VARIABLES

CLASES DE VARIABLES:

1. **NATIVAS O PRIMITIVAS.**

* **NUMBER:** valores de tipo numérico.
* **STRING:** cadenas de caracteres; deben estar entre “”.
* **BOOLEAN:** valores binarios usados para validaciones, pueden contener dos posibles valores: “**true”** o **“false”.**

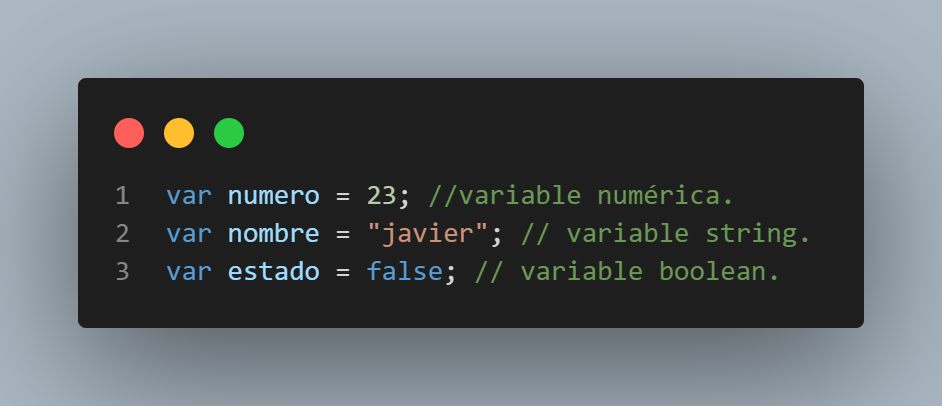
1. **DE TIPO OBEJO.**

* **ARRAY:** valor de tipo objeto, puede contener cualquier tipo de información. Está compuesto por dos **[].**
* **OBJETO:** valor de tipo objeto, puede contener cualquier tipo de información con diferentes parámetros. Está compuesto por dos **{}.**

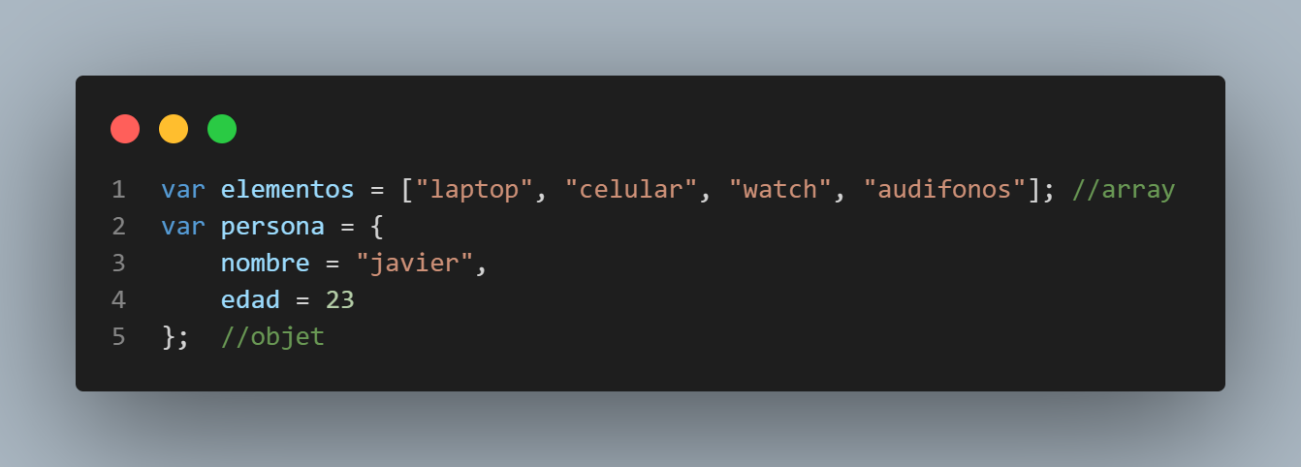
Para declara una variable JavaScript tiene la palabra designada **var.**

****

Para inicializar la variable se le asigna algún valor con el operador **=.**

Algunos ejemplos de variables:

Las variables de tipo objeto guardan más información:



