Interview Questions

Java Script

#1 Question: Explain what is Javascript? List some data types supported by Javascript?

Javascript
Javascript is an object-oriented computer programming language commonly used to create interactive effects within web browsers.It is first used by the Netscape browser, that provides access to the HTML document object model (DOM), provides access to the browser object model (BOM). Javascript syntax looks a lot like java, c or c++ syntax.

Below is the list of data types supported by Javascript:-

- Undefined
- Null
- Boolean
- String
- Symbol
- Number
- Object

#2 Question: What close() does in Javascript?

In Javascript close() method is used to close the current window. You must write window.close() to ensure that this command is associated with a window object and not some other JavaScript object.

#3 Question: What is the difference between let and var?

Both var and let are used for variable/ method declaration in javascript but the main difference between let and var is that **var** is function scoped whereas **let** is block scoped.

#4 Question: Explain Closures in JavaScript?

Closures are the combination of lexical environment and function within which the function was declared. This allows JavaScript programmers to write better, more creative, concise and expressive codes. The closure will consist of all the local variables that were in-scope when the closure was created.

Sure, closures appear to be complex and beyond the scope, but after you read this article, closures will be much more easy to understand and more simple for your everyday JavaScript programming tasks. JavaScript is a very function-oriented language it gives the user freedom to use functions as the wish of the programmer.

#5 Question: Explain JavaScript Event Delegation Model?

In JavaScript, there is some cool stuff that makes it the best of all. One of them is Delegation Model. When capturing and bubbling, allow functions to implement one single handler to many elements at one particular time then that is called event delegation. Event delegation allows you to add event listeners to one parent instead of specified nodes. That particular listener analyzes bubbled events to find a match on the child elements. Many people think it to be complicated but in reality, it is very simple if one starts understanding it.

#6 Question: Describe negative infinity in JavaScript?

NEGATIVE_INFINITY property represents negative infinity and is a number in javascript, which is derived by 'dividing negative number by zero'. It can be better understood as a number that is lower than any other number. Its properties are as follows:
– A number of objects need not to be created to access this static property.
– The value of negative infinity is the same as the negative value of the infinity property of the global object.

The values behave differently than the mathematical infinity:

1. Any positive value, including POSITIVE_INFINITY, multiplied by NEGATIVE_INFINITY is NEGATIVE_INFINITY.
2. Any negative value, including NEGATIVE_INFINITY, multiplied by NEGATIVE_INFINITY is POSITIVE_INFINITY.
3. Zero multiplied by NEGATIVE_INFINITY is NaN.
4. NaN multiplied by NEGATIVE_INFINITY is NaN.
5. NEGATