

INTRODUCCIÓN A LA LÓGICA DE PROGRAMACIÓN

**SUB PROCESOS, ARROW FUNCTIONS, DECLARACIÓN, INVOCACIÓN,
PARTICULARIDADES DE LAS ARROW FUNCTIONS.**

TRABAJO PRÁCTICO NRO. 3

Objetivo:

El Objetivo del presente trabajo práctico es que el alumno practique varios y diversos ejercicios de programación y lógica de manera tal que domine los conceptos de:



Listado de Ejercicios a Realizar:

Ejercicio Nro. 51: Arrow function

Realizar una función que reciba como parámetro dos números (X, PORCENTAJE).

El primero será un número que lo llamaremos x y al segundo lo llamaremos porcentaje. La idea es que la función tome el número x y le calcule el porcentaje que recibe como segundo parámetro; a ese resultado devolverlo al programa principal. Luego (INVOCAR Ó LLAMAR) esa función desde el programa PRINCIPAL.

Realice la función en forma de arrow function, invoque la función pasándole como parámetro diferentes valores.

Ejercicio Nro. 52: Arrow function

Realizar una arrow function que reciba como parámetros la altura y el peso y en función de esos valores calcule el IMC (índice de masa corporal), la función debe devolver el mensaje dependiendo de los valores obtenidos.

$IMC = \text{peso (kg)} / (\text{altura} * \text{altura})$

Nota: el peso esta expresado en kilogramos y la altura en metros.

- Bajo peso: IMC menor de 18.5
- Peso normal: IMC entre 18.5 y 24.9
- Sobrepeso: IMC entre 25 y 29.9
- Obesidad grado I: IMC entre 30 y 34.9
- Obesidad grado II: IMC entre 35 y 39.9
- Obesidad grado III (Obesidad mórbida): IMC de 40 o más

Realice la función en forma de arrow function, invoque la función pasándole como parámetro diferentes valores.

Ejercicio Nro. 53: Arrow function

Un banco de plaza local, desea realizar un simulador de plazos fijos, con la intención de captar clientes y fomentar el ahorro, para ello contratará un programador FULL STACK que permita construir una aplicación web con el objetivo de simular las siguientes reglas de negocio. El gerente del Banco nos indica que la aplicación debería solicitar que ingrese

1) El capital a Ingresar

2) La cantidad de meses en los que realizará el plazo fijo

para Montos de Capital de:

- 500.000 a 1.500.000 la rentabilidad será del 6% mensual



- 1.500.000 hasta 5.000.000 la rentabilidad será del 7,5% mensual
- 5.000.000 hasta 25.000.000 la rentabilidad será del 8,5% mensual
- para valores que superen esa magnitud la tasa mensual será del 9,5%

IMPORTANTE – RESTRICCIONES DEL EJERCICIO (Desde el ejercicio 54 al 65)

⚠ En estos ejercicios NO se permite el uso de funciones nativas de JavaScript para manipulación de fechas, como Date(), getDate(), getMonth(), getFullYear(), getTime() u otras similares. Debe Construir sus propias funciones.

- El objetivo es que el alumno implemente sus propias funciones aplicando lógica de programación y estructuración de código, sin depender de herramientas predefinidas del lenguaje.
- Se espera que el alumno desarrolle todas las operaciones de cálculo de fechas, validaciones y diferencias de días utilizando únicamente estructuras condicionales (if, switch), operadores aritméticos y funciones personalizadas.

Ejercicio Nro. 54: Arrow function – **validar si el año es bisiesto**

Realizar una función que determine si un año enviado por parámetro es año bisiesto, la función debe devolver un valor booleano (true o false).

