TRABAJO PRÁCTICO NRO. 03 – JAVASCRIPT – MANEJO DEL DOM y ARROW FUNCTION

Especificaciones:

- Crear documento HTML
- Crear el controlador.js que tendrá el programa principal
- Si el programa, necesita una ó mas funciones debe (expresarlas como arrow function). Y si son funciones específicas de cálculos, funciones específicas de la resolución del problema debe aislarlas en un archivo separado denominado "modelo.js". allí deberá tener su conjunto de funciones y exportarlas para que el controlador.js pueda consumirlas.
- Desde el archivo "controlador.js" deberá importar esas funciones que están en el archivo "modelo.js".
- En Resumen cada trabajo práctico tendrá
    - Vista.html
    - Controlador.js
    - Modelo.js => donde estarán las funciones (TODAS).

Ejercicio Nro. 13:

Realizar una arrow function que reciba dos parámetros de entrada. El parámetro 1 será el importe de compra de un producto, el segundo será el margen de ganancia que se aplicará sobre ese producto, la función debe retornar el precio de venta que será igual a aplicarle el % de margen de ganancia. La función debe retornar el precio de venta (sin impuestos).

Nota: Debe devolver un número

Ejercicio Nro. 14:

Realizar una arrow function que reciba como parámetro las 3 notas que obtuvo un alumno en los distintos trabajos prácticos de una materia y que a partir de esas notas obtenga el promedio de las mismas

Nota: Debe devolver un número

Ejercicio Nro. 15:

Realizar una arrow function que reciba como parámetro una nota promedio y que según la nota recibida como parámetro devuelva un texto que diga los siguientes mensajes.

- Si la nota <= 4 debe devolver **"Desaprobado"**
- Si la nota > 4 y nota <= 7 debe devolver **"Aprobado"**
- Si la nota >7 y nota <=9 debe devolver **"Muy Bueno"**
- Si la nota = 10 debe devolver **"Excelente"**

Nota: Debe Devolver un Texto

Ejercicio Nro. 16:

El Gobierno Nacional desea aplicar un impuesto (Sobre Tasa) a las bebidas en función de la siguiente clasificación y tipo.

1 – Bebidas Agua en envases plásticos = 5 ‰ (cinco por mil)

2 – Bebidas Agua en envases retornables = 1 ‰ (uno por mil)

3 – Bebidas Gaseosas Azucaradas en envases plásticos = 7 ‰ (siete por mil)

4 – Bebidas Gaseosas Azucaradas en envases retornables = 2 ‰ (dos por mil)

5 – Bebidas Energéticas = 15 ‰ (quince por mil)

6 – Cualquier otra bebida no clasificada = 1 ‰ (uno por mil).

La función debe recibir el Importe Base de la Bebida, debe calcular y retornar la sobre Tasa, la recaudación de ese impuesto tendrá destino a la protección del medio ambiente.

Nota: Debe devolver un número

Ejercicio Nro. 17:

Una empresa de distribución de agua potable "Aguas de Catamarca ECSAPEM" utiliza un esquema de facturación por bloques de consumo, donde el precio por metro cúbico aumenta a medida que el usuario consume más. Este tipo de sistema se aplica con el objetivo de promover el uso responsable del recurso y penalizar el consumo excesivo.

El cálculo del importe base se realiza a partir de los metros cúbicos leídos en el medidor del cliente. Para ello, la empresa define tres bloques de facturación. El primer bloque corresponde a los primeros 50 metros cúbicos, que se facturan a razón de 350 pesos por metro cúbico. El segundo bloque comprende los siguientes 20 metros cúbicos, es decir desde el metro cúbico número 51 hasta el 70, los cuales se facturan a un valor de 555,20 pesos por metro cúbico. Finalmente, todo consumo que supere los 70 metros cúbicos se considera dentro del bloque excedente o de castigo, y se factura a razón de 1.552,20 pesos por metro cúbico.

Cabe aclarar que los clientes cuyo consumo mensual sea inferior a 50 metros cúbicos igualmente abonan un mínimo de 50 metros cúbicos, de modo que ese sea el valor base de cálculo.

La tarea consiste en definir una arrow function llamada calcularImporteAgua que reciba como parámetro el total de metros cúbicos leídos y devuelva el importe base a pagar por el cliente según el esquema tarifario antes descripto.

