

INTRODUCCIÓN A LA PROGRAMACIÓN CON JAVASCRIPT – PLANIFICACIÓN y CONTENIDOS

El presente documento es una guía de temas, clases, contenidos, tiempos y orden de dictado de los conceptos teóricos y prácticos; con el objetivo de educar e introducir al alumno del presente módulo en el amplio mundo de la programación con JavaScript; intentando dar mayor prioridad a los temas que tiene mayor uso en el campo real.

Los módulos tienen una duración de 1 (una hora) reloj, teniendo dos módulos continuos por clase con un descanso entre módulo y módulo de 10 minutos.

Haciendo un resumen de clases, temas el siguiente esquema dará una idea conceptual de la forma en la que se darán los contenidos. |

A continuación, describiremos clase por clase los temas y posteriormente cada tema tendrá su desarrollo completo y cada tema tendrá un link de descarga de la clase correspondiente; de esta forma tendremos un documento principal como columna vertebral de todos los temas a brindar en clases.

CLASE NRO. 01: CONCEPTOS PREVIOS + INTRODUCCIÓN A LA PROGRAMACIÓN

- Introducción a la programación.
- Qué es la programación.
- Lenguajes de programación.
- Tipos de lenguajes de programación
- Componentes básicos de un programa
 - . variables, tipos de datos
 - . estructuras de control
 - . estructuras condicionales
 - . estructuras repetitivas
- Depuración y manejo de errores
- Introducción al protocolo HTTP y HTTPs
- Esquema de Petición / Respuesta
- Diferencias entre conceptos
 - . Página Web
 - . Sitio Web
 - . Aplicación Web
- Arquitectura de una aplicación Web
- Patrón de diseño Modelo Vista Controlador

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

CLASE NRO. 02: PATRON MODELO VISTA CONTROLADOR

- creación de primer proyecto con JavaScript
- creación del documento HTML
- incorporación de hoja de estilos
- incorporación del controlador, código JavaScript
- incorporamos las primeras sentencias
 - . alert()
 - . console.log()
 - . prompt()
 - . confirm()
 - . console.error()
 - . console.warn()
 - . realizar líneas de comentarios

CLASE NRO. 03: TIPOS DE DATOS – OPERACIONES CON VARIABLES

- que es una variable?.
- formas de declarar una variable
- particularidades al momento de asignarle un nombre
- diferentes tipos de Variables
 - . tipo numérico
 - . tipo string
 - . tipo boolean
 - . valores nulos
 - . valores indefinidos
- Bloques de código {}
- JavaScript lenguaje débilmente tipado
- operaciones entre variables
 - . suma
 - . resta
 - . multiplicación
 - . división
 - . suma de cadenas
 - . interpolación de cadenas \${variable}

CLASE NRO. 04: ESTRUCTURAS CONDICIONALES Y COMPARACIONES LÓGICAS

- Creación del documento HTML
- Incorporación del código JavaScript como controlador de las interacciones del usuario
- Declaración de variables desde el teclado a través de prompt
- Inicialización de variables
- Realizar operaciones numéricas simples entre las variables numéricas
- Concatenación y/o interpolación de valores strings.
- Mostrar valores por consola.

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

- Visualizar el problema de dividir por cero
- Visualizar el problema de realizar una operación con una variable no inicializada
- Visualizar en el archivo controlador.js cuando escribo o trato de realizar o mostrar por consola una variable que no existe.
- Respetar la indentación de código.
- Introducción a las estructuras condicionales
- Las operaciones de comparación lógicas básicas >, <, =, >=, <=
 - . A > B => se lee A es mayor a B
 - . A < B => se lee A es menor a B
 - . A = B => se lee A es igual a B
 - . A >= B => se lee A es mayor o igual que B
 - . A <= B => se lee A es menor o igual que B
- Introducción a las estructuras condicionales (lado verdadero, lado falso)

CLASE NRO. 05: ERRORES TÍPICOS Y TRY/CATCH

- No inicializar variables
- Utilizar el "==" en comparaciones en lugar de utilizar el "==="
- Utilizar una variable no inicializada dentro de un condicional.
- Olvidarse de cerrar paréntesis (), corchetes [], llaves {} en lugares donde son obligatorios.
- Ser desprolijo en la estructura del programa, no respetar la IDENTACIÓN.
- Olvidarse de colocar comillas dobles o comillas simples, en algún mensaje que sea por consola por ejemplo
- Introducción a la sentencia try/catch

