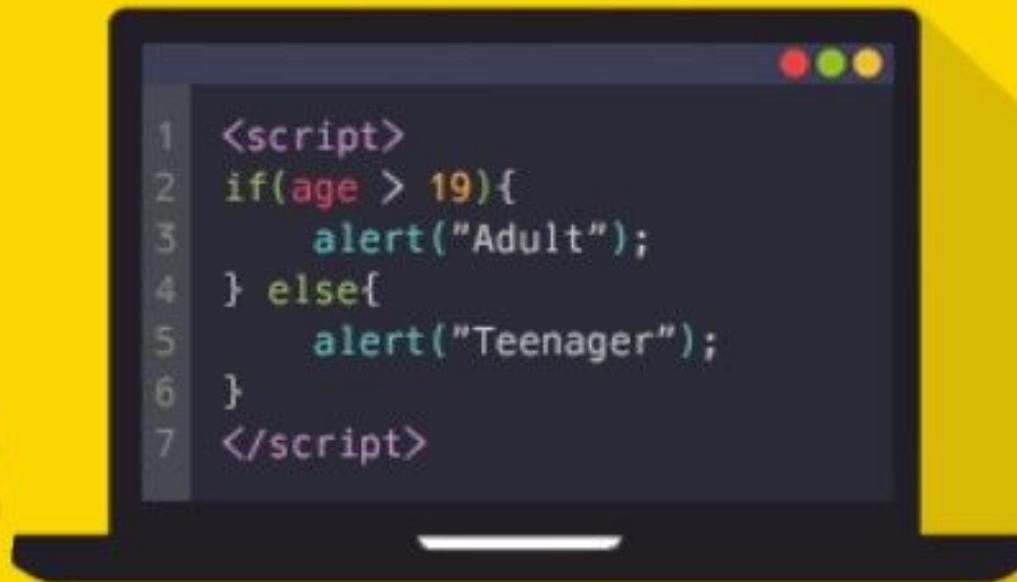
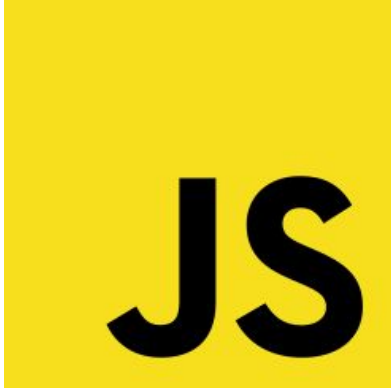




JavaScript

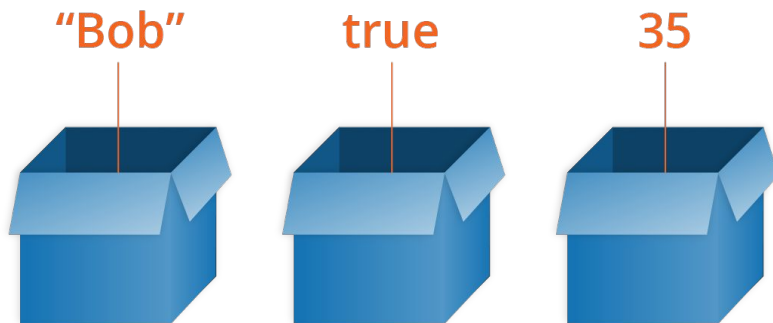




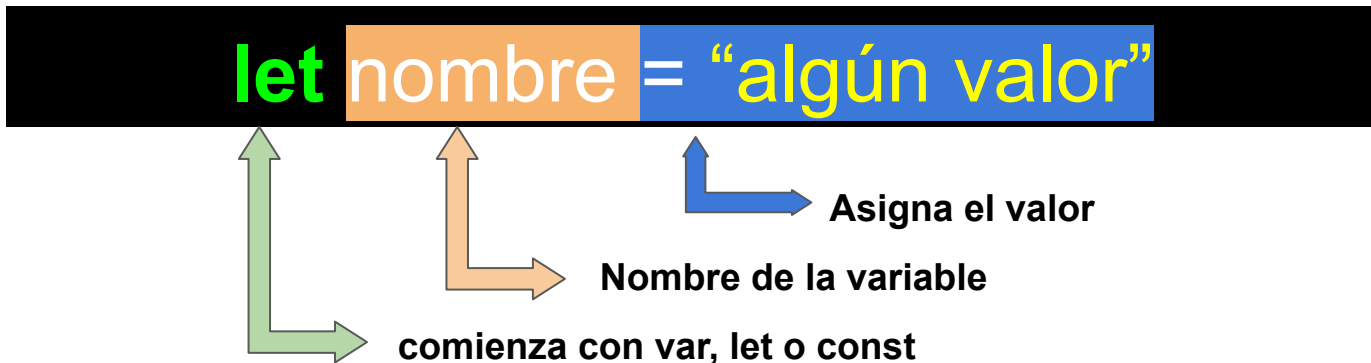
Variables

¿Qué es una variable?

- Es un contenedor de un valor
- Podemos guardar un valor temporalmente.
- Puede almacenar distintos tipos de datos.



Inicializar una variable



- Podemos usar las variables para hacer cálculos, mostrar su valor por pantalla, etc.

```
let nombre = "Robert John Downey Jr."  
let edad = 54
```



Tipos de datos

Tipos de datos

Cadenas de texto, números, booleanos (verdadero o falso), valor no definido y nulos, son tipos de datos que más se usan en JavaScript

 **STRING**
`var name = "John Doe";`

 **NUMBER**
`var x = 123;`

 **BOOLEAN**
`var pass = true;`

 **ARRAY**
`var animals = ["cat", "dog"];`

 **OBJECT**
`var person = {name:"John", age:"99"};`

 **EMPTY**
`var foo = null;`

Variables

Hay tres tipos de declaraciones de variables en JavaScript.

