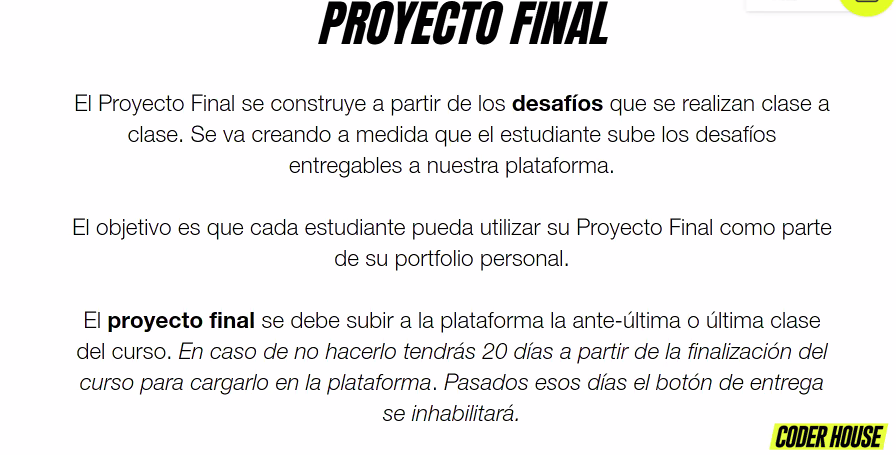
***JAVASCRIPT CLASE 1***

***Sintaxis y variables***



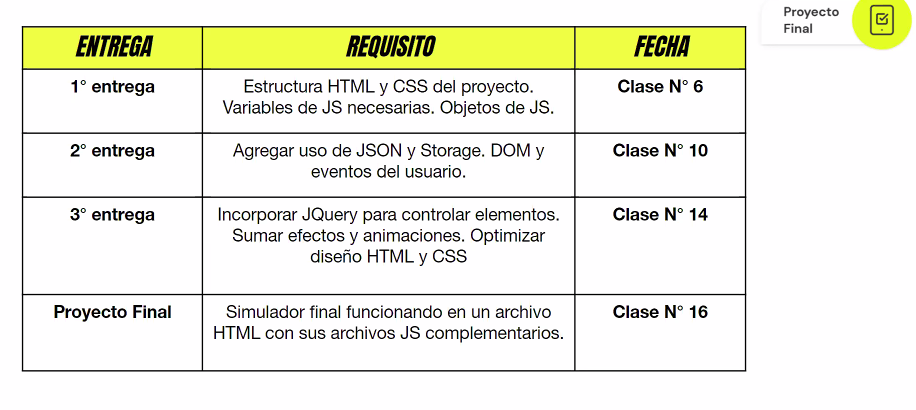


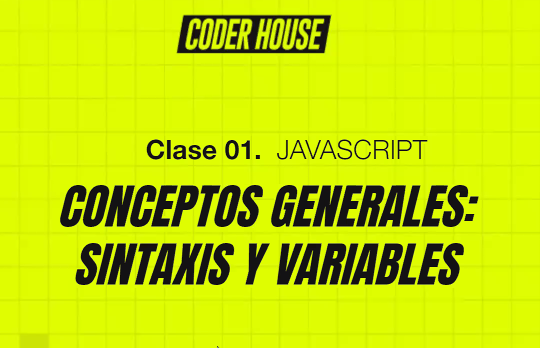


Carrito de compras por ej

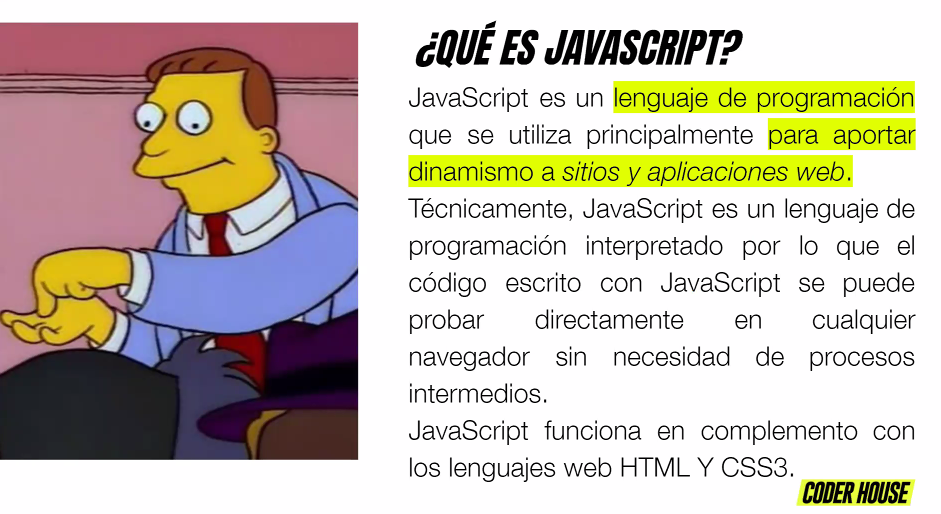
20 días cuando termina la cursada para entregar proyecto final





