

Exposición

Grupo #3

- *Leanne Caba #7
- *Adrian Cordero #12
- *Elianna Jimenez #22
- *Zoilibel Novas #27
- *Isaac Perez #33
- *Keyver Reyes #34
- *Jose Rosado #35

An abstract background pattern of blue and yellow lines resembling a circuit board or data flow, with some lines ending in small blue dots.

JavaScript

Definicion de JavaScript:



JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, **JavaScript** (abreviado comúnmente **JS**) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Historia de JavaScript

Se trata de un lenguaje de programación que permite realizar acciones de diversos grados de complejidad en sitios web sin necesidad de compilación. El hecho de que los mismos navegadores lean y asimilen el código para efectuar las instrucciones indicadas por éste, ha convertido a JavaScript en un lenguaje de programación muy utilizado y apreciado por los desarrolladores.

A día de hoy, es el lenguaje en el que se desarrolla una página web cuando los mecanismos que se desean implantar encierran un mayor nivel de dificultad de los que se suelen plantear a menudo. Pero, antes de continuar hablando de la actualidad de JavaScript, vamos a hacer un breve regreso al pasado para entender qué circunstancias dieron origen a este lenguaje de programación.

Esa época en la que usábamos los módems que se conectaban a la línea telefónica, fue también el momento en el que a las aplicaciones para web se le empezaba a añadir complejidad. Esto se trasladó a las páginas web, que incluían formularios cada vez más completos y complicados, lo que afectó a la calidad de la navegación online.

Con una velocidad de navegación tan lenta y unas aplicaciones que avanzaban sin parar, resultó evidente que HTML se estaba quedando corto, así que los programadores tuvieron que pararse a pensar cómo solucionar este desfase. La respuesta llegó en forma de un lenguaje de programación que podía ejecutarse directamente en el navegador del lado del usuario.

Adaptando tecnologías de programación ya existentes, el navegador Netscape desarrolló LiveScript, un lenguaje que permitía crear programas pequeños en las páginas y que fuese más sencillo que Java, creado por Sun Microsystems. Ambas compañías, Netscape y Sun Microsystems, unieron fuerzas y conocimiento para desarrollar juntas la programación JavaScript.

Y así, en líneas muy generales, nació un lenguaje de desarrollo web fácil de utilizar y accesible a personas con nociones básicas de programación o, incluso, con ningún conocimiento previo en la materia, lo cual explica por qué la popularidad de JavaScript gozó de tan buena salud desde sus comienzos y también por qué lo sigue haciendo a día de hoy

Origen:

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó compatibilidad con la tecnología Java en su navegador web Netscape Navigator en la versión 2002 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en el ámbito de los nuevos lenguajes de programación web.



Elemento de JavaScript:

Posee algunos objetos predefinidos u objetos intrínsecos como son: Array, Boolean, Date, Function, Global, Math, Number, Object, RegExp, y String. Además, es posible crear objetos nuevos, con sus propios métodos y propiedades, adaptados a las necesidades concretas de cada aplicación

- **Array**

El objeto Array de JavaScript es un objeto global que es usado en la construcción de arrays, que son objetos tipo lista de alto nivel.

Los arrays son objetos similares a una lista cuyo prototipo proporciona métodos para efectuar operaciones de recorrido y de mutación. Tanto la longitud como el tipo de los elementos de un array son variables. Dado que la longitud de un array puede cambiar en cualquier momento, y los datos se pueden almacenar en ubicaciones no contiguas, no hay garantía de que los arrays de JavaScript sean densos; esto depende de cómo el programador elija usarlos. En general estas características son cómodas, pero si, en su caso particular, no resultan deseables, puede considerar el uso de arrays con tipo.

- **Date**

Permite trabajar con fechas y horas.

El constructor Date:

- **Math**

Math es un objeto incorporado que tiene propiedades y métodos para constantes y funciones matemáticas. No es un objeto de función. Math funciona con el tipo Number

- **Descripción**

A diferencia de los demás objetos globales, el objeto Math no se puede editar. Todas las propiedades y métodos de Math son estáticos. Usted se puede referir a la constante pi como Math.PI y puede llamar a la función seno como Math.sin(x), donde x es el argumento del método. Las constantes se definen con la precisión completa de los números reales en JavaScript.

- **Number**

Number es un objeto primitivo envolvente que permite representar y manipular valores numéricos como 37 o -9.25. El constructor `Number` contiene constantes y métodos para trabajar con números. Valores de otro tipo pueden ser convertidos a números usando la función `Number()`.

