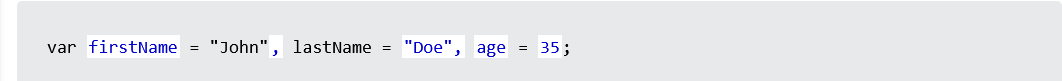
APUNTES FDLP

Variables

Crear una variable en JavaScript se llama “declarar”

Para declarar variables en una sola línea, podemos poner una “,”:



Var firstName- declaramos una variable

Var firstName=”John” – le asignamos un valor

= asigna un valor (no es igual a)

== es igual a (valor)

=== igual valor y tipo

Si ponemos el primer número entre comillas “” el sistema concatenará el resto de los num:



El resultado será 523

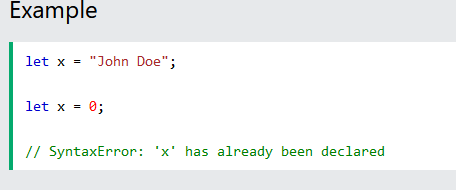
En cambio si lo hacemos con el último numero, el sistema solo concatenará el resultado de la suma de los primeros dos con el último:



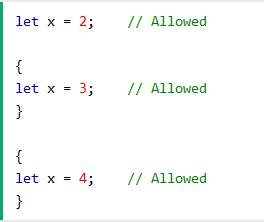
El resultado es 55.

**Variable let**

Las variables declaradas con let no pueden ser redeclaradas:

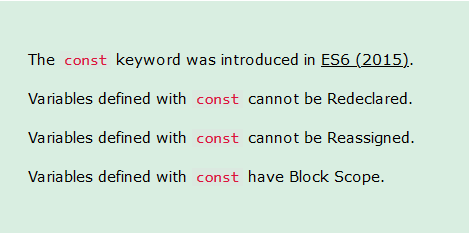


Ojo! Pueden ser redeclaradas dentro de un bloque {}

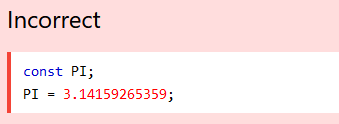


Variables const

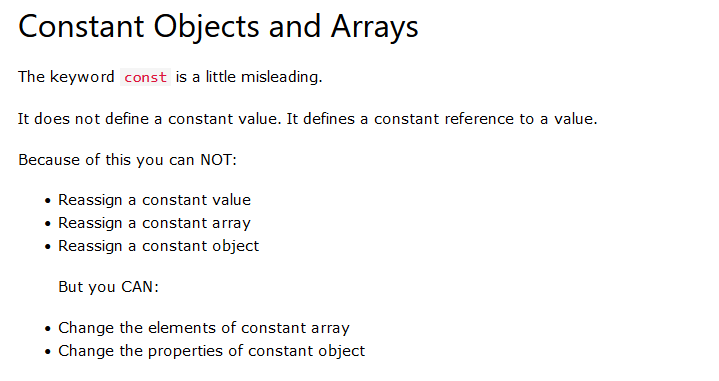
No puden ser redeclaradas ni reasignadas:



Tienes que asignarles un valor a la hora de declararlas:



Como regla general declara las variables con const a no ser que sabes que va a cambiar.

. 

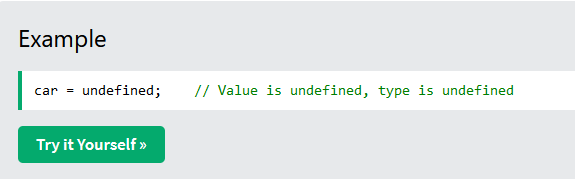
Scope de una variable hace referencia a su vida últil o visibilidad. Hay 3 tipos de ámbitos en Java Scrip:

Ámbito global: toda variable que no está dentro de un bloque, estará en un ámbito global.

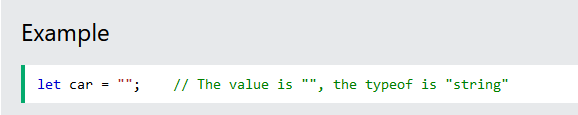
Ámbito local o de función: el ámbito de una variable declarada dentro de una función solo estará accesible dentro de esta función.

Ámbito bloque: el ámbito de una variable declarada dentro de un bloque solo estará accesible dentro de este bloque.

