

Relación A

Variables, tipos básicos, operaciones e interacción con el usuario

Nota

En esta relación se trabaja con la declaración de variables y constantes (`let`, `const`, `var`), los tipos primitivos de JavaScript (`number`, `string`, `boolean`, `undefined`, `null`), operadores aritméticos, de comparación y lógicos, así como las funciones de interacción básica `document.write()`, `console.log()` y `window.alert()`. Los alumnos que hayan trabajado con Python o R encontrarán similitudes, pero deberán prestar atención a las diferencias en tipado dinámico y coerción de tipos.

Ejercicio 1. Cree una página HTML básica e incluya en ella un bloque `<script>` en el `<head>`. Declare las siguientes variables y constantes, y muestre su valor y tipo (usando `typeof`) con `console.log()`:

- Una constante entera con el año actual.
- Una variable de tipo flotante con el número π (use al menos 5 decimales).
- Una variable de tipo cadena con su nombre.
- Una variable booleana inicializada a `true`.
- Una variable declarada pero no inicializada.

¿Qué tipo devuelve `typeof` para la variable no inicializada?

Ejercicio 2. A partir de las variables del ejercicio anterior, realice y muestre por consola las siguientes operaciones, prestando especial atención a la coerción de tipos que aplica JavaScript:

- Concatene su nombre con una cadena que diga " , la raíz de 2 es: " y el resultado de $\sqrt{2}$ (use `Math.sqrt`).
- Sume el año (entero) y la variable booleana. ¿Qué resultado obtiene?
- Sume el año (entero) y la cadena con su nombre. ¿Qué resultado obtiene? ¿Por qué?
- Compare el año con `==` y con `===` respecto a su versión como cadena de texto. Comente la diferencia.

Ejercicio 3. Declare tres variables numéricas `a`, `b` y `c` con los valores 7, 3 y 0 respectivamente. Calcule y muestre con `document.write()` los resultados de:

- División entera de `a` entre `b` (use `Math.floor`).
- Resto de la división (`%`).
- Potencia a^b (use el operador `**` y también `Math.pow`).
- División de `a` entre `c`. ¿Qué valor devuelve JavaScript?
- Raíz cuadrada de `-1`. ¿Qué valor devuelve?

Asegúrese de que cada resultado aparece en una línea separada, usando etiquetas `
` en el `document.write()`.

Ejercicio 4. Cree una página que solicite al usuario, mediante `window.prompt()`, dos números. Calcule su suma, resta, producto y cociente. Muestre los cuatro resultados en un

mensaje emergente con `window.alert()`, con formato legible. ¿Qué ocurre si el usuario introduce texto en lugar de un número? Pruébelo y explique el resultado.

Ejercicio 5. Amplíe el ejercicio anterior para que, antes de operar, convierta explícitamente las cadenas recibidas de `prompt()` a número usando `Number()`, `parseInt()` y `parseFloat()`. Compruebe con `console.log()` si los tres métodos producen el mismo resultado para las entradas: `"3.7"`, `"3abc"` y `""`.

Ejercicio 6. Declare una variable de tipo cadena que contenga una frase larga (mínimo 10 palabras). Usando los métodos de `String`, muestre por consola:

- El número de caracteres de la cadena.
- La cadena en mayúsculas y en minúsculas.
- La posición de la primera vocal `'a'` en la cadena.
- Los primeros 5 caracteres y los últimos 5 (use `slice`).
- La cadena con todas las apariciones de la letra `'e'` reemplazadas por `'3'`.

Ejercicio 7. Usando interpolación de cadenas (template literals con backtick ```), construya y muestre con `document.write()` una pequeña ficha de un estudiante con los siguientes campos: nombre, edad, grado, nota media y si aprobó o no (booleano mostrado como `"Sí"` / `"No"`). Defina cada campo en una variable o constante separada. El resultado debe aparecer como una lista HTML (``) con un `` por campo.

Ejercicio 8. Declare las siguientes expresiones booleanas y evalúelas con `console.log()`, justificando el resultado esperado:

- `5 > 3 && 10 < 20`
- `"hola" === "Hola" | 1 == "1"`
- `!false && (3 >= 3)`
- `null == undefined` y `null === undefined`
- `NaN === NaN` y `isNaN(NaN)`

Ejercicio 9. Cree una página que solicite al usuario su año de nacimiento mediante `prompt()`. A partir de él, calcule su edad y muestre con `document.write()` un párrafo indicando: su año de nacimiento, su edad actual, si tiene más o menos de 18 años (usando el operador ternario condición `? valorSi : valorNo`) y si el año de nacimiento es par o impar.

Ejercicio 10. Explore los valores especiales de JavaScript. Declare variables que contengan `Infinity`, `-Infinity`, `NaN`, `null` y `undefined`. Para cada una, muestre por consola su valor, su tipo según `typeof`, y el resultado de compararla con `==` y `===` con `0` y consigo misma. Comente qué resultados le sorprenden respecto a lo que cabría esperar en Python o R.

Ejercicio 11. Escriba un programa que, usando únicamente variables, operadores y las tres salidas vistas (`alert`, `console.log`, `document.write`), calcule el índice de masa corporal (IMC) de una persona a partir de su peso (kg) y altura (m) dados mediante `prompt()`. Muestre el valor del IMC redondeado a 2 decimales (`toFixed`) en un `alert`, imprímalo también por consola e insértelo en el documento con `document.write()` dentro de un párrafo HTML.

Ejercicio 12. Dado el siguiente fragmento de código, analice sin ejecutarlo el resultado de cada `console.log()` y explique el porqué en cada caso. A continuación ejecútelo para verificar sus predicciones.

```
let x = "5";
let y = 2;
console.log(x + y);
console.log(x - y);
console.log(x * y);
console.log(x / y);
console.log(+x + y);
console.log(typeof (x + y));
console.log(typeof (x - y));
```

Ejercicio 13. Cree una página que realice las siguientes operaciones de redondeo y muestre cada resultado con `document.write()` y `console.log()` simultáneamente:

- Redondeo al entero más cercano de 3,5, 3,4 y $-3,5$.
- Redondeo hacia arriba y hacia abajo de 7,1 y 7,9.
- Truncamiento de $-4,9$.
- Un número aleatorio entre 1 y 100 (inclusive), generado con `Math.random()`.

Utilice los métodos apropiados del objeto `Math`.

Ejercicio 14. Investigue la diferencia entre `let`, `const` y `var` en cuanto a ámbito (*scope*) y reasignación. Cree un archivo JS externo `scope.js` que contenga código que demuestre:

- Que `let` y `const` tienen ámbito de bloque y `var` tiene ámbito de función.
- Que `const` no puede ser reasignado (intente hacerlo y capture el error con `try...catch`).
- El fenómeno de *hoisting* con `var`: declare una variable después de usarla y observe la salida.

Importe el archivo desde una página HTML y compruebe los resultados en consola.