

Desarrollo Web en Entornos de Cliente

Hoja de Ejercicios - JavaScript fundamentals

Descripción

Esta hoja de ejercicios está diseñada para que los estudiantes practiquen conceptos fundamentales de JavaScript, incluyendo variables, tipos de datos, operadores, estructuras de control, bucles y la documentación del código.

Ejercicio 1: Variables y Tipos de Datos

1. Declara tres variables, una para cada tipo de dato primitivo que hemos visto (cadena, número y booleano). Asigna valores a las variables y, usando la consola, muestra el tipo de cada una con `typeof`.
2. Realiza una conversión de tipo para una de las variables, por ejemplo, convierte un número a cadena.

Objetivos:

- Practicar la declaración de variables y el uso de diferentes tipos de datos.
- Convertir entre diferentes tipos de datos en JavaScript.

Ejercicio 2: Modificación de Script

Modifica el siguiente script:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>El primer script</title>

<script type="text/javascript">
  console.log("Hola Mundo!");
</script>
</head>

<body>
<p>Esta página contiene el primer script</p>
</body>
</html>
```

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. Añade algunos comentarios que expliquen el funcionamiento del código.

4. Añade en la página HTML5 un mensaje de aviso para los navegadores que NO tengan activado el soporte de JavaScript.

Objetivos:

- Externalizar el código JavaScript.
- Incluir comentarios en el código para la documentación.

Ejercicio 3: Operadores y Asignaciones

1. Declara dos variables numéricas. Usa los operadores aritméticos básicos (+, -, *, /) y muestra los resultados en consola.
2. Utiliza un operador de incremento y uno de decremento en las variables, mostrando el resultado en consola.

Objetivos:

- Practicar el uso de operadores aritméticos y asignaciones compuestas.
- Familiarizarse con el incremento y decremento de valores.

Ejercicio 4: Conversión de Tipos Implícita

1. Declara una variable de tipo cadena que contenga un número, por ejemplo "123", y otra de tipo número. Suma ambas variables y observa el resultado en la consola. Explica lo que ha sucedido en un comentario.

Objetivos:

- Comprender la conversión de tipos implícita en JavaScript.

Ejercicio 5: Operadores Comparativos y Lógicos

1. Declara dos variables numéricas. Usa operadores de comparación (<, >, ==, !=) para comparar los valores y muestra los resultados en consola.
2. Utiliza operadores lógicos (&&, ||, !) para combinar condiciones y muestra los resultados en consola.

Objetivos:

- Reforzar el uso de operadores de comparación y lógicos.

Ejercicio 6: Operadores Ternarios

Escribe un script que pida al usuario un número y utilice el operador ternario para determinar si el número es positivo, negativo o cero. Muestra el resultado en un `alert()`.

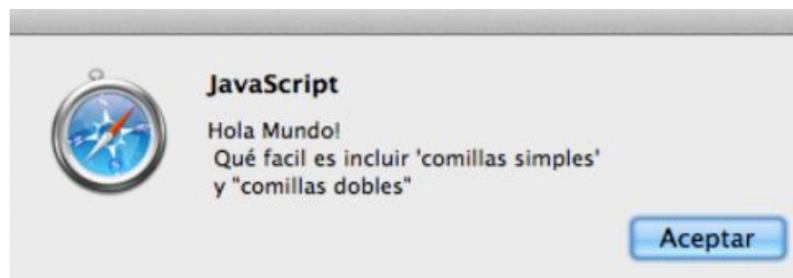
Objetivos:

- Introducir el uso del operador ternario (`condición ? expr1 : expr2`).

Ejercicio 7: Modificación de Script Avanzada

Modifica el script del ejercicio 2 para que:

1. El mensaje que se muestra al usuario se almacene en una variable llamada **mensaje** y aparezca en un pop-up.
2. El mensaje mostrado incluya:
 - Un salto de línea.
 - Comillas simples y comillas dobles.



Objetivos:

- Gestionar correctamente las cadenas de texto, incluyendo caracteres especiales.

Ejercicio 8: Condicionales y Decisiones

1. Escribe un script que pida al usuario su edad y determine si es mayor o menor de edad. Usa una estructura **if...else**.
2. Añade una segunda condición que verifique si el usuario tiene entre 18 y 25 años, mostrando un mensaje específico en ese caso.

Objetivos:

- Practicar el uso de estructuras condicionales y tomar decisiones con base en los datos introducidos.

Ejercicio 9: Estructuras Condicionales Anidadas

1. Escribe un script que pida al usuario su edad y su nacionalidad. Si el usuario tiene 18 años o más y es de nacionalidad "española", mostrar un mensaje que diga "Puedes votar". En caso contrario, muestra un mensaje que indique que no puede votar.

Objetivos:

- Practicar con condicionales anidados (**if...else**).

Ejercicio 10: Bucles con Contadores

Escribe un bucle **for** que cuente del 1 al 100 e imprima solo los números que sean divisibles por 5.

Objetivos:

- Reforzar el uso de bucles y condicionales dentro del bucle.

Ejercicio 11: Bucle **while** con Condición de Parada

Escribe un bucle **while** que pida al usuario un número entero y siga pidiendo números hasta que el usuario introduzca un número negativo.

Objetivos:

- Practicar el uso de bucles con condición de parada.

Ejercicio 12: Bucle **do...while**

Crea un bucle **do...while** que pida al usuario que introduzca una contraseña. El bucle seguirá ejecutándose hasta que el usuario introduzca la contraseña correcta ("1234"). Si se introduce la contraseña correcta, muestra un mensaje de éxito.

Objetivos:

- Introducir y practicar el bucle **do...while**.

Ejercicio 13: Control de Errores en Entrada de Datos

Escribe un script que pida al usuario un número y verifique si la entrada es realmente un número. Si no lo es, muestra un mensaje de error y vuelve a pedir la entrada. Repite hasta que el usuario introduzca un número válido.

Objetivos:

- Controlar y validar la entrada del usuario.

Ejercicio 14: Números Pares e Impares

Escribe un script en el que el usuario introduzca un número entero y muestre por pantalla una cadena de texto que indique si el número es par o impar.

Objetivos:

- Practicar el uso de operadores de módulo y condicionales.

Ejercicio 15: Completa las Condiciones en los **if**

Completa las condiciones de los **if** en el siguiente script para que los mensajes de los **alert()** se muestren 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");
}
```

