

Práctica 4

Fecha límite de entrega: 17 de marzo de 2021

Desarrolle los siguientes ejercicios los cuales se deben tener en cuenta funciones, condicionales, desglose, enumeración e intervalos utilizando el paradigma funcional y lenguaje de programación JavaScript. Usted debe enviar el código fuente y pasar los test a través de la plataforma INGINIOUS M-IDEA (<http://ingin.ddns.net/courselist>). Puede apoyarse de la herramienta repl.it (<https://repl.it/~>)

Documento de repaso

https://docs.google.com/document/d/1kWkNKLZ9dak7IEcAwWuF0Eq4uah1AJ4UhZOO9Oz0G_U/edit?usp=sharing

IMPORTANTE

Operadores de Comparación, si usted tiene problemas para identificar estos símbolos ($<$, $>$, $!=$, $==$, $===$, $<=$), debe revisar el siguiente link

https://developer.mozilla.org/es/docs/conflicting/Web/JavaScript/Reference/Operators_5a0acbbb60ea37d0fdc52e3bd4c3fae8

Librería Math, si usted no sabe para que funciona el `Math.cos()` o el `Math.sqrt()` o el `Math.log()` o `Math.PI` o `Math.E()`, debe revisar el siguiente link

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Math









Compuertas logicas, Si usted tiene problema en identificar estos símbolos ($\&\&$, $\|\|$), debe revisar el link <https://www.logicbus.com.mx/compuertas-logicas.php>

Operadores Aritméticos, si usted tiene problemas con identificar estos símbolos ($+$, $-$, $/$, $*$, $\%$), debe revisar el link

<https://developer.mozilla.org/es/docs/conflicting/Web/JavaScript/Reference/Operators>

La precedencia de operadores determina el orden en el cual los operadores son evaluados. Los operadores con mayor precedencia son evaluados primero. revisar el link

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Operator_Precedence

| NOMBRE | SÍMBOLO | SIGNIFICADO | REPRESENTACIÓN |
|-----------------------|----------------|--|--|
| Intervalo abierto | (a,b) | $\{ x / a < x < b \}$ Nº comprendidos entre a y b |  |
| Intervalo cerrado | $[a,b]$ | $\{ x / a \leq x \leq b \}$ Nº comprendidos entre a y b, éstos incluidos. |  |
| Intervalo semiabierto | $(a,b]$ | $\{ x / a < x \leq b \}$ Nº comprendidos entre a y b, incluido b |  |
| | $[a,b)$ | $\{ x / a \leq x < b \}$ Nº comprendidos entre a y b, incluido a |  |
| Semirrecta | $(-\infty, a)$ | $\{ x / x < a \}$ Números menores que a |  |
| | $(-\infty, a]$ | $\{ x / x \leq a \}$ Nº menores o iguales que a |  |
| | (a, ∞) | $\{ x / a < x \}$ Números mayores que a |  |
| | $[a, \infty)$ | $\{ x / a \leq x \}$ Nº mayores o iguales que a |  |

Recuerde: abierto: $()$ es $>$ o $<$, cerrado: $[]$ es \geq o \leq

<https://www.matematicasonline.es/pdf/Temas/1BachCT/Intervalos,%20semirrectas,%20entornos%20y%20valor%20absoluto.pdf>

0. Número en intervalo

Desarrolle una función que reciba como parámetro la presión de un adulto, si esta es mayor a 120/80 y menor a 180/80, el adulto está sano, si es menor el adulto debe tomar medicina, si es mayor debe ir al médico.

código/solución

```
function presionAdulto(valor) {  
  if (valor >= (120 / 80) && valor <= (180 / 80)) {  
    return "adulto sano"  
  } else if (valor < (120 / 80)) {  
    return "debe tomar medicina"  
  } else {  
    return "debe ir al médico"  
  }  
}
```

Comprueba las salidas

Entrada

```
presionAdulto(121/80)  
presionAdulto(172/80)  
presionAdulto(119/80)  
presionAdulto(7/80)  
presionAdulto(181/80)  
presionAdulto(199/80)
```

Salidas

```
adulto sano  
adulto sano  
debe tomar medicina  
debe tomar medicina  
debe ir al médico  
debe ir al médico
```

