Java Script syllabus (09-Jan-2025)

1.Introduction of Javascript

2.Features of javascript

3.Difference between java and javascript

4.Tokens

5.Variables

6.Data Types

7.Condiational Statements

8.Looping Statements

9.Functions

10.String

11.Array

12.Objects

13.Ways of creating object

14.Introduction to classes

15.this keyword

16. BOM (Browser object model)

17. DOM (Document object model)

18.Promises

19.ES 6 Concepts

JAVASCRIPT:

It is a programming language which is directly understood by browser.

We can use java script for both front-end and back-end also.

JavaScript is also used for developing mobile applications, games, testing, to perform slideshows etc.,

It is also used for making servers.

History:

In early 1990's all the webpages were made by using HTML and CSS, they were static in nature, to bring interactivity java script into picture.

In 1993, mosqic was a browser which supports graphical user interface, was available to people.

The creator of Mosqic that is Netscape corporation created more polishes version Mosqic in 1994 and named it Netscape navigator.

Further, to add more interactivity to web pages they wanted to integrate a programming language inside their browser, they hired BRANDON EICH who created a new language within 10 days which was understood or supported by browser.

First, this language was known as Mocha after that live script and finally java script.

Java Script is now maintained by ECMA (European computers manufactures association).

JavaScript:

Java script is a scripting(It is used to automate another languages) language which is used to have interactivity to the web pages.

It is a dynamically typed language.

*Note:

Dynamically types languages: DTL are those whose data type is determined during the runtime.

Java script is a weakly typed language.

It is interpreted language that is line by line execution of code is done.

It is synchronous and single-threaded.

*Note:

By using some advanced concepts of java script like call backs, promises we can make it as synchronous.

It is object -based language.

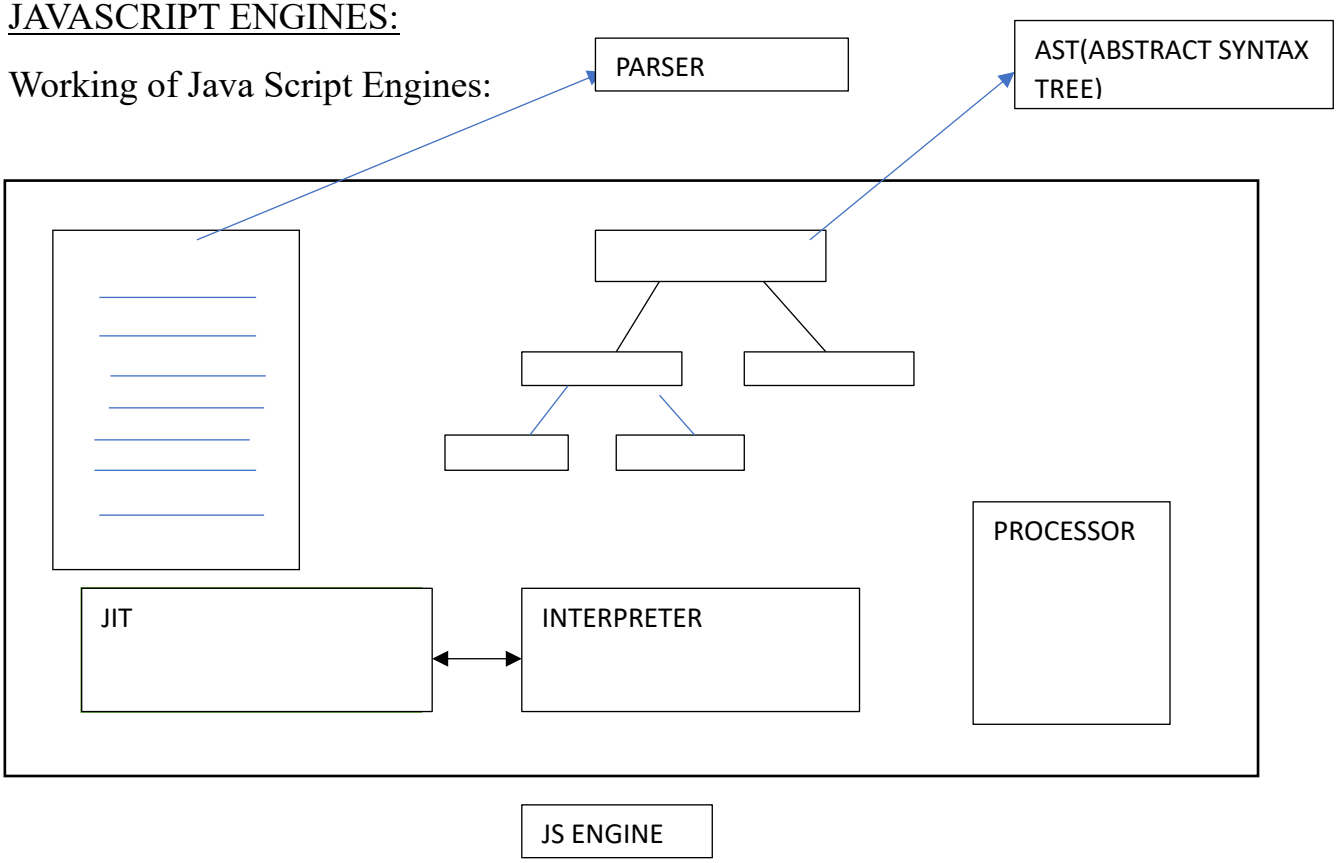
It is also object-oriented language.

