JavaScript

Daniela Ospina Salamanca

*Facultad de ingenierías, Universidad Tecnológica de Pereira, Colombia*

Correo-e: daniela.ospina4@utp.edu.co

***Resumen*—**

**Los documentos HTML permiten incrustar fragmentos de código JavaScript, bien dentro del propio archivo HTML o bien realizando una carga de ese código indicando el archivo donde se encuentra el código JavaScript. Dentro de un documento HTML puede haber ninguno, uno o varios scripts de JavaScript. El uso de JavaScript es muy habitual en la programación web. En este curso no entramos a explicar el lenguaje de programación JavaScript, pero hemos considerado adecuado dejar indicado que JavaScript es un lenguaje que la mayor parte de los programadores web conocen porque permite hacer muchas cosas que no se pueden hacer con HTML.**

***Palabras clave—***

***Código, programación web, HTML, lenguaje.***

***Abstract*—**

**HTML documents allow you to embed JavaScript code fragments, either inside the HTML file itself or to load that code indicating the file where the JavaScript code is located. There may be none, one or several JavaScript scripts within an HTML document. The use of JavaScript is very common in web programming. In this course there is no explanation about the language of JavaScript programming, but we have been adequate to stop using it is a language for most programmers.**

***Key Word***

***Code, Web programing, HTML, Language***

1. INTRODUCCIÓN

JavaScript, al igual que Flash, Visual Basic Script, es una de las múltiples maneras que han surgido para extender las capacidades del lenguaje HTML (lenguaje para el diseño de páginas de Internet). Al ser la más sencilla, es por el momento la más extendida.

Javascript es un **lenguaje de programación** que permite llevar a cabo actividades tanto simples como complejas en páginas web. Es un sistema que goza de la peculiaridad de no necesitar compilación alguna, ya que son los propios navegadores los que se encargan de leer el código para asimilarlo y llevar a cabo las acciones que este indica. A día de hoy es el estándar cuando se quieren plantear mecanismos más complicados de lo habitual en una web.

Es un error habitual confundir este lenguaje, sobre todo por parte de los menos expertos en la materia, con Java. Debido a las similitudes entre nombres, es fácil caer en la confusión de que ambos guardan algún tipo de relación, que tienen funciones similares o que uno es un subconjunto del otro; pero no tienen conexión alguna más allá de la casualidad en su nombre. Son completamente distintos y, de hecho, Java nació en el 91, mientras que Javascript hizo lo mismo varios años más tarde.

Ambos se elaboraron con fines totalmente distintos, aunque es cierto que el lenguaje orientado al entorno web, el que nos ocupa, tiene ese nombre porque sus creadores intentaron aprovechar la seriedad y popularidad de Java para arrojar un aire de confianza y profesionalidad con esta forma de programar tan joven en su momento.

A día de hoy, Javascript puede presumir de ser el lenguaje para programar en web más importante que ha existido, debido sobre todo al grado de interactividad que puede incluir en cualquier página en la que se utilice correctamente.

1. CONTENIDO

**1) Bases del lenguaje**.

A continuación veremos algunas de las funciones básicas del lenguaje JavaScript para que puedas comprender mejor cómo funciona todo. Mejor aún, estas características son comunes para todos los lenguajes de programación. Si puedes entender esos fundamentos, deberías ser capaz de comenzar a programar en casi cualquier cosa.

* **Variables:**

Las variables son contenedores en los que puedes almacenar valores. Primero debes declarar la variable con la palabra clave var, seguida del nombre que le quieras dar:

var nombreDeLaVariable;

Tras declarar una variable, puedes asignarle un valor:

nombreDeLaVariable = 'Bob';

Puedes hacer las dos cosas en la misma línea si lo necesitas:

var nombreDeLaVariable = 'Bob';

Puedes obtener el valor de la variable llamándola por su nombre:

nombreDeLaVariable;

Después de haberle dado un valor a la variable, puedes volver a cambiarlo:

var nombreDeLaVariable = 'Bob'; nombreDeLaVariable = 'Steve';

* **Comentarios:**

Puedes escribir comentarios entre el código JavaScript, igual que puedes en CSS. En JavaScript, los comentarios de una sola línea se escriben así:

// Esto es un comentario

Pero también puedes escribir comentarios en más de una línea, igual que en CSS:

