

Desarrollo Web en entorno cliente

Sintaxis de Javascript

Ejercicios propuestos - 2

1. De un operario se obtiene mediante prompt su sueldo (sin decimales) y en otro los años de antigüedad. Se pide un programa que lea los datos de entrada e informe:
 - Si el sueldo es inferior a 500 y su antigüedad es igual o superior a 10 años, triplicar el sueldo.
 - Si el sueldo es inferior a 500 pero su antigüedad es menor a 10 años, doblar sueldo.
 - Si el sueldo es mayor o igual a 500 mostrar sin cambios.

Sacar por pantalla el sueldo resultante.

2. Escribe un programa que lea 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
3. Escribe un programa que pida el número de horas trabajadas y luego el precio por hora.

Calcular el salario neto semanal de un trabajador en función del número de horas trabajadas y la tasa de impuestos de acuerdo a las siguientes reglas:

- Las primeras 35 horas se pagan a tarifa normal.
- Las horas que pasen de 35 se pagan a 1.5 veces la tarifa normal.

Al sueldo bruto se le aplicarán las siguientes tasas de impuestos:

- Los primeros 500 euros son libres de impuestos.
- Los siguientes 400 tienen un 25% de impuestos.
- Los restantes un 45% de impuestos.

Escribe en el navegador el salario calculado.

4. Realiza un programa que lea un numero N y muestre en el navegador todos los pares desde 2 hasta N (inclusive). Cada número se mostrará en una línea distinta. N siempre sera mayor o igual a 2.
5. Realiza un programa que lea un numero N y muestre en el navegador todos los impares desde N(inclusive) hasta 1. Cada número se mostrará en una línea distinta. N siempre sera mayor o igual a 1.
6. Desarrollar un programa que muestre la tabla de multiplicar del numero X, hasta el valor Y. Los valores de X e Y deben cumplir que $-50 \leq X \leq 50$, $1 \leq Y \leq 20$
7. Realiza un programa que lea una secuencia de notas (con valores que van de 0 a 10, sin decimales). La entrada acabará con el valor -1. El programa debe mostrar por salida estándar “SI” si hay algún 10, “NO” en caso contrario.
8. Realiza un programa que lea un número y diga si es primo o no. 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.

9. Realiza un programa que lea una cantidad de dinero (múltiplo de 5) y tras ello desglose el cambio en billetes de 500,200,100,50,20,10,5 intentado dar el mínimo número de billetes.

El programa mostrará en el navegador el número mínimo de billetes.

10. Realiza un programa que lea una secuencia de números enteros, uno cada vez.

El programa deberá mostrar como resultado el número mas alto, seguido del número mas bajo, separados por un espacio.

El programa incluirá un primer dato X, $1 \leq X \leq 100$, indicando cuantos números a leer (esa línea no se tendrá en cuenta). Tras ello se leeran X números en las próximas X veces.

11. Realiza un programa que lea una secuencia de números enteros en una misma línea y muestre como resultado la línea en orden inverso.

El programa incluirá una primera línea X, $1 \leq X \leq 100$, indicando cuantos números a leer (esa línea no se tendrá en cuenta). Tras ello en la siguiente línea se leerán X números separados por espacios y se mostrará finalmente en una sola línea el resultado.

12. Realiza un programa que pinte un tablero de ajedrez con código HTML.

Ejercicios avanzados de sintaxis

Sudoku

Realiza una aplicación web que compruebe si una solución de un Sudoku es correcta o no.

Una sugerencia de diseño para la aplicación web es que debe tener 9x9 campos de texto y un botón “Comprobar”. Se recomienda tener una solución de un Sudoku válido para hacer pruebas.

La explicación de las reglas del Sudoku las tenéis aquí <https://es.wikipedia.org/wiki/Sudoku>

Internamente el programa realizará la comprobación de si el Sudoku es o no correcto en una función definida como **function esSudokuCorrecto(miArrayBi)** que devolverá true si es correcto, false en caso contrario.

Patrones

Realiza una aplicación web que solicite una cadena de texto.

El programa debe decir cuántas veces ocurre cada uno de estos patrones sin distinguir mayúsculas y minúsculas: “00” “101”, “ABC”, “HO”.

Un carácter puede formar parte de más de un patrón encontrado. Por ejemplo: En la cadena “000” el patrón “00” aparece dos veces (una empieza en la posición 0 y otra empieza en la posición 1).

Internamente el programa realizará la cuenta de patrones con una función definida como **function numeroPatrones(texto)** que devolverá un número entero con el número de patrones encontrados.

Buscaminas

Realiza una aplicación web que reciba en código mediante un array bidimensional (de longitud variable) un escenario de Buscaminas, donde haya un 0 donde no hay minas y un -1 donde si hay.

Para cada casilla que no tenga una mina, diga cuantas minas adyacentes hay (en diagonal, horizontal y vertical).

Internamente el programa realizará las acciones con una función definida como **function contandoMinas(miCampo)** que devolverá un array bidimensional con el número de minas adyacentes en cada posición.