CLASE NRO. 06: ESTRUCTURAS REPETITIVAS

- Estructuras repetitivas
 - . Ciclo for
 - . Ciclo while
 - . Ciclo do while
- Declaración e inicialización de variables
- Estructuras condicionales if/else dentro de ciclos
- Estructuras condicionales anidadas
- Contadores
- Acumuladores
- Cálculo de porcentajes sobre el total
- Salir de estructuras repetitivas con break
- Saltar una iteración en estructuras repetitivas con continue

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

CLASE NRO. 07: FUNCIONES

- Que es una función => sinónimos (subprocesos, subrutinas, procedimientos, métodos).
- Definición ó declaración de una función
 - . Palabra function
 - . Parámetros ó argumentos
 - . Cuerpo de la función
 - . Retorno de la función
- Funciones que devuelven valor
- Invocación de Funciones.
- Mostrar como las funciones pueden participar dentro de expresiones matemáticas
- Funciones que no devuelven valor => procedimientos.
- Funciones y bibliotecas de funciones estándar
- Funciones como constantes => Expresión de función
- Funciones Flecha ó arrow functions
 - . Forma de declaración
 - . Omitir paréntesis, llaves y la palabra return cuando se recibe un solo parámetro de entrada. Ejemplo.
- Funciones que reciben como parámetro otras funciones.

CLASE NRO. 08: MANEJO DEL DOM

- Introducción y explicación del DOM
- Estructura de árbol del DOM y sus ELEMENTOS
 - . Elementos
 - . Atributos
 - . Texto
 - . Comentarios
- Interacción con el DOM
 - Los selectores por ID y Nombre
 - . getElementById
 - . getElementsByClassName
 - . getElementsByTagName
 - . querySelector("selector")
 - . querySelectorAll("selector")
- Modificación de Elementos del DOM
 - innerHTML
 - textContent
- Eventos Principales en una página WEB
 - . window.onload
 - . click
 - . Incorporamos dos botones y capturamos el evento click en dos botones
 - . Incorporamos el Evento click al documento y observamos que se produce un efecto de propagación de a dentro hacia a fuera.
 - . stopPropagation()

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

CLASE NRO. 09: MANEJO DEL DOM

- Manejo del DOM
- integración HTML, CSS y JavaScript
- declaración de variables y constantes
- captura de elementos / objetos de la interfaz con getElementById
 - . Mostramos cuando una variable o constante es nula
 - . Mostraremos cuando una variable es undefined
- captura de valores de esos elementos
- declaración de funciones
- invocación de funciones
- captura de los eventos que surgen en la pantalla
- interpolación de string para mostrar resultados por consola

CLASE NRO. 10: MANEJO DEL DOM

- Manejo del DOM
- integración HTML, CSS y JavaScript
- declaración de variables y constantes
- captura de elementos / objetos de la interfaz con getElementById
 - . Mostramos cuando una variable o constante es nula
 - . Mostraremos cuando una variable es undefined
- captura de valores de esos elementos
- declaración de funciones
- invocación de funciones
- captura de los eventos que surgen en la pantalla

interpolación de string para mostrar resultados por consola

CLASE NRO. 11: ARRAYS / VECTORES

- Declarar un vector literal
- Mostrar el contenido de un vector
- Acceder a una posición del vector y modificarlo
- Agregar al final push()
- Eliminar el último valor y devolverlo pop()
- Eliminar el primer elemento del vector y devolverlo shift()
- Agregar al principio unshift()
- Eliminar elementos contiguos splice()
- Devolver la ubicación de un elemento indexOf() y lastIndexOf(). Uno retorna la primera ubicación y el otro devuelve la última posición.
- Devolver un elemento y la ubicación, pero con criterios más flexibles y definidos por el usuario como find y findIndex.
- Filtrar y seleccionar y devolver un conjunto de elementos del vector que cumplan con una condición específica. Filter
 - Forma 1: con una arrow function que no tiene la palabra los () para el parámetro, no tiene las llaves {} del cuerpo y no lleva la palabra return;

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

- Forma 2: con una arrow function completa que tiene los () y la palabra return y las llaves del cuerpo {}
 - Forma 3: con una función anónima clásica definida dentro del parámetro de filter
 - Forma 4: con una función declarada previamente y pasarle como parámetro la función dentro del filter.
- Recorrer tradicionalmente un vector con un ciclo for
 - Recorre un vector con vector.forEach()
 - Transformación de los elementos de un vector con vector.map() recorre los elementos, aplica la función pasada como parámetro y devuelve cada elemento transformado, a todos ellos los devuelve y los agrega en un vector de salida.
 - Funciones reductoras con reduce(). Que permiten acumular valores, encontrar el mayor, el menor, etc.
 - Función de ordenamiento sort() con las particularidades cuando son valores string o cuando son valores numéricos.
 - Particularidad de copiar variables o constantes que contienen vectores. Es decir, se crean referencias a los elementos y no se realiza una copia del elemento. Dado este problema surge la necesidad de hacer una clonación de esos elementos y sale spreadOperator también se puede usar slice() pero es más completo y flexible spread. El problema de la copia por referencia ocurre con Vectores y con Objetos.
 - spreadOperator ...

