Taller de Ejercicios JS

**Programación web 1**

Profe. Jossy Tello

Aplicar los fundamentos del lenguaje Javascript realizando los siguientes ejercicios.

1. ***Investigue que es ecmascript y cuales son sus actualizaciones recientes.***

ECMAScript específicamente es el **estándar**que a partir del año 2015 a la actualidad se encarga de **regir como debe ser interpretado y funcionar el lenguaje JavaScript**, siendo este (JS – JavaScript) interpretado y procesado por multitud de plataformas, entre las que se encuentran los navegadores web, NodeJS u otros ambientes como el desarrollo de aplicaciones para los distintos sistemas operativos que actualmente existen en el mercado. Los responsables de dichos navegadores y JavaScript deben encargarse de interpretar el lenguaje tal como lo fija ECMAScript.

En el mes de diciembre del año 1995 Brendan Eich uno de los empleados de Netscape diseñó y desarrolló un lenguaje llamado “**Mocha**” posteriormente lo renombro a **LiveScript**hasta que con el transcurso del tiempo debido a que la empresa Netscape fue adquirida por Sun Microsystems como estrategia de marketing se cambió su nombre a lo que hoy en día conocemos como **JavaScript**.

A continuación te daremos un resumen de la evolución que ha tenido **ECMAScript:**

* **Año 1995**: El programador y trabajador Brendan Eich de la empresa Netscape como se mencionó anteriormente creó un lenguaje de programación llamado “Mocha”, posteriormente la empresa Sun Microsystem adquirió Netscape y este pasó a llamarse JavaScript hasta la actualidad.
* **1997**: Nace el estándar Document Object Model (DOM) para evitar las incompatibilidades entre los navegadores. Este mismo año, surge el estándar ECMA-262 y sale al mercado ECMAScript 1.
* **1998**: Sale ECMAScript 2 con la actualización del formato en la especificación para alinearla con los estándares ISO.
* **1999**: Es lanzado ECMAScript 3, cuya novedad era el soporte para expresiones regulares, gestión estructurada de excepciones y otras mejoras puntuales.
* **2009**: Luego de una década nace ECMAScript 5 ya que la versión 4 no llegó a feliz término puesto que le querían agregar una cantidad considerable de cambios y perdería el propósito inicial de este. En ECMAScript 5 se añadió el soporte nativo para JSON o los getters y setters para propiedades, entre otras pequeñas mejoras.
* **2011**: Es puesta en marcha la versión ECMAScript 5.1, esta simplemente alineaba el estándar de ECMA con el formato correspondiente de ISO (ISO/IEC 16262:2011).
* **2015**: Sale a producción ECMAScript 6 la cual llegó con una serie de mejoras de las cuales podemos mencionar la mejora de la sintaxis y actualización de la misma ya que trajo consigo los símbolos, las lambdas y tipos de datos que no existían en las versiones anteriores, así como también fueron mejoradas las estructuras iteración.
* **2016**: ECMAScript 7 esta versión trajo consigo mejoras básicamente el operador de exponenciación y un método nuevo para las matrices que permite comprobar si existen ciertos elementos dentro de éstas.
* **2017**: La octava edición ECMAScript 8 inclyó constructores async/await , los cuales funcionan usando generadores.
* **2018**: ECMAScript 9 agrego los operadores rest/spread para variables.
* **2019**: Este año se agregaron nuevas características entre las cuales podemos destacar las siguientes la función Array.flat() devuelve una nueva matriz con cualquier sub-matriz. Por su parte, String.trimStart() puede utilizarse para recortar el espacio en blanco desde el inicio de una cadena. Por otra parte, para el manejo de errores con try / catch se aplica el “error opcional en catch” ya que permite a los desarrolladores utilizar el try / catch sin la obligatoriedad de aplicar el parámetro de error dentro del bloque correspondiente.

## ES6 vs ES5: Novedades de ECMAScript 6

A continuación te presentamos algunas de las novedades de ECMAScript 6 (ES6) en cuanto a es5 (ECMAScript 5):

* **Función Arrow**: Conocidas como expresiones lambda en C# y Java, arrows o flechas son abreviaciones de funciones utilizando el operador => Por ejemplo:

*// ES5*

var sum = function(y, z){

return y+z;

}

*// ES6*

var sum = (y, z) => y + z;

* El código de la sintaxis de las clases ha sido actualizado de forma que sea **más sencillo y de fácil comprensión** al momento de su implementación.
* Para indicar que una variable solo va a estar definida en un bloque en particular se agregara let a su inicio y al terminar el bloque la variable dejará de existir, siendo de esta forma muy útil a la hora de evitar errores desde el punto de vista lógico, en el momento en que se altere una variable que no debería haberse modificado.
* Las variables constantes van definidas por const en su inicio esto permitirá prevenir que una variable declarada cambie de valor, convirtiéndola efectivamente en una constante.

1. ***Escoja mínimo 5 novedades ecmascript y explique cada novedad en 2 ejemplos.***

Constantes: Me permite crear variables constantes que no cambia su valor (const)

Variables: Me permite crear variables, con un alcance de bloque o scope (let)

Arrow Function: () => {}

Template Strings: Me permite concatenar cadena de string, código HTML entre otro

For EACH: Me permite recorrer un array (forEach)