Objetivos:

- Comprender el uso de condiciones dentro de bucles y estructuras de control.

Ejercicio 16: Documentación y Pruebas de Código

Toma uno de los ejercicios anteriores, añade comentarios explicando cada parte del código, y realiza pruebas en diferentes escenarios. Documenta en comentarios si el comportamiento es el esperado.

Objetivos:

- Reforzar la importancia de los comentarios y las pruebas de código en la programación.

Ejercicio 17: Bucle con Condicionales

Crea un bucle que pida al usuario su edad 5 veces y calcule cuántas veces el usuario ha introducido un valor mayor o igual a 18.

Objetivos:

- Practicar la interacción entre bucles y condicionales.



Ejercicio 18: Sumar Dígitos

Escribe un script que pida al usuario un número entero de varios dígitos y sume todos los dígitos del número. Muestra el resultado en un `alert()`.

Objetivos:

- Practicar la manipulación de números y el uso de bucles.

Ejercicio 19: Factorial de un Número

1. Escribe un script que calcule el factorial de un número introducido por el usuario. El factorial de un número n es el producto de todos los números enteros desde 1 hasta n .

Objetivos:

- Reforzar el uso de bucles para cálculos matemáticos.

Ejercicio 20: Verificación de Palíndromos

Escribe un script que determine si una cadena de texto es un palíndromo, es decir, si se lee de la misma forma desde la izquierda y desde la derecha.

Ejemplo de palíndromo:

"La ruta nos aporoto otro paso natural".

Objetivos:

- Practicar el manejo de cadenas de texto.
- Utilizar estructuras de control para verificar condiciones específicas.

Entrega

En Moodle Centros, abre la tarea llamada "Entrega actividad A01" y entrega tus soluciones con la siguiente estructura y nombres:

- e1-[apellido1][apellido2]-[nombre].js
- e1-[apellido1][apellido2]-[nombre].html