

TAREA UT2

ESTRUCTURA DEL LENGUAJE JAVASCRIPT

La tarea consiste en la implementación de varios scripts en lenguaje Javascript. Se valorará positivamente la claridad del código, el uso de las convenciones pedidas para el nombrado de ficheros, funciones y variables y el uso de comentarios en el código, así como la corrección de los resultados obtenidos.

Es obligatorio implementar las funciones que se detallan en cada uno de los ejercicios, así como ceñirse a los nombres propuestos para las mismas, sus argumentos y tipos de entrada y valores de salida esperados.

Salvo que se indique lo contrario, todos los mensajes se mostrarán por la consola de depuración del navegador.

1. Crea un script que pida al usuario dos reales. A continuación se mostrará un mensaje que indique si uno de ellos es positivo y el otro es negativo.

Función	Recibe	Devuelve	Descripción
esNumero	c: cadena	booleano	Devuelve verdadero si c puede convertirse a un valor numérico y falso en caso contrario.
compruebaNumeros	numero1: real numero2: real	booleano	Devuelve verdadero si uno de los números reales pasados como parámetros ¹ es positivo y el otro negativo, y falso en caso contrario.

2. Crea un script que pida al usuario un número entero positivo. A continuación se mostrará en un mensaje por consola el sumatorio de dicho número.

Función	Recibe	Devuelve	Descripción
esEntero	c: cadena	booleano	Devuelve verdadero si c puede convertirse a un número entero y falso en caso contrario.
esPositivo	numero: entero	Booleano	Devuelve verdadero si numero
Sumatorio	numero: entero	entero	Devuelve el sumatorio de numero

En caso de que el usuario no haya introducido un número entero positivo, se mostrará un mensaje de error indicativo y se pedirá de nuevo. Dicho proceso se repetirá hasta que el número introducido sea correcto.

3. Crea un script que pida al usuario cuatro números enteros. A continuación se mostrará un mensaje que indique si tanto el primer como el segundo número están en el intervalo cerrado formado por el tercer y el cuarto número. Nótese que los números pueden ser positivos o negativos.

Función	Recibe	Devuelve	Descripción
enIntervalo	numero: entero maximo: entero minimo: entero	booleano	Devuelve verdadero si numero está en el intervalo cerrado formado por maximo y minimo y falso en caso contrario.

En caso de que alguno de los números introducidos no sea entero, se mostrará un mensaje de error indicativo y se pedirá de nuevo. Dicho proceso se repetirá hasta que el número introducido sea correcto.

¹ Nótese que la función asume que se están pasando números reales, de modo que las validaciones pertinentes deben hacerse antes de llamar a la función. Esto se aplicará a todos los ejercicios.

4. Crea un script que pida tres números reales e indique el mayor de dichos números. En caso de haber varios mayores se devolverá uno de ellos de manera arbitraria.

Función	Recibe	Devuelve	Descripción
numeroMayor	n1: real n2: real n3: real	-	Imprime el mayor de los números introducido como parámetro.

En caso de que alguno de los números introducidos no sea real, se mostrará un mensaje de error indicativo y se pedirá de nuevo. Dicho proceso se repetirá hasta que el número introducido sea correcto.

5. Crea un script que pida 4 números enteros positivos e indique si dichos números están en orden estrictamente creciente.

Función	Recibe	Devuelve	Descripción
ordenCreciente	n1: entero n2: entero n3: entero n4: entero	boolean	Devuelve verdadero si los números recibidos como parámetros están en orden estrictamente creciente, y falso en caso contrario.

En caso de que alguno de los números introducidos no sea entero positivo, se mostrará un mensaje de error indicativo y se pedirá de nuevo. Dicho proceso se repetirá hasta que el número introducido sea correcto.

6. Crea un script que pida al usuario un número entero entre 1 y 100. A continuación se mostrará un mensaje que indique si el número:
- Es divisible entre 2.
 - Es divisible entre 3.
 - Es divisible entre 5.
 - Es divisible entre 10.

Indicaciones:

- Un número es divisible entre N cuando el resto de dividir el número entre N es 0.
- Se mostrará un único mensaje que indique entre qué números es divisible el número. Además, en caso de que no sea divisible entre ninguno de los citados, se mostrará un mensaje indicativo.

Función	Recibe	Devuelve	Descripción
esDivisible	dividendo: entero divisor: entero	booleano	Devuelve verdadero si el primer parámetro es divisible entre el segundo y falso en caso contrario

En caso de que el número no esté entre 1 y 100 o no se introduzca un número, se mostrará un mensaje de error indicativo y se pedirá de nuevo. Dicho proceso se repetirá hasta que el número introducido sea correcto.

7. Implementa un script con una pequeña calculadora que haga las siguientes operaciones con enteros:
- Pedir un operando.
 - Pedir otro operando
 - Pedir la operación a realizar (puede ser suma, resta, multiplicación o división o FIN para finalizar).

A continuación se mostrará en consola el resultado de la operación.

Es necesario validar:

- Que los operandos y el operador no sean vacíos.
- Que el operador sea uno de los permitidos (+,-,*,/,FIN).
- En caso de que el operador sea la división, que el segundo operando no sea nulo.