1. Semáforo

Desarrolle una función que reciba un estado que indique cuál es la luz que debería encender de acuerdo a la que está encendida actualmente. Si la luz es “rojo” entonces debe mostrar “verde”, si es “verde” debe mostrar “amarillo” y finalmente si es “amarillo” debe mostrar “rojo”.

código/solución

```
function semaforo(estadoActual) {  
  if (estadoActual == "amarillo") {  
    return "rojo";  
  } else if (estadoActual === "rojo") {  
    return "verde";  
  } else if (estadoActual === "verde") {  
    return "amarillo";  
  } else {  
    return "no existe un color en el semáforo"  
  }  
}
```

Comprueba las salidas

Entrada

```
semaforo("amarillo")
semaforo("rojo")
semaforo("verde")
```

Salidas

```
rojo
verde
amarillo
```

2. Intervalo

Desarrolle una función que reciba un número y determine (false o true) si está en el intervalo (3,7].

código/solución

```
function intervaloAC(num) {
    if (num > 3 && num <= 7) {
        return true;
    } else {
        return false;
    }
}
```

Comprueba las salidas

Entrada

```
intervaloAC(3)
intervaloAC(5)
intervaloAC(6.89)
intervaloAC(7.00000)
```

Salidas

```
false
true
true
true
```

3. Número en intervalo

Desarrolle una función que reciba un número. Si el número está en uno de los siguientes intervalos: $(3, 10) \cup (32, 57) \cup (100, \infty)$ debe retornar "válido" o en caso contrario retorna "NO válido"

Entrada punto 1

```
intervalo(734543234)
intervalo(13)
intervalo(42)
```

Salida punto 1

```
válido
NO válido
?
```

4. Intervalo con string

Desarrolle una función que determine si un valor está entre el intervalo [3,7] si está retorna la cadena "True", si no está, retorna la cadena "Error".

Entrada punto 3

```
intervalo(3)
intervalo(8)
intervalo(5)
```

Salida punto 3

```
?
Error
?
```

5. Espacio abierto y cerrado

Desarrolle una función que determine si un valor está entre el intervalo $[-1,13]$, si está retorna true, si no está, retorna false.

Entrada punto 4

```
intervalo(-1)
intervalo(13)
intervalo(-5)
```

Salida punto 4

```
?
?
false
```

6. intervalo

Desarrolle una función que determine si un valor está entre el intervalo $(0,1]$, si está, retorna true, si no está, retorna false.

Entrada punto 5

```
intervalo(1)
intervalo(0)
intervalo(0.5)
```

Salida punto 5

```
true
?
?
```

7. Intervalo anidado

Desarrolle una función que determine si un valor está entre el intervalo $(-5,-2] \cup (7,10) \cup (5,6]$, si está, retorna true, si no está, retorna false.

Entrada punto 5

```
intervalo(1)
intervalo(0)
intervalo(0.5)
```

Salida punto 5

```
true
?
?
```

8. Función aumentada

Desarrolle una función que reciba un parámetro, si este es un número se le suma 3, sino es un número, devuelve 3.

Entrada

```
devuelveNum("hola");
devuelveNum(true);
devuelveNum(3);
devuelveNum("khe");
devuelveNum(10);
devuelveNum(false);
```

Salidas

```
3
?
?
?
13
3
```

9. Interes

Desarrolle la función intereses que devuelve el porcentaje de interés para un monto dado. Así, el banco paga un 4% para depósitos inferiores o iguales a \$1.000, 4.5%, para depósitos inferiores a \$5.000 (y superiores a \$1.000), y 5% para depósitos mayores o iguales a \$5.000.

Entrada

```
intereses(500)
intereses(2000)
intereses(10000)
```

Salidas

```
20
?
?
```

10. Operaciones

Desarrolle una función que reciba tres números enteros (a, b, i), si i es 1 que eleve (a) a la potencia (b), si i es 2 que calcule la raíz cuadrada de (a) y la raíz cúbica de (b), si i es 3 que devuelva el número menor entre (a) y (b), en caso que no se encuentre en las condiciones retorne la siguiente cadena: "No tengo operación para el número ingresado"

Entrada

```
operacion(3,3,1)
operacion(9,17,1)
operacion(7,13,3)
operacion(4,9,2)
operacion(13,26,2)
operacion(52,13,3)
operacion(19,2,7)
```

Salidas

```
27
?
713
2.00 2.08
? ?
?
?
```

11. Video Juego

Desarrolle una función, que usará en un juego de video, para despedir el juego cuando se ha terminado. La función debe indicar si el jugador merece una medalla de oro, plata o bronce, dependiendo del puntaje (siempre positivo) alcanzado. Si este puntaje es menor o igual que 10, la medalla es de bronce; si es mayor que 10 pero menor o igual que 20, medalla de plata y si es superior a 20, medalla de oro.

