

A yellow square containing the letters 'JS' in a bold, black, sans-serif font.

Desarrollo web en entorno cliente

UD – 2

Manejo de la sintaxis del lenguaje

Actividades



1.INTRODUCCIÓN

2.COMENTARIOS

3.VARIABLES

4.MOSTRAR POR PANTALLA Y LEER TECLADO

Act 1 Crear un programa en javascript que realice la lectura de 3 valores numéricos enteros . Además del nombre y apellido del usuario. Debes utilizar 5 variables, cada una con su tipología correspondiente.

5.OPERADORES

Act 2 Crear un programa en javascript que realice la lectura de 3 valores numéricos enteros . Además del nombre y apellido del usuario.

- Nombre y apellido del usuario(concatenación de cadenas).
- Operaciones :
 - Suma de los 3 valores
 - Multiplicación de los 3 valores
 - División del primer y tercer valor

6.ESTRUCTURAS DE CONTROL Y BUCLES

6.1 IF Y ELSE

Act 3 Crear un programa en javascript que realice la lectura de 3 valores numéricos enteros. Indicar si hay algún de los números mayor que 10.

Act 4 Leer por pantalla 2 valores numéricos correspondientes al día y mes. Indicar si corresponde al día de navidad.

Act 5 De un operario se obtiene por teclado una línea con su sueldo (sin decimales) y en otro línea los años de antigüedad.

Se pide un programa que lea los datos de entrada e informe:

- a) Si el sueldo es inferior a 500 y su antigüedad es igual o superior a 10 años, triplicar el sueldo.
- b) Si el sueldo es inferior a 500 pero su antigüedad es menor a 10 años, doblar sueldo.
- c) Si el sueldo es mayor o igual a 500 mostrar sin cambios.
Imprime el sueldo resultante. (Por ejemplo)

Entrada	Salida
500 10	500
400 29	1200
499 22	1497

Act 6 Escribe un programa que lea por entrada estándar una calificación numérica entre 0 y 10 (con decimales) y la transforma en calificación alfabética según la siguiente tabla.

- de 0 a <3 Muy deficiente
- de 3 a <5 Insuficiente
- de 5 a <6 Bien
- de 6 a <9 Notable
- de 9 a 10 Sobresaliente

Entrada	Salida
0	Muy deficiente
3	Insuficiente
5,3	Bien

6.2 FOR

6.3 WHILE

Act 7 Realiza un programa que lea un número N y muestre por la salida estándar todos los pares desde 2 hasta N (inclusive). Cada número se mostrará en una línea distinta. N siempre será mayor o igual a 2.

Entrada	Salida
4	2 4
5	2 4
6	2 4 6

Act 8 Realiza un programa que lea un número N y muestre por la salida estándar todos los impares desde N(inclusive) hasta 1. Cada número se mostrará en una línea distinta. N siempre será mayor o igual a 1.

Entrada	Salida
1	1
2	1
3	3 1

6.4 DO-WHILE

6.5 BREAK y CONTINUE

Act 9 Desarrollar un programa que muestre la tabla de multiplicar del numero X, hasta el valor Y. Ambos los leerá de la entrada estándar y estarán en una sola línea separados por un espacio, siendo $-50 \leq X \leq 50$, $1 \leq Y \leq 20$. La tabla la mostrará con una entrada por línea. Entre cada elemento habrá un único espacio de separación.

Entrada	Salida
2 5	$2 \times 1 = 2$ $2 \times 2 = 4$ $2 \times 3 = 6$ $2 \times 4 = 8$ $2 \times 5 = 10$
44 2	$44 \times 1 = 44$ $44 \times 2 = 88$
-3 1	$-3 \times 1 = -3$

Act 10 Realiza un programa que lea una secuencia de notas (con valores que van de 0 a 10, sin decimales). Se leerá una nota por línea. La entrada acabará con el valor -1. El programa debe mostrar por salida estándar "SI" si hay algún 10, "NO" en caso contrario.

Entrada	Salida
1, 4, 6, 10, 5, 2, 2, 10, 3, -1	SI
5, 5, 5, 5, -1	NO
1, 2, 3, 4, 5, 6, 7, 8, 9, -1	NO

Act 11 Crear una aplicación que solicite al usuario un número. Escribirá un triángulo con tantos asteriscos como indique el valor introducido

→ valor Introducido: 7

```
*
**
***
****
*****
*****
*****
```

7. FUNCIONES

8. VARIABLES LOCALES Y GLOBALES

Act 12 Escribe una función que reciba dos enteros y devuelva su suma. La función no debe leer nada de la entrada estándar ni mandar ningún dato por la salida estándar. La función debe definirse con esta cabecera:

```
function suma(a, b)
```

Entrada	Salida
2,3	5
100 , 2212	2312

Act 13 Un año es **bisiesto** si cumple los siguientes criterios:

- Es divisible entre 4.
- Si termina en 00, se comprueba si es divisible entre 400 (2000 y 2400 sí son bisiestos. 2100, 2200 y 2300 no lo son).

Mas información sobre años bisiestos en:

https://es.wikipedia.org/wiki/A%C3%B1o_bisiesto

Escribe una función que reciba un año y devuelva 1 si es bisiesto, 0 en caso contrario. La función no debe leer nada de la entrada estándar ni mandar ningún dato por la salida estándar. La función debe definirse con este estilo:

```
function esBisiesto(anyo)
```

Entrada	Salida
4	1
2000	1
2011	0

Act 14 Escribe una función que reciba dos decimales con la base y altura de un rectángulo y devuelva su perímetro. La función no debe leer nada de la entrada estándar ni mandar ningún dato por la salida estándar. **Nota:** se devolverá un decimal (float). La función debe definirse con este estilo:

```
function perimetroRectangulo(a, b)
```

Entrada	Salida
11.50 , 2.00	27.00
1.10 , 2.20	6.60

Act 15 Escribe una función que reciba un número y diga si es primo (devolviendo true) o no (devolviendo false).

El número puede tomar los valores $1 \leq \text{numero} \leq 10000$.

Recuerda que un número primo es aquel que solo es divisible por si mismo y por la unidad y que por convención el número 1 no se considera primo. La definición de primo la tienes en:

https://es.wikipedia.org/wiki/N%C3%BAmero_primo

Utiliza una forma optimizada para reducir el número de operaciones.

La función debe definirse con este estilo:

```
function esPrimo(num)
```

Entrada	Salida
1	false
2	true