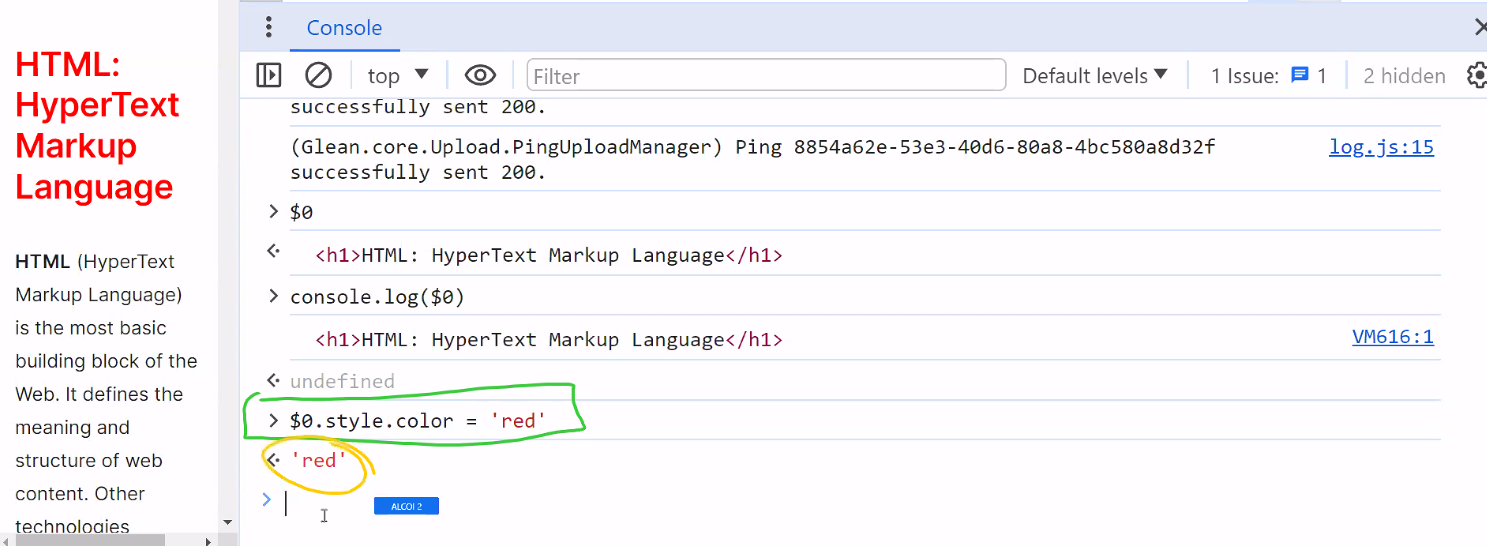
Marcando $0 cogiendo un elemento de una web podemos acceder a el.

El primero te devuelve el elementos en si al poner solo $0 y al poner console.log($0) te dice lo que contiene el elemento pero te devuelve indefinido.

Con esto podemos cambiar los estilos del elemento por ejemplo.

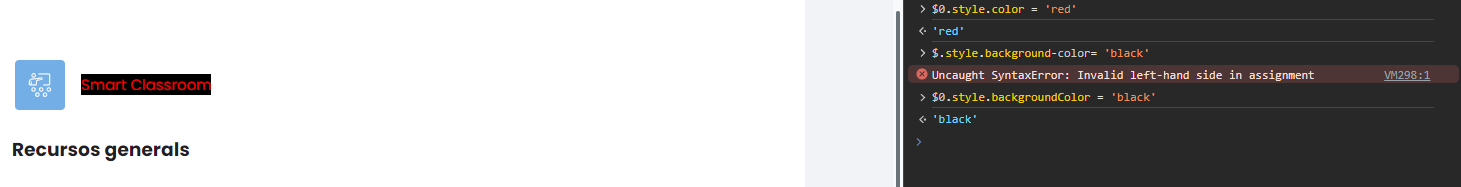


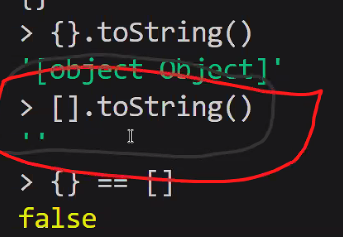
$0.style.color = ‘red’ y nos devuelve el color rojo y pone el titulo de color rojo.

Vamos a aplicar esto a **la pagina del campus** =>



Aplicamos al enlace Smart classroom el style.color = ‘red’ y nos cambia de color.

