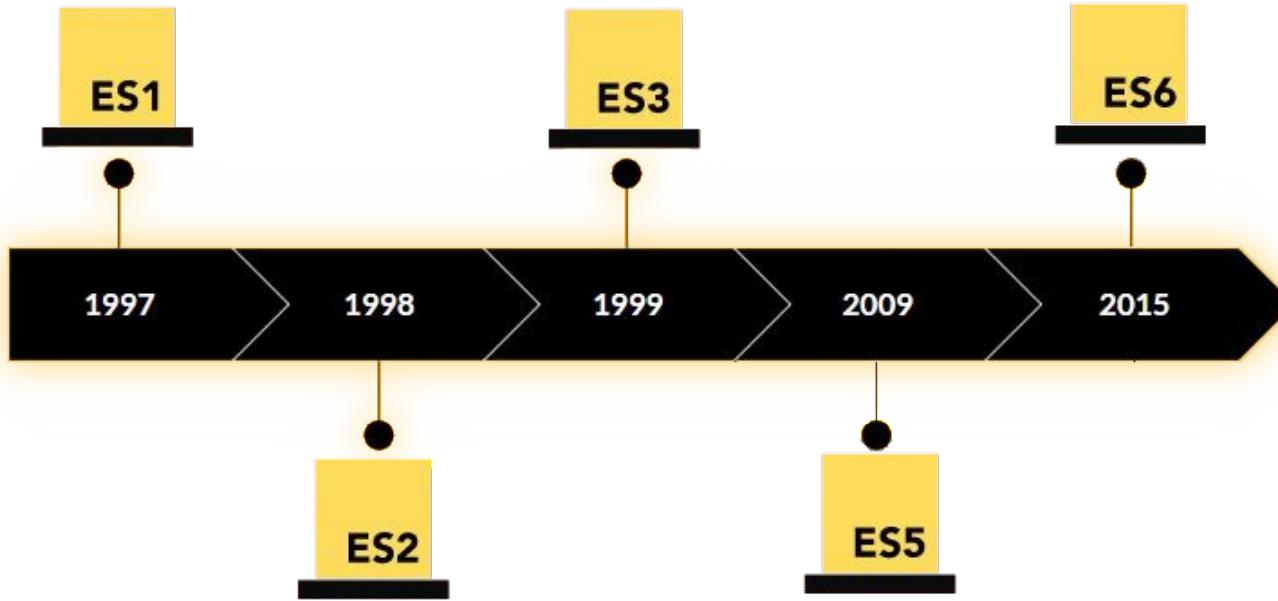
Backend



Backend y Frontend

- JavaScript se usa tanto para construir aplicaciones de Frontend como de Backend. Por Frontend se entiende a la parte de la aplicación que corre en el navegador y con la cual interactúan los usuarios.
- Las aplicaciones Frontend también consumen datos y servicios ofrecidos por algún Backend.

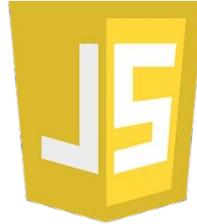
Evolución de JavaScript



La primera versión de Javascript estandarizado se lanza en 1997 y fue cambiando con el tiempo. Cada año el estándar **ES** lanza una versión nueva.

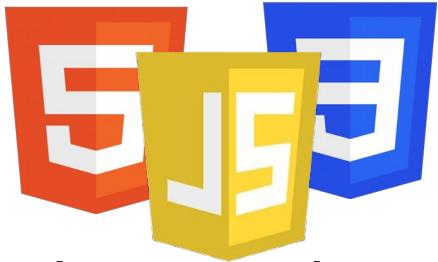


Diferencia con Java



- Si bien, tanto Java como JavaScript son lenguajes de programación que comparten base de su sintaxis, el uso de cada uno es distinto.
- Java funciona a través de declaraciones que se establecen a partir de un sistema de clases en tiempo de compilación. En cambio, JavaScript se basa en un sistema de tiempo de ejecución donde cada dato distinto puede representar valores booleanos, de cadena o numéricos.
- Conclusión: son dos lenguajes diferentes y no es correcto mencionarlos como sinónimos

Relación entre HTML, CSS y JS



- HTML aporta a través de sus etiquetas, la estructura básica de un sitio web. Funciona en consonancia y en relación con CSS y JS.
- CSS se utiliza para aplicarle estilos al sitio web a través de la definición de formato y diseño.
- JavaScript usa ambos lenguajes (HTML y CSS), además de su propia lógica, para controlar el comportamiento y funcionalidad de los elementos del sitio web

¿Como escribir código JS?