Para vaciar una variable hay que asignarle el valor “undefined”:

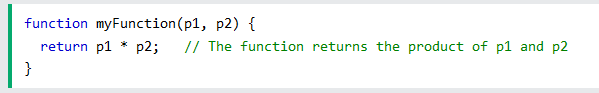


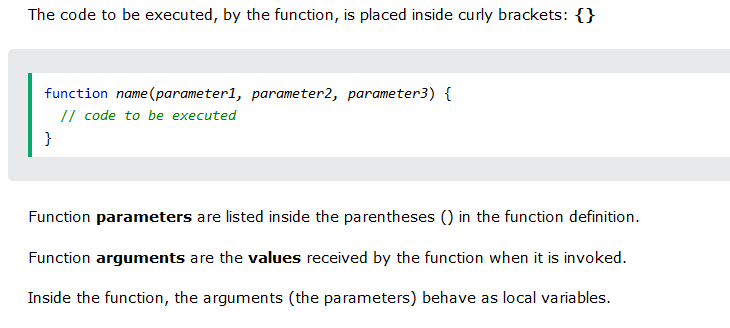
Otra cosa es cuando una variable está vacía: es este caso tiene tanto un valor como un tipo:



**Funciones:**

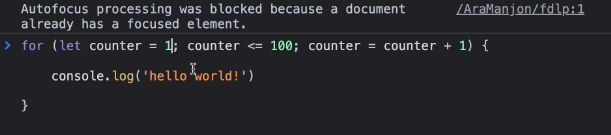
Una función es un bloque de código diseñado para llevar a cabo una tarea. En Java Script, se ejecuta cuando algo le “llama”:





Loop For:

Sirve para elementos como arrays (y todos los elementos iterable que tienen .length)

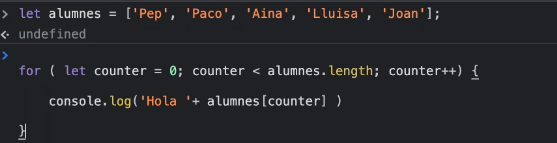


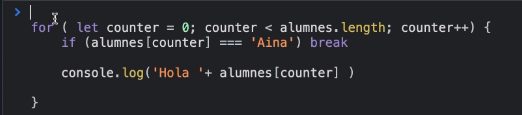




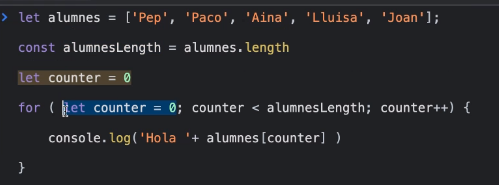
alumnes.lenght te da el num de elementos

counter





Break: para que no siga

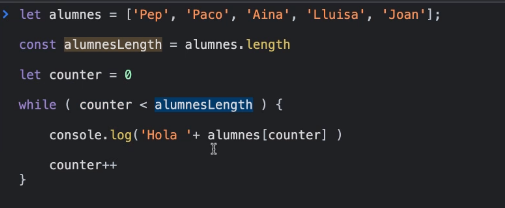


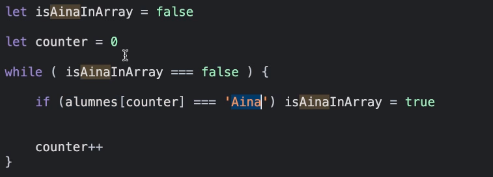
Creamos una constante para que el sist no tenga que calcular los elem de array cada vez

Se puede declarar counter fuera del bucle let counter= 0, el counter también puede estar fuera

BUCKLE WHILE

Tenemos el contador fuera. Se usa cuando no sabemos cuántas veces se va a ejecutar.



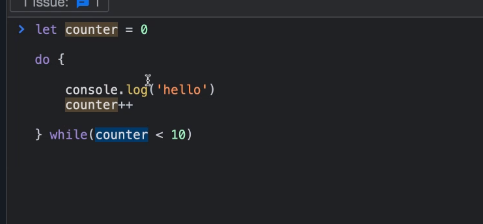


Tenemos que buscar Aina en Array

Al principio no sabemos si Aina está dentro así que counter= 0, cuando isAinaInArray=true (la hemos encontrado)  sale de la función si es false vuelve a incrementas (counter++)

Si no la encuentra entra en un bucle infinito

DO WHILE: para hacer algo una vez, el programa dice ‘Hello’ por lo menos una vez



Ejercicios: 1,2,3,4

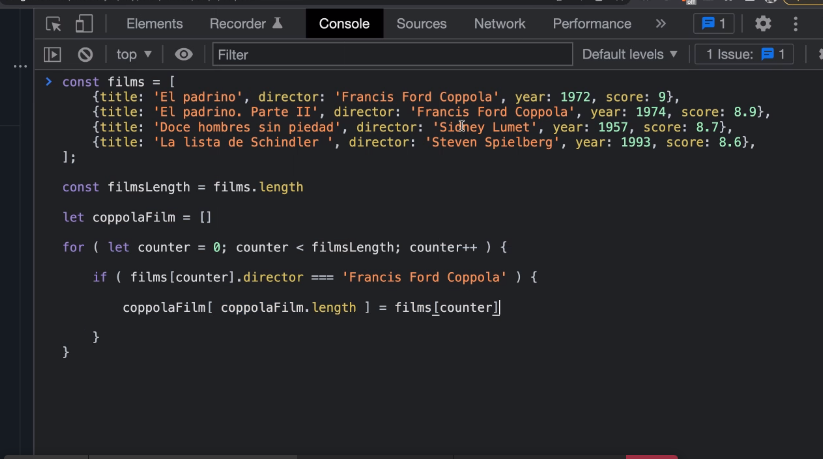
Quiero que el bucle acabe cuando encuentre la peli de SS o que pase por el array entero:



