Programación Web – 5 JavaScript

Por: Ing. Luis Daniel Lepe Rodríguez

JavaScript

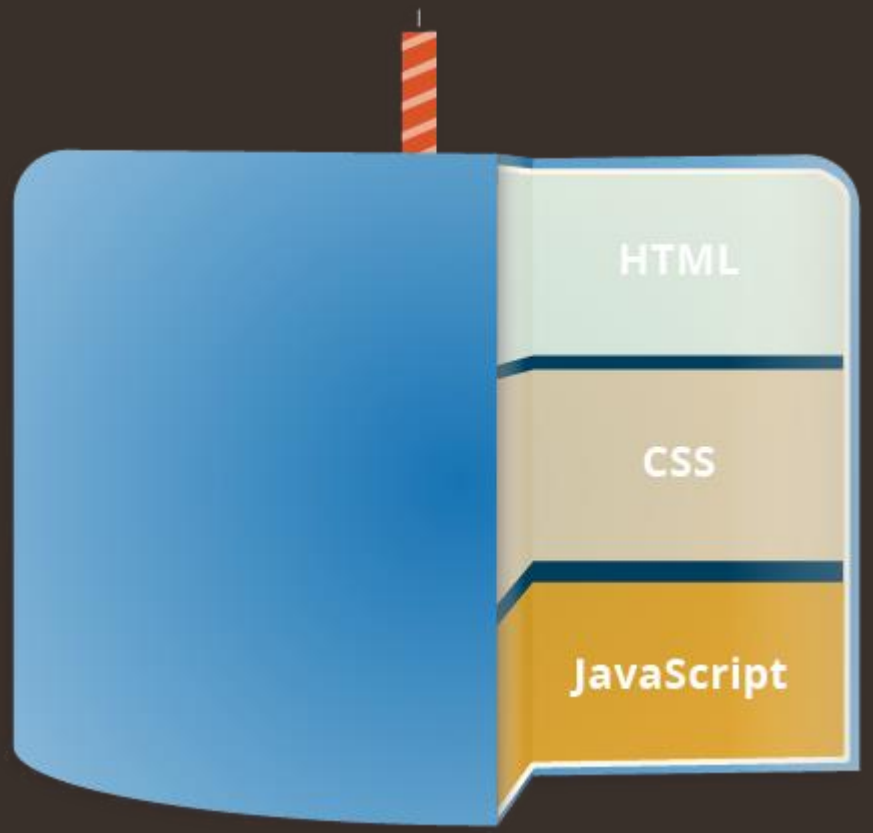
Es un lenguaje de programación a diferencia de HTML y CSS.

Es un lenguaje de programación interpretado, su principal uso es el de hacer páginas web interactivas.

JavaScript utiliza la ***programación basada en prototipos***, el cual es un estilo de POO en la cual los objetos no se definen por clases sino por clonación de otros objetos o la escritura de código del programador.

JavaScript

- JavaScript es un lenguaje de programación que te permite implementar cosas complejas en tu página web, cada vez que veas una página web que haga más que solo desplegar información estática para que tu la veas sentado probablemente significa que se está utilizando JavaScript, desplegar actualizaciones periódicas, mapas interactivos, animaciones, etc.
- JavaScript es la tercera capa del pastel de las tecnologías estándares web.



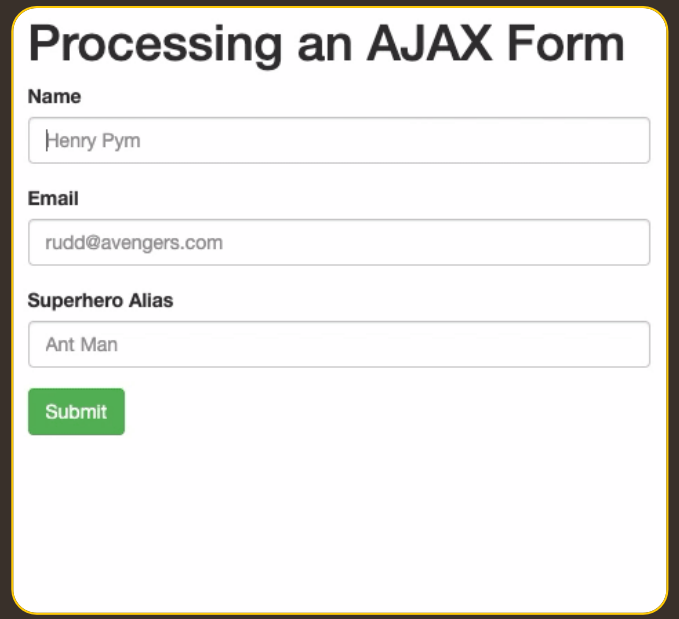
What is JavaScript? - Learn web development | MDN (mozilla.org)