- **var:** Declara una variable, inicializando opcionalmente a un valor. Es la forma tradicional que se venía usando para declarar una variable.
- **let:** Declara una variable local en un bloque de ámbito, inicializando opcionalmente a un valor. Es la nueva forma de declarar variables.
- **const:** Declara una constante de sólo lectura en un bloque de ámbito.



var



const



let


Variables

var

var apple = 



a thing in a box
named "apple"

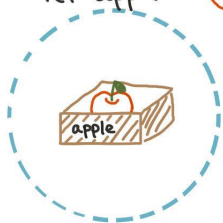
apple = 




you can swap
item later

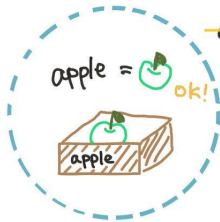
let

let apple = 



a thing in a box
named "apple" w/
protection shield

apple =  OK!



you can swap item
only if you ask
inside of the shield

~~apple =  NG~~

const

const apple = 



a thing in
LOCKED cage
named "apple"



~~apple =  NG~~

you can't
swap item
later.



apple.multiply(3) OK!

... but you can ask
the item to change itself
(if the item has method
to do that)

Cadenas de texto (Strings)

Una cadena de texto es un conjunto de caracteres encerrados entre comillas simples (') o dobles ("). Por ejemplo:

"Texto entre comillas dobles"

'Texto entre comillas simples'

Tipos de operadores

¿Qué es un operador?

Los operadores son símbolos especiales que sirven para ejecutar una determinada operación.

Una expresión es, normalmente, una ecuación matemática, tal como $3 + 5$. En esta expresión, el símbolo más (+) es el operador de suma, y los números 3 y 5 se llaman operandos. En síntesis, una expresión es una secuencia de operaciones y operandos que especifica un cálculo.

Tipos de operadores

Existen diferentes tipos de operadores:

1. Operador de asignación
2. Operadores aritméticos
3. Operadores unitarios
4. Operadores Condicionales
5. Operadores relacionales
6. Operadores lógicos

Operador de asignación

Es el operador más simple que existe, se utiliza para asignar un valor a una variable o a una constante. El signo que representa la asignación es el **=** y este operador indica que el valor a la derecha del **=** será asignado a lo que está a la izquierda del mismo.

```
let edad = 20
```

```
let precio = 25.45
```

Operadores aritméticos

Son operadores binarios (requieren siempre dos operandos) que realizan las operaciones aritméticas habituales: suma (+), resta (-), multiplicación (*), división (/) y resto de la división entera (%)

Operador	Significado
+	Suma
-	Resta
*	Producto
/	División
%	Resto de la división entera

Expresión	Operador	Operandos	Resultado arrojado
5 * 7	*	5 , 7	35
6 + 3	+	6 , 3	9
20 - 4	-	20 , 4	16
50 % 8	%	50 , 8	2
45/5	/	45,5	9

Los operadores unitarios

Los operadores unitarios requieren sólo un operando; que llevan a cabo diversas operaciones, tales como incrementar/decrementar un valor de a uno, negar una expresión, o invertir el valor de un booleano.

Operador	Descripción	Ejemplo	Resultado
++	operador de incremento; incrementa un valor de a 1	<pre>int suma=20; suma++;</pre>	suma=21
--	operador de decremento; Reduce un valor de a 1	<pre>int resta=20; resta--;</pre>	resta=19
!	operador de complemento lógico; invierte el valor de un valor booleano	<pre>boolean a=true; boolean b= !a;</pre>	b=false

Operadores Condicionales

Son aquellos operadores que sirven para comparar valores. Siempre devuelven valores booleanos: TRUE O FALSE. Pueden ser Relacionales o Lógicos.

Operadores relacionales

Operador	Significado
==	igual a
!=	diferente de
===	estrictamente igual
!==	estrictamente diferente
>	mayor que
<	menor que
>=	mayor o igual que
<=	menor o igual que

Los operadores relacionales sirven para realizar comparaciones de igualdad, desigualdad y relación de menor o mayor.

Determinan si un operando es mayor que, menor que, igual a, o no igual a otro operando.

La mayoría de estos operadores probablemente le resultará familiar. Tenga en cuenta que debe utilizar " == ", no " = ", al probar si dos valores primitivos son iguales.

Operadores relacionales

Expresión	Operador	Resultado
a > b	>	true: si a es mayor que b false: si a es menor que b
a >= b	>=	true: si a es mayor o igual que b false: si a es menor que b
a < b	<	true: si a es menor que b false: si a es mayor que b
a <= b	<=	true: si a es menor o igual que b false: si a es mayor que b.
a == b	==	true: si a y b son iguales. false: si a y b son diferentes.
a != b	!=	true: si a y b son diferentes false: si a y b son iguales.

Operadores lógicos

Los operadores lógicos (AND, OR y NOT), sirven para evaluar condiciones complejas. Se utilizan para construir expresiones lógicas, combinando valores lógicos (true y/o false) o los resultados de los operadores relacionales.

Expresión	Nombre Operador	Operador	Resultado
a && b	AND	&&	true: si a y b son verdaderos. false: si a es falso, o si b es falso, o si a y b son falsos
a b	OR		true: si a es verdadero, o si b es verdadero, o si a y b son verdaderos. false: si a y b son falsos.