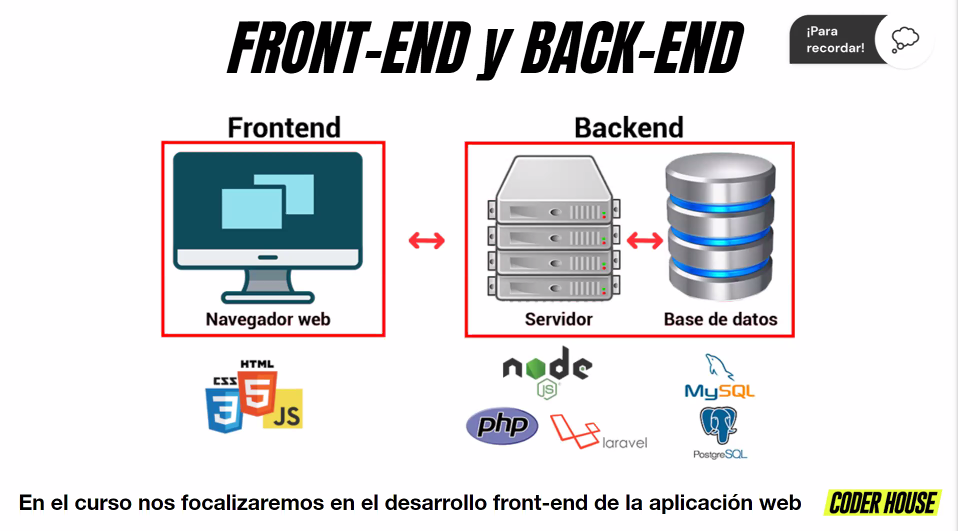


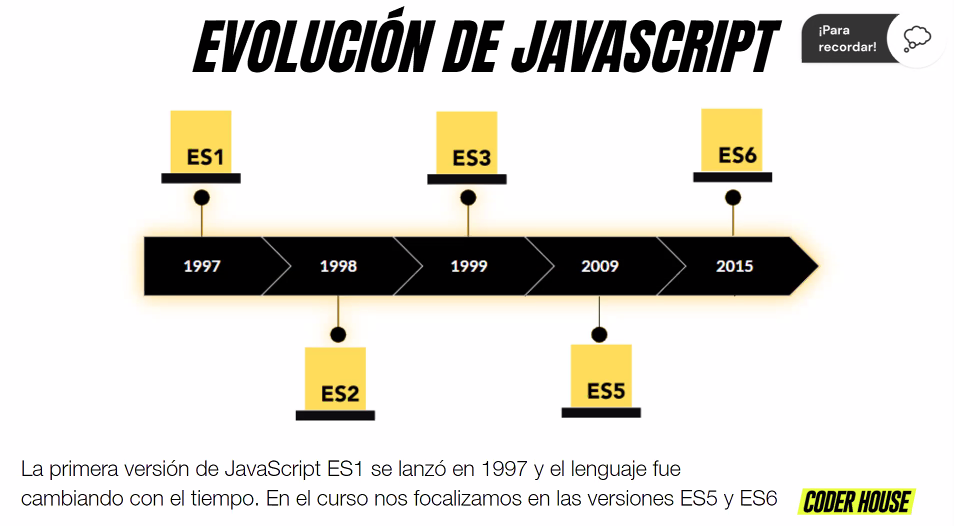
Ver de hacer página nueva y meter JS ahí para ir entendiéndolo y después pasarlo a Lumiere.





Plataformas dinámicas e interactivas





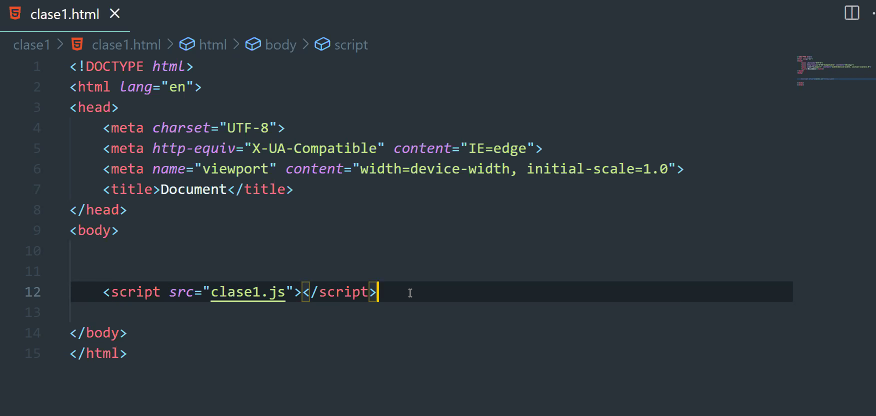
Javascript no es lo mismo que Java, no tienen nada que ver

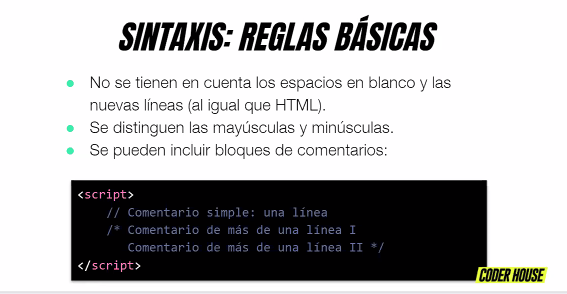
Los archivos de js son scripts. En visual estudio llevan .js

Para ver scripts🡪 linkearlos a un html. No se pueden ejecutar en carpeta

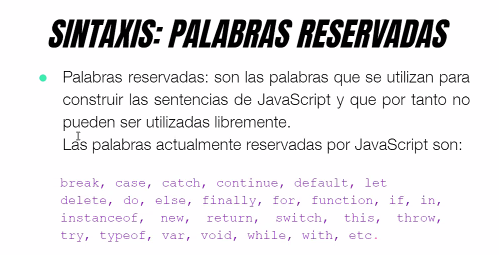
<script> Dentro de HTML no es recomendable</script>

