Java Script

1. What is Java script and its Behaviour?

Ans: Java script is scripting, high-level, Interpreterd language which is used for the developing or building the Web applications which is more interactive and to add the functionality for Mobile applications, Desktop Applications, Web Applications.

Basically, Java script act as both scripting and programming language. Based, on the situation behaviour as Scripting language when it is used in front-end and Programming language in Back-end. It will be performed by the help of APIs.

* Java script is high-level language. It is also used for the server side programming and in game development.
* Front-end: User-interface - Use case: Perfumes Logic building for API integration purpose.
* Back-end: Logic Building for API building purpose.

1. How many kinds of Applications can be build using Java script?

Ans: Using Java script we can build:

1. Single Page Application: In these page app. A web page open within the same page when action is performed.
2. Hybrid Page Application: It will act as both the Single and Multiple page app. Based on the user requirement it will changes the functionality of application.
3. Multi page Application: Here, App. behaviour like open in multiple pages when the actions performed.

Note: So, we can’t say directly as an application is the single or multiple or Hybrid app.

1. Is Java script uses the compiler or Interpreter or both?

Ans: Interpreting language: It is executed in the browser directly without being complied into machine code.

Interpreter: Here, Code is executed line by line. So, it is called as Interpreter language.

* Java script is also uses complier but for partial time purpose only

Complier: It executes complete code at a time which converts high level to low level language.

So, recently JS used JIT complier called JUST IN TIME compiler. It can be used with Interpreter along with JIT compiler.

Usecase: It used when the block of code have to be executed frequently JIT complier is used.

\*\* JS can be called as Single Threaded Language. Because, code will be executed line by line.(uses interpreter were browser will be executed)

1. Mention some JS Engines and their information.

Ans: Chrome-uses: V8 Js engine

Mozilla firefox-uses: Spider Mockey

Microsoft(Edge)-uses: Chakra

Facebook- own developed Hermes

Safari-uses: JScore.

1. Explain concept of synchronous and asynchronous programming language and their operations in JS.

Ans: Synchronous programming language mean when is task is performed the task will executed line by line. So, it will takes time to complete the large tasks such the perform an action in a game or application. For example: To get reponsiveness from server (HTTP) and update in UI. it will consume more time to update the action requested so far. But, here we have to perform the multiple actions at a time to get responsiveness to get rendering(display) the content.

Ex: 5 4 1 3 2 (set on order from here to )🡪 1 2 3 4 5(here)

\*\* Java Script is also called as Synchronous programming language (it will performed when server side programming)

For, these Asynchronous came into action.

Asynchronous Programming language: It can handle multiple tasks at a time were multiple problems ca be executed at the same time without blocking the main thread(main block of programms)

Thread: It is lightweight unit of processing unit which can run concurrently with other threads. These are performes CPU intiated tasks without blocking main thread, these are responsible for the handing the user input and rendering to the UI.

To perform the multiple operations we need some operations to perform. There are a number of ways to implement asynchronous operations in JavaScript.. Those are:

1. Call by function
2. Call back function
3. Promises function
4. Asynchronous await function
5. Explain interpreter stages.

Ans:

1. Lexical analysis 🡪 acts a Required analysis to perform
2. Semantic analysis 🡪 check the errors and bugs to handle
3. Syntax analysis 🡪checks errors to solve with syntax
4. Execution 🡪then here, runs execution for a problem.
5. Repeat 🡪 if any thing problem raises same process will repeat to resolve it.

Let’s take an example to understand:

I want to cook an curry I need to prepare the items , check the required items, need ingridents and have to add particular quantity of ingrridents after then I start the process. If is not enough I have to repeat he process. Same way.

1. What defer in js and its uses?

Ans: Defer is an attribute of Boolean attribute which is used along with source (src=””)in a script tag.

<script src=”example.js” defer>

//code goes here

</script>

Uses: It majorly used for the downloads the script after the html parsing (means the script content won’t execute and waits till the html entire document execution completes by this can achieve the responsive and faster page loading to render in the browser)

\*\* it is used in external javascript only.

Along, render-blocking behaviour during html parsing. Gives gurrented execution order for deffered scripts in the order they appear in the document. By this can achieve improve in page loading performance by preventing render-blocking

1. What is console and console.log() and why it is used?

Ans: Console: it is debugging tool it is an object which present in the web browser and development environment like Vs code.

Uses: it shows errors messages, inspect the code of javascript, output information during program execution. We can access in browser using key word : F12

Console.log(): it is method of which is provided by the console object in javascript

Uses: it will prints the messages , variable vales, any data which want to inspect during code execution. Output will displayed console of developer

Ex:

let name="Sai Kiran"

let age=25;

console.log("My name is:", name, "My is age:",age);

\n used to start in a newline for strings

Output: My name is: Sai, My age: 21

Other ways

Let name=”Sai\n”;

Let age=21;

But here we use: only singlequotes to enter value and $ symbole to the particular let variable inoutput

let name="Sai Kiran"

let age=25;

console.log(`my name is ${name}, My age :${age}`);

Output: my name is Sai , My age: 21

1. Explain declaration, Assigning/Assiginment, initiallisation

Ans: Declaration: it will introduce the variable to store the given value to it and along with datatype(var, let or const).

Commonly, we consider variable to declare the varibales like a, b, x, y…. or name, age… // declaration of varable

Mean, it will store value passed given to it.

Assigning/Assignment: Here, actually we give the value to the variable to store in it.

Like X=4; or name =”Sai” //assigning value to variable might be number or string

Initiallisation: consider with datatype, declared a variable then given a particular value to it all come in a combination known initiallisation

Like var x=5; //initilisation

10. How many types we can declare a variable?

Ans: we can declare a variable in 4 types

1. Automatically or implicit decleration:

This declaration of variable highly not recommended. Because when we assign value to a variable with out declaration it becomes the global variable.

Ex: example()

X=10; // didn’t declare variable so, same value applies to all the variables taking further

Console.log(x);

1. Let: it a most and common used variable to declare

Here, we can can’t re-declare the varible, can’t re-initialize the

Variable but we ca re-assign value to the varible.

So, it is blocked-scope function varible.

Ex: example()

X; // we can’t re declare

X=10; // we can re-assgin value later

X=20;// re-assigning

Var x=10; we can’t re-initialize

let x=0; // no declare

x=15; //re-assign can be done

console.log(x); output: 15

1. Var: it function scope varible. We can re-decalre, re-assign, re-initialise to varible.
2. Const: cant be re-decalre, re-assign-, re-intiliase to the variable
3. const x=0;
4. // const x=15;
5. console.log(x);

Datatypes in java script:

1. Primitive datatypes:

We can assign only single value, mutable(we can’t change the value again). Takes new memory space each time for allocation of value. Like a=0, a=5;

\*\* we have ot access with keyname\

Consle.log(a);

1. Number
2. String
3. Un-defined
4. Boolean
5. Null
6. Big int
7. Symbol
8. Non-primitive data types: this are accept multiple values at a time. Immutable (we can change the value not address of value were new value will replace previous address position of memory)

Ex: a=[1,2,3,4,5]

A=[1,1,2,6,6]

\*\* we have to access with the index value like cosole.log(a[0])

It is for array.

1. Functions
2. arrays
3. objects
4. regular expressions