Ahora aplicamos un background color para hacerlo mas visible…



En este caso {} == [] va a dar Falso porque al pasarlo a string son cosas distintas.

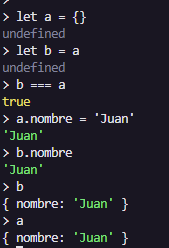
En cambio por ejemplo :

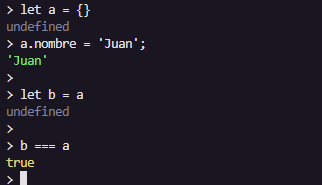


Eso va a dar igual porque al pasar el primer 3 a string va a ser igual al segundo ‘3’ == ‘3’.

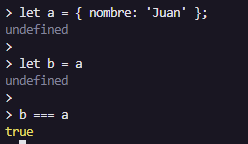
Pero con el triple igual..

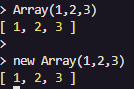
 Aquí da false porque uno es un número y otro es un string.



Al poner una variable igual a otra que tiene un objeto vacío va a ser === .

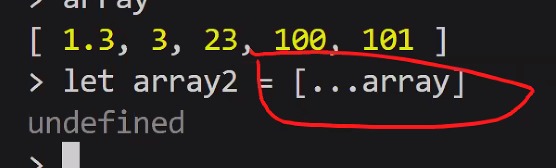
Pasa igual si es un objeto con un valor dentro.



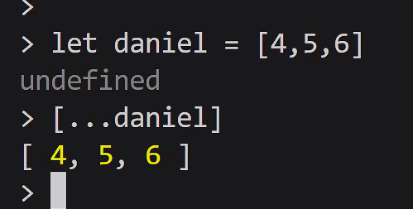
Formas de crear un **array**:  


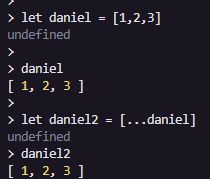
A la función Array o new Array podemos pasarle los parámetros con los elementos que queremos meter dentro del array. Por el contrario también podemos escribirlo con el numero de elementos que queremos pero con el array vacío si ponemos un solo parámetro:



Podemos copiar un array de un array anterior con esto:  


Ahora aquí array2 es igual a array poniendo […array]

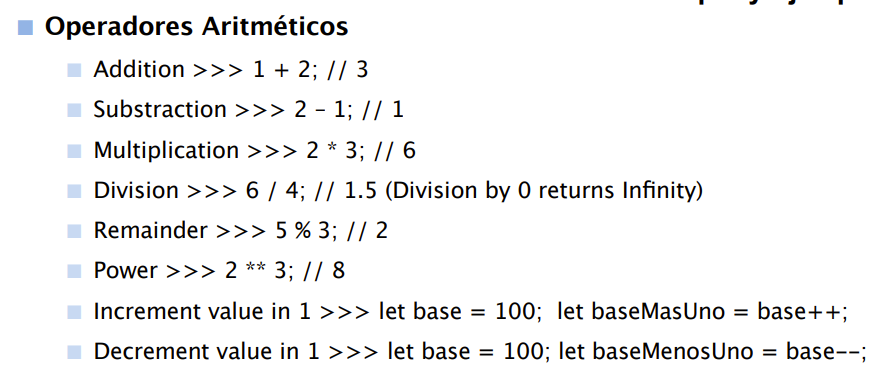
aquí también se hace una duplica del array con el nombre de la variable donde tenemos el array.



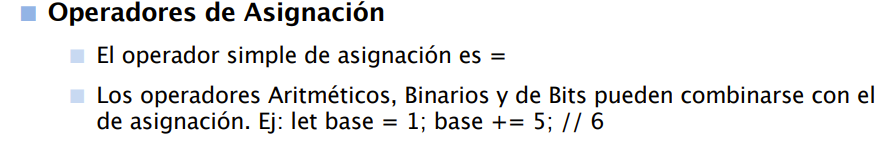
Aquí la copia se la asignamos a otra variable.

