UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Centro de e-Learning

Diplomatura en programación web full stack con React JS

Centro de e-Learning SCEU UTN - BA.
Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Módulo 2: JavaScript

Unidad 1: El lenguaje JavaScript, variables y estructuras

Presentación:

Comenzamos un nuevo módulo donde empezamos a adentrarnos en el mundo de la programación de la mano del lenguaje JavaScript.

JavaScript en un lenguaje multipropósito, sumamente difundido y es la base de otros lenguajes como NodeJS, Angular y React por lo que aprender JavaScript facilita el aprendizaje de otros lenguajes a la vez que se obtienen nuevas habilidades para mejorar la visualización de las páginas web.

Objetivos:

Que los participantes:

- Comprendan los fundamentos esenciales del lenguaje JavaScript.
- Sean capaces de desarrollar un programa “Hola Mundo”.

Bloques temáticos:

1. Fundamentos del lenguaje JavaScript
2. Introducción a los conceptos básicos de JavaScript
3. Variables en JS
4. Estructuras de datos en JS
5. Ejemplo de programa en JS.”Hola Mundo!”
6. Trabajo Práctico

Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.

** El MEC es el modelo de E-learning colaborativo de nuestro Centro.*

Tomen nota:

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

1. Fundamentos del lenguaje JavaScript

JavaScript surge como lenguaje “del lado del cliente”, es decir, que se ejecuta en el navegador del usuario. Tiempo después, JavaScript fue envuelto en una capa C++ y de esta forma se hizo posible su uso “del lado del servidor” mediante NodeJs.

Cada navegador trae dentro de sí, la capacidad de interpretar y ejecutar el código JavaScript. Desde JavaScript es posible modificar la visualización y comportamiento de la página web que lo incluye mediante el DOM que es la interpretación de la página por parte del navegador.

En el presente documento trataremos la sintaxis del lenguaje JavaScript. El lenguaje JavaScript mantiene una sintaxis similar a lenguajes como C, PHP, JAVA, C#. Si bien la sintaxis es similar, tiene algunas particularidades que iremos viendo en el documento.

2. Introducción a los conceptos básicos de JavaScript

JavaScript es un lenguaje de programación orientado a objetos, donde no hay distinción entre los tipos de objetos. Es un lenguaje con tipado dinámico donde la responsabilidad de la asignación del tipo no es del desarrollador sino que el lenguaje la elige de forma automática en función del valor que tome la variable. Posee una biblioteca de objetos estándar como Array, Date y Math y también un conjunto central de elementos del lenguaje tales como operadores, estructuras de control, etc.

Arrancar con JavaScript es muy fácil, solo necesitas un navegador y un editor de texto enriquecido como Visual Studio Code, Atom, Sublime, etc.

Del navegador vamos a utilizar también la consola que muestra información sobre la página web que se está ejecutando en ese momento.

Gramática y tipos

JS es case-sensitive (distingue entre mayúsculas y minúsculas). Por ejemplo,

var **E**dad = 18;
var **e**dad = 21;

Edad no es la misma variable que **e**dad.

El final de cada sentencia lleva un **;** (punto y coma)

Comentarios

Hay 2 tipos de comentarios: de una sola línea y de varias líneas.

Ejemplo comentario de una sola línea:

// este es un comentario de una sola línea

/*
Esto es un comentario de varias líneas
linea 1
linea 2
linea 3
Ultima linea!
*/

3. Variables en JS

Definición de variables

La definición de variables en JavaScript se realiza por medio de la palabra reservada var. El nombre de la variable debe comenzar con una letra o guión bajo, luego puede contener letras, números y/o guiones bajos. Tomemos de ejemplo las siguientes definiciones

Ejemplo

```
var  nombre = ‘Juan’;
var edad = 30;
var esMayor = true;
```

Al definir las variables en JavaScript no especificamos un tipo para las mismas. Esta forma de trabajo permite una mayor flexibilidad al momento de utilizar las variables, pero nos exige que el control de su contenido lo realicemos nosotros. Es decir, la variable nombre la podemos inicializar con un tipo de datos de cadena (texto), pero podríamos luego asignarle un valor numérico.

Ejemplo

```
var nombre = ‘Juan’;
nombre = 30;
```

Este código es válido en JavaScript, pero recomendamos no realizar este tipo de operaciones (cambiar el tipo de variable).

Cambio de valores a una variable

Podemos cambiar el valor de una variable con solo poner la variable del lazo izquierdo de una asignación. Ejemplo:

Ejemplo

```
var nombre = 'Juan';
nombre = 'Pedro'; // A partir de aquí la variable nombre contiene el valor Pedro
```

Vectores

La definición de vectores se realiza de la siguiente forma

Ejemplo

```
var vectorEdades = [10,20,18,33,84];
```

Los índices de los vectores comienzan en 0, por lo que el primer elemento (valor 10) tiene como índice 0 y para acceder al mismo debemos usar vectorEdades[0].
De esta misma forma, el último elemento del vector tiene la posición 4, y para acceder al mismo debemos utilizar vectorEdades[4];

4. Estructura de datos en JS

Condicionales

Los condicionales en JavaScript tienen el mismo formato que lenguajes como PHP, C, JAVA, C# y se representan con la siguiente estructura

```
If (<condiciones>) {
    // Código a ejecutar cuando se cumplen las condiciones
} else {
   // Código a ejecutar cuando NO se cumple alguna de las condiciones
}
```

La condición else es opcional y es solo utilizada cuando queremos ejecutar un código específico cuando NO se cumple la condición.

