JS has two versions

1. ES5(only var)(old e)
2. ES6(let & const included)

Undefined - Exist but no value Not- Defined - Does not Exist

If there exist Brackets {} or [] or () then variale is refrence if not Bracket exist then variable is primitive

function = name of the set of code,executed when it is called function ke chlte hi sbse pehle execution context bnta hai

Execution Context :-

1. variable
2. Functions inside the parent function
3. Lexical Environment - It tells us that, what we can access or what we cannot access

arguments = real values jo hum dete hai parameters = variables jinme values store hoti hai

[] = array = to store more then one value to a variable

{} = object = Object properties are written as name:value pairs, separated by commas

method = property of object whose value is a function

let= it doesnot itself in window

to copy an array(( var a = [1,2,3,4,5] var b = [...a] //(three dots are spread operator) ))

to copy an obj(( var a = {Harsh} var b = {...a} ))

Falsy (( 0 false undefined null NAN document.all )) Truthy (( all except falsy ))

loops(( ( foreach( works for looping on array works on each and every value "Har Ek" ) forin( works for looping on objects key = key ki value ) ) ))

callback function = esa function jo tb chlta hai jb kaam complete ho jaaye

first class functions = iske according hum functions ko as a value use kr skte hai

heap memory is used to store the data

**Variables** = Containers to store data

Cannot use the reserved keywords as the name of the variables

**Scope in JS**

Block Scope

Global scope

Types of variables:-

to add JS in html file,

use <script> </script> tag inside <head> or <body> tag

**JS is case sensitive**

"<br>" **line breaks in JS**

**Declares**  
var name;

**initialize**  
name = "John Doe";

getElementById("demo")

to access the value of any element by its id

getElementById('demo').innerHTML  
changes the element content (innerHTML)

" document.getElementById('myImage').src = 'pic\_bulbon.gif' "

Can change the image

getElementById("demo").style.fontSize = "35px";

can change the css

Writing into the HTML output using document.write().

Using document.write() after an HTML document is loaded, will delete all existing HTML  
The document.write() method should only be used for testing.

alert(5 + 6);

Writing into an alert box,

console.log(5 + 6);

print/ display the output

print()

open print preview to print the page via printer

JavaScript syntax defines two types of values:

* Fixed values
* Variable values

Fixed values are called Literals.

Variable values are called Variables.

Comments

Single line comments //

Multiline comments /\* and \*/

JavaScript Identifiers

Identifiers are used to name variables and keywords, and functions.

JavaScript name must begin with:

* A letter (A-Z or a-z)
* A dollar sign ($)
* Or an underscore (\_)

One Statement, Many Variables

let person = "John Doe";

let carName = "Volvo";

let price = 200;

OR

let person = "John Doe", carName = "Volvo", price = 200;

**Re-Declaring JavaScript Variables**

If you re-declare a JavaScript variable declared with var, it will not lose its value.

var carName = "Volvo";

var carName = "Tata";

You cannot re-declare a variable declared with let or const.

If you put a number in quotes, the rest of the numbers will be treated as strings, and concatenated.

let x = 5 + 2 + 3;

# this gives 10

let x = "5" + 2 + 3;

# this gives 523

**Let**

* Variables defined with let cannot be Redeclared
* Variables defined with let must be Declared before use
* Variables defined with let have Block Scope
* Can be initialized later

**Const**

* It is a read only keyword
* We have to initialize(give its value) at the time of declare.
* it can be used anywhere (function scoped)

**Var**

* It ignores all the block scopes but does not work outside the Function
* Var is hoisted to the top and can be initialized at any time.

Variable Shadowing

Not Defined = Does not exist

Un Defined = Defined but Does not have any value

**Skipped**

# Unsigned right shift (>>>)

## Bitwise AND

Operators in JS

Arithmetic Operators

+ Addition

- Subtraction

\* Multiplication

\*\* Exponentiation / power x\*\*2 is same as Math.pow(x,2)

/ Division

% Modulus (Division Remainder)

++ Increment by 1

-- Decrement by 1

Assignment Operators

= x = y x = y

+= x += y x = x + y

-= x -= y x = x - y

\*= x \*= y x = x \* y

/= x /= y x = x / y

%= x %= y x = x % y

\*\*= x \*\*= y x = x \*\* y

Shift Assignment Operators( works on 32bit form of numbers) ()

<<= Left Shift Assignment Operator remove the left bit and add a 0 to the right side

>>= Right Shift Assignment Operator remove the right bit and add 0 to the left side

Short Trick <https://youtube.com/shorts/7kzXJ434L84?si=usz8J1Kf_IYI1hCS>

>>>=

Data Types

let x = 16 + 4 + "Volvo";

// Output = 20Volvo

let x = "Volvo" + 16 + 4;

Volvo164

Empty value ("") = String

**Functions**

Executed only when it is invoked / called

The code to be executed, by the function, is placed inside curly brackets: {}

Hoisting works

You can write code that can be used many times