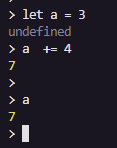
Hay varios **OPERADORES** en JS …

**Aritmeticos**

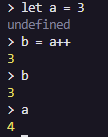


**Asignación.**

****

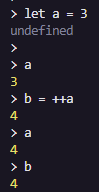
**El += lo suma y luego le da el valor a la variable.**

**Ahora los de incremento:**

****

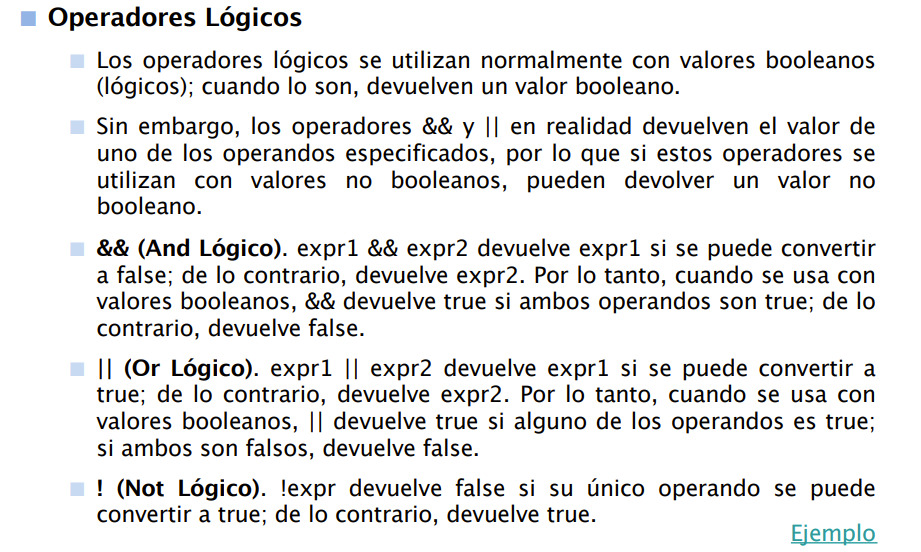
**A vale 3 y al declarar b como a ++ primero evalua que b = a y luego le añade 1 a la variable a por lo que b vale 3 y a vale 4.**

**Por el contrario…**

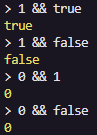
****

**Si primero hacemos el incremento y luego ponemos la variable, el incremento se hace antes de la evaluación por lo que a será 4 y b también.**

**Operadores lógicos: Se utilizan para hacer comparaciones.**

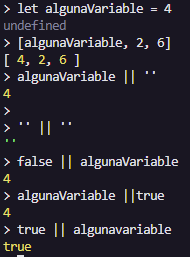


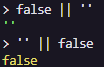
**En el operador && evalua los dos valores a ambos lados, si ambos son verdaderos devuelve el de la derecha, en cambio si uno es falso, nos devuelve el falso**

**En el operador & si ambos son verdaderos como el primer caso, nos devuelve el de la derecha.**

**Si uno es verdadero y el otro falso, nos devuelve el falso (2º y 3r caso)**

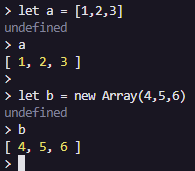
**Si ambos son falsos nos devuelve el valor de la izquierda.**

**el operador || devuelve siempre el valor verdadero, si los dos son verdaderos da el valor de la izquierda.**

**Si los dos son falsos…**

**…En este caso devuelve el valor de la izquierda, el primero que lee .**

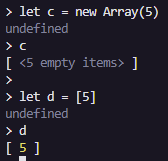
**Para crear un array hay dos formas declarar la variable con [] o el método new Array().**

****

**Ambas son formas sencillas pero tienen casos en los que son necesarios el uso de una u otra.**

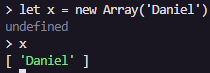
**Por ejemplo si queremos iniciar un array con un solo numero es obligatorio usar el primer método…**

**Y esto se hace porque…**

****

**Si creamos el array con un valor entre paréntesis solamente, nos crea un array con ese valor de elementos vacíos en cambio con el método de aplicar [] a la variable si se consigue hacer.**

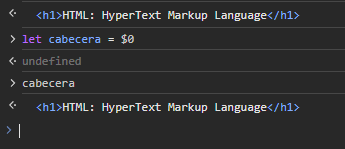
**-----ESTO SOLO APLICA A NUMEROS !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!**

****

**Vamos a trabajar con el navegador…**

**Inspeccionamos el h1 de la pagina** [**https://developer.mozilla.org/en-US/docs/Web/HTML**](https://developer.mozilla.org/en-US/docs/Web/HTML) **y al hacer $0 nos coge el h1**

