

# PÁCTICAS DE JAVASCRIPT

## Ejercicio 1

Modificar el primer script para que:

1. Todo el código JavaScript se encuentre en un archivo externo llamado `codigo.js` y el script siga funcionando de la misma manera.
2. Después del primer mensaje, se debe mostrar otro mensaje que diga *"Soy el primer script"*
3. Añadir algunos comentarios que expliquen el funcionamiento del código
4. Añadir en la página XHTML un mensaje de aviso para los navegadores que no tengan activado el soporte de JavaScript

## Ejercicio 2

Modificar el primer script del capítulo anterior para que:

1. El mensaje que se muestra al usuario se almacene en una variable llamada `mensaje` y el funcionamiento del script sea el mismo.
2. El mensaje mostrado sea el de la siguiente imagen:

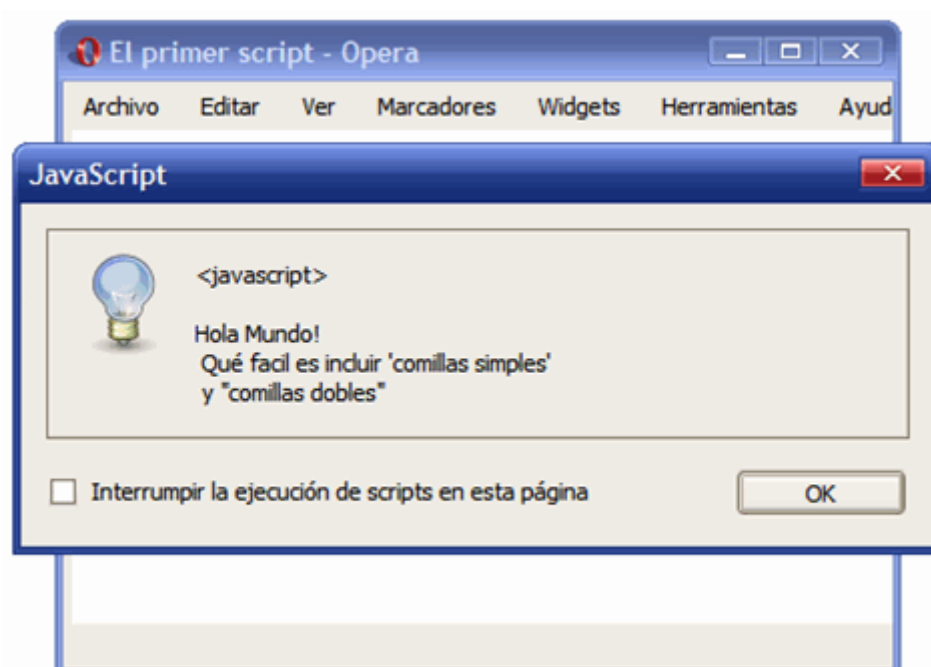


Figura 3.1 Nuevo mensaje que debe mostrar el script

## Ejercicio 3

Crear un array llamado `meses` y que almacene el nombre de los doce meses del año. Mostrar por pantalla los doce nombres utilizando la función `alert()`.

## Ejercicio 4

A partir del siguiente array que se proporciona: `var valores = [true, 5, false, "hola", "adios", 2];`

1. Determinar cuál de los dos elementos de texto es mayor
2. Utilizando exclusivamente los dos valores booleanos del array, determinar los operadores necesarios para obtener un resultado `true` y otro resultado `false`
3. Determinar el resultado de las cinco operaciones matemáticas realizadas con los dos elementos numéricos

## Ejercicio 5

Completar las condiciones de los `if` del siguiente script para que los mensajes de los `alert()` se muestren siempre de forma correcta:

```
var numero1 = 5;
var numero2 = 8;

if(...) {
  alert("numero1 no es mayor que numero2");
}
if(...) {
  alert("numero2 es positivo");
}
if(...) {
  alert("numero1 es negativo o distinto de cero");
}
if(...) {
  alert("Incrementar en 1 unidad el valor de numero1 no lo hace mayor o igual que numero2");
}
```

## Ejercicio 6

El cálculo de la letra del Documento Nacional de Identidad (DNI) es un proceso matemático sencillo que se basa en obtener el resto de la división entera del número de DNI y el número 23. A partir del resto de la división, se obtiene la letra seleccionándola dentro de un array de letras.

El array de letras es:

```
var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'I'];
```

Por tanto si el resto de la división es 0, la letra del DNI es la T y si el resto es 3 la letra es la A. Con estos datos, elaborar un pequeño script que:

1. Almacene en una variable el número de DNI indicado por el usuario y en otra variable la letra del DNI que se ha indicado. (Pista: si se quiere pedir directamente al usuario que indique su número y su letra, se puede utilizar la función `prompt()`)
2. En primer lugar (y en una sola instrucción) se debe comprobar si el número es menor que 0 o mayor que 99999999. Si ese es el caso, se muestra un mensaje al usuario indicando que el número proporcionado no es válido y el programa no muestra más mensajes.
3. Si el número es válido, se calcula la letra que le corresponde según el método explicado anteriormente.
4. Una vez calculada la letra, se debe comparar con la letra indicada por el usuario. Si no coinciden, se muestra un mensaje al usuario diciéndole que la letra que ha indicado no es correcta. En otro caso, se muestra un mensaje indicando que el número y la letra de DNI son correctos.