Pero llamas a la etiqueta script ahí





Usar shift alt a



Reservadas para JS

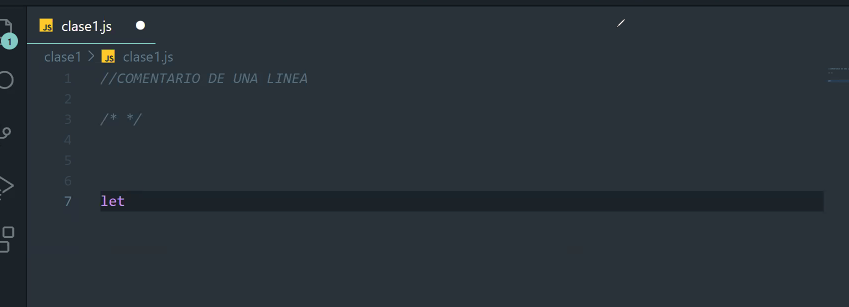


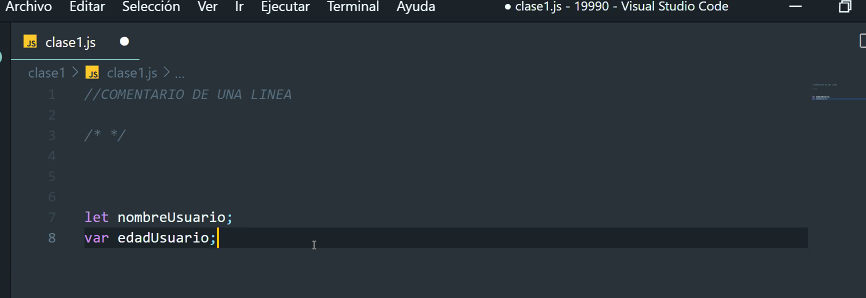


Llamo a una variable para referenciar el valor almacenado dentro

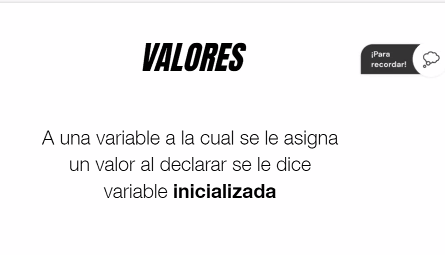
Let 🡪 declaro variable

Var🡪 forma más antigua de declarar variable

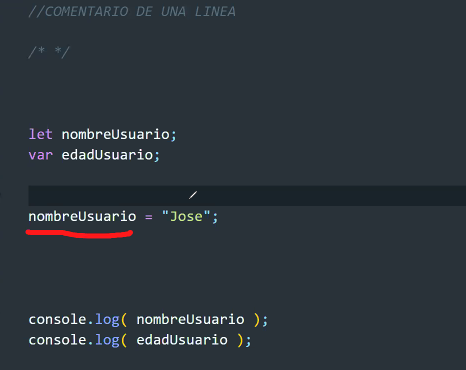




Usar let

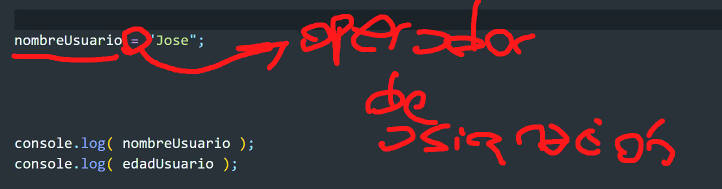


Cuando uso let le digo a js que reserve ese espacio en memoria.

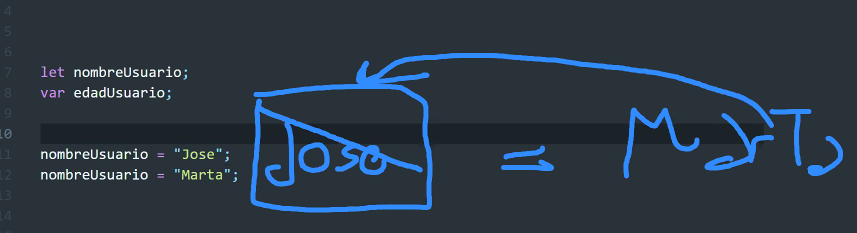


El valor Jose se almacena dentro de variable nombreUsuario

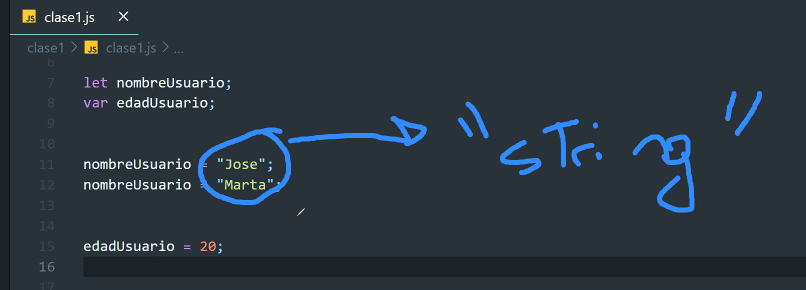
= 🡪 operador de asignación



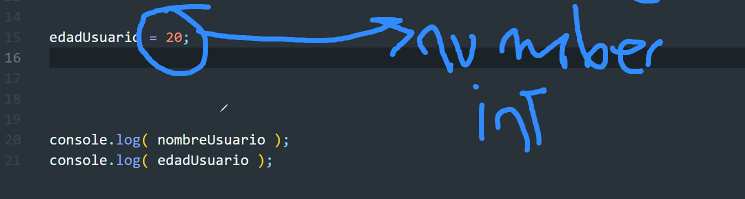
**Lo de la derecha se asigna a la izquierda. Este dato** lo guardo en esta variable



Se sobreescriben



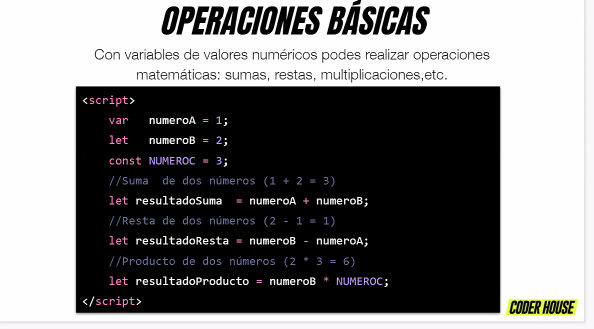
Cadena de caracteres

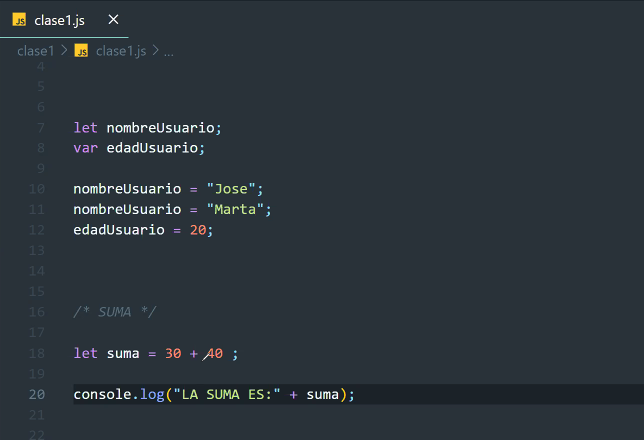


Número (entero o int)