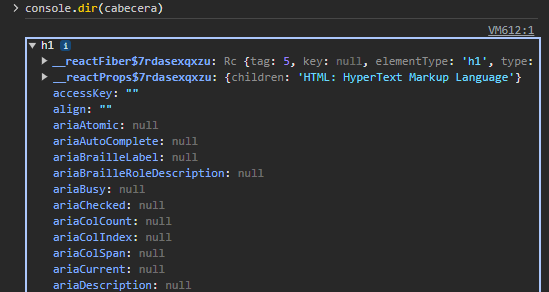
**Lo guardamos en una variable cabecera…**

****

****

**Al intentar convertirla a string nos dice que es un objeto**

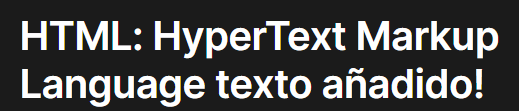
**Para ver los detalles de un objeto console.dir… nos da TODOS los atributos de ese elemento incluso los que no están definidos.**

****

Podemos añadir texto…

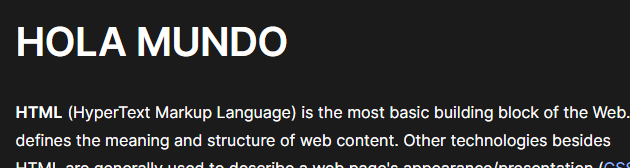


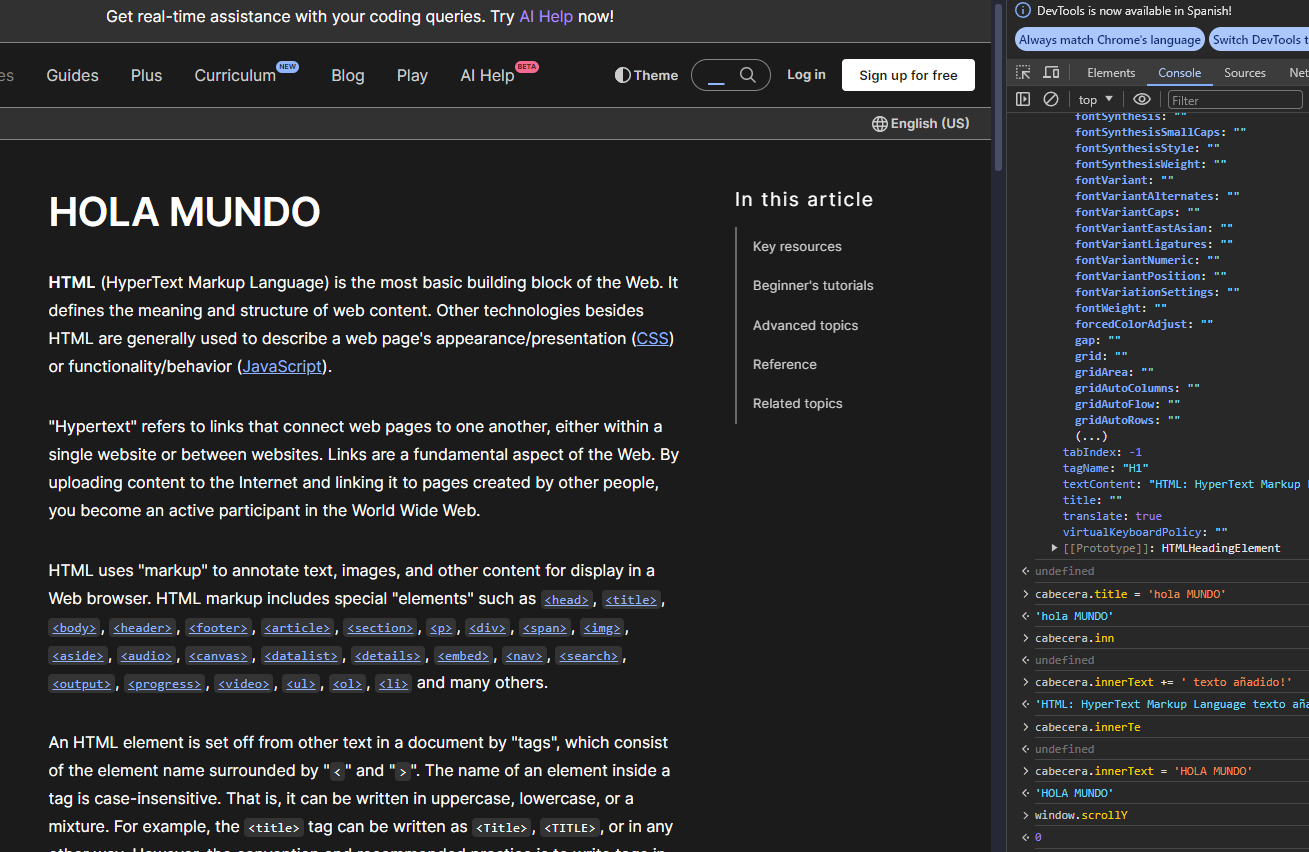
El += añade texto a continuación del elemento