## Ejercicio 7

El factorial de un número entero  $n$  es una operación matemática que consiste en multiplicar todos los factores  $n \times (n-1) \times (n-2) \times \dots \times 1$ . Así, el factorial de 5 (escrito como  $5!$ ) es igual a:  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Utilizando la estructura `for`, crear un script que calcule el factorial de un número entero.

## Ejercicio 8

Escribir el código de una función a la que se pasa como parámetro un número entero y devuelve como resultado una cadena de texto que indica si el número es par o impar. Mostrar por pantalla el resultado devuelto por la función.

## Ejercicio 9

Definir una función que muestre información sobre una cadena de texto que se le pasa como argumento. A partir de la cadena que se le pasa, la función determina si esa cadena está formada sólo por mayúsculas, sólo por minúsculas o por una mezcla de ambas.

## Ejercicio 10

Definir una función que determine si la cadena de texto que se le pasa como parámetro es un palíndromo, es decir, si se lee de la misma forma desde la izquierda y desde la derecha. Ejemplo de palíndromo complejo: "La ruta nos aportó otro paso natural".

## Ejercicio 11

A partir de la página web proporcionada y utilizando las funciones DOM, mostrar por pantalla la siguiente información:

1. Número de enlaces de la página
2. Dirección a la que enlaza el penúltimo enlace
3. Numero de enlaces que enlazan a <http://prueba>
4. Número de enlaces del tercer párrafo

## Ejercicio 12

Completar el código JavaScript proporcionado para que cuando se pinche sobre el enlace se muestre completo el contenido de texto. Además, el enlace debe dejar de mostrarse después de pulsarlo por primera vez. La acción de pinchar sobre un enlace forma parte de los "Eventos" de JavaScript que se ven en el siguiente capítulo. En este ejercicio, sólo se debe saber que al pinchar sobre el enlace, se ejecuta la función llamada `muestra()`.

## Ejercicio 13

Completar el código JavaScript proporcionado para que se añadan nuevos elementos a la lista cada vez que se pulsa sobre el botón. Utilizar las funciones DOM para crear nuevos nodos y añadirlos a la lista existente. Al igual que sucede en el ejercicio anterior, la acción de pinchar sobre un botón forma parte de los "Eventos" de JavaScript que se ven en el siguiente capítulo. En este ejercicio, sólo se debe saber que al pinchar sobre el botón, se ejecuta la función llamada `anade()`.

## Ejercicio 14

A partir de la página web proporcionada, completar el código JavaScript para que:

1. Cuando se pinche sobre el primer enlace, se oculte su sección relacionada
2. Cuando se vuelva a pinchar sobre el mismo enlace, se muestre otra vez esa sección de contenidos
3. Completar el resto de enlaces de la página para que su comportamiento sea idéntico al del primer enlace
4. Cuando una sección se oculte, debe cambiar el mensaje del enlace asociado (pista: propiedad `innerHTML`)

## Ejercicio 15

Completar el código JavaScript proporcionado para que:

1. Al mover el ratón en cualquier punto de la ventana del navegador, se muestre la posición del puntero respecto del navegador y respecto de la página:



**Figura 6.1** Información que se muestra para los eventos del ratón

Para mostrar los mensajes, utilizar la función `muestraInformacion()` deduciendo su funcionamiento a partir de su código fuente.

2. Al pulsar cualquier tecla, el mensaje mostrado debe cambiar para indicar el nuevo evento y su información asociada:



**Figura 6.2** Información que se muestra para los eventos del teclado

3. Añadir la siguiente característica al script: cuando se pulsa un botón del ratón, el color de fondo del cuadro de mensaje debe ser amarillo (#FFFFCC) y cuando se pulsa una tecla, el color de fondo debe ser azul (#CCE6FF). Al volver a mover el ratón, el color de fondo vuelve a ser blanco.



**Figura 6.3** El color de fondo del cuadro de información cambia en función del tipo de evento

## Ejercicio 16

Crear un script que informe al usuario en que zona de la pantalla ha pulsado el ratón. Las zonas definidas son las siguientes: izquierda arriba, izquierda abajo, derecha arriba y derecha abajo. Para determinar el tamaño de la ventana del navegador, utilizar la función `tamanoVentanaNavegador()` proporcionada.

## Ejercicio 17

Crea un formulario con un `textArea` que contenga como máximo 100 caracteres, que indicando en todo momento al usuario el número de caracteres que aún puede escribir. Y además, se debe permitir pulsar las teclas `Backspace`, `Supr.` y las flechas horizontales cuando se haya llegado al máximo número de caracteres.