**1.why JS is called scripting language?**

The interpretation and execution takes place at the runtime, hence it is called as scripting language.

**2.Why JS is called interpretted language**

In order to convert the high level language or low level language to machine level language so many translators are available eg: compiler, assembler and interpreter. In this case java script uses **interpreter as an translator** , so they are called as interpreted language

**3.** **What are the advantages/ uses of JavaScript language**

**Advantages:**

Javascript can be used as a coding language to develop the 3 layers of the application. They are

1. Client browser (client verification) – uses core js (vanilla js)
2. Webserver side -- uses node js
3. Database side – uses Mongo DB or Couch DB.

**4.** **what are the 3 types of programming languages.Explain briefly**

3 types of programming languages are

1. High level language
2. Assembly level language
3. Machine level language

**Machine level language:**

1. This language is in the form of binary digits like 0 and 1.
2. The processor can understand only this machine or binary language.
3. The codes which are written in high level language should be converted to machine level language in order for the processing unit to process the code

**Assembly level language**

1. The assembly level language has to be converted to machine level language by use of any translators.
2. Translators are used to translate the low level language to machine level language
3. Assemblers are used as an translators to convert low level language to machine level language

**High level language**

1. The codes which is written in javascript are called as high level language
2. In order to convert this high level language to machine level language we make use of compilers as an translators

**5.** **What is a translator. What are the different types of translators and what are the difference b/w them**

Translator is used to convert high level or low level language to machine level language because the processor understands only the machine level language

Types of translators are:

1. Compilers
2. Interpreters
3. Assemblers

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Compilers** | **Assemblers** | **Interpreters** |
| 1 | It converts high level language to machine level language | It converts low level language to machine level language | It converts high level language to machine level language |
| 2 | Syntax parsing of the file is done in one run | Syntax parsing of the file is done in one or two run | Syntax parsing is done by line by line |
| 3 | Creates one class file after the successful compilation | Creates one or two object file after the successful compilation | No files has been created after the successful compilation |
| 4 | Execution is faster | Execution is faster | Execution is slower |
| 5 | Suitable of huge set of codes | Suitable for huge set of codes | Suitable for low set of codes |

**6.** **What is Node JS**

Node JS provides an environment for the javascript code to run

**7.** **What is VS code**

It provides an environment to write our javascript code

**8.** **What is the difference b/w var, let, const**

|  |  |  |
| --- | --- | --- |
| **Var** | **Let** | **Const** |
| Its used for the variable declaration | Its used for the variable declaration | Its used for the variable declaration |
| Variable declared using var keyword can be re-declared multiple times | Variable declared using let keyword cannot be re-declared multiple times | Variable declared using const keyword cannot be re-declared multiple times |
| Variable declared using var keyword can be re-intialized multiple times | Variable declared using let keyword can be re-intialized multiple times | Variable declared using const keyword cannot be re-intialized multiple times |
| Eg: var a =20  Var a = 30  a=40  all ways are possible | Eg let a = 40  a=60 (possible)  let a=30(not possible) | Eg const a = “hello”  a=”hi” (not possible)  const a=30(not possible) |

**9.** **What is the difference b/w == and ===. Example related to your project**

**==:** It check only for the matching literals in the variable

Eg: if(10==”10”) 🡪 returns true

**===:** It check for the matching literals as well as the datatype of the variables.

Eg if(10===”10”) 🡪 returns false

**10.** **What are the different type of datatypes available.**

There 2 types of datatypes available ,they are primitive and non primitive data type

1. Primitive data type
   1. Undefined
   2. Null
   3. String
   4. Numbers
   5. Boolean
2. Non primitive data type
   1. Object
   2. Function
   3. Arrays
   4. Math
   5. Date

**11. What is the predefined method available to find the datatype**

**Type of** is the predefined method available to find the data type of the variable

Eg. Log.console(typeof(“10”))

Output:- string

Eg. Log.console(typeof(10))

Output:- Numbers

**12.** **Explain client server validation with an example(Provide the requirement for the element which is being validated)**

Client server validation can be achieved in javascript by using interpreters as their translators.