Consumo (m³)	Cálculo aplicado	Importe final ($)
30	50 × 350,00	17.500,00
55	(50 × 350,00) + (5 × 555,20)	20.276,00
85	(50 × 350,00) + (20 × 555,20) + (15 × 1.552,20)	57.214,00

Nota: Debe devolver un número

Ejercicio Nro. 18:

Realizar una arrow function que reciba como parámetro el Importe Base de una factura de “Servicios Públicos de Aguas de Catamarca” y a partir de ese importe base calcule y devuelva la Tasa de Subsuelo, que es un importe que corresponde al 3% del importe Base ingresado como parámetro de la arrow function.

Nota: Debe devolver un número

Ejercicio Nro. 19:

Realizar una arrow function que reciba como parámetro el Importe Base de una factura de “Servicios Públicos de Aguas de Catamarca” y partir de ese importe base calcule y devuelva la Tasa de Fiscalización ENRE (Ente Regulador de Servicios Públicos) que corresponde al 1,2 % del importe Base ingresado como parámetro de la arrow function.

Nota: Debe devolver un número

Ejercicio Nro. 20:

Realizar una aplicación web que permita calcular y determinar la dosis de insulina recomendada para un paciente diabético; Basada en tres datos importantes para el cálculo.

1) Nivel de glucosa en sangre
2) Peso Corporal (en kilogramos)
3) Tipo de diabetes
    a. Tipo 1

b. Tipo 2

Para Tipo 1: El cálculo es el 50% del Peso corporal del paciente + el 50% del nivel de glucosa en sangre, este último término solamente si la glucosa es mayor a 180.

Para Tipo 2: El cálculo es el 20% del Peso corporal del paciente + el 50% del nivel de glucosa en sangre, este último término solamente si la glucosa es mayor a 180.

La función debe retornar la dosis de insulina recomendada y recibir como parámetros de entrada (argumentos) nivel de glucosa, peso corporal y tipo de diabetes.

Salidas de la aplicación: la aplicación debe indicar la insulina recomendada para el paciente.

Ejercicio Nro. 21:

Realizar una arrow function que reciba como parámetro una cadena de texto y que devuelva la cantidad de vocales “mayúsculas y/o minúsculas” que tiene la misma.

Debe recorrer la cadena con un ciclo for, desde el primer carácter hasta el último, analizar cada uno de los caracteres y determinar si es una vocal o no. No debe utilizar métodos de strings como replace(), split() o expresiones regulares.

Nota: Debe devolver un número.

Ejercicio Nro. 22:

Realizar una arrow function que reciba como parámetro una cadena de texto y devuelva cuántas consonantes contiene (mayúsculas o minúsculas). Es decir consideramos consonante a todo carácter que NO SEA VOCAL.

Debe recorrer la cadena con un ciclo for, analizar cada carácter y determinar si es una letra que no sea vocal.

No debe utilizar métodos de strings como replace(), split() o expresiones regulares.

Nota: Debe devolver un número.

Ejercicio Nro. 23:

Definir una arrow function que reciba una palabra y determine si contiene al menos dos letras “s” (mayúsculas o minúsculas).

La función debe recorrer la palabra con un ciclo for, utilizando un contador o una bandera booleana.

Debe devolver true si se cumplen las dos apariciones y false en caso contrario.

No puede utilizar métodos como includes() o indexOf().

Nota: Debe devolver un boolean (true ó false).

Ejercicio Nro. 24:

Definir una arrow function que reciba como parámetro una cadena de texto y determine si la misma contiene al menos tres espacios en blanco.

La función debe recorrer la cadena carácter por carácter utilizando un ciclo for.

Durante el recorrido, deberá contar la cantidad de espacios que se encuentren.

En el momento en que se detecte el tercer espacio en blanco, la función debe interrumpir inmediatamente el ciclo mediante la instrucción break y devolver el valor lógico true.

Si al finalizar el recorrido completo no se encontraron tres espacios, la función debe devolver false.

Nota: Debe devolver un boolean (true ó false).

Ejercicio Nro. 25:

Definir una arrow function que reciba como parámetro una cadena de texto y determine si la misma no contiene ningún dígito numérico (del 0 al 9).

La función debe recorrer la cadena carácter por carácter utilizando un ciclo for. Durante el recorrido, analizar cada carácter y comprobar si se encuentra dentro del rango de los números '0' a '9'.

Si se detecta algún número, se debe interrumpir inmediatamente el bucle con break y devolver el valor lógico false, ya que la cadena deja de cumplir la condición “no contiene números”.

En caso de recorrer la cadena completa sin encontrar ningún número, la función deberá devolver true.

Nota: Debe devolver un boolean (true ó false).