- **Object**

La clase `Object` representa uno de los tipos de datos de JavaScript. Es usado para guardar una colección de datos definidos y entidades más complejas. Los objetos pueden ser creados utilizando el constructor `Object()` o la sintaxis literal de objeto. El constructor `Object` crea una envoltura al objeto.

- **String — Cadena de caracteres**

El objeto **String** se utiliza para representar y manipular una secuencia de caracteres.

- **Descripción**

Las cadenas son útiles para almacenar datos que se pueden representar en forma de texto. Algunas de las operaciones más utilizadas en cadenas son verificar su length, para construirlas y concatenarlas usando operadores de cadena + y +=, verificando la existencia o ubicación de subcadenas con indexOf() o extraer subcadenas con el método substring().

- **Crear cadenas**

Las cadenas se pueden crear como primitivas, a partir de cadena literales o como objetos, usando el constructor String():

Aquí podemos ver 3 ejemplos de cadenas String().

Función de JavaScript (POO)

Un estilo de programación orientada a objetos (POO), en el que la herencia se produce mediante la definición de clases de objetos, en lugar de que la herencia se produzca únicamente a través de los objetos.

El modelo más popular de POO está basado en clases, pero como mencioné, JavaScript no es un lenguaje basado en clases, es un lenguaje basado en prototipos.

Según la documentación de Mozilla:

Un lenguaje basado en prototipos toma el concepto de *objeto prototípico*, un objeto que se utiliza como una plantilla a partir de la cual se obtiene el conjunto inicial de propiedades de un nuevo objeto.

Características de la Programación Orientada a Objeto:

La abstracción

Abstracción denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características.

Encapsulación

La encapsulación significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción.

Herencia

Herencia, las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen.

Modularidad

Modularidad se denomina modularidad a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.

Polimorfismo

Polimorfismo comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.

Ventajas y desventajas:

***Ventajas**

Es de fácil aprendizaje

JavaScript es un lenguaje muy sencillo de dominar, ya que su curva de aprendizaje es baja.

Es muy versátil

JavaScript es un lenguaje estándar en la industria web, por tanto, se puede integrar con otras tecnologías.

Carga del servidor

Como el programa se ejecuta del lado del cliente se reduce la carga en el servidor de la página web.

Crea interfaces dinámicas

Es otra ventaja notable, con JavaScript puedes desarrollar elementos como menús desplegables, botones, formularios de registros, encuestas, agregar efectos al texto, cambiar el color de la fuente, etc.

Es multiplataforma

JavaScript es un lenguaje de programación web multiplataforma, es decir, se ejecuta en distintos sistemas operativos, como Mac, Linux y Windows.

Es compatible con los distintos CMS

JavaScript es un lenguaje muy sencillo de dominar, ya que su curva de aprendizaje es baja.

Se actualiza con frecuencia

Debido a la flexibilidad del lenguaje, empresas como Google, Microsoft han desarrollado frameworks agregando el modelo vista controlador (MVC), esto facilita construir aplicaciones web en poco tiempo.

Desventajas de usar JavaScript

A continuación, te indico las desventajas de usar JavaScript en la programación web.

Sus scripts lo hacen vulnerable

Como es un programa que se ejecuta en el lado del cliente, sus códigos pueden ser leídos por otros usuarios.

Es posible desactivar el JavaScript

A veces los usuarios por desconocimiento pueden desactivar la funcionalidad de JavaScript en el navegador, esto genera que no se ejecuten los códigos dinámicos en la página web.

Sintaxis:

Clases

Las clases de JavaScript, introducidas en ECMAScript 2015, son una mejora sintáctica sobre la herencia basada en prototipos de JavaScript. Las clases de JavaScript proveen una sintaxis mucho más clara y simple para crear objetos y lidiar con la herencia.

o Constructor

El método constructor es un método especial para crear e inicializar un objeto creado con una clase. Solo puede haber un método especial con el nombre "constructor" en una clase. Si esta contiene más de una ocurrencia del método constructor, se arrojará un Error SyntaxError

o this.

La palabra clave `this` de una función se comporta un poco diferente en Javascript en comparación con otros lenguajes.

En general, el valor de `this` está determinado por cómo se invoca a la función. No puede ser establecida mediante una asignación en tiempo de ejecución, y puede ser diferente cada vez que la función es invocada.

- o Subclases con `extends`

La palabra clave `extends` es usada en declaraciones de clase o expresiones de clase para crear una clase hija.

¡Gracias por su atención!