‘’20’’ es un string, 20 es un number. No tengo que ponerle comillas al número

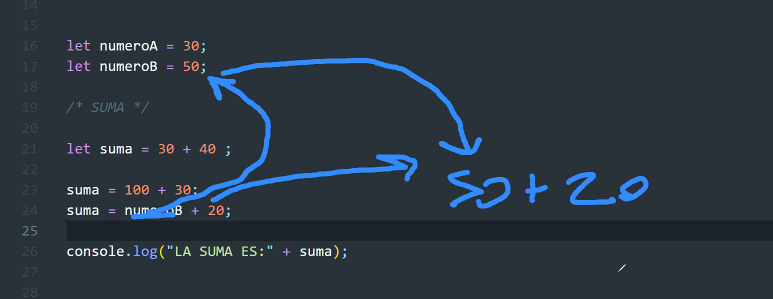
Number se puede hacer operaciones aritméticas (sumas, restas, divisiones)

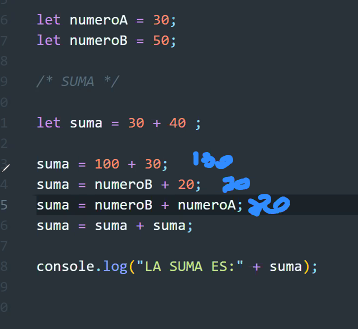




Se asigna a suma el resultado de la suma

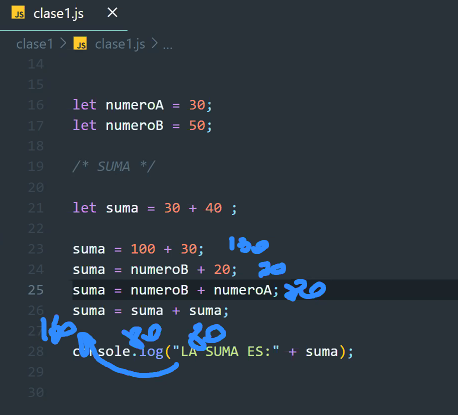


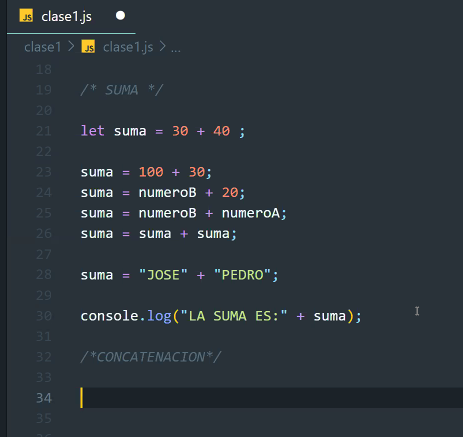




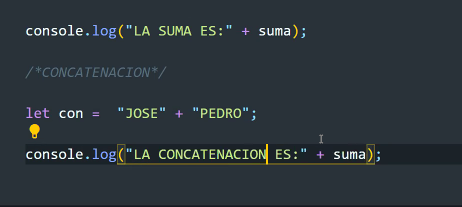
Se pisan los valores. Usa el último, o sea , 80.

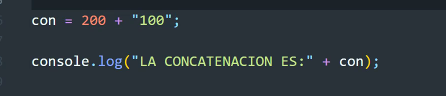
Por lo que suma + suma es 80 + 80 = 160



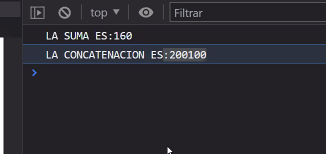


Concatenación de strings. Saldrá JosePedro





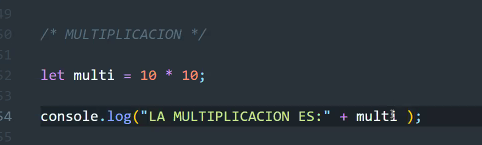
String y número se concatenan: 200100

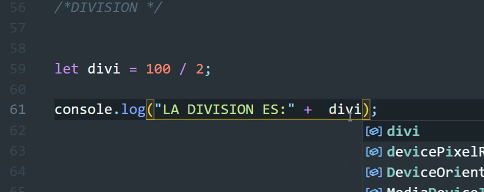


el + en el console concatena el mensaje "la concatenacion es…." con ese resultado del let

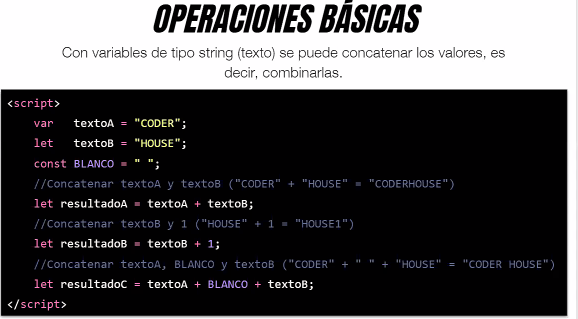


80





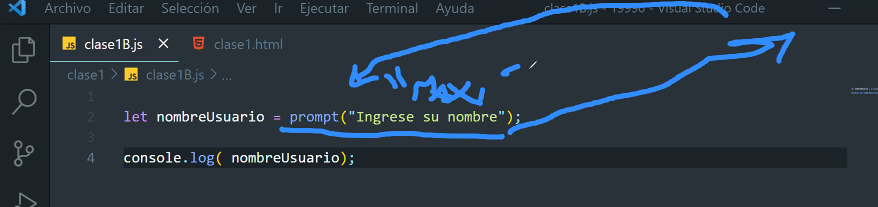
Lo que hace es que el let ‘divi’ , valga 50





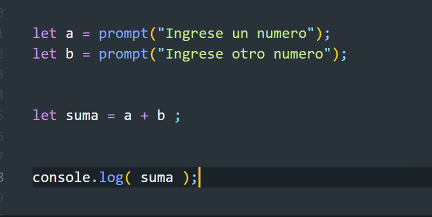
Console.log = da un dato en la consola

Prompt : es una entrada de datos. Forma en que le pido al usuario que ingrese información

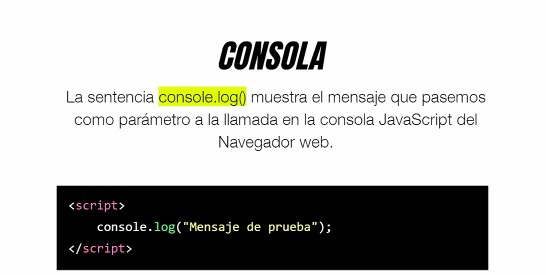


Se ejecuta **prompt**: se pide el dato. Se ejecuta y se envía dentro de la variable. Lo que retorna prompt es un **string**.

Prompt siempre es un STRING.

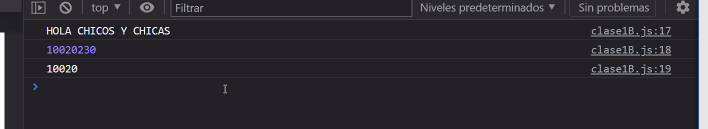


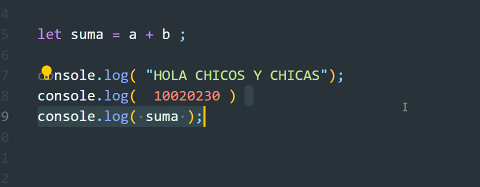
Esto da 10050 (a=100 b=50), ya que se concatenan porque prompt siempre es un STRING.



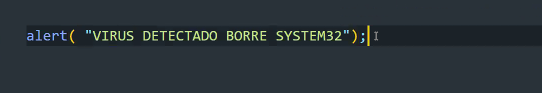
Log= mensaje dentro de la consola

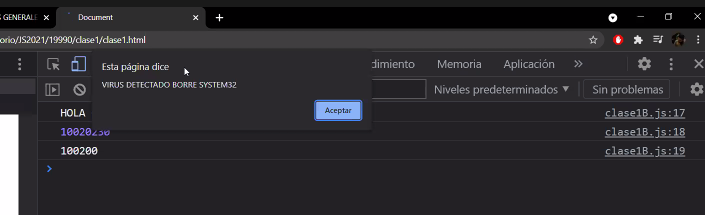
Cuando hago console.log (‘variable’) muestra valor de la variable





Alert(‘’) es otra forma de salida

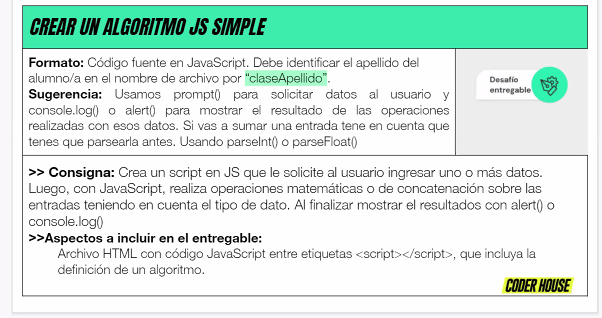


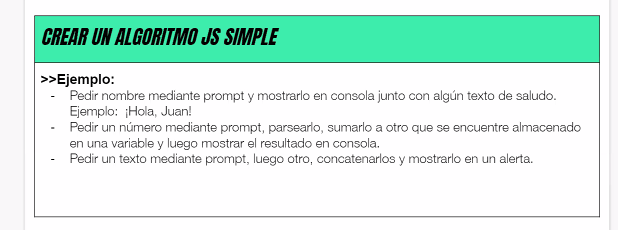


Alerta al usuario

Se ejecuta primero script, después HTML







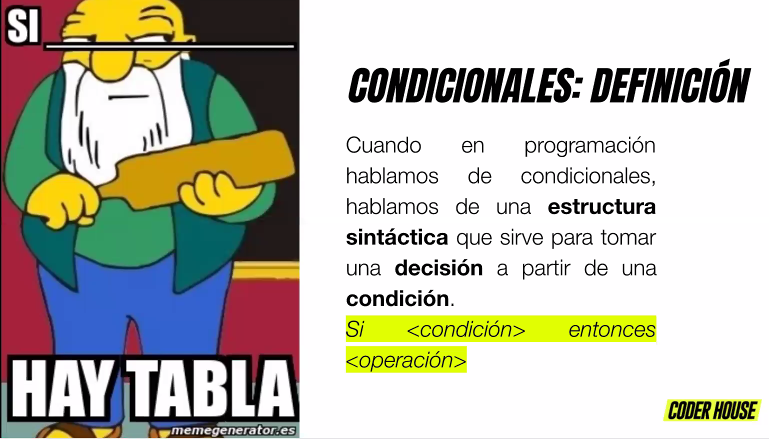
<https://drive.google.com/drive/u/0/folders/1sJaJKu9Tj31w2NtEeOjRNx-cgdHeTlu4>

Link al drive

Ver ejemplos de desafíos de preentregas

Booklet 🡪 ejercicios de práctica





Si la condición es verdadera, se ejecuta esa estructura.

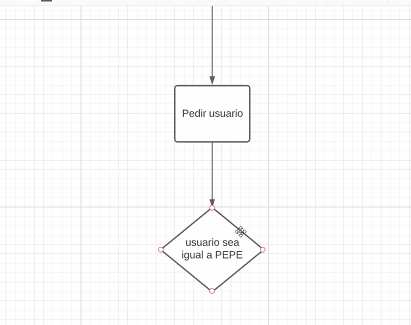
Si hoy a la tarde sale el sol, me voy a tomar mates al parque. Si no hay sol, no.

Si es mayor de 16, vota, si no, no.

Diseño de sistema 🡪 Bajar requerimientos, pensar como se realiza cada proceso

**Primer proceso** log in 🡪 Pedir usuario (prompt) diagrama de flujo.

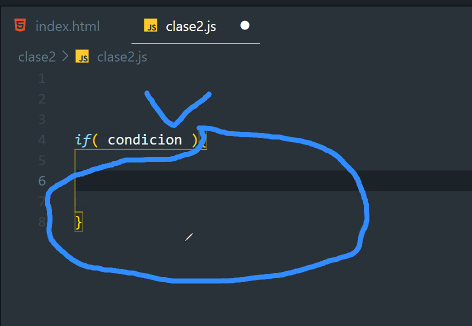
Condición-> usuario sea igual a PEPE



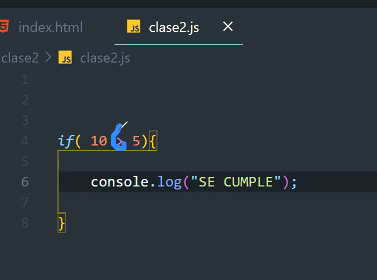
If🡪 (condición) 🡪 {

se ejecuta

}

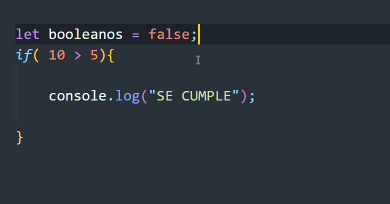


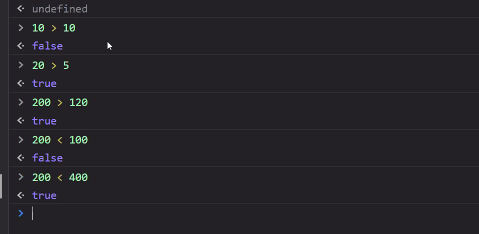
If siempre da verdadero o falso, no otra cosa.