CLASE NRO. 12: POO – PROGRAMACIÓN ORIENTADA A OBJETOS

- Introducción al Concepto de la POO
- Nueva estructura de datos
- Las clases, objetos, atributos, constructor, métodos get y set
- El constructor
- Métodos de una clase
 - o públicos
 - o estáticos
 - o privados
- Los Objetos
- Abstracción
- Encapsulación
- Roles de un programador (Constructor de clases ó Consumidor de clases)
- Objetos literales
- JSON => JavaScript Object Notation
- **Destructuring**

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

CLASE NRO. 13: MÓDULOS - BIBLIOTECAS

- que son los módulos?
- exportación / importación de elementos primitivos anónimos (variables / constantes)
- exportación / importación de vectores
- exportación / importación de funciones anónimas
- exportación / importación funciones tradicionales
- exportación / importación de una clase anónima
- exportación / importación de objetos
- exportación / importación de un botón
- exportación / importación de un fragmento
- exportaciones / importación por defecto
- exportaciones / importación de elementos con nombre
- importar todo un módulo completo

CLASE NRO. 14: MANIPULACIÓN DINÁMICA DEL DOM

- que es la manipulación dinámica del DOM
- tres técnicas de manipulación dinámica del DOM
 - o con métodos propios del DOM como createElement
 - o con innerHTML
 - o con Template
- práctica intensiva
- generación de componentes
- módulos que almacenarán funciones que generan componentes
- ejercicios varios

CLASE NRO. 15: FUNCIONES. COMO ENTIDADES DE PRIMERA CLASE

- Ser asignadas a una variable ó constante.
- Ser pasadas como argumentos a funciones.
- Ser devueltas como resultado de funciones.
- Ser almacenadas en estructuras de datos.
- Particularidades que tenemos con las funciones
 - pueden declararse de forma clásica
 - pueden guardarse dentro de variables o constantes
 - pueden declararse como arrow functions y guardarlas en constantes. Expresiones
 - En algunos casos, las arrow functions pueden no llevar (), {} y la palabra return
 - Funciones pueden ser pasadas como argumentos o parámetros de otra función
 - Funciones que retornan otras funciones
 - El Resultado de una función puede ser pasado a otra función
 - Pueden establecerse valores por defecto en los parámetros de entrada
 - Pueden ser almacenadas en cualquier estructura de datos
 - Funciones con Rest Parameters
 - Funciones con Spread Operator

DIPLOMATURA EN DISEÑO WEB FULL STACK CON JAVASCRIPT

MÓDULO 02 - JAVASCRIPT

CLASE NRO. 16: ASINCRONISMO

- Que es el asincronismo.
 - Diferencias entre procesos síncronos y asíncronos
 - Event Loop
 - Manejar el asincronismo para conseguir un proceso síncrono
- Ejemplo 01: de dos procesos aislados y asíncronos

Funciones callBack

Ejemplo 02: utilizando una función callback

Ejemplo 03: la función llamadora recibe como parámetro la próxima función a llamar

Ejemplo 04: la función callBack al momento de la invocación se declara de forma anónima como una arrow function

Ejemplo 05: en este quinto ejemplo la función llamadora pasa resultados a la callBack

Promesas

Ejemplo 07: visualización de como instanciar el objeto Promise

Ejemplo 08: crear una función que retorne una promesa

Ejemplo 09: crear una función que retorne una promesa, pero la promesa se pasa como parámetro a la función

Ejemplo 10: En este ejemplo la función lanzadora anterior, se utiliza para enganchar dos promesas que se desean ejecutar de forma síncrona

Async / await

CLASE NRO. 17: APIs + FETCH + JSON + TRY/CATCH

CLASE NRO. 18: ESTADO DE LA APLICACIÓN

CLASE NRO. 19: PERSISTENCIA DE DATOS EN EL NAVEGADOR

- Cookies
- Web Storage
 - o Local Storage
 - o Session Storage