/\* Esto es un comentario de varias líneas. \*/

* **Operadores:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operador** | **Explicación** | **Símbolo** | **Ejemplo** |
| **Suma**  **/concatenación** | Se usa para sumar dos números, o juntar dos cadenas en una. | + | 6 + 9;  "Hola " + "mundo!"; |
| **Resta, multiplicación, división** | Éstos hacen lo que esperarías que hicieran en las matemáticas básicas. | -, \*, / | 9 - 3;  8 \* 2; // La multiplicación en JS es un asterisco  9 / 3; |
| **Operador de asignación** | Los has visto anteriormente: asigna un valor a una variable. | = | var miVariable = 'Bob'; |
| **identidad**  **/igualdad** | Comprueba si dos valores son iguales entre sí, y devuelve un valor de true/false (booleano). | === | var miVariable = 3;  miVariable === 4; |
| **Negación, distinto**  **(no igual)** | En ocasiones utilizado con el operador de identidad, la negación es en JS el equivalente al operador lógico NOT — cambia true por false y viceversa. | !, !== | a expresión básica es true, pero la comparación devuelve falseporque lo hemos negado:  var miVariable = 3;  !miVariable === 3;  Aquí estamos comprobando "es miVariableNO igual a 3". Esto devuelve false, porque ES igual a 3.  var miVariable = 3;  miVariable !== 3; |

* **Condicionales:**

Las Condicionales son estructuras de código que permiten comprobar si una expresión devuelve true o no, y después ejecuta un código diferente dependiendo del resultado. La forma de condicional más común es la llamada if ... else. Entonces, por ejemplo:

var helado = 'chocolate'; if (helado === 'chocolate') { alert('¡Si, amo el helado de chocolate!'); } else { alert('Awwww, pero mi favorito es el de chocolate...'); }

La expresión dentro de if ( ... ) es el criterio — este usa al operador de identidad (descrito arriba) para comparar la variable helado con la cadena chocolate para ver si las dos son iguales. Si esta comparación devuelve true, el primer bloque de código se ejecuta. Si no, ese código se omite y se ejecuta el segundo bloque de código después de la declaración else.

* **Funciones:**

Las funciones son una manera de encapsular una funcionalidad que quieres reutilizar, de manera que puedes llamar esa función con un solo nombre, y no tendrás que escribir el código entero cada vez que la utilices.

Ya has visto algunas funciones más arriba, por ejemplo:

1. var nombreDeLaVariable = document.querySelector('h1');
2. alert('Hola!');

Estas funciones están integradas en el navegador para poder utilizarlas en cualquier momento.

Si ves algo que parece un nombre de variable, pero tiene paréntesis — () — al final, probablemente es una función. Las funciones con frecuencia toman [arguments](https://developer.mozilla.org/es/docs/Glossary/Argumentos) — pedazos de datos que necesitan para hacer su trabajo..

Por ejemplo, la función alert() hace aparecer una ventana pop-up dentro de la ventana del navegador, pero nosotros necesitamos asignarle una cadena como un argumento para decirle qué mensaje se debe escribir en la ventana pop-up.

Las buenas noticias son que podemos definir nuestras propias funciones — en el siguiente ejemplo escribimos una función simple que toma dos números como argumentos y los multiplica entre si:

function multiplica(num1,num2) { var resultado = num1 \* num2; return resultado; }

Trata de ejecutar la función anterior en la consola. Después trata de usar la nueva función algunas veces, p.ej:

multiplica(4,7); multiplica(20,20); multiplica(0.5,3);

* **Eventos:**

Para crear una interacción real en tu sitio WEB, debes usar eventos — Estos son unas estructuras de código que captan lo que sucede en el navegador, y permite que en respuesta a las acciones que suceden se ejecute un código. El ejemplo más obvio es un click (click event), que se activa al hacer click sobre algo. Para demostrar esto, prueba ingresando lo siguiente en tu consola, luego haz click sobre la pagina actual:

document.querySelector('html').onclick = function() { alert('Ouch! Deja de pincharme!'); }

Hay muchas maneras de enlazar un evento a un elemento; aquí hemos seleccionado el elemento HTML y le asignamos a su propiedad onclick una función anónima (función sin nombre) que contiene el código que se ejecutará cuando el evento suceda.

Nota que

document.querySelector('html').onclick = function(){};

es equivalente a

var miHTML = document.querySelector('html'); miHTML.onclick = function(){};

es solo un modo mas corto de escribirlo

**2) Ejemplos**

* Nuestro primer programa será el famoso "Hola Mundo", es decir un programa que muestre en el documento HTML el mensaje "Hola Mundo".