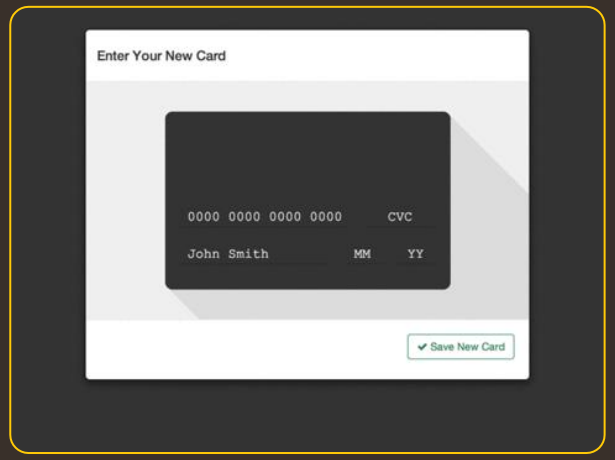
Lenguaje interpretado

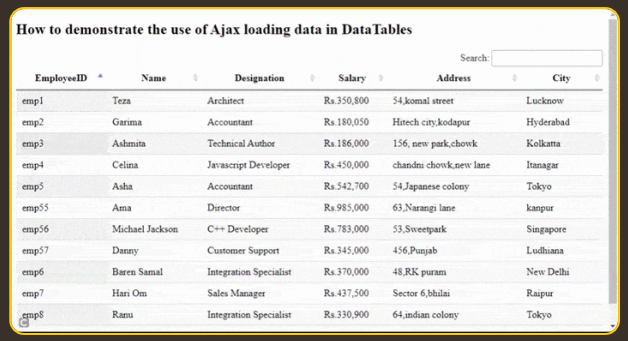
- JavaScript es en lenguaje interpretado, quiere decir que no necesita un compilador sino que se irá ejecutando durante la marcha a través de un interprete (en este caso el navegador) que irá generando las instrucciones línea por línea.

Páginas interactivas

- Con JavaScript puedes generar páginas web interactivas como cambiar diseños dependiendo de condiciones complejas, crear validaciones del lado del cliente, mandar y recibir información del servidor, etc.







¡Aclaración!

≠

Historia

- JavaScript fue creado alrededor del año 1997. Antes de la creación de JavaScript todas las páginas web eran simplemente documentos donde tu consultabas texto y te sentabas a leer, la interacción era casi nula.
- El internet empezó a crecer y empezaron a crearse las primeras aplicaciones web. Ante el obstáculo de un internet tan lento (30 kbps la media) se empezó a necesitar una manera de ejecutar lógica de programación del lado del usuario. Así si el usuario escribía un campo incorrectamente, no se le hacía esperar mucho tiempo hasta que el servidor respondiera indicando los errores del formulario.

