

## Desarrollo Web en entorno cliente

# Sintaxis de Javascript

## Ejercicios propuestos - 2

### Condiciones

#### 1.

Leer por la entrada estándar (normalmente teclado) tres números usando `readline()`. Si al menos uno de los valores introducido es mayor a 10, mostrar por la salida estándar (normalmente el teclado) “ALGUNO MAYOR QUE 10”. En caso contrario mostrar “NINGUNO MAYOR QUE 10”.

El comando `readline()` te permite leer una línea completa.

#### 2.

Leer por la entrada estándar (normalmente teclado) día y mes en formato numérico. Realizar un programa que imprima SI si es navidad (el 25 del 12) y NO si es no es navidad.

#### 3.

De un operario se obtiene por teclado una línea con su sueldo (sin decimales) y en otra línea 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.

#### 4.

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

#### 5.

Escribe un programa que lea en una línea el número de horas trabajadas y en otra línea el precio por hora.

Calcula 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.

Imprime por la salida estándar el salario calculado.

### Bucles

#### 1.

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.

## 2.

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.

## 3.

Desarrollar un programa que muestre la tabla de multiplicar del número  $X$ , 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.

## 4.

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.

Se leerá una nota por línea.

## 5.

Realiza un programa que lea número y diga si es primo o no. El número puede tomar los valores  $1 \leq \text{número} \leq 10000$ .

Recuerda que un número primo es aquel que solo es divisible por sí 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](https://es.wikipedia.org/wiki/N%C3%BAmero_primo)

Utiliza una forma optimizada para reducir el número de operaciones.

## 6.

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 por la salida estándar el número mínimo de billetes.

## 7.

Realiza un programa que lea una secuencia de números enteros, uno por línea.

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

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 se leerán  $X$  números en las próximas  $X$  líneas.

## 8.

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.

## 9.

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.