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Ejemplo de JavaScript</title>**

**<meta charset="UTF-8">**

**</head>**

**<body>**

**<script>**

**document.write('Hola Mundo');**

**</script>**

**</body>**

**</html>**

* Mensaje de alerta

<html>

<head>

<meta charset="utf-8">

<title>Embeber JavaScript – aprenderaprogramar.com</title>

<script type="text/javascript" src="archivo.js"></script>

</head>

<body>

Página web de prueba – aprenderaprogramar.com

</body>

</html>

* Mensaje de prueba

html>

<head>

<title>Neoattack.com – Usando JavaScript</title>

</head>

<body>

<script type = “text/javascript”>

document.write (‘Texto de prueba de NeoAttack en JS’);

</script>

</body>

</html>

**4) Aplicaciones**

Javascript sirve para que en una página web se consiga un mayor índice de interactividad con los usuarios y, por tanto, la experiencia de estos sea mucho mejor y más enriquecida. Es algo que se emplea para dotar de más funciones, facilitar la comunicación y, además, otorgar todas las mejoras posibles sin apenas incidir en el peso de la página, lo que garantiza una mayor velocidad de carga y una mejor reputación a ojos de Google.

Para entender lo que es JavaScript consideremos lo siguiente. Un usuario escribe una dirección web en su navegador, por ejemplo http://www.aprenderaprogramar.com. El servidor recibe la petición y como respuesta a esa petición envía al ordenador del usuario código HTML junto a código JavaScript. El código HTML se encarga de que en la pantalla se muestre algo, por ejemplo una imagen, un menú, etc. El código JavaScript se puede encargar de crear efectos dinámicos en respuesta a acciones del usuario, por ejemplo que se despliegue un menú tipo acordeón cuando el usuario pasa el ratón por encima de un elemento del menú.

La ventaja de JavaScript es que al estar alojado en el ordenador del usuario los efectos son muy rápidos y dinámicos. Al ser un lenguaje de programación permite toda la potencia de la programación como uso de variables, condicionales, bucles, etc. También podemos citar algún inconveniente: por ejemplo si el usuario tiene desactivado JavaScript en su navegador, no se mostrarán los efectos. No obstante, hoy día la mayoría de los usuarios navegan por la web con JavaScript activado.

**5) Conclusiones**

No es nada nuevo de que la gente hable constantemente de un “developer multiplataforma”.

En algún momento JAVA fue la mejor solución para todo esto, mas alla del sistema operativo podríamos hacer aplicaciones de escritorio ya que java corre sobre una maquina virtual (JVM) y esa misma maquina virtual tambien estaba en una version “minimalista” en algunos celulares (J2Me).

Pero como todo en el mundo, y por sobre todo en la tecnología, los avances son cada vez mas y mas rápidos.

Entonces ya no bastaba con ser un programador JAVA para ser un multi-plataforma, las tecnologias fueron avanzando como así la WEB y teníamos 2 posibilidades, FLASH o JAVASCRIPT, de esta forma se hacían conexiones en la web y éramos interactivos, mas alla que con JSP (javaserver page) o PHP, podríamos interactuar con el cliente, las paginas en su contenido eran estáticas.

Con la aparición de Flash y Javascript se empezó a interactuar con las paginas y como así tambien se empezó a poner movimientos y conexiones con el backend de mejor forma.

No es un secreto de estado que Javascript gano ampliamente esta pelea por ser el movimiento del frontend ya que con la salida del IPad murió flash poco a poco.

Entonces, de ser un simple conector entre backend y frontend, poco a poco fue ganando funciones y seguidores en la web, los menus y movimientos se dejaron de hacer sobre flash y se empezaron a hacer sobre javascript, vale aclarar que fue un gran beneficio (aunque vino de la mano) HTML5 CSS3, para que javascript sea el lenguaje del frontend, con grandes framework iniciando por JQuery y con cosas tan poderosas como AngularJS React y otros.

REFERENCIAS

1. ¿Qué es JavaScript?. [Online].

Available: <https://www.tutorialesprogramacionya.com/javascriptya/temarios/descripcion.php?inicio=0&cod=2>

1. (2019, Mar.). Fundamentos de JavaScript

[Online].

Available: <https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics>

1. ¿Qué es y para qué sirve JavaScript?

Online].

Available: <https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=590:ique-es-y-para-que-sirve-javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69&Itemid=192>

1. .Concepto de JavaScript

[Online].

Available:

<https://neoattack.com/neowiki/javascript/>

1. J. Cano (2016, Ene.). La importancia de JavaScript [Online].

Available:

<https://medium.com/@jorgeucano/la-importancia-de-javascript-914f5cf8dc68>