Se realizarán operaciones mientras el operador introducido sea distinto de FIN, en cuyo caso el script finalizará. Además, en caso de que alguna de las validaciones sea incorrecta, se mostrará un mensaje indicativo y se seguirá pidiendo el dato hasta que este sea correcto.

Función	Recibe	Devuelve	Descripción
validaOperador	operador:cadena	booleano	Valida que el operador sea permitido
validaOperando	numero:entero	booleano	Valida que el operando sea válido
divisionValida	dividendo:entero divisor:entero	booleano	Valida que la división sea válida.

8. Implementa un script gestione un control de acceso básico a una aplicación. Se pedirán (en dos ventanas distintas) un login y una contraseña:

En función de la validez de las mismas, se informará al usuario acerca de uno de los siguientes mensajes:

- El nombre de usuario y la contraseña son correctas*
- El nombre de usuario no existe*
- El nombre de usuario no coincide con la contraseña.*

Para comprobar los datos, se usarán dos constantes globales de tipo array. La primera de ellos contiene 5 nombres de usuario, mientras que la segunda de ellas contiene las contraseñas de dichos usuarios, La contraseña en la posición i-ésima del array de contraseñas se corresponderá con el login ubicado en la posición i-ésima del array de logins.

Por ejemplo, benito tendrá la contraseña 67890.

pepe	paco	juan	benito	manuel
------	------	------	--------	--------

12345	23456	09876	67890	666666
-------	-------	-------	-------	--------

El usuario tendrá 5 intentos para introducir correctamente el nombre de usuario y la contraseña.

Función	Recibe	Devuelve	Descripción
existeLogin	login:cadena	booleano	Valida que el login exista
passwordCorrecto	password:cadena login:cadena	Booleano	Valida que el password correspondiente al login proporcionado sea correcto.

9. Crea un script que pida un número real que representa una cantidad en euros. A continuación se mostrará el número mínimo de monedas necesario para pagar dicha cantidad.

Para representar las monedas se usará un array de 8 posiciones, donde cada posición del array almacena el número de monedas de cada tipo, ordenadas por valor decreciente.

- Posición 1: Número de monedas de 2 euros
- Posición 2: Número de monedas de 1 euro
- Posición 3: Número de monedas de 50 céntimos.
- Posición 4: Número de monedas de 20 céntimos.
- Posición 5: Número de monedas de 10 céntimos.
- Posición 6: Número de monedas de 5 céntimos.
- Posición 7: Número de monedas de 2 céntimos.
- Posición 8: Número de monedas de 1 céntimo.

Ejemplo: ¿Qué cantidad desea convertir? → 5,26

5,26 euros son 2 x 2 euros, 1 x 1 euro, 1 x 20 céntimos, 1 x 5 céntimos, 1 x 2 céntimos.

2	1	0	1	0	1	2	0
---	---	---	---	---	---	---	---

Función	Recibe	Devuelve	Descripción
pasoAMonedas	cantidad:real	entero[]	Devuelve un array de enteros que representan las monedas necesarias para representar la cantidad.

En caso de que la cantidad introducida no sea un número real, se pedirá de nuevo. Dicho proceso se repetirá hasta que el número introducido sea correcto.

El script se repetirá hasta que el usuario introduzca la palabra “FIN” en lugar de un número real.

10. Se realizará una modificación del script anterior. En este caso, tenemos una **variable global**² llamada **cajaMonedas** que consiste en la cantidad de monedas de cada tipo de la cual disponemos.

Al comenzar el script se preguntará al usuario la cantidad de monedas de la que disponemos y se inicializará cada uno de los valores de **cantidadMonedas** a esa cantidad.

Función	Recibe	Devuelve	Descripción
inicializaMonedas	cantidadInicial	-	Inicializa los valores del array cajaMonedas al entero introducido como parámetro.
pasoAMonedas	cantidad:real	entero[]	Devuelve un array de enteros que representan las monedas necesarias para representar la cantidad. Se actualizarán los valores de cajaMonedas .
maximoDisponible	-	real	Devuelve la cantidad máxima disponible en cajaMonedas

² El uso de variables globales es una práctica que vamos a intentar evitar en la medida de lo posible, pero se puede de manera controlada (solo con ciertas variables que vamos a usar desde todo nuestro script).

Ejemplo: ¿Cuántas monedas de cada tipo tenemos inicialmente? → 5.

cajaMonedas

5	5	5	5	5	5	5	5
---	---	---	---	---	---	---	---

¿Qué cantidad desea convertir? → 9,99

9,99 euros son 4 x 2 euros, 1 x 1 euro, 1 x 50 céntimos, 2 x 20 céntimos, 1 x 5 céntimos, 2 x 2 céntimos

cajaMonedas

1	4	4	3	5	4	3	5
---	---	---	---	---	---	---	---

¿Qué cantidad desea convertir? → 4,51

4,51 euros son 1 x 2 euros, 2 x 1 euro, 1 x 50 céntimos, 1 x 1 céntimo.
Se han agotado las monedas de 2 euros.

cajaMonedas

0	2	4	3	5	4	3	4
---	---	---	---	---	---	---	---

¿Qué cantidad desea convertir? → 9

No tenemos suficiente dinero para convertir dicha cantidad a monedas.