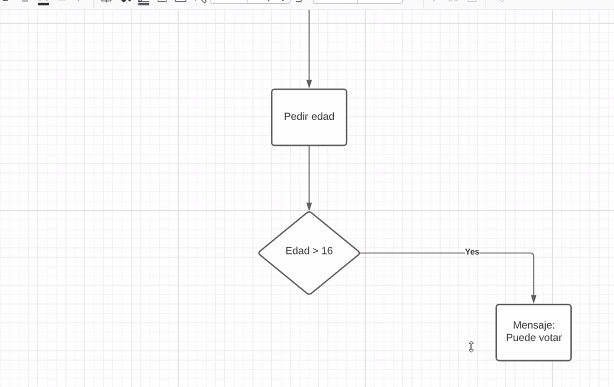
Operadores relacionales ( > < mayor o menor) comprobación entre dos elementos y retornan algo. Retornan valor booleano (0 y 1)

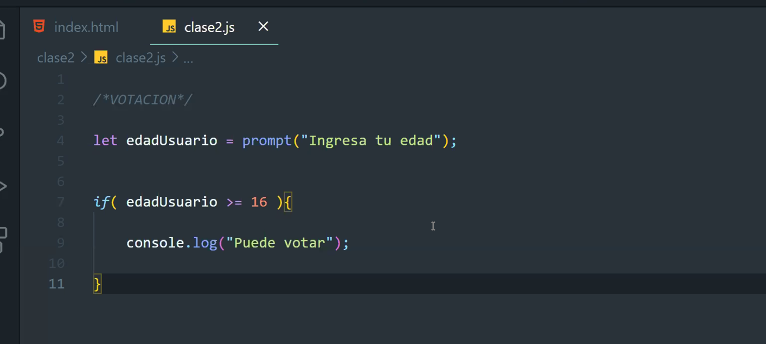
Booleanos = true o false





Mayor o igual >=

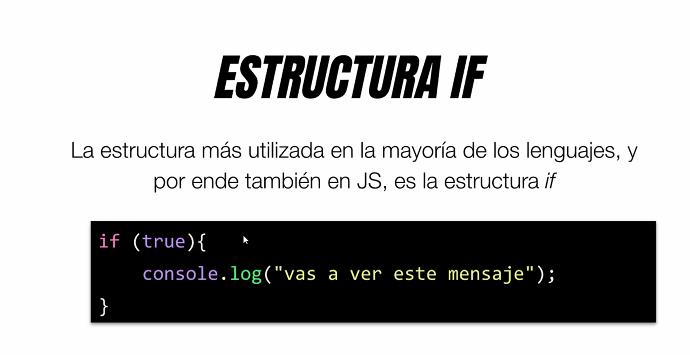




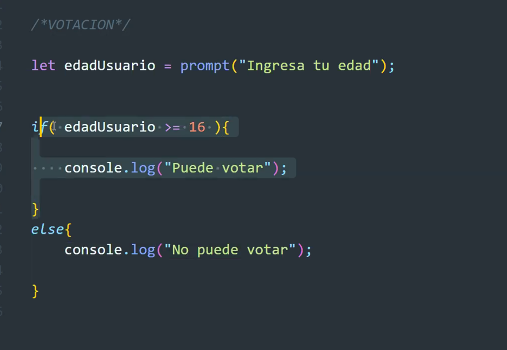
== Signo relacional para decir igual



!= (desigual)

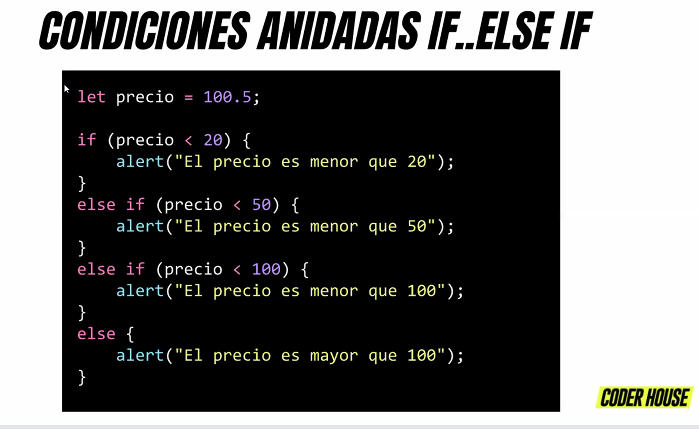


Else, si no se cumple la condición anterior

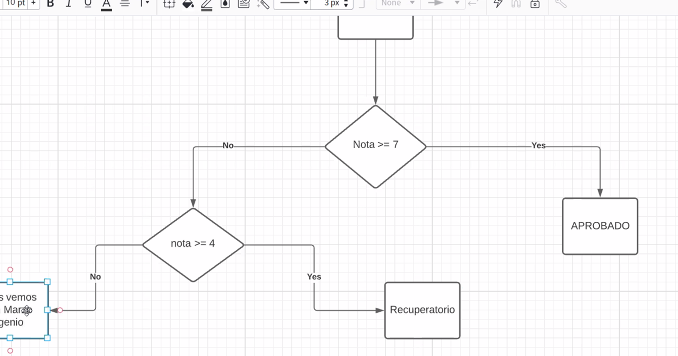


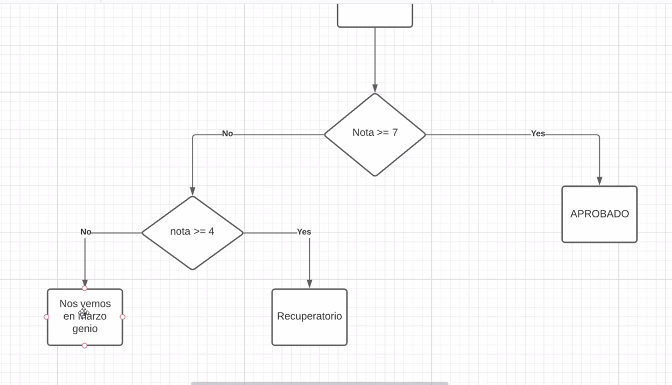
Si la condición es negativa, se ejecuta el else