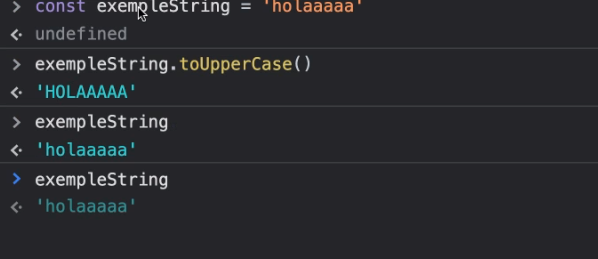
Para encontrar pelis de Coppola:

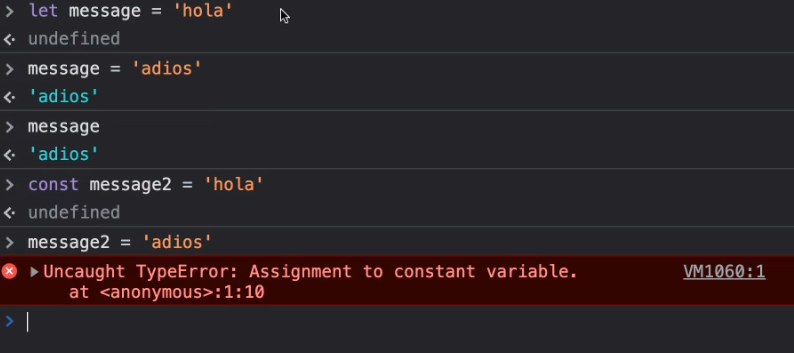
* Con push:

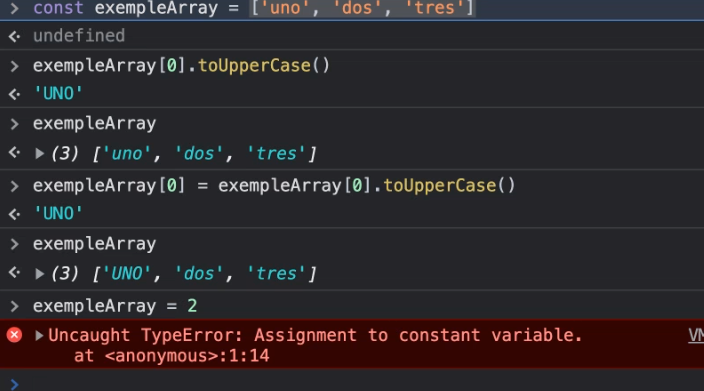


* Con counter:
* 
* 

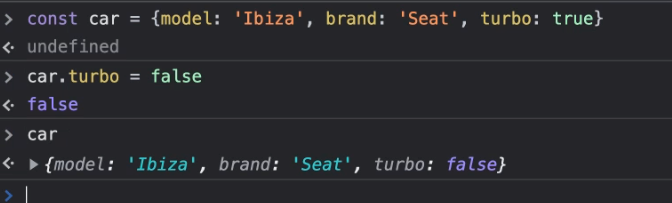
Valores no primitivos (arrays y objetos) son mutables a diferencia de los primitivos (números, booleanos, strings)

Los primitivos se pueden reasignar, pero no mutar

en arrays accedimos a un valor por posición, en objetos accedemos a un valor por key

en el ejemplo de arriba estamos mutando una variable .toUpperCase no la estamos reasignando. Si la intentamos reasignar (exampleArray = 2) nos sale un error por es una constante.

ExempleArray.join-, ExempleArray.-,mirar en mozilla docs métodos de Array



Con .push podemos añadir un elemento al final

Con .shift añadimos elementos al principio

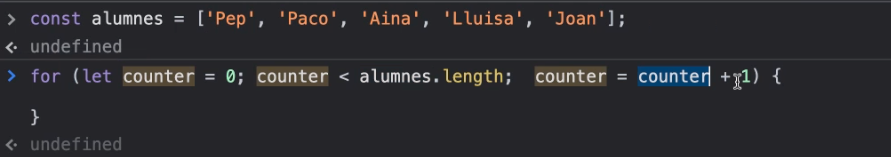
Con .pop quitamos un elemento del final – si solo queremos ver el elemento lo solicitamos por posición, con pop lo eliminamos del array.

.sort – ordena el array

.reverse

Buscar tipos de input

ejemplo: input=’password’

Alumnes.lenghth nos dice cuantas veces hay que repetir el bucle, counter++ actualiza informacion cuantas veces ya ha ocurrido

Bucle FOR EACH – para arrays, solo funciona con una función;

greeting en realidad es una variable, let greeting

Es el bucle más lento