- Desde el año 1998 la [W3C](#) definió que HTML, CSS y JS, tuvieran sus archivos independientes, por lo tanto, el código JS debe escribirse en un archivo aparte. El agregado de código JS dentro de un archivo HTML es una mala practica inaceptable en cualquier empresa de software.
- Se vincula con la etiqueta **<script>** y el atributo “src”, y se referencia el archivo.js dentro del apartado **<body>** del documento HTML.

```
  ,head
  <body>
    |   <script src="archivo.js"></script>
  </body>
```

Variables

- Las variables son un espacio reservado en la memoria que, como su nombre indica, puede cambiar de contenido a lo largo de la ejecución de un programa.
- En las variables se almacenan diversos tipos de datos que se usan en la aplicación tales como: numéricos, texto, booleanos.

Inicializar variables

- Se puede declarar una variable y asignarle un valor inicial en el mismo proceso. Las declaraciones con **let** y **const** tienen controles adicionales para las variables. Principalmente impiden que se puedan crear dos variables con el mismo nombre. Una variable **let** puede recibir múltiples asignaciones. Una variable **const** recibe una única asignación al momento de su declaración

```
//declaro y asigno variables:  
  
const nombre= "Maria"  
  
let edad = 30
```

```
//se puede inicializar con un valor,  
//y luego asignarle otro  
let edad = 30  
edad=35  
  
const nombre= "Maria"  
//no se puede cambiar el valor  
const nombre= "Lucas"
```

Funciones nativas

- **prompt(), alert(), confirm() y console.log()**
- Las funciones nativas son sentencias específicas del lenguaje Javascript, con instrucciones específicas y vienen dadas por defecto. Cada vez que utilizamos una de estas funciones podemos hacer una consulta al usuario con un espacio para responder, enviar un alerta, dar la opción de confirmar o no una acción y realizar una consulta por la consola, respectivamente.

Funciones nativas: prompt(), console.log(), alert(), confirm()

```
//funciones nativas  
  
prompt("ingrese su nombre por favor");
```

⊕ 127.0.0.1:5500

ingrese su nombre por favor

Maria

Cancelar

Aceptar

```
alert("¡Bienvenido a mi pagina web!");
```

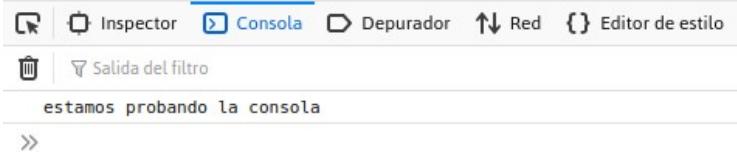
⊕ 127.0.0.1:5500

¡Bienvenido a mi pagina web!

No permitir que 127.0.0.1:5500 pregunte nuevamente

Aceptar

```
console.log("estamos probando la consola");
```



```
confirm("¿necesitas ayuda?");
```

⊕ 127.0.0.1:5500

¿necesitas ayuda?

No permitir que 127.0.0.1:5500 pregunte nuevamente

Cancelar

Aceptar

Operadores

OPERADORES DE COMPARACIÓN	DESCRIPCIÓN	EJEMPLO
<code>==</code>	Es igual	<code>a == b</code>
<code>===</code>	Es estrictamente igual	<code>a === b</code>
<code>!=</code>	Es distinto	<code>a != b</code>
<code>!==</code>	Es estrictamente distinto	<code>a !== b</code>
<code><, <=, >, >=</code>	Menor, menor o igual, mayor, mayor o igual	<code>a <= b</code>
OPERADORES LÓGICOS	DESCRIPCIÓN	EJEMPLO
<code>&&</code>	Operador AND (y)	<code>a && b</code>
<code> </code>	Operador OR (o)	<code>a b</code>
<code>!</code>	Operador NOT (no)	<code>!a</code>

El uso de “estRICTAMENTE”, no solo valida que los valores comparados sean o no iguales, sino que también valida que el tipo de dato sea o no igual.

`(1 == "1")` //esto da como resultado: TRUE

`(1 === "1")` //esto da como resultado: FALSE

Esto se debe, en parte, a que JS posee un tipo débil de datos.