**JAVASCRIPT**

Javascript was created in 1995 by Brendan Eich. Initially it was called as Livescript and later it was renamed as javascript.

It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document

It is used to create interactive content on sites and applications and enhances.

1. **Machine level language** –> **MLL**🡪 binary transactions like 0’s & 1’s
2. **Assembly level language** 🡪 **ALL**—>some abbreviation like sub, add was used here(An assembly language is a type of low-level programming language that is intended to communicate directly with a computer's hardware)

Unlike machine language, which consists of binary and hexadecimal characters, assembly languages are designed to be readable by humans.

1. **High level language**🡪**HLL**🡪 Javascript, c#, python etc.

Installation:-

1. **Node.js** – provides runtime environment
2. **Visual studio** – IDE

**Translators**🡪 we use javascript which is high level language, it should be converted to machine level language so we use translators

1. **Assemblers**🡪 convert **ALL** to the **MLL**
2. **Compilers**🡪 **HLL** to MLL(syntax parsing.. if there is no errors only there will be .exe file)
3. **Interpreters**🡪 **HLL** to **MLL**(line by line syntax parsing)

**Note:-**

**V8 js engine for chrome**

**Spider Monkey Js engine for firefox**

**Chakra js engine for edge**

**Tokens in Javascript :- Tokens are the smallest unit of a program**

1. **Keywords**
2. **Identifiers**
3. **Operators**
4. **Literals**
5. **Punctuators**

Int a = 10;

Int 🡪 datatype

a 🡪 variable

= 🡪 operator

10 🡪 literal

**Variables in Javascript**

1. **Var**
2. **Let**
3. **Const**

**var Variable**

* It can hold different types of values, including numbers, strings, arrays, objects, functions, and more.
* Re-declaration and re-initialisation is possible
* var a = 10
* var a = 20
* var a = 'javascript'
* var a = 1+6

Here output is 7. Here it is not showing any error because re-declaration and re-initialisation is possible

**let Variable**

* It can hold different types of values, including numbers, strings, arrays, objects, functions, and more.
* Re-declaration is not possible whereas re-initialisation is possible
* let a = 98
* a = 47
* a= 'javascript'

Here output is javascript. Here if we re-declare, will get an error. Only re-initialisation is possible.

let a = 98

let a =47

a= 'javascript'

Here we will get an error because we are trying to re-declare.

**Const variable**

* It can hold different types of values, including numbers, strings, arrays, objects, functions, and more.
* Re-declaration and re-initialisation is possible. Once it is declared with const it is final and it cannot be changed.
* const a = 250
* const a= 'javascript'

Here error will come because we are trying to re-declaration and re-initialisation.

* const a = 250
* a= 'javascript'

Here error will come because we are trying to re-initialising.

const a = 250

Here output will be 250. No error will be there because no re-declaration and no re-initialisation.

**NOTE:-**

JavaScript is an example of a loosely typed language, where variables can hold different types of values at different points in time. This means that a variable can be assigned a string value at one point, and then later assigned a numeric value without any explicit type conversion.

let x = "hello";

console.log(x); // outputs "string"

x = 42;

console.log(x); // outputs "number"

In the above example, the variable **x** is initially assigned a string value, but later assigned a numeric value. Because JavaScript is loosely typed, the type of operator can be used to determine the current type of **x**.

**typeof**

typeof is a keyword which is used to determine the type of a value or a expression.

const a = 250

console.log(typeof(a));

console.log(typeof(String(a)))

Output is :-

PS C:\Users\User\Desktop\JavaScript> node Variables\variables.js

number

string(explicitly we can covert the type here)

**Libraries**

It is used to simplify a complex task i.e. Javascript can alone perform a task in bulk code, but the same task can be performed with minimal/optimal code by using libraries.

* JS libraries has coded or functions that developers can reuse or repurpose.
* These methods, functions are used to perform same tasks on a webpage.
* To have optimised code we will have libraries

Ex:- jQuery, load#, Bootstrap, js etc..

**Framework built using JS**

* **JS + Enhancements-Node JS** (used for developing web applications)
* **JS + Enhancements-React JS** (used for developing web applications)
* **JS + Enhancements-React Native**(used for developing client server applications(mobile applications))
* **JS + Enhancements-Angular JS** (used for developing standalone applications)
* **JS + Enhancements-Electron** JS (used for developing web applications)
* **JS + Enhancements-Tensor flow**(Artificial Intelligence, Machine Learning Applications).