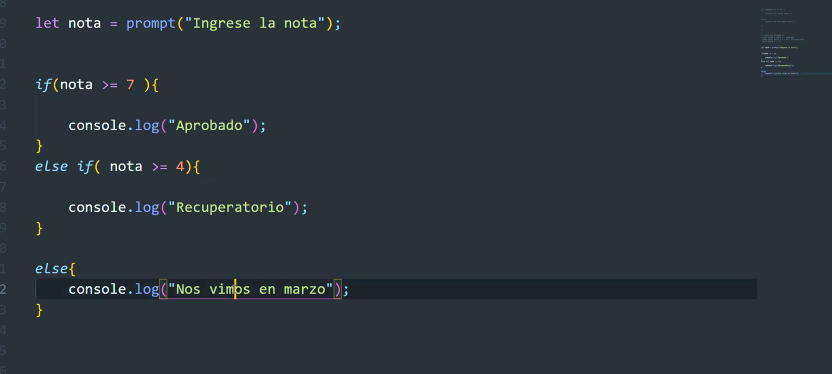




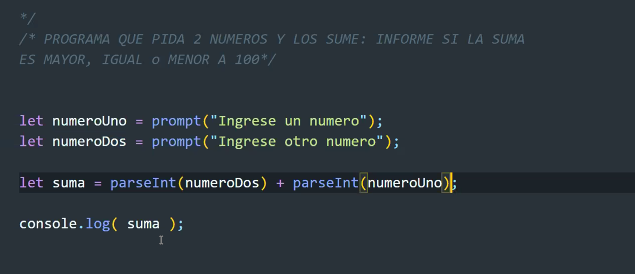
Else if ‘’en cambio si’’



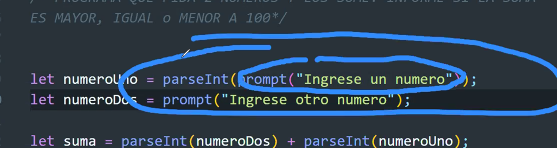


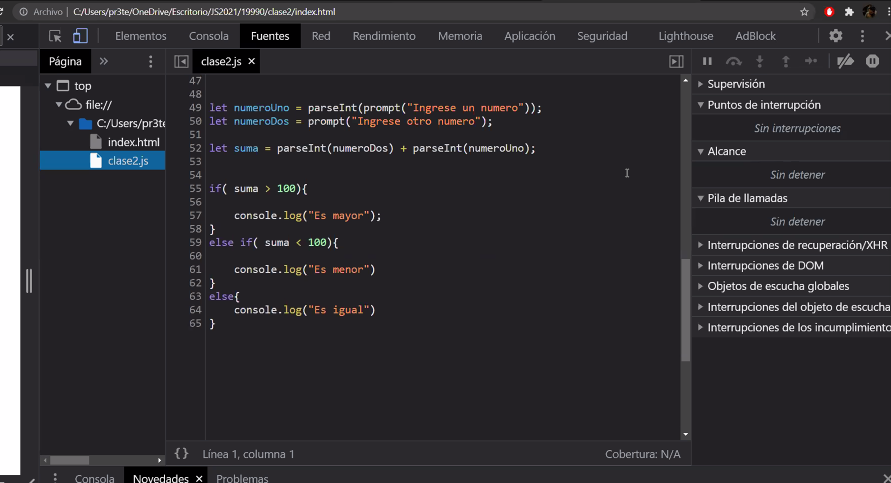


Pasar **string a entero**, para estas sumas. parseInt (convertir a entero)

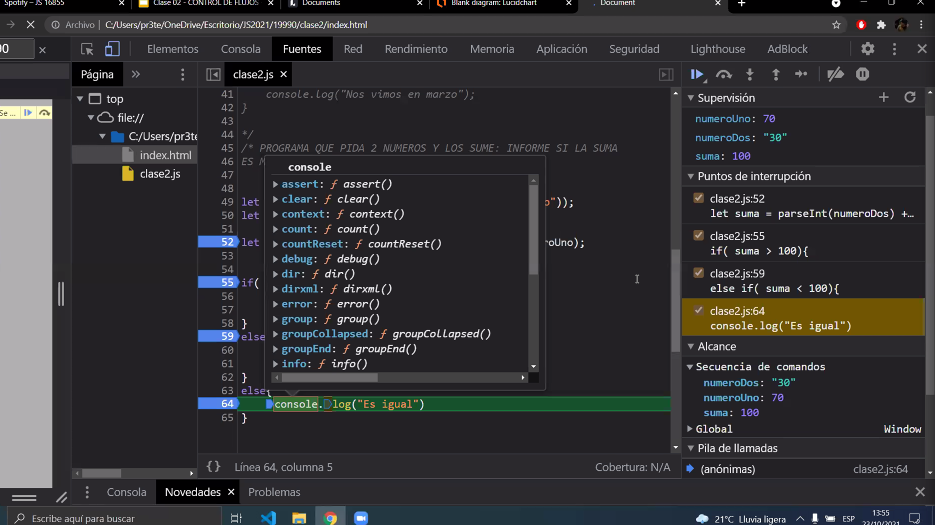


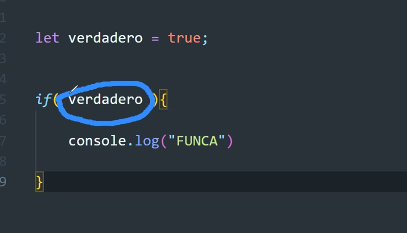
‘Jose’ no se puede convertir a int porque no tiene nro