10-01-2025

- Rendering Engine-is the component of browser is used to read html and css code
- JavaScript Engine-is a component of browser is used to read Java Script code.

JAVASCRIPT ENGINES:

Working of Java Script Engines:



Parser:

This is the first stage of the engine, every time we run a JavaScript program, our code is first received by the “parser” inside JS Engine.

The parser’s job is to check the JavaScript code for syntactic errors in line by line manner because JavaScript is a interpretive language, so whenever an error is detected by the parser, it throws a kind of error and stops execution of the code.

AST:

Once the parser checks all the JavaScript code and get satisfies that there are no mistakes /errors in the code, it creates the data structure called AST (it stands for Abstract Syntax tree).

JIT:

Modern Java Script engines use JIT compilation to improve performance. The bytecode is further compiled into at runtime, allowing the engine to optimize code based on how it is actually used.

Interpreter:

In addition to JIT compilation, Java Script engines also use an interpreter to execute bytecode or parts of the code directly. This allows for quick execution of code without waiting for full compilation.

Processor:

The processor’s role in a java script engine involves executing instructions generated by the engine, whether they are interpreted bytecode or JIT-compiled machine code.

The processor handles the low-level execution of these instructions, manages memory operations and supports concurrency and synchronous tasks.

<u>Browser of JS Engine:</u>

Chrome-v8

Mozilla Firefox-Spider Monkey

Internet Explorer- Chakra (Discontinued)

Edge- v8 and blink, (Chakra-before 2020)

Brave- v8

Node- v8

Safari- Javascriptcore

*Note:

Machine code is the binary code directly understood and executed by the CPU, specific to a hardware architecture, and optimized for performances.

Byte code is an intermediate representation executed by aa virtual machine or interpreter, designed for portability and flexibility and can be further optimized at runtime by the virtual machine.

Difference between JAVA and JAVASCRIPT

JAVA	JAVASCRIPT
It is a statically typed language.	It is a dynamically typed language.
It is a strongly typed language.	It is a weakly typed language.
It is compiled language.	It is a interpreted language(execution is done line by line).
It is independent or Stand alone language.	It depends on HTML file.
It is a programming language	It a scripting language(used to understand another language)
It is multi-threaded language	It a single- threaded language
It uses more memory	It uses less memory
It is object oriented language	It is object based language(Java Script is both object oriented and also based)

*NOTE:

Weakly typed: Pertains to the language's handling of type conversions and flexibility in operations different types;

Dynamically Typed: Pertains to when type checking occurs, with types being resolved at runtime. (we can change data types while runtime)

Strongly Typed: Pertains to the language's handling of type conversions and flexibility in operations involving different types.