**Variables String:**

Existen atributos especiales con funciones específicas como:

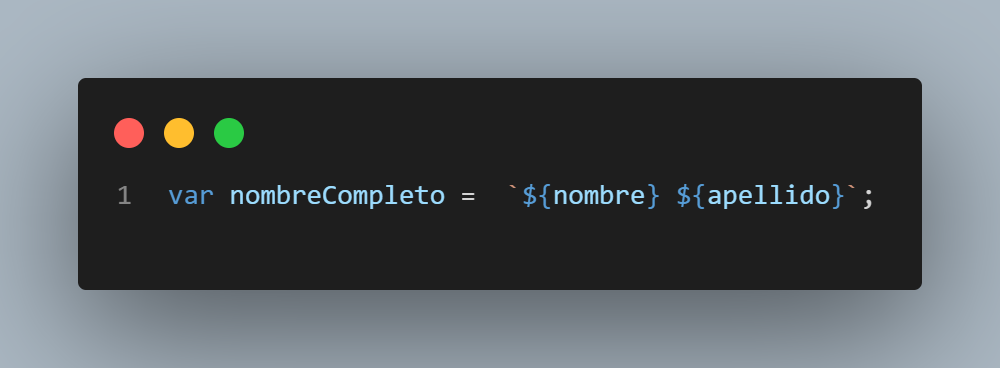
**.touppercase ():** convierte la variable (**string)** en mayúsculas.

**.tolowercase ():** para convertir en minúsculas.

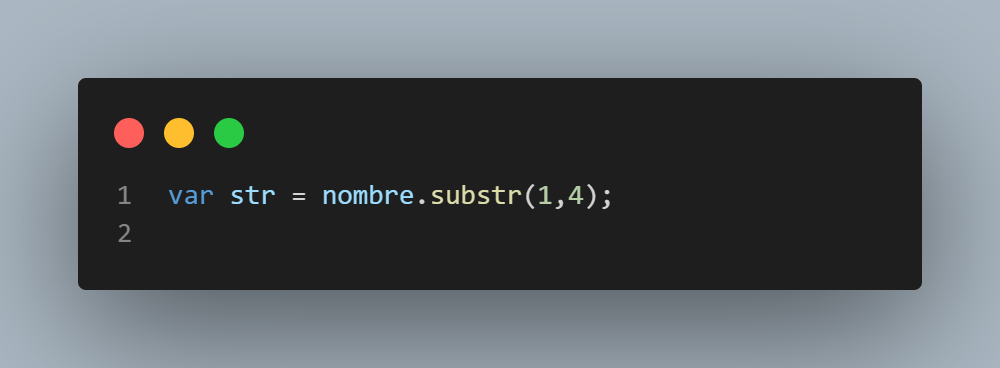
También existe atributos como el **charAt (#poscicion del carácter)** para saber qué carácter se posiciona en la ubicación dada.

**Variable.length:** nos dice el número de caracteres que compone al string.

Para concatenar dos o más string se usa la declaración:



**Substr:** para escoger caracteres dentro de un string. Su estructura va así:

****

En donde los valores de **Substr** son: el primero el carácter desde donde comenzará a escoger y el segundo el número de caracteres que tomará.

**Variables Número:**

Para incrementar el valor de una variable se puede usar la sentencia: **variable += 1;** en donde el 1 es el valor que incrementara la variable, este valor puede ser cualquier número. Para el caso de la disminución es lo mismo solo que con el operador **(-=).**

Para trabajar con decimales se suele usar el truco de multiplicar el valor \*100 luego \*el número a multiplicar y luego /dividir entre 100.

