JavaScript

* Every Browser will have a JavaScript Engine, and this can execute the JavaScript code.
* Node.js is just the server written on JavaScript.
* Install Visual Studio code.

How JavaScript works?

Every Browser has JavaScript engine internally E.g. inspect in Chrome and go to Console and type Console.log(‘Hello’) …It will print Hello

**Data Types:**

Refer to DataTypes.js for below examples.

* There are eight data types in JavaScript.
  + Primitive Data Types
    - String
    - Number
    - Boolean
    - Undefined
    - Null
  + Reference Data Types
    - Object
    - Array
    - Functions

**Operators:**

Refer to Operators.js for below examples.

* Arithmetic (Pre-increment, Post increment, Pre-Decrement, Post-Decrement)
* Assignment
* Ternary (?)
* Comparison (Strict Equality and Loose Equality)
* Logical (&&,||.!, Logical non Boolean)
* Bitwise (not that much imp)

**Control Flow Statements:**

Refer to ControlFlows.js for below examples.

* If-else
* Switch

**Loops:**

Refer to Loops.js for below examples.

* For
* while
* do-while
* for.. in
* for.. of

**Objects:**

* Objects are nothing but the entity. Which has properties.
* Best Practice to define the Object is using the const keyword.
* Variables and functions can be defined in the Object as a key and value format check below file.
* Factory functions are those which create different objects.
* For a Factory function we need to use Camel notation (factoryFunctions)

Refer to Object.js for below examples.

Refer to FactoryFunctions.js for below examples.

**Constructor Functions:**

* This is also used to create several objects.
* In simple terms, It is simply a function which acts as a pattern or template for creating objects.
* With Constructor function we can instantiate objects and implement **inheritance**.
* For Creating Constructor Functions, we need to use Pascal Notation (ConstructorFunctions)
* To call the Construction function we need to use new Keyword.
* New keyword does 3 things.
  + It first creates an empty object.
  + Ensures this variable is pointing to newly created empty object.
  + Finally, it also returns the objects from the function constructor.

Refer to ConstructorFunctions.js for below examples.

**Dynamic Nature in JavaScript:**

* During runtime (after creating Object) we can add or delete the properties of the object hence it is very dynamic in nature.

Refer to DynamicNature.js for below examples.

**Value Type and Reference Type:**

**Value Type**

* All the Primitive data types are Value Type
  + String
  + Number
  + Boolean
  + Undefined
  + Null
  + Symbol

**Reference Type**

* All the Non-Primitive Data Types are Reference Type
  + Objects
  + Arrays
  + Functions

Refer to ValueandReferenceTypes.js for below examples.

**Object Enumeration:**

* Object Enumeration is nothing but reading the properties of the Object
* Properties can be read by
  + For in
  + For of
  + In operator

Refer to ObjectEnumeration.js for below examples.

**Object Cloning:**

* Object Cloning can be done in 3 ways.
  + For in loop
  + …spread operator
  + Using Object.assign

Refer to ObjectCloning.js for below examples.

**MATH and DATE Objects:**

* Math and Date Objects have lot of methods.
* Math.random()
* Const date= new Date() // to get the current date
* Date.toDateString() //to get date in particular format
* Date.toTimeString() //to get the time in particular String

Refer to MathDateObject.js for below examples.

**Template Literals:**

* After ES6 template literals are introduced
* Backtick ``
* Using this we can print the same way as we are writing.
* To skip the particular line, we need to use $ before it

Refer to TemplateLiterals.js for below examples.

**Arrays:**

* Arrays are dynamic and not strict and can store different data types.
* Arrays can have objects as well.
* To add the value last in the array .push()
* To add the value first in the array .unshift()
* To add in the middle of the array .splice(starting point, deletion count, value)
* To delete the value in the last .pop()
* To delete the first value .shift()
* To delete the particular value .splice(starting index,delete count)
* To know the length of the array .length()
* To know the index .index()

Refer to ArrayConcepts.js for below examples.