Statically Types: Pertains to when type checking occurs with types being resolved at compile time.

A language can be both weakly typed and dynamically typed, as is the case with JavaScript.

It can be strongly typed and statically types such as java and it can also be one or the other independently.

For example: Python is dynamically typed but strongly typed, meaning it doesn't perform implicit type conversions in a way that could be lead to unexpected behaviour.

Object-Oriented Languages: Typically use classes, support inheritance, encapsulation, polymorphism and abstraction.

Object-Based Language: May use prototypes or structures and often focus more on practical use of objects rather than strict adherence to object-oriented principles.

Interpreted language: line by line execution is done.

Difference between Programming language or Scripting language

PROGRAMMING LANGUAGE	SCRIPTING LANGUAGE
Programming languages are those languages which are used to communicate with computers using set of instructions.	Scripting languages are those languages designed to automate the execution of task.
Most of the programming languages are compiled or compiler-based languages.	Most of the scripting languages are interpreted languages or interpreter-based languages.
It is used to develop an application from scratch.	It is used to combine existing components and automate a specific task.
All programming languages are not scripting languages.	But all scripting languages are programming languages.
Example: C, C++, Java	Example: pearl, python, java Script etc..

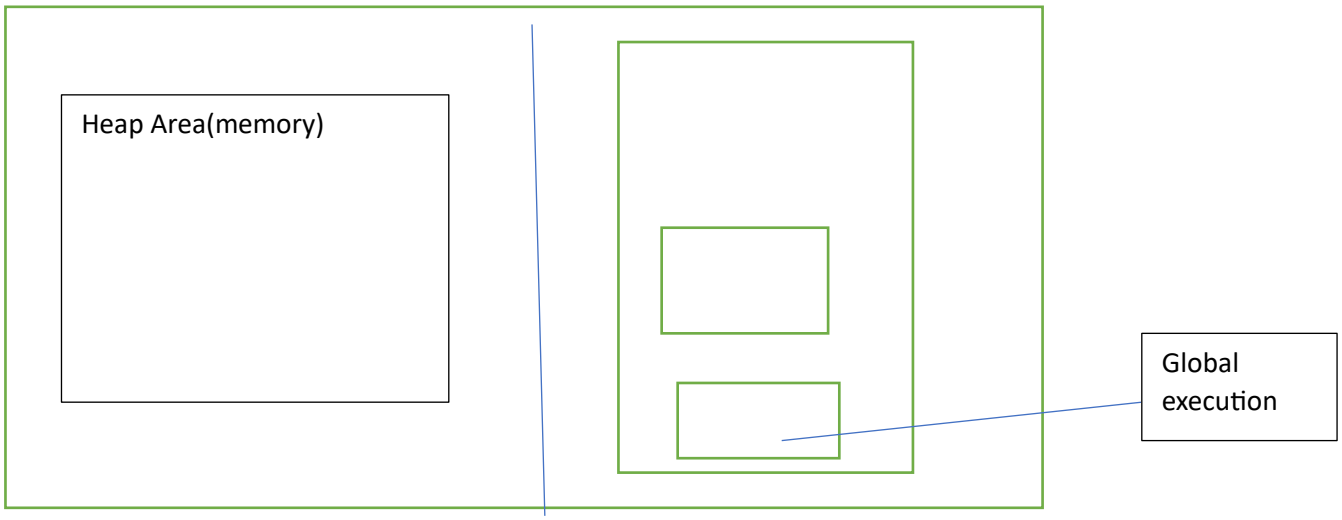
JS engine is divided into 2 Areas: Heap Area and Call Stack.

Heap Memory: This is the place where all the objects are stores which are necessary in the application.

Call Stack: It is place where your javascript code gets executed.

Execution Context: when the Java Script code it executes withing an execution context this context includes the global context for code outside of the functions and function context for code inside the functions.

-Each context had its own scope variable and functions.



Ways to execute Java Script code:

1.Directly on the browsers console.