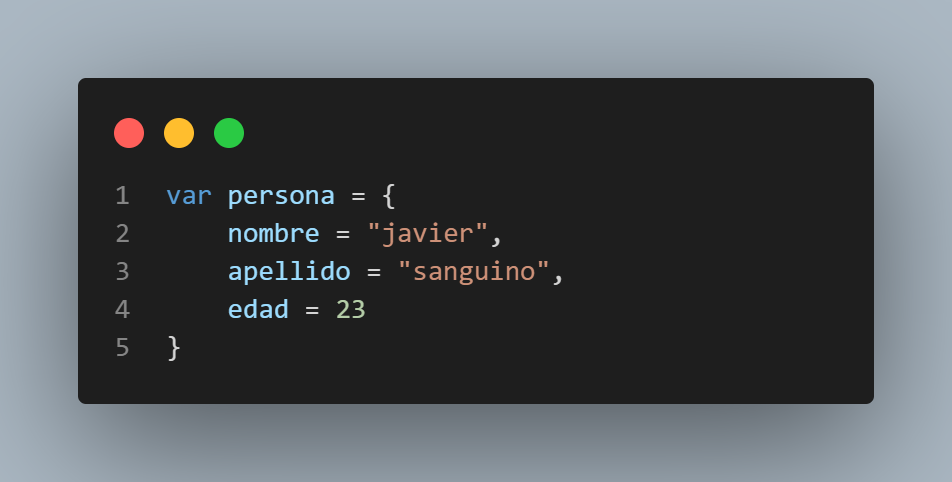
Para mostrar más de un decimal se usa declarando una variable y asignándole el valor a expresar seguido del atributo **toFixed(n)** donde **n** es la cantidad de decimales que mostrará.

Para cambiar el valor de una variable string a número se crea una variable y se le asigna por valor el atributo **parseFloat (var)** en donde **var** es el string a convertir.

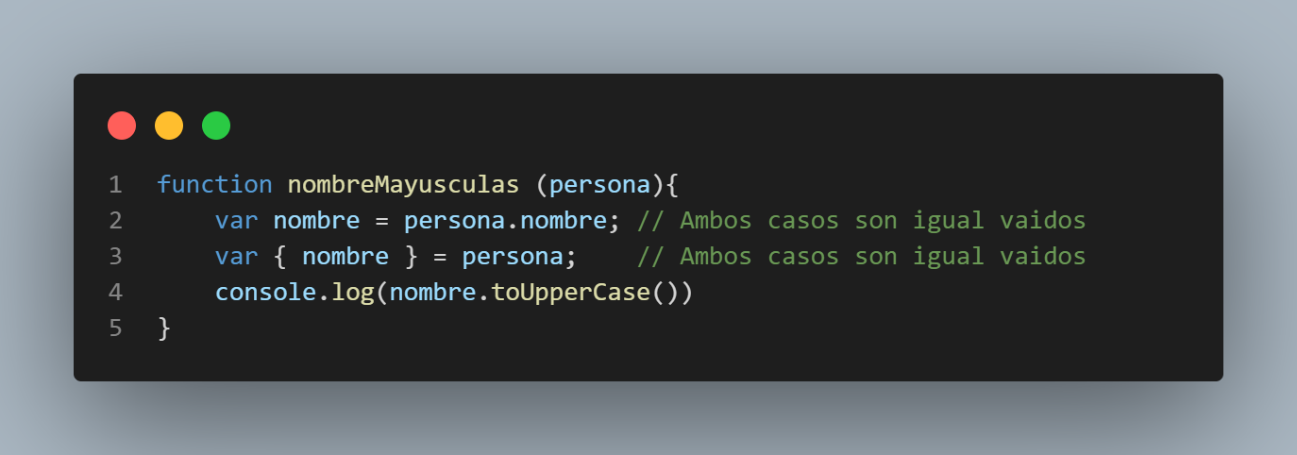
Las operaciones aritméticas básicas se realizan normalmente con su respectivo operador: **+ - / \*.**

**Objetos:**

Son variables capaces de contener diferentes datos, están compuestos por **llaves (**atributos**)** y **valores (**son los valores que se le asignan a las llaves**)**

****

Los objetos se pueden desestructurar:



**COMPARACIONES:**

En JavaScript el **=** se usa para asignar un valor a una variable, pero existen dos operadores usados para hacer comparaciones:

**==:** para comparar si dos valores poseen el mismo valor.

**===:** para comprar si dos valores son exactamente los mismos, incluyendo su tipo y valor.

**COERCIÓN:** es cuando se cambia el tipo de una variable. Puede ser implícita: cuando la genera el lenguaje; o explicita: cuando uno fuerza el cambio.

ARRAYS:

Los arrays son variables que pueden contener diferentes datos, se declaran con los corchetes [].

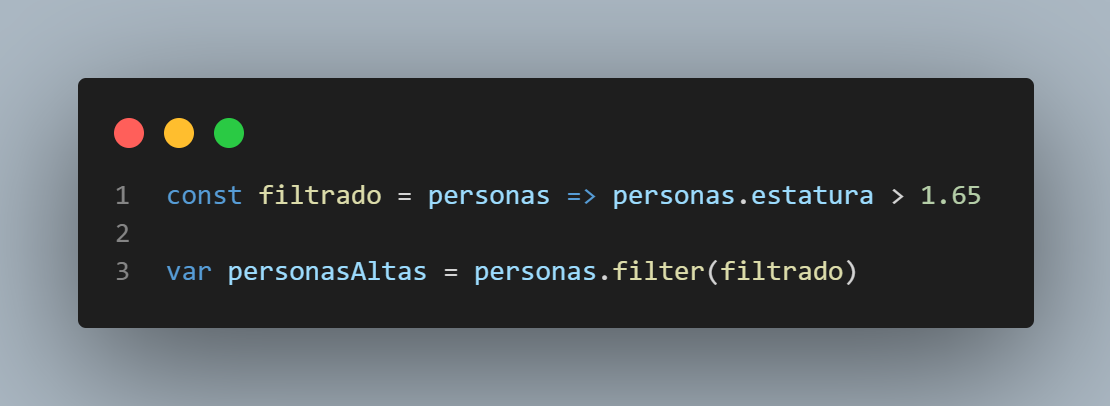
Tienen varios métodos para mutarlos y para recorrerlos:

**Métodos:**

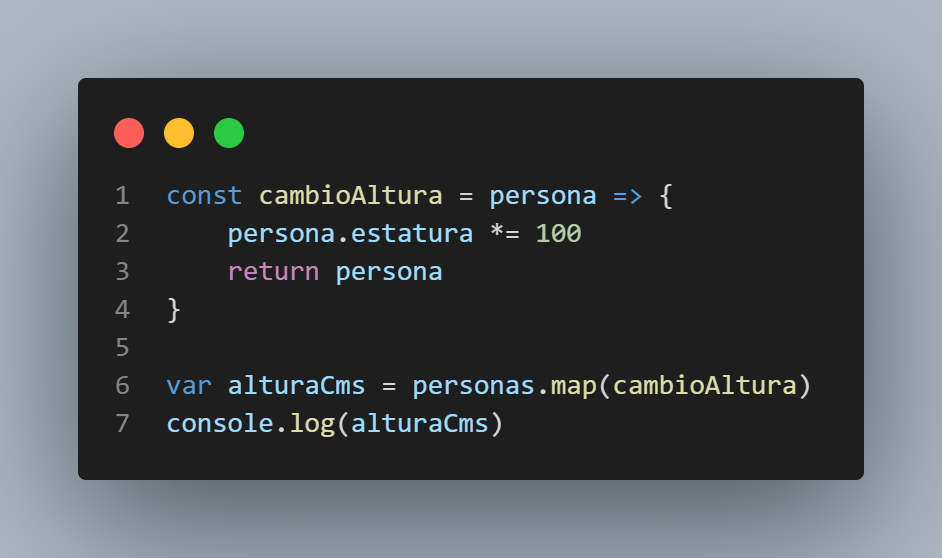
* **Push:** agrega un elemento al final del array.
* **Pop:** elimina el último elemento del array.
* **Unshift:** agrega un elemento al comienzo del array.
* **IndexOf:** para averiguar la posición de un elemento del array.

