

Programación JavaScript

Objetivos generales

En este curso aprenderás los fundamentos del lenguaje de programación más usado en la actualidad, con el cual es posible crear aplicaciones de todo tipo. Explorarás inicialmente herramientas propias del mismo, indagando casos prácticos de aplicación.

Conocimientos previos necesarios

Buen manejo de PC.

Destinatarios

Dirigido a todos aquellos interesados en sentar las bases teóricas y prácticas del desarrollo de software, y que cuenten o no con conocimientos y/o experiencia previa.

Temario

Introducción:

- Qué es Javascript.
- Para qué se usa Javascript.
- Porque se utiliza Javascript.
- Javascript NO es lo mismo que JAVA.
- Características del lenguaje de Javascript.
- Herramientas o software necesario para programar en Javascript (VS Code – Navegador web).
- Descargar, instalar y configurar Visual Studio Code (extensiones para programar en Javascript).

Primeros pasos:

- Crear un proyecto o carpeta de contenido y configurarla en VS Code.
- Que es un archivos HTML y un archivo JS.
- Crear un achivo HTML.
- Estructura basica de un archivo HTML.
- Ejecutar y configurar Live Server para ver un archivo HTML en el navegador.
- Cómo insertar código Javascript en HTML (etiqueta script).
- Uso básico del método alert().

- Crear un contenedor en HTML (uso de la etiqueta div).
- Cómo manipular un contenedor HTML desde Javascript.
- Crear un archivo JS y enlazarlo con HTML.
- Importancia del orden del código Javascript dentro de HTML.
- Mención: para que se usa CSS.

Tipos de datos en Javascript:

- Datos Primitivos: string, number, boolean, null.
- Datos Compuestos: object, array, function, Class.

Variables y Constantes:

- Qué es una variable y para qué se utiliza (ejemplos).
- Cómo declarar una variable y asignarle un valor de texto en Javascript (instrucción var).
- Crear una variable en un archivo JS.
- Cómo ver el valor de una variable dentro del navegador – activar y usar la consola del navegador.
- Enviar datos a la consola desde Javascript. Uso del objeto console.log().
- Cómo agregar comentarios en código Javascript (comentar una o varias líneas).
- Cómo modificar el valor de una variable (asignación).
- Alcances de la instrucción var (variables con el mismo nombre).
- Declarar un variable con la instrucción let.
- Ambito de una variable declarada con var.
- Ambito de una variable declarada con let.
- Declarar una variable numerica.Cómo sumar variables numericas.
- Sumas de variables de texto (concatenar variables).
- Concatenar texto y variables dentro otra variables. Uso del acento invertido y el marcadoror `${variable}`.
- Qué es una constante y para qué se utiliza(ejemplos).
- Cómo declarar una constante y asignarle un valor en Javascript (instrucción const).
- Crear una constante en un archivo JS.
- Qué ocurre si creamos más de una constante con el mismo nombre.
- Modificar el valor de constante.

Tipos de variables básicas en Javascript:

- Cómo declarar una variable string.
- Cómo declarar una variable number.

- Cómo declarar una variable boolean.
- Cómo saber el tipo o tipología de una variable (typeof).

Uso de cadena de texto:

- Distintas formas de declarar una variable string (comillas simples, dobles y el constructor String).
- Cómo asignar un salto de línea dentro de un string (uso de \n).
- Funciones y métodos más utilizados de un dato de tipo string:
- length.
- toUpperCase.
- toLowerCase.
- indexOf.
- includes.
- slice.
- trim.
- replace.
- split.
- Convertir o pasar una variable string a una variable numerica (método Number).

Uso de datos numéricos:

- Distintas formas de declarar una variable numerica (por igualdad y con el constructor Number).
- Declarar números entero y números con decimales.
- Operaciones aritméticas entre variables (suma, resta, multiplicación, división – operadores aritméticos +, -, *, /).
- Cómo obtener el resto o residuo de una división (operador %).
- Declarar una variable como acumular.
- Declarar una variable como contador (con la variable, con ++, con --).
- Sumar una variable numérica con una variable de texto (number + string).
- Funciones y métodos más utilizados de un dato de tipo number:
- toFixed.
- parseInt.
- Funciones y métodos más utilizados de un dato de tipo number:
- parseFloat.

Uso de datos booleanos (verdadero – falso)::

- Distintas forma de declarar una variable boolean (por igualdad y con el constructor Boolean).

- Uso de operadores lógicos dentro de una asignación de una variable (operadores >, <, ==, !=).