Historia

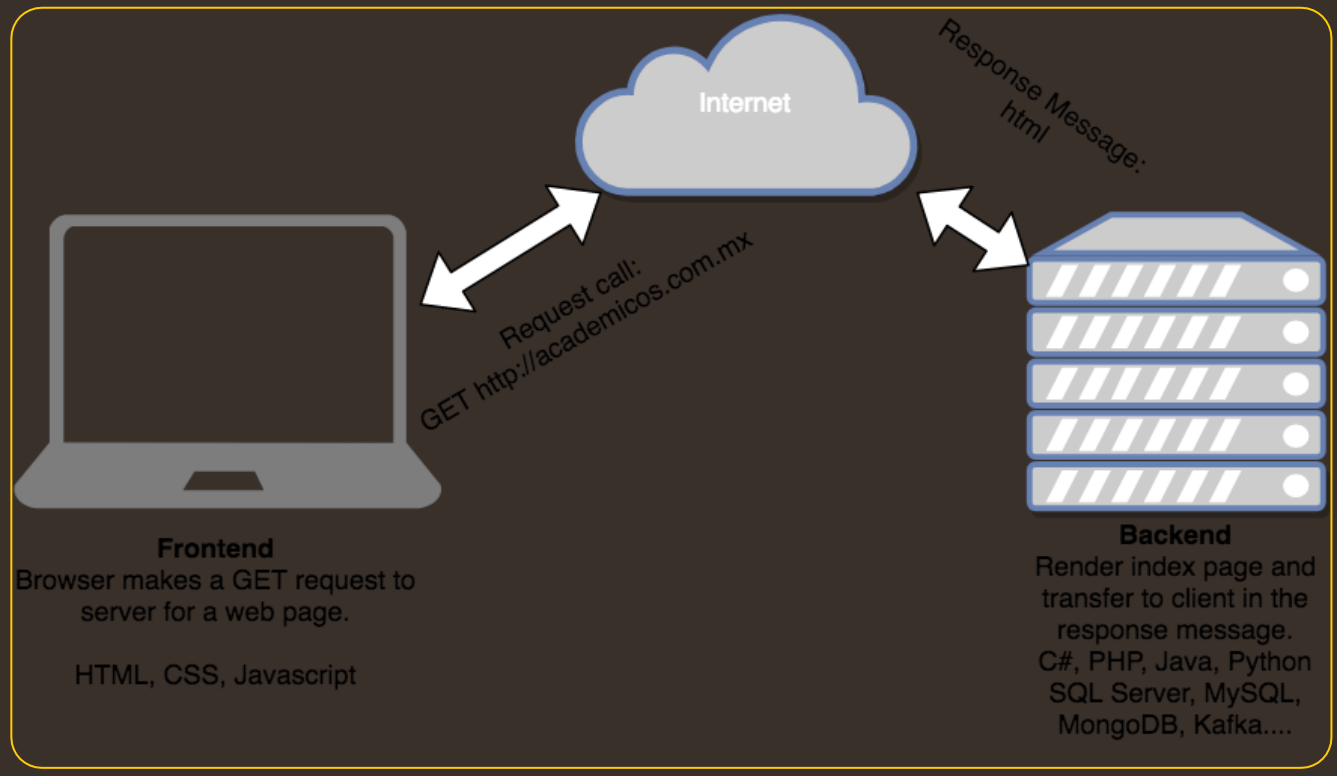
- **Brendan Eich**, un programador que trabajaba en Netscape, pensó que podría solucionar el problema adaptando otras tecnologías existentes al navegador Netscape, a este lenguaje que se generó de las adaptaciones de Eich se le llamó LiveScript.
- Después Netscape firmo alianzas con Sun Microsystems para terminar de desarrollar la tecnología y decidieron cambiarle el nombre a JavaScript. La razón del cambio fue simplemente por marketing ya que en aquellos tiempos la palabra "Java" estaba de moda en el mundo de la informática, también es la razón por la que Java y JavaScript comparten la palabra en sus nombres.