**Object Oriented Programming:**

* **An Object**  is something which has both data(properties) and the behaviour(Methods)
* **Object Oriented Programming** is also known as OOP which is used to the program in an organised way.
* There are 4 pillars of OOPS.
  + Inheritance
    - mechanism that allows us to create new classes from the existing classes
    - With inheritance we can reuse the existing code without repeating.
    - **Constructor Function** is the example.
    - **Adv: Code Re-Usability.**
  + Encapsulation
    - It is the process of hiding the data from Outside world.
    - It is achieved by using the private keyword.
  + Abstraction
    - Abstraction is the way of hiding the implementation details and showing only the functionality to the User.
    - In other words, it ignores the irrelevant details and shows only the required one.
  + Polymorphism
    - Poly –many and morphism – forms
    - Writing in different forms
    - Write methods with same name with different implementations

**INHERITANCE:**

* mechanism that allows us to create new classes from the existing classes
* With inheritance we can reuse the existing code without repeating.
* Using Inheritance we can inherit the property of one class from another class
* **Constructor Function** is the example.
* **Adv: Code Re-Usability.**
* **Inheritance** in JavaScript can be achieved by Prototype.
* **Prototype**
  + Each and every object in JavaScript has the prototype property.
  + **Inheritance** is achieved by using the prototype property.

**N:B:**

Every newly created Object in java is the instance of Object object.

Refer to Inheritance.js for below examples.

**JavaScript Class and Class Constructor:**

* JavaScript allows us to create blueprint and using this we can instantiate objects.
* It is like function constructor and prototype inheritance (internally)
* Class are Executed in Strict mode.
* Classes cannot be hoisted (cannot be used before declaration)

Refer to ClassesConstructor.js for below examples.

**Encapsulation:**

* Hiding the data from outside world is encapsulation.
* Getters and Setters are used for it.

**Getters and Setters:**

* **Getters** are used to get the property of the Object
* **Setters** are used to set the property of the Object.

Refer to Encapsulation.js for below examples.

**Static Methods:**

* Static methods are those methods which are associated with class or objects .

Refer to StaticMethods.js for below examples.

**Object.create() Method:**

* This is used to achieve prototype inheritance

**INTERVIEW QUESTIONS:**

**1.**Java vs JavaScript

* Java is a Programming Language
* JavaScript is the Scripting language.
* Java will have .java file
* JS will have .js
* Runs only on JVM
* JS runs in all web browsers
* Java is multithreaded
* JS is Single Threaded
* JAVA consumes more memory compared to JS
* JS consumes less memory since it runs on browser.

2.What is JavaScript?

JavaScript is programming language (Client Side Scripting language) that is used to make server dynamically interactive. Using JavaScript we can built WebApp(Facebook), MobileApp, WebSites, DesktopApps(slack, WhatsApp)

3.How JavaScript Works?

* Initially JavaScript works only on the browser engine due to this some limitation where there so RYAN did wrap JavaScript engine with C++ and its called Node.
* Using Node we can build web application , mobile etc .

4.What is Echma Script?

* Also called as ES , It is nothing but the set of rules. Currently ES6 is the version.

5.What is a Variable?

* Temporary location used to store the value.
* Can be declared using var,const,let.
* Eg. var num=5;
* But var is not much recommended.
* N:B Variable name should not be a keyword (like if,switch,let)
* Variable name should not start from number (eg. 1hub)
* Variable name should be unique and should be in camel notation.
* Variable name is case sensitive.
* Variable name should be meaningful.

6.What are JavaScript Constants?

* When using let keyword we can re-assign/modify the value where as using const we cannot re-assign/modify the value
* Const are mostly used for static ones.
* Const can be used where values will not be changed further.

7.What are JavaScript DataTypes?

* Two types of DataTypes
  + Primitive
    - String
    - Number
    - Boolean
    - Null
    - Undefined
    - Symbol //After ES6
  + Reference
    - Arrays
    - Objects
    - Functions

8.What is a JavaScript Objects?

* An Object is the entity with properties and type.
* Eg. let person ={ }
* Object property should be in key value format.
* Object Property can be accessed by two types
  + . notation
  + [] notation
* {} -Object

9.Array in JavaScript?

* In JavaScript Arrays are dynamic and can store dissimilar data
* Arrays are based on index.
* To find the length .length
* To find the index .indexof(‘’)
* To push the value to last .push()
* To remove the last .pop()

10.functions in JavaScript?

* It is a block of code designed to perform particular task.
* It executes when something invokes or calls it.
* Every Function is the Object in JavaScript

11.Operators in JavaScript?

* Arithmetic Operators
* Comparison
* Assignment
* Bitwise
* Logical

12.Equality Operator?

