

### Mensajes del navegador

Combina todos los mensajes del navegador vistos hasta ahora. Para ello, crea un fichero JavaScript que tras ejecutarlo en el navegador:

- Pida el nombre al usuario. El usuario deberá introducir su nombre.
- Le pregunte al usuario si quiere abandonar el programa. El usuario deberá pulsar Aceptar o Cancelar.
- Lance una alerta con la decisión del usuario.

Tipo de salida:

Busca que tipo de salida dan los métodos usados en el ejercicio anterior.

Ejercicio +:

Crea un programa que pida a un usuario dos números y si quieres sumarlos o concatenarlos. Saca el resultado por consola de lo que se le haya pedido.

Ejemplo de uso:

Dame el primer número:

➔ 2

Dame el segundo número:

➔ 3

¿Deseas sumar?:

➔ No

¿Deseas concatenar?

➔ Sí

El resultado es 23.

Posibles modificaciones (Dif):

- El usuario da los dos números de la forma 2, 3 y el programa entiende automáticamente que se refiere a primer número 2, segundo número 3. Pista: tendrás que investigar sobre el método `.split()` de javascript.
- Busca cómo se usa el operador ternario y modifica la condición de sumar o concatenar.

### Estructura de control if

Crea un fichero JavaScript e incluye las instrucciones necesarias que te permitan ejecutar la siguiente lógica:

- Pídele su nombre y edad al usuario.
- Utiliza la estructura de control if para determinar en qué franja de edad se encuentra, de manera que consideres lo siguiente:
  - Niño: hasta 12 años.
  - Adolescente: de 13 a 17 años.
  - Trabajador: de 18 a 64 años.
  - Jubilado: de 65 años o más.
- El programa debe terminar mostrando este tipo de mensaje en verde y negrita por consola: «Juan tiene 21 años y por tanto es Trabajador».

### Estructura de control switch

Crea un fichero JavaScript e incluye las instrucciones necesarias que te permitan ejecutar la siguiente lógica de gestión de horarios de apertura:

- Pídele al usuario la inicial del día de la semana (L, M, X, J, V, S, D).
- Por medio de la estructura de control switch muestra por consola el horario de apertura del día elegido. Por supuesto, el horario debe ser diferente cada día.

### Estructura de control while

Vamos a jugar a adivinar la letra.

- Crea un programa con un bucle while cuya condición de salida sea el valor 's'.
- Dentro del bucle while pídele al usuario una letra.
- Si la letra elegida es la «s» el programa finaliza y muestra un mensaje de éxito por consola. Si la letra elegida no es la «s» se muestra un mensaje por consola de no haber habido suerte y se le vuelve a pedir una letra al usuario.
- El bucle se ejecutará indefinidamente hasta que el usuario introduzca la letra «s».

Sustituye el bloque anterior para que utilice el bucle do while.

DNI:

Crea un programa que evalúe si un DNI introducido con el prompt es correcto (comprobando la letra).

### Estructura de control for

Crea un programa que muestre en la consola todos los números múltiplos de 7 que hay del 1 al 100. Al finalizar debe indicar cuántos son.

### Liga local de fútbol

Quieres crear una estructura de datos que almacene la tabla de clasificación de una liga local de fútbol con estos datos:

- La liga la disputan 10 equipos.
- Para cada equipo necesitas estos datos:
  - Nombre.
  - Puntos.
  - Partidos jugados, ganados, empatados y perdidos.
  - Goles a favor y goles en contra.

Crea la estructura de datos que consideres más útil, rellénala con datos coherentes y muestra en pantalla toda la información de la clasificación del equipo ganador.

Crea un mapa vacío y añade los DNI de diez personas ficticias, usando el DNI como clave y el nombre como valor. A continuación, muestra en pantalla la lista de todos los DNI junto con el nombre de las personas. Modifica el nombre de la tercera persona y vuelve a mostrar todos los datos en pantalla para comprobar que la operación se ha realizado correctamente.

Crea un mapa con cinco pares de elementos de manera que tanto las claves como los valores sean cadenas de caracteres. Luego idea algún mecanismo para ordenar alfabéticamente los valores del mapa. Muestra el resultado en pantalla.

Escribe un programa para comprobar si dos arrays unidimensionales contienen los mismos elementos.

Utiliza las estructuras de datos que consideres oportunas para generar un tablero del tres en raya con todas las posiciones rellenas e indique si alguno de los jugadores ha ganado la partida.

Repaso prompt y arrays:

Leer una cadena de texto mediante el método `prompt()` y generar un array con las palabras que contiene. Posteriormente, mostrar la siguiente información: número de palabras, primera palabra y última palabra.

Dif:

Crea un programa que detecte cuántos elementos repetidos hay en un array unidimensional e indique en qué posiciones se encuentran.