Example: In my project there is an page called apply credit card page where in customers apply for the credit card. In that page there is an mandatory field to enter account number. The requirement of text field accepts only 11 digit numbers only. If customer makes any mistake in the syntax it will shows an error immediately saying please enter correct credentials and if customers enter correct syntax and the data which has been entered will be verified with the data which is already been stored in the database. If the account number entered by customer is wrong it shows invalid data entered.

**13.** **Why JS is called full stack data file**

Because javascript can be used as a coding language to develop the 3 layers of the application. They are

1. Client browser (client verification) – uses core js (vanilla js)
2. Webserver side -- uses node js

3.Database side – uses Mongo DB or Couch DB

**14.** **Give the different examples for libraries developed using JS**

1. JQueries
2. Bootstrap
3. Load#
4. \_js

**15.** **Give the different examples for frameworks developed using JS**

1. Core JS
2. React JS
3. React native
4. Angular JS
5. Electron JS
6. Tensor Flow

**16.** **What are the different types of tokens. Write the conditions for writing the keyword and identifier**

1. Keywords
2. Identifiers
3. Literals
4. Operators
5. Seperators

**Keywords:**

1. Keyword are the set of predefined words where it has own meaning given already.
2. Keywords should not be used as an identifier
3. Keywords are case sensitive
4. Eg: break, continue, var, let, const, function etc

**Identifier**

1. Identifiers are names given by the programmer to identifies the variable and function in the code
2. Keyword should not be used as an identifier
3. Numbers and special characters should not be used as an identifier except \_ and $ Eg. \_5name and $4add can be used
4. Its also an case sensitive Eg:- arr and Arr are two different variables

**17.** **What do you mean by JS engine. Name of the JS engine for chrome browser**

Every browser has an own inbuilt translators which is called as an JS engine. So these translators convert the high level or low level language to the machine level language

**Chrome browser uses v8 as a JS engine**

**18.** **Why variable is loosely typed in JS.What is the advantage**

In java Script all variables are loosely typed because data type declaration to the variable happens at the run time.

**Advantages:**

**1.** Can change the data type at the run time

2. explicitly we can change the data type of variables

3.Implicitly takes the data type of variable based on the literals or values assigned to the variable.

**19.** **How can we change the datatype explicitly**

Explicitly the data type can be changed like

Number(“10”) 🡪 it converts string data type “10” to number data type 10.

String(10) 🡪 it converts number date type 10 to string data type “10”

number to string and string to number is the only possible explicit conversion.

**20.** **Write pgrms along with the output showing the difference b/w pre and post increment**

**Progaram1:**

Var a = 10

Log.console(a)

Log.console(a++)

Log.console(a)

**Output:**

**10**

**10**

**11**

**Program 2:**

Var b = 30

Log.console(b)

Log.console(++b)

Log.console(b)

**Output:**

**30**

**31**

**31**

**21. What are the different types of JS execution. Explain briefly**

1. On browser execution
   1. In line embedded execution
   2. External line embedded execution
2. Off browser execution

1.**On browser execution**: It uses browser for the execution of the java script code Eg:- browsers like chrome, firefox, IE can be used to execute the java script code because each has own JS engine

**a.In line embedded execution**

It uses html code for execution in the browser where in scripts can be written in html using document write which will be displayed on the web page

**b. External line embedded execution**

It also uses html code to run the scripts in the browser. We can add the source .js file to html code for the execution

Eg: <script src=”filename> <script/>

**2.** **Off browser execution**

It does not uses browser for the execution of the java script. Eg:- Visual studio used node Js to run the java script

**22.** **Write the progms and demonstrate the execution of all looping and conditional statements.Program and with one example of your project**

**Looping statements:**

1. For loop
2. For in
3. For of
4. For each

**For loop**

Var a = 10

For(var i=0; i<4; i++){

a++

Log.console(i,a)