* Strict Equality ===
  + Should have same Datatype and Value
* Loose Equality ==
  + Will take the first operand value
* Eg. console.log(‘1’ ===1) //false
* Console.log(‘1’ == 1) //true

13.Ternary Operator?

(Condition)?’option1’:’option2’

14.Logical Operators with Non Boolean..

* Falsy:
  + Undefined
  + Null
  + 0
  + False
  + ‘’
  + NAN
* Truthy:
  + Anything that doesn’t fall under falsy is truthy.

15. What are Conditional Statements?

* If-else
  + If condition is true it will execute if block else it will execute else block
* Switch.
  + It checks the condition and switch to the case and executes it.

16.What is a Factory Function?

* Factory function is used to create a function that is generic and using this we can call and make use of it.

N:B: Every Object has the default constructor in it.

Eg. let name=’’anbu’’

Let name =new String “anbu” //internally it will show like this

17.Reference Type and Value Type?

* Value Type:
  + Primitive DataTypes
    - Let x=10;
    - Let Y=x;
    - X=20; //while printing x=20,but y=10
    - This is because its primitive and x and y are independent variable and this will copy only the value
* Reference Type:
  + Reference DataType
    - Let x={ value:10};
    - Let y=x;
    - x.value=20; // while printing x=20,y=20
    - This is because it’s a reference type and both shares the same memory location

18. Enumeration Property of Object:

* Using for in loop we can get the property of the Object whereas using for of loop we cant get directly since it is not iteratable as like arrays.so we will be using as below.
* For (let key of person)
* Log(key) //it wont work since it is an Object so we need to use another method
* For(let key of Object.keys(person)
* Log(key)
* Here Object.keys will convert it to Array and then display the results.

19.Object Cloning in JavaScript:

* Object cloning can be done by three ways
  + For in loop
    - For(let key in user){
    - Another[key]=User[key]
  + Object.assign
    - Let another=Object.assign({},user)
  + Spread operator //after ES6
    - Let another={…user}

20.JavaScript Garbage Collection?

* JavaScript Engine (V8) will take care of automatic garbage collection once the usage is over.

21. JavaScript Internal Object(MATH) ?

* MATH is not like a constructor.
* Math methods are static and have various methods
* Math.random()
* Math.sqrt()
* Math.PI -🡪Properties in Math Object
* Math.floor()

22.JavaScript String Object ?

* Set of characters enclosed within quotes/template literals is called String.
* It has a property of .length
* It doesn’t uses index number so if there are 4 characters it will show length as 4.
* .chatAt() –but we have to use the char index based.
* .concat(‘’) used for concatenation.
* .includes(‘’) ---results as Boolean value and ensure to give with correct case sensitive
* .startsWith(‘’)
* .endsWith(‘’)
* .indexOf(‘’) –to find the index of in a string.
* .replace(‘’,’’) -it will replace the given value.
* .split()
* .substr()
* .trim() –used to remove the spaces

23.Template Literals?

* Back tick `` is Template literals
* It is used to maintain the same format with help of backtick
* To include the variable we can use ${}.

24. Date Object ?

* Let date =new Date()
* Using date we can call other methods like
* getFullYear()
* Date.now will give the current date
* .getDate()
* .getDay() ---starts from Sunday and gives value as 0.

25.Empty an Array?

* Array can be empty by
  + Reinitialise the array to blank array .Even though different memory is used garbage collector will take care and remove one.
  + By setting .length =0 …This will empty the array
  + .splice(0,num.length())
  + And using while loop use .pop() //but not recommended due to loop

26.Combine and Extract Array?

* To Combine two arrays we can use .concat()
* To Extract the values from the array .slice (1,3) //1 starting index 3-ending index
* The above methods are used old days after ES6 we can use Spread operator
* Let receipe= […shopping\_Cart,…receipe\_Cart]

27.Iterate an Arrary?

* Using for of loop
* Using for in loop
* Using forEach loop
* vegetables.forEach((key)=>{
* console.log(key)
* })

28.Joining the Array and Split ?

* Using .join(‘-’) -it will join the string with –
* Using .split(‘ ’)-it will split the string based on the space.

29.Iterating each and Every Element in Array ?

* After ES6 they introduced some and every
* Using .every(function(value){
* Return value%2===0
* }) //returns true if all the value is true //else it will go to false
* Using .some() ---same as every but it will throw true if one element is true like or function.