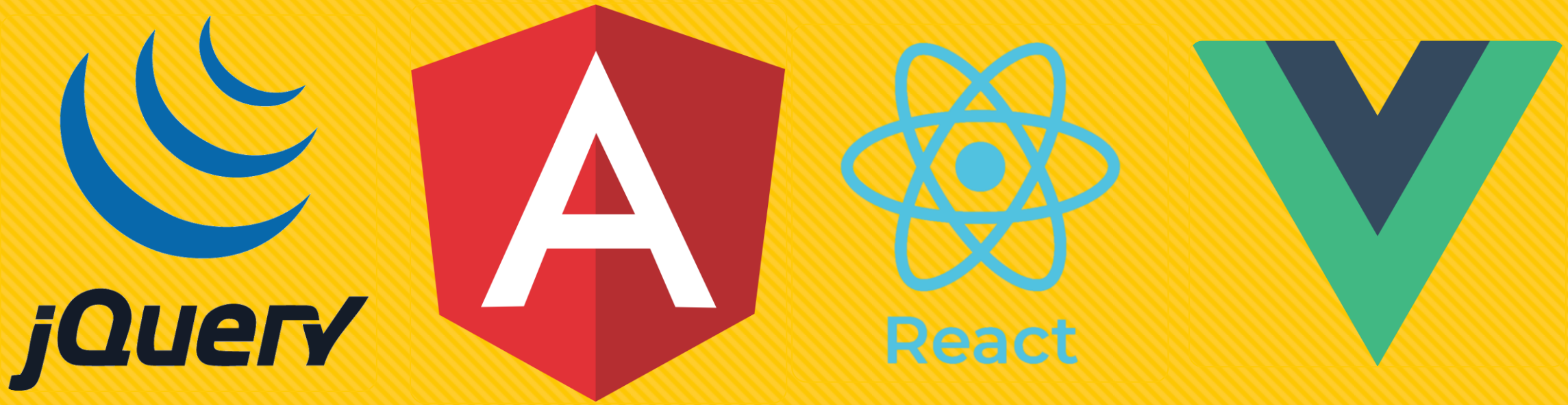
Historia

- Microsoft al ver la funcionalidad de JavaScript la copió e integró en su navegador Internet Explorer bajo el nombre de Jscript para evitar problemas legales. Al ver esto Netscape decidió estandarizar el lenguaje y evitar una guerra de tecnologías. De esta forma se creó en 1997 la especificación de JavaScript 1.1 al organismo ECMA (European Computer Manufacturers Association).
- ECMA creó el comité TC39 con el objetivo de ***"estandarizar de un lenguaje de script multiplataforma e independiente de cualquier empresa"***. El primer estándar que creó el comité TC39 se denominó **ECMA-262**, en el que se definió por primera vez el lenguaje ECMAScript.

Código Server-side vs Client-side

- Habrás escuchado alguna vez los términos **server-side** y **client-side**, especialmente en el contexto del desarrollo web. El código del lado del cliente es código que corre en la computadora del usuario – cuando una página es visitada, el código del lado del cliente de la página es descargado luego ejecutado y desplegado en el navegador.
- El código del lado del servidor por el contrarios se ejecuta en el servidor, luego los resultados son descargados y desplegados por el navegador. Ejemplos de lenguajes para desarrollo web del lado del servidor son PHPm Python, Ruby y ASP.NET. ¡Pero también JavaScript! JavaScript puede se utilizado como lenguaje del lado del servidor en un ambiente conocido como Node.js.





Librerías de JavaScript

¿Cómo utilizar JavaScript?

- JavaScript se enlaza al HTML de manera muy similar al CSS. Mientras el CSS utiliza <link> para enlazar hojas de estilo externas y <style> para aplicar hojas de estilos internas, JavaScript solo necita un amigo en el mundo del HTML – La etiqueta <script>.

```
<script>
   //El código JavaScript va aquí
</script>
```

What is JavaScript? - Learn web development | MDN (mozilla.org)

Variables

Hay 3 maneras de crear una variable de JavaScript:

- Usando `var`
- Usando `let`
- Usando `const`

```
<script>
    var x = 1;
    let y = 2;
    const PI = 3.1415926;
</script>
```

JavaScript Variables – w3school

Javascript Datatypes

JavaScript es un lenguaje dinámico o de tipo flexible. Las variables en JavaScript no están asociadas directamente con ningún tipo de valor en particular, y a cualquier variable se le pueden asignar (y reasignar) valores de todos los tipos.

El último estándar ECMAScript define siete datos tipos:

- Boolean type
- Null type
- Undefined type
- Number type
- BigInt type
- String type
- Symbol type

Y los Objects (collections of properties)

```
<script>
    let foo = 42;    // foo is now a number
    foo = 'bar';     // foo is now a string
    foo = true;      // foo is now a boolean
</script>
```

JavaScript data types and data structures – Learn web development | MDN (mozilla.org)

Matemáticas básicas

```
<script>
        let num1 = 10 + 7
        let num2 = 9 * 8
        let num3 = 60 % 3
        let num4 = 9 * num1;
        let num5 = num1 ** 3;
        let num6 = num2 / num1;
        num1++;
        num1--;
    </script>
```

Operator	Name	Purpose	Example
+	Addition	Adds two numbers together.	6 + 9
-	Subtraction	Subtracts the right number from the left.	20 - 15
*	Multiplication	Multiplies two numbers together.	3 * 7
/	Division	Divides the left number by the right.	10 / 5
%	Remainder (sometimes called modulo)	Returns the remainder left over after you've divided the left number into a number of integer portions equal to the right number.	8 % 3 (returns 2, as three goes into 8 twice, leaving 2 left over).
**	Exponent	Raises a base number to the exponent power, that is, the base number multiplied by itself, exponent times. It was first Introduced in EcmaScript 2016.	5 ** 2 (returns 25, which is the same as 5 * 5).