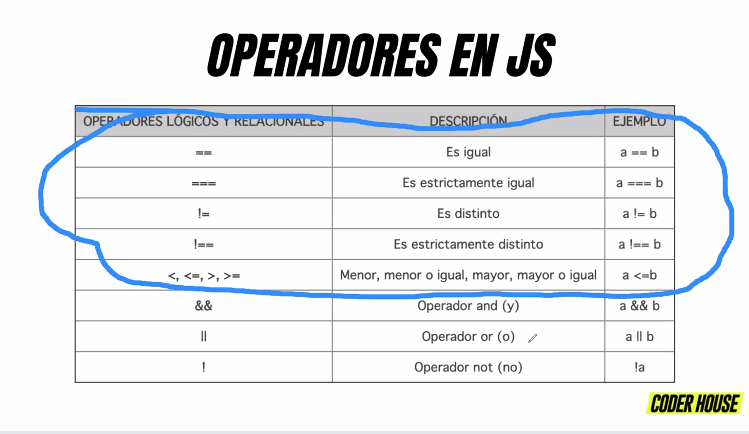
Ver código desde el navegador (Source)





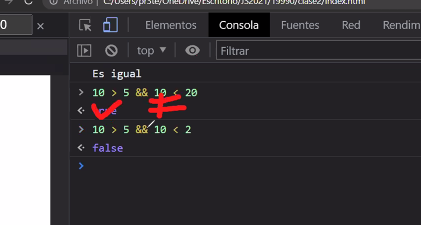
Se evalua por true o por false.

Si verdadero es true, se ejecuta.



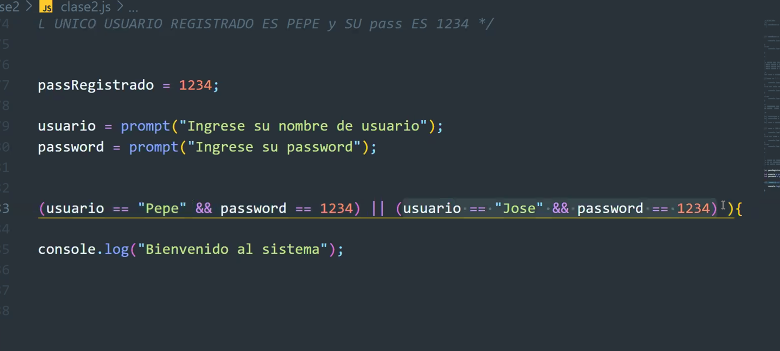
Azules son los operadores relacionales, los otros se llaman lógicos

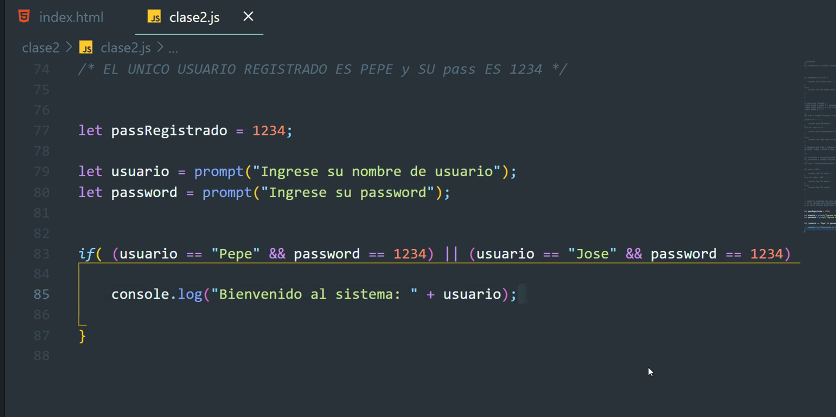
Operadores lógicos && (and) es verdadero cuando las 2 o más condiciones se cumplan

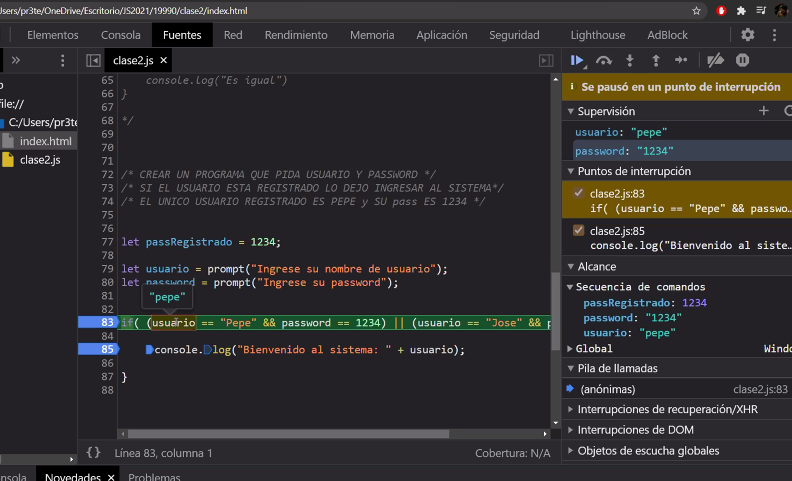


Para que retorne V se tienen que cumplir todas las condiciones

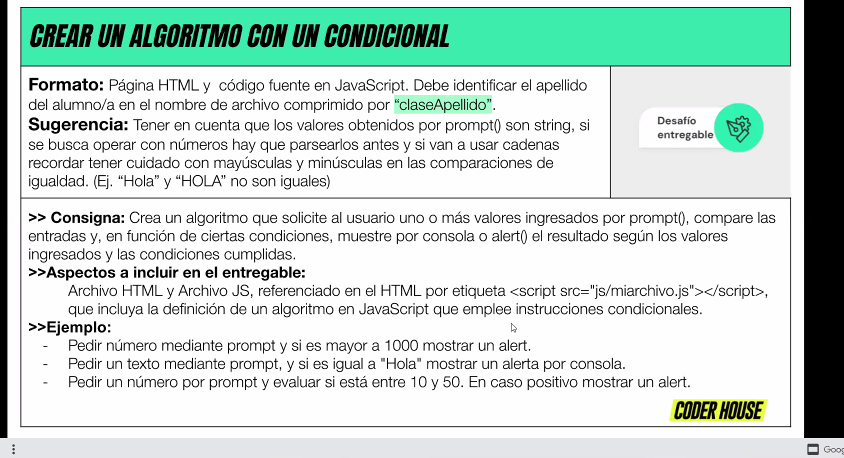
|| ‘’o’’ a la izquierda de 1 se escribe







Pepe =/ pepe



A través de github ideal