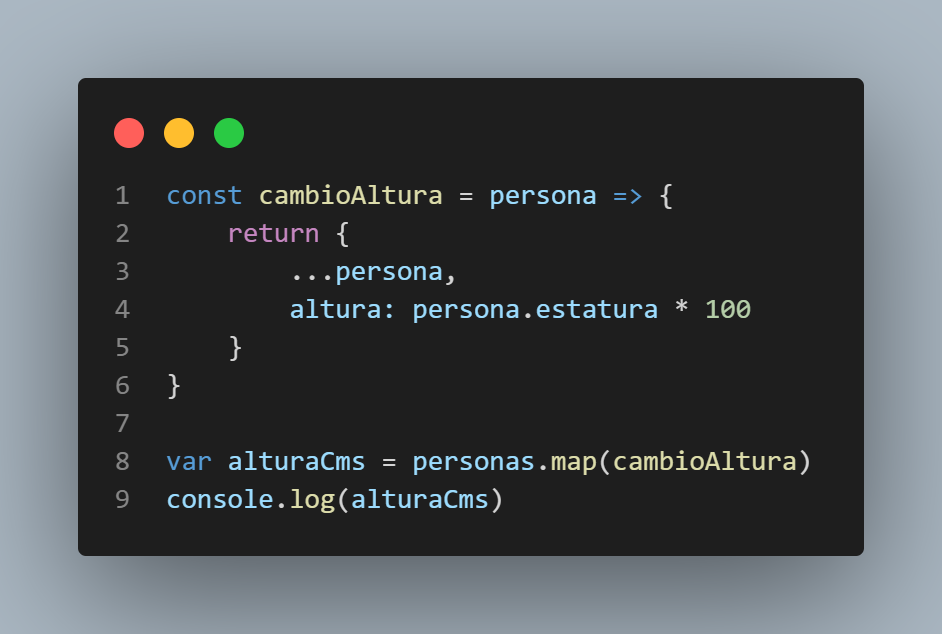
**Métodos de recorrido:**

* **Filter:** El atributo **Filter** crea un nuevo array que contendrá los elementos que cumplan con la condición que éste impone:

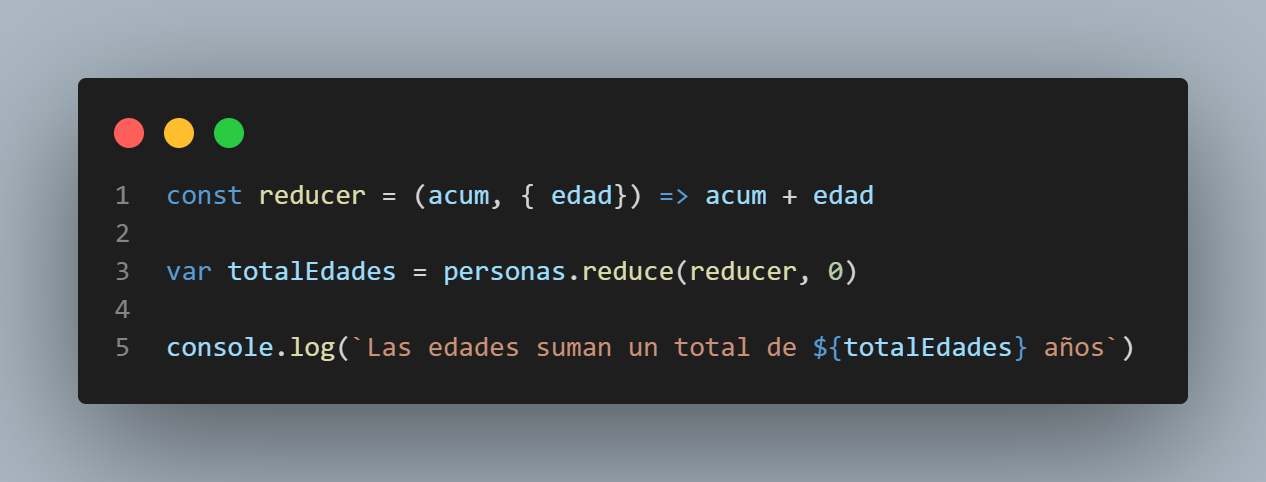


* **Map:** Crea un nuevo array que tendrá el array original con los valores modificados según se lo indiquemos (modifica también el array original).



Para que no modifique también el array original se debe indicar que cree un nuevo objeto: 

* **Find:** genera un array que contiene el valor de un elemento solicitado.
* **Reduce:** Reduce el array a un valor:



* **For each:** filtra los elementos según el parámetro que uno solicite.
* **Some:** determina mediante **false** o **true** si existe o no un elemento con las características dadas.