Basic math in JavaScript — numbers and operators - Learn web development | MDN (mozilla.org)

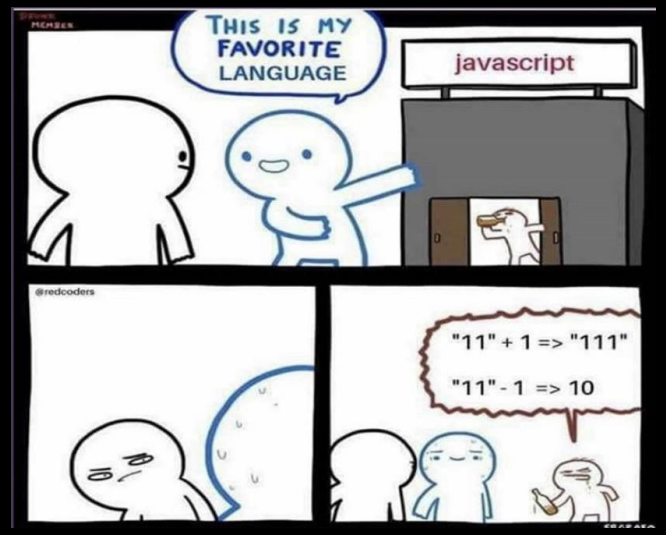
Condicionales

Para las condicionales tenemos las siguientes opciones:

- **===** y **!==** Para probar si un valor es idéntico o no idéntico a otro.
- **<** y **>** Para probar si un valor es menor que o mayor que otro.
- <= y >= Para probar si un valor es menor o igual que y mayor o igual que otro.

```
<script>
    var cheese = 'Cheddar';
    if (cheese === 'Cheddar') {
        console.log('Yay! Cheese available for making cheese on toast.');
    } else {
        console.log('No cheese on toast for you today.');
    }
</script>
```

Making decisions in your code — conditionals - Learn web development | MDN (mozilla.org)




```
> "b" + "a" + NaN + "a"
"ba" + NaN + "a"
"baNaN" + "a"
"baNaNa"
```

Funciones

- Otro concepto esencial en la codificación son las funciones, que le permiten almacenar un fragmento de código que hace una sola tarea dentro de un bloque definido, y luego llamar a ese código cuando lo necesite usando un solo comando corto, en lugar de tener que escribir el mismo.

```
<script>
    function random(number) {
        return Math.floor(Math.random() * number);
    }
</script>
```

Functions — reusable blocks of code - Learn web development | MDN (mozilla.org)

Bucles

```
<script>
    for (let i = 0; i < 100; i++) {
        // code a ejecutar
    }
    for (initializer; condition; final-expression) {
        // code a ejecutar
    }
</script>
```

Looping code - Learn web development | MDN (mozilla.org)

Manipulación del DOM

- Al escribir páginas web y aplicaciones, una de las cosas más comunes que querrá hacer es manipular la estructura del documento de alguna manera. Por lo general, esto se hace mediante el Document Object Model (DOM), un conjunto de API para controlar HTML y la información de estilo que hace un uso intensivo del objeto de Document.

```
<script>
    var x = document.getElementById("demo"); // Obtiene el elemento con id="demo"
    x.style.color = "red"; // Cambia el color del elemento
</script>
```

Manipulating documents - Learn web development | MDN (mozilla.org)

Tarea

- Hacer una presentación sobre qué es JSON y XML para que sirven, en donde se utilizan, sintaxis, ejemplos, datos históricos, etc. Describir las diferencias principales entre JSON y XML.
- Completar el ejercicio encargado en clase.
- Hacer un resumen sobre lo visto en clase.