}

**Output:**

**0 11**

**1 12**

**2 13**

**3 14**

**For in**

Var arr = [1,2,3]

for (let index = 0; index < arr.length; index++) {

    console.log(index+"    "+arr[index]);

}

Output:-

0    1

1    2

2    3

**For of**

Var arr = [1,2,3]

for (let index = 0; index < arr.length; index++) {

    console.log(arr[index]);

}

Output:-

1

2

3

**For each**

arr.forEach((element, index) => {

    console.log(index+"       "+element);

});

Output:-

0   1

1  2

2   3

**Conditional statements:**

1. **If**
2. **If else**
3. **Nested if**
4. **Switch**

**1.If**

Var a = 10

If(a>5){

Log.console(“larger number)

}

Output:- larger number

**2.If else**

Var a = 10

If(a>15){

Log.console(“larger number)

}

Else{

Log.console(“smaller number”)

}

Output:- smaller number

**3.Nested if**

Var a = 11

If(a>15){

Log.console(“larger number)

}

Else if(a<10){

Log.console(“ number is below 10”)

}

Else {

Log.console(“ number is between 10 to 15”)

}

Output:- number is between 10 to 15

**4.switch**

Var a =2

Switch(a){

Case 1:

Log.console(“inside case 1”)

Break;

Case 2:

Log.console(“inside case 2”)

Break;

Case 3:

Log.console(“inside case 3”)

Break;

Default:

Log.console(“not an number”)

Break;

}

Output:

Inside case 2

**23.** **Write the programe to demonstrate the concept of break and continue**

**Continue**

var a = 20

for(var i=0; i<4; i++){

if(a<26){

a++

log.console(i+” “+a)

continue;

}

Log.console(“outside if statement”)

}

Output:

0 21

1 22

2 23

3 24

**Break**

var a = 20

for(var i=0; i<4; i++){

if(a<26){

a++

log.console(i+” “+a)

break;

}

Log.console(“outside if statement”)

}

Output:

0 21

**25.** **What are functions and different types of functions (with syntax)**

Functions are the set of codes that performs specific operation

It can be reused multiple times whenever it is invoked

Different types of functions are

1. Function declaration/ standard function
2. Function expression
3. Immediate Invocable function (IIF)
4. Arrow function

**1.Function declaration / standard function**

**Syntax:** function ref\_name(parameters){

// code to be executed

}

Ref\_name(parameters)

**2.Function expression**

**Syntax:** var ref\_name =function (parameters){

// code to be executed

}

Ref\_name(parameters)

**3.Immediate invocable function**

**Syntax: (**function ref\_name (parameters){

// code to be executed

}) (parameters)

**4.Arrow Function**

**Syntax: multiline arrow function**

var ref\_name = function (parameters) =>{

// code to be executed

}

Ref\_name(parameters)

**Syntax: single line arrow function**

Var ref\_name = function(parameters)=>(//code)

Ref\_name(parameters)

**26.** **What are objects and mention the different types of objects. (with syntax)**

Objects have state and behavior which can be used accessed in the other objects

In java script object have property and value which can be used in another objects

Different type of object declaration are

1. Object literals
   1. Syntax: var obj\_ref = {prop1: value1 , prop2:value2,..}
2. New object keyword
   1. Syntax: var obj\_ref = new object()
   2. Obj\_ref.prop1=value1
   3. Obj\_ref.prop2=value2
3. Constructor
   1. Function ref\_name(prop1, prop2,..){
   2. This.prop1=prop2
   3. This.prop2=prop2
   4. }
   5. Var obj\_ref = new obj\_ref(value1, value2, value3,..)
4. Class
   1. Class ref\_name{
   2. Constructor(prop1,prop2,){
   3. This.prop1=prop2
   4. This.prop2=prop2
   5. }
   6. Var obj\_ref = new obj\_ref(value1, value2, value3,..)
5. Object .create() method