30.Filtering the value by iterating each value in array?

* .filter(function(value){
* Return value>=18} //returns list of values greater than 18

31.find vs filters?

* Find will show only the first occurrence (only one)
* Filter will show all the value occurrence. )all occurrence)

32.Reduce() in JavaScript?

* It returns the single value like the function accumulated result
* It doesn’t execute for empty array.
* It doesn’t change the Original Array.
* Use case : Sum of all elements in the array
* Reduceright() is also used for same it takes value from right

33.Function Expression vs Function Declaration ?

//Functional Declaration and Function Expression

//1.Functional Declaration

//no need semicolon at the end of function declarations

let x=10;

let y=10;

function addNumbers(){

console.log(x+y);

}

addNumbers();

//Function Expression

//need to end the statement at the end of function Expression.

//two types

    //Function Named Expression

    //Anonymized Function Expression

//function Named Expression since function name a is on both sides

const a=function a(){

    console.log(x+y);

};

a();

//Anonymized Function Expression

const b=function (){

console.log(x+y);

};

b();

34.Function Hoisting?

* Hoisting is nothing but the interpreter considers declaration of variable/function as first before execution of code.

**Important:**

* **Calling the Function Declaration before initializing is possible but calling the function expression before initializing will throw error.**

**35.** Immediate Invoked Function Expression or Self Invoking Expression?

* Instead of manually calling a function which has actions to be completed it can be done by making it as self invoking using like below. (covering function with () and ()
* let x=10;
* let y=20;
* const addNumbers = (function(){
* console.log(x+y);
* })();

36.Rest parameter in JavaScript?

* After ES6 Rest operator is added to the JavaScript.
* Rest Operator are used in the functions
* And it will be used as a arguments for a function and should be always the last parameter.

function sum(...theArgs) { //… is the rest parameter

    let total = 0;

    for (const arg of theArgs) {

      total += arg;

    }

    return total;

  }

  console.log(sum(1, 2, 3));

  // Expected output: 6

  console.log(sum(1, 2, 3, 4));

  // Expected output: 10

37.Default Parameter in JavaScript?

* If value is not passed as a argument in the function then we can use default value as a parameter.
* Always default parameters should be declared at last.

function calcualteTax(cost,tax=18){ //while calling function tax is not defined so it will take

    //default parameter

taxAmount=cost\*(tax/100);

    console.log(

`The Total Cost is ₹${cost}

The Total tax ${tax}% for the Amount is ₹${taxAmount}

The Total netPay Amount is ₹${cost-taxAmount}`)

}

calcualteTax(2000,)

38. Difference Between let/var/const keywords?

//declaration of var/let/const

//except var ---let and let/const cannot be redeclared

var x=100;

var x=100;

let z=100;

let z=100;

const m=100;

const m=100;

//Reinitialize

//except const --- var/let can be reinitialized

var i=100;

i=200;

let g=100;

g=400;

const n=100;

n=200; //throws error Re-Assignment is not allowed

39. String Methods?

* String Methods are as follows
* let day = "mondaya ";
* console.log(day.length); //to find the length of the string (Includes spaces as well)
* console.log(day.charCodeAt(0)); //to find the ASCII character value
* console.log(day.substring(1)); //to create a new string from the starting index given //onday will be output
* console.log(day.toLowerCase());//to convert to lowerCase
* console.log(day.toUpperCase());//to convert to upperCase
* console.log(day.split('o'))//to split the given string based on the given value
* console.log(day.replace('o','a'))//to replace the given char to diff one
* console.log(day.trim())//to remove the spaces in front and last
* console.log(day.indexOf('a')); //to find the first occurance index of the given value
* console.log(day.indexOf(‘a’,3); // this will start from 3 index
* console.log(day.lastIndexOf('a')); //to find the last occurance index of the given value
* console.log(day.slice(0,2));//it will cut the string from the starting index till given //mo is the result
* let datenew =parseInt(newDate) - parseInt(oldDate);
* console.log(datenew); //to convert string to date
* console.log(datenew.toString()); //to convert number to string //'4'

40.Occurance of each character in a String ?

//2. Method

let str ="Hi this is Jagadish";

 let result=[...str].reduce(function(a,b){

a[b]=a[b]?a[b]+1:1;

return a;

 },{}

 )

 console.log(result)