Entrada Ejemplo 1

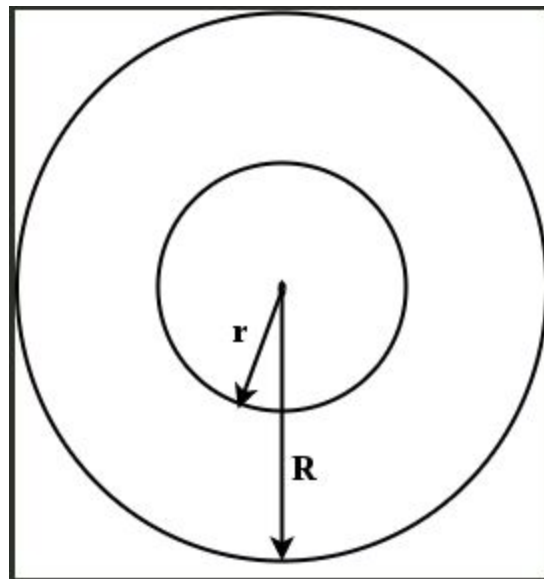
```
7
?
457
1?
```

Salida Ejemplo 1

```
Bronce
Bronce
?
Plata
```

12. Circulo

Calcula el área de un círculo cuyos radios exterior e interior son $R=19$ cm y $r=16$ cm respectivamente.



Calcule: Área Externa, Área Interna y Área del círculo

Entrada

```
areaOuter(19)
areaInner(16)
areaRing(19,16)
```

Salidas

```
1134.1149
804.2477
329.8672
```

13. Semáforo de estado (extra)

Desarrolle una función que reciba un color y un estado que indique cuál es la luz que debería encender de acuerdo a la que está encendida actualmente. Si la luz es “**amarillo**” con true debe mostrar “**verde**”, si es “**amarillo**” con false, debe mostrar “**rojo**”, si es “**verde**” o “**rojo**” con true, debe mostrar “**amarillo**” y finalmente, si hay otro color, debe mostrar “color no válido”.

Entrada

```
semaforo("amarillo",true)
semaforo("verde",true)
semaforo("rojo",true)
semaforo("amarillo",false)
semaforo("azul",false)
```

Salidas

```
verde
?
amarillo
?
color no válido
```

14. Reforma tributaria (extra)

La nueva reforma tributaria ha definido un nuevo modelo de impuesto a las ventas. Los productos, que cuestan 4 millones o más, pagarán un impuesto del 8%. Cualquier otro producto pagará un impuesto del 5%. Diseñe una función que dado el precio de un producto calcule el valor del impuesto.

Entrada Ejemplo 1

```
7200000
9000000
1800000
```

Salida Ejemplo 1

```
?
?
?
```

14. IMC

Desarrolle un programa que tenga dos funciones, una para calcular el índice de masa corporal (IMC) y otra para clasificar el IMC dependiendo estos valores:

- Si el IMC es menor a 18.5 quiere decir que el peso es inferior al normal
- Si el IMC es mayor que 18.5 y menor a 24.0 quiere decir que el peso es normal
- Si el IMC es mayor que 25 y menor a 29.9 quiere decir que el peso superior al normal
- Si el IMC es mayor que 30 quiere decir que hay obesidad.

Entrada

```
imc(85,1.66));
clasifique(85,1.66));
imc(53,1.55));
clasifique(53,1.55));
imc(52,1.80));
clasifique(52,1.80));
imc(90,2.0));
clasifique(90,2.0));
```

Salidas

```
30.85
obesidad
22.06
peso es normal
16.05
peso es inferior al normal
22.50
peso es normal
```