Uso de arreglos o arrays en Javascript:

- Qué es un array.
- Cómo declarar un array numérico.
- Cómo mostrar el contenido de un array completo por la consola.
- Cómo mostrar un elemento o posición de un array por la consola.
- Cómo declarar un array de texto.
- Cómo declarar un array mixto o de distintos tipos de datos.
- Cómo saber o informar la cantidad de elementos o posición de un array.
- Cómo mostrar el último elemento de un array.
- Cómo agregar un elemento o posición a un array:
- Agregar al final (push).
- Agregar al principio (unshift).
- Como quitar o eliminar un elemento de un array:
- Quitar un elemento (shift).
- Quitar el último elemento (pop).
- Cómo buscar y saber la posición de un dato dentro de un array (indexOf).
- Modificar o cambiar el valor de un elemento o posición de un array (asignación).
- Cómo declarar un array vacío.
- Cómo declarar un array e inicializarlo con el mismo valor para todas las posiciones de dicho array (fill).
- Cómo mostrar el contenido completo de un array en orden invertido (reverse).
- Cómo mostrar en forma ordenada el contenido de un array (sort).
- Cómo unir o concatenar dos arrays (concat).

Uso de objetos en Javascript:

- Qué es un object u objeto en Javascript.
- Distintas formas de declarar un objeto (con llaves {} y el constructor Object).
- Cómo mostrar el contenido completo de un objeto por la consola.
- Cómo mostrar una propiedad o atributo de un objeto (variables y arrays).
- Cómo ejecutar un método de un objeto (funciones – function).
- Uso de la palabra clave this dentro de un objeto.
- Cómo agregar propiedades o atributos de un objeto.
- Cómo agregar un método un objeto.

- Cómo eliminar una propiedad y/o método de un objeto (delete).
- Cómo modificar o cambiar el valor de una propiedad o atributo de un objeto.
- Cómo redefinir o modificar un método de un objeto.
- Cómo agregar propiedades o atributos vacío o nulo, y luego asignarle un valor (uso del tipo o valor null).
- Cómo mostrar o saber cuáles son las propiedades y métodos de un objeto (llaves o keys).

Uso de condicionales y conmutadores:

- Qué es y para qué se usa un condicional simple (if).
- Operadores lógicos o de comparación (operadores >, <, ==, >=, <=, !=).
- Cómo usar un condicional compuesto (if, else).
- Qué es y cómo se usa un operador condicional u operador ternario (condición ? expr1 : expr2).
- Conectores lógicos And y Or (&&, ||).
- Qué es y para qué se usa un conmutador (switch).
- Switch con datos de textos y datos de números.
- Uso de condicionales y conmutadores con arrays y objetos.

Uso de ciclos o bucles:

- Qué es y para qué se usa un bucle o ciclo.
- Cómo usar un ciclo for o para.
- Cómo usar un ciclo while o mientras.
- Cómo usar un ciclo do while o hacer mientras.
- Cómo romper y continuar un ciclo (break, continue).
- Cómo recorrer o buscar dentro de un array con el ciclo for (length).
- Cómo recorrer un array con el ciclo for of.
- Cómo recorrer un objeto con el ciclo for in.
- Cómo recorrer un array con el método forEach.

Uso de funciones:

- Qué es una función en Javascript.
- Cómo declarar una función. Estructura o sintaxis de una función en Javascript.
- Cómo invocar, llamar o ejecutar una función.
- Funciones que utilizan parámetros o argumentos.
- Funciones con retorno.
- Qué es una función declarada.

- Qué es una función expresada, anónima o de flecha. Distintas formas de declararla.

Uso de funciones de Javascript desde HTML:

- Cómo ingresar un valor desde HTML. Uso de la etiqueta input.
- Cómo capturar o tomar el valor de un input desde Javascript (getElementById).
- Cómo crear un botón con el etiqueta input desde HTML.
- Cómo ejecutar o asignar una función de Javascript a un botón de HTML.
- Cómo pasar o mostrar un valor desde Javascript a un input (getElementById).
- Ejercicio: crear una calculadora que informe el área de un círculo.
- Uso de la función global eval.

Ejercicio final

- Implementación de todos los temas vistos para crear una aplicación web de una calculadora.
- Diseño de la aplicación: uso de las etiquetas HTML.
- Diseño de la lógica: crear funciones Javascript.
- Diseño visual de la aplicación: aplicar estilos CSS básicos.