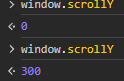


Si por el contrario solo escribimos = , cambiara todo el contenido por el nuevo que le asignemos…





Tambien podemos mirar con scrollY la posición del scroll en la que nos encontramos en la página. En este caso en el inicio el scrollY es = 0 y en cuanto bajamos un poco…

**…cambia a 300**

**Tambien podemos medir la distancia entre un elemento y el principio de la pagina…**

** con offsetTop**

**Con classList.add le podemos añadir una clase y con classList.remove se la podemos quitar.**

****

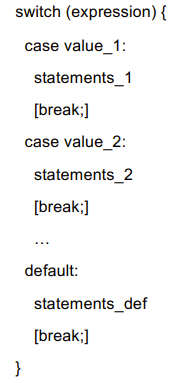
****





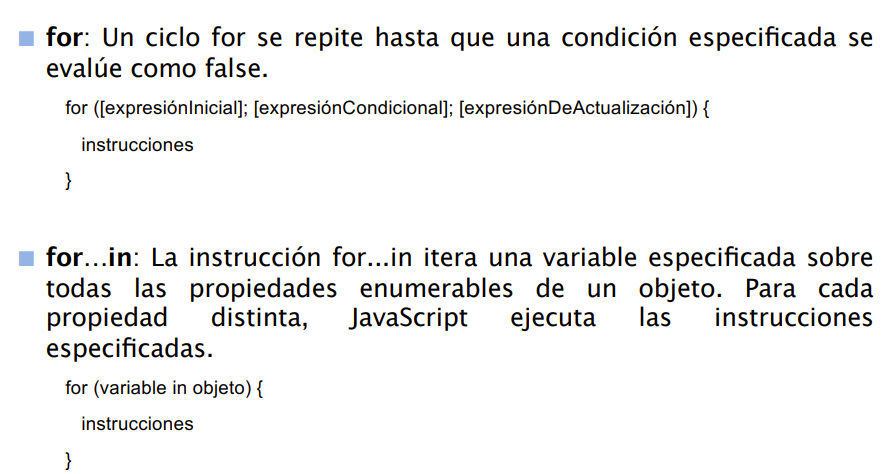
En ambos casos hay que especificar el nombre de la clase para añadir obviamente pero también para removerla.

ESTRUCTURAS DE CONTROL DE FLUJO (diapo 54+)

**SWITCH:**

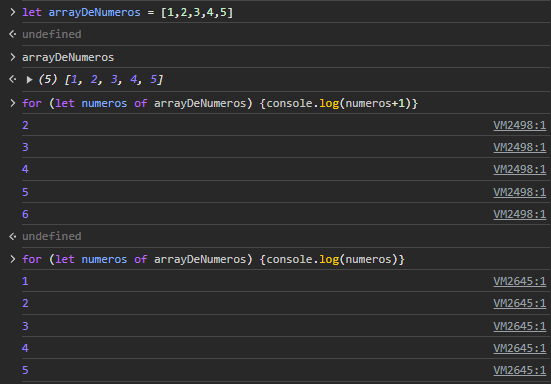
**BUCLES**

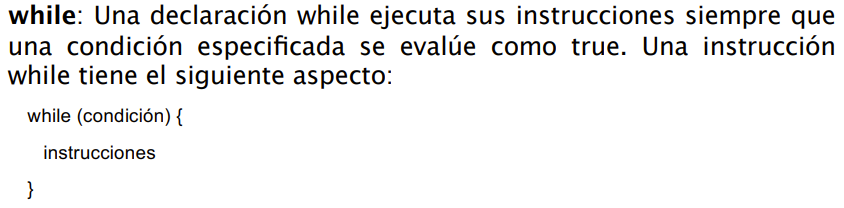
**FOR:**

****

con **FOR OF**, podemos recorrer de esta manera cada item del array siendo la variable declarada dentro del paréntesis (**numeros** en este caso) cada uno de los elementos del array.

Para hacer un recorrido con una condición se deberá utilizar el FOR normal o preferiblemente WHILE, breaks… (diapositivas 60+)



**WHILE**