Nota: Los años bisiestos son aquellos cuyo número cumple las siguientes características

- Son divisibles por 4
- y
- (No son Divisibles por 100 salvo que sean divisibles por 400)

((EsDivisiblePor4 == true) y (NoEsDivisiblePor100 ó EsDivisiblePor400))

Esta sería la condición que debe evaluar en el condicional

`(anio % 4 === 0) && (anio % 100 !== 0 || anio % 400 === 0)`

Ejercicio Nro. 55: Arrow function

Realizar una función que retorne la cantidad de días de un mes determinando, teniendo la particularidad de que el Mes de Febrero de los años bisiestos tiene 29 días. Es decir, para poder devolver la cantidad de días de un mes necesitará pasar como parámetro dos parámetros; el mes y el año. Dentro de esta función debe invocar la función anterior del año bisiesto, dado que febrero de un año bisiesto tiene 29 días y sino tiene 28 días.

Ejercicio Nro. 56: Arrow function – **validar únicamente el día**



Realice una función que reciba como parámetro una fecha en formato (DIA, MES, AÑO) y valide si el día es correcto.

Por ejemplo:

- El día de la siguiente fecha es inválido 35/02/2015
- El día de la siguiente fecha es inválido 74/03/2000
- El día de la siguiente fecha es inválido 29/01/2001
- El día de la siguiente fecha es inválido 31/04/2001

La función debería devolver un valor booleano (true ó false) dependiendo si el día es válido ó no para el mes que se esté analizando y el año.

Debe utilizar la función que declaró en el ejercicio anterior, función que devuelve la cantidad de días de un mes y de un año específico.

Ejercicio Nro. 57: Arrow function – **validar únicamente el mes de una fecha**

Realice una función que reciba como parámetro una fecha en formato (DIA, MES, AÑO) y determine si el mes ingresado es válido.

Por ejemplo:

- El mes de la siguiente fecha es inválido (31/14/2015)
- El mes de la siguiente fecha es inválido (02/17/2015)

La función debería devolver un valor booleano (true ó false) dependiendo si el día es válido ó no para la fecha que se está analizando.

Ejercicio Nro. 58: Creación, Definición e Invocación de Funciones

Realice una función que reciba como parámetro una fecha en formato (DIA,MES, AÑO) y determine si el año ingresado es válido. Para que el año sea válido, el mismo debe ser mayor o igual a cero.

Ejercicio Nro. 59: Creación, Definición e Invocación de Funciones

Realice una función que determine si una fecha en formato (DIA, MES, AÑO) es una fecha válida y para ello deberá utilizar las funciones desarrolladas en los ejercicios anteriores.

Nota: Una fecha válida sería aquella que



- El día es válido ya posee una función que hace esto.
- El mes es válido ya posee una función que hace esto.
- El año es válido ya posee una función que hace esto.

Ejercicio Nro. 60: Creación, Definición e Invocación de Funciones

Realice una función que a partir de una fecha en formato (DIA, MES, AÑO) retorne la cantidad de días que transcurrieron desde el principio de ese mes hasta la fecha ingresada.

Ejercicio Nro. 61: Creación, Definición e Invocación de Funciones

Realice una función que a partir de una fecha en formato (DIA, MES, AÑO) retorne la cantidad de días que faltan para llegar a fin de mes. Para ello debería utilizar la función realizada en el ejercicio 57 donde se obtiene la cantidad de días para un mes y año determinado.

Ejercicio Nro. 62: Creación, Definición e Invocación de Funciones

Realice una función que a partir del año de una fecha (AÑO) retorne la cantidad de días que tiene ese año. Por ejemplo:

- El año 2016 tiene 366 días (Porque es Bisiesto)
- El año 2015 tiene 365 días (Porque no es Bisiesto)

Ejercicio Nro. 63: Creación, Definición e Invocación de Funciones

Realice una función que a partir de una fecha en formato (DIA, MES, AÑO) determine la Cantidad de días que faltan para llegar a fin de año.

Ejercicio Nro. 64: Creación, Definición e Invocación de Funciones

Realice una función que a partir de una fecha en formato (DIA, MES, AÑO) determine la cantidad de días que hay desde principio de año a esa fecha.

Ejercicio Nro. 65: Creación, Definición e Invocación de Funciones

Realice una función que a partir de dos fechas en formato (DIA1, MES1, AÑO1, DIA2, MES2, AÑO2) determine la cantidad de días que hay entre esas dos fechas. Para ello debería reutilizar algunas funciones de las que ya tiene definidas.