Ejemplo

```
var edad = 30;
if (edad >= 18) {
   console.log('Es mayor de edad');
} else {
   console.log('Es menor de edad');
}
```

La función console.log(<mensaje>) muestra un mensaje por consola

Bucles

Disponemos de varias formas de realizar bucles en JavaScript, las principales es por medio de la sentencia **for** y de la sentencia **while**.

Veamos un contador del 1 al 10 por medio de for y luego su implementación por medio de while.

Ejemplo

```
for(var contador=1; contador<=10; contador++) {
    console.log(contador);
}
```

La salida del código anterior será

```
1
2
3
4
5
6
7
8
9
10
```

Ahora veamos el mismo código realizado con un ciclo while.

Ejemplo

```
var contador = 1;
while(contador<=10) {
    console.log(contador);
    contador++;
}
```

La salida por pantalla será la misma que en el ejemplo del bucle **for**.

Funciones

Para la definición y uso de funciones debemos utilizar la siguiente sintaxis
function <nombre de la función>(<listado de parámetros separados por ,>) {
  <código de la función>
}

Ejemplo

```
function sumar(valor1, valor2) {
    return valor1 + valor2;
}

var total = sumar(10,20);
console.log(total);
```

La salida por pantalla del código será 30.

5. Ejemplo de programa en JS: "Hola mundo!"

Desarrollar un programa en JS que dado un vector de 10 posiciones con números entre 1 y 100, determine el mayor valor y su ubicación y también el menor valor y su ubicación.

Paso 1: necesitamos un html para poder ejecutar el código en la consola del navegador. Agregamos la llamada al código javascript con la etiqueta script puesta en el body de la página.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <script src="mayorMenorVector.js"></script>
</body>
</html>
```

Paso 2: escribimos el programa en js

```
var vector = [22,25, 60, 98, 11, 78, 4, 33, 85, 10];

var mayor, posicionMayor;
var menor, posicionMenor;

// Inicializo con un valor cualquiera del vector
// para asegurarme de que el resultado este
// dentro del conjunto de datos del "universo" del vector.
mayor = vector[0];
posicionMayor = 0;

menor = vector[0];
posicionMenor = 0;


for(let i = 0; i < 10; i++) {
    if(mayor < vector[i]) {
        mayor = vector[i];
        posicionMayor = i;
    }

    if(menor > vector[i]) {
        menor = vector[i];
        posicionMenor = i;
    }
}

console.log("El mayor es " + mayor + " y se encuentra en la posicion " + posicionMayor);
console.log("El menor es " + menor + " y se encuentra en ela posicion " + posicionMenor);
```

Definimos las variables a utilizar. Por un lado el vector conteniendo los números y luego 4 variables para almacenar el mayor y su posición en el vector y el menor y su posición.

Inicializamos mayor y menor con la primer posición del vector (podría haber sido cualquier valor dentro del vector), esto lo hacemos para asegurarnos que el resultado final esté dentro de los datos que el vector contiene. Pensemos qué pasaría si iniciamos a mayor con 100 y a menor con 0… Como ninguno de los 2 valores está dentro del vector que 100 es mas grande que cualquier valor del vector, va a quedar como el mayor número aunque no pertenezca al vector. Lo mismo ocurriría con el 0 inicializando la variable menor.

Con un ciclo for recorremos cada posición del vector y comparamos el valor con el almacenado en las variables mayor y menor. En caso de que el valor de la posición del vector evaluada sea más grande que el mayor almacenado o mas chico que el valor almacenado como mínimo, entonces se reasigna respectivamente y se guarda la posición. Finalmente se muestra por consola los datos obtenidos.


Inspector
Consola
Depurador
Red
Editor de estilo
Almacenamiento
DOM
Salida del filtro
GET http://127.0.0.1:5500/js/varios/mayorMenorVector/index.html
GET http://127.0.0.1:5500/js/varios/mayorMenorVector/mayorMenorVector.js
El mayor es 98 y se encuentra en la posicion 3
El menor es 4 y se encuentra en ela posicion 6
GET ws://127.0.0.1:5500/js/varios/mayorMenorVector/index.html/ws
GET http://127.0.0.1:5500/favicon.ico

6. Trabajo Práctico

Teniendo:

var meses = [“Enero”, “Febrero”, “Marzo”, “Abril”, “Mayo”, “Junio”, “Julio”, “Agosto”, Septiembre”, “Octubre”, “Noviembre”, “Diciembre”];

var diasDelMes= [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];

// diasDelMes tiene cantidad de días que tiene cada mes ordenado según vector meses, es decir, a posición 0 de meses corresponde Enero y a posicion 0 de diasDelMes corresponde 31 que son los días de Enero

Desarrollar un programa que muestre por consola todos los meses con 31 días y todos los meses con 30 días.

Utilizar ciclo for y ambos vectores.

Bibliografía utilizada y sugerida

MDN - JavaScript. (n.d.) Recuperado de: https://developer.mozilla.org/es/docs/Web/JavaScript

Udemy. (n.d.) Recuperado de: https://udemy.com

Wikipedia - JavaScript. (n.d.) Recuperado de https://es.wikipedia.org/wiki/JavaScript

w3schools.com - JavaScript. (n.d.) Recuperado de https://www.w3schools.com/js/

Lo que vimos:

- Fundamentos de JavaScript.
- Sintaxis del lenguaje.

Lo que viene:

- Integración con HTML.
- Manejo del DOM.