**La coerción y operadores de comparación**

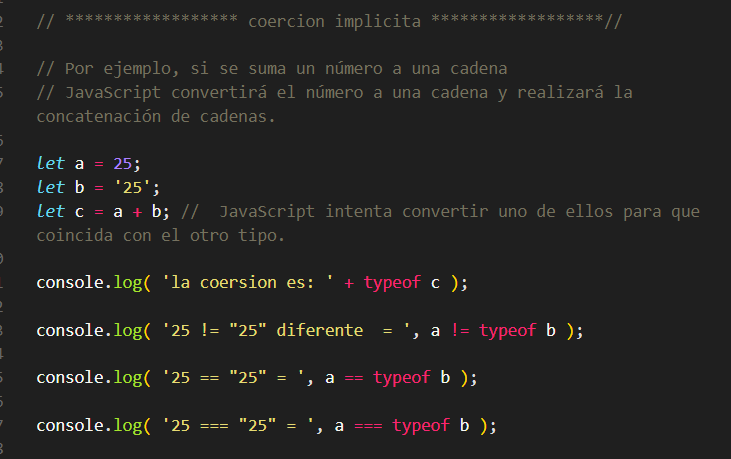
**La coerción**

En JavaScript, la coerción de tipos se refiere a la conversión automática de valores de un tipo de dato a otro. Esto sucede cuando se utilizan diferentes tipos de datos en una operación y JavaScript intenta convertir uno de ellos para que coincida con el otro tipo. La coerción puede ser implícita (automática) o explícita (realizada por el desarrollador).

**Coerción implícita:**

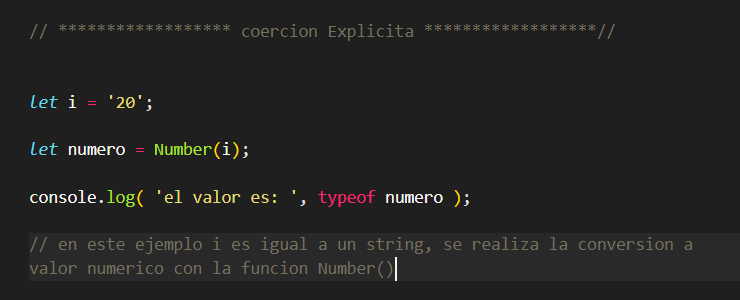
* Se produce automáticamente por JavaScript en ciertas situaciones, como cuando se utilizan operadores como **+, ==,** etc. con diferentes tipos de datos.

Por ejemplo, si se suma un número a una cadena, JavaScript convertirá el número a una cadena y realizará la concatenación de cadenas.



**Coerción explícita:**

La coerción explícita se realiza intencionalmente por el desarrollador, **utilizando funciones como Number(), String(), Boolean(), etc.** Por ejemplo:



**Ejemplos comunes de coerción implícita:**

* Suma con el operador +:

Si se utiliza el operador + con una cadena y un número, se produce la concatenación de cadenas.

* Comparación con ==:

El operador == realiza la coerción de tipos antes de comparar, mientras que === realiza una comparación estricta sin coerción.

* **Conversión a booleano:**

En contextos booleanos (como en condicionales), los valores se convierten a booleanos.

* **Coerción implícita:**
  + "5" + 9 -> "59" (el número 9 se convierte a cadena)
  + 5 == "5" -> true (el string "5" se convierte a número)
* **Coerción explícita:**
  + Number("5") + 9 -> 14 (la cadena "5" se convierte a número)
  + String(10) + 20 -> "1020" (el número 10 se convierte a cadena)

**operadores de comparación**

**En JavaScript, los símbolos =, == y === representan diferentes tipos de operaciones:**

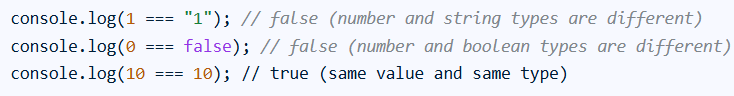
**= (Operador de asignación**): Este operador asigna un valor a una variable.

****

**== (Operador de igualdad flexible):** Este operador compara dos valores para determinar su igualdad, aplicando la conversión de tipos si es necesario. Esto significa que, si los operandos son de tipos diferentes, JavaScript intenta convertir uno o ambos operandos a un tipo común antes de realizar la comparación.

****

**=== (Operador de Igualdad Estricta):** Este operador compara dos valores para determinar su igualdad sin realizar conversión de tipos. Para que la comparación sea verdadera, tanto el valor como el tipo de dato de los operandos deben ser idénticos.



**Recomendación:** Generalmente, se recomienda usar el operador === (igualdad estricta) en JavaScript siempre que sea posible para evitar comportamientos inesperados que pueden surgir de la coerción de tipos con el operador ==. El uso de igualdad estricta promueve un código más predecible y robusto.

* Template string: (también llamado literal de plantilla) es una forma de definir cadenas de texto que permite la interpolación de expresiones y la creación de cadenas multilínea de manera más sencilla y legible que la concatenación tradicional. Se delimitan con comillas invertidas (`` ` ``) en lugar de comillas simples o dobles. Ejemplo:  **`EL MAYOR ${ a }`**
* Interpolación de expresiones:

Puedes insertar variables y expresiones dentro de la cadena utilizando la sintaxis ${...}. El valor de la expresión se evaluará y se insertará en la cadena en tiempo de ejecución.

${ a }