Steps:

Open any browser of your choice

Move to developer's tool, that is right on browser and click on inspect

Move to console, here you can directly write your java script code.

2.Internal Way.

You can write code inside html page by using script tag.

Ex: <script>

Console.log('hello') </script>

3.External Way

Steps:

Create a java script file with .js extension.

Link your java script file to your html page by using script tag along with src attribute.(you need to specify the path of your java script file src attribute ./)

Ex: <script src=./hello.js(filepath)></script>

4.Node js environment

Node js is a runtime environment for java script.

By using node we can execute our java script files without html file.

This node environment also uses v8 engine internally.

Steps for executing in node js

Install node js

Check the version by using the command node –version

To run java script files we use command node filename.

*Note:

Script tag should always be written inside body tag or at the end of the body tag.

If your writing script tag inside head tag you will get unexpected output as head tag is executes before body tag.

If you want to write script inside head tag you can use defer attribute

This defer attribute will parallelly pass the script and execute it once the body is loaded.

TOKENS:

Tokens are the smallest individual units of any programming language ,

There are 6 types for tokens

1.Keywords

2.Identifiers

3.Literals

4.Separators

5.Operators

6.Comments

1.Keywords:

Keywords are the pre-defined words which have some special meaning.

Key words are always written in lower case.

It is understood by lowe cas.

Ex: var, let, const, async, break, continue, functions.

2.Identifiers:

Identifiers is nothing but name provided to any variable. Class, function of an object.

Rules to write identifiers:

→we can't use keywords as identifiers.

→identifier name start with a number can have number b/w it.

→it cannot have any special character except underscore(_) and $

→it does not contain any space in between

3.Literals:

Literal is nothing but the data provided by user

i.number→Int

decimal

ii.String→' '

" "

` ` template literals

iii.boolean → true *undefined *function

false *null * array

iv. object→{}

4.Separators:

Separators are used to differentiate the statement written in a javascript file .

Ex: semi-colon(;), single-quotes(‘ ‘), double-quotes(“ “), comma(,), colon(:).

5.Comments:

Comments are non-executable lines.

Comments are the line of code that are not executed in the browser.

Comments are used to add the notes, description and explaination about the js code.

Java script supports 2 types of comments:

1. Inline comments(//)
2. Multi-line comments(/* Sanjana */)

6.Operators:

Are pre-defined symbols which are performing some specific tasks.

Ex: x+y=z, x, y are operands and + is an operator z is a result.

VARIBALES:

Variables are the names that are given to memory locations where your data is stored

Important table

Data Type	Initialization	Declaration	Re-Initialization	Re-Initialization
var	possible	possible	possible	possible

let	Possible	possible	no	Possible
const			no	No

In const both declaration and initialization together are possible.

*Variable is nothing but name given to the block of memory.

We can create a variable in java script using variable declaration followed by variable name.

Syntax: variable_declaration variable_name=value;

There are 3 types of variable declaration

1.var

2.let (added in ES6)

3.const (added in ES6)

var a; →Declaration of a variable

a=20; Intialiasation of a variable

var b=30; →declaration and initialization of variable

var =40; →reinitialization

var a=50; →re initialization

1.var :

→var is the traditional way of declaring the variables.

→The var statement declare function scope or global scope variables optionally →initializing the value to the variable.

→It can be redeclared and updated within its scope.

→It can be declared without initialization.

→It can be accessed without initialization as its default value as undefined.

2.let:

→the scope let variable is the block scope.

→it can be updated but cannot be redeclared in the same scope.

→it can be declared without initialization.

→it cannot be accessed before initialization.

→if you are trying to access after the declaration without initialization, we will get value as the undefined.

3.const:

→the scope of the const variable is block scope.

→it can neither be updated or redeclared in any scope.

→it cannot be declared without initialization.

Bare Declaration : **

When ever are declaring a variable without using var, let, const declarations then JS engine automatically treats the declarations as Bare Declarations.

Bare declaration works only when the variable is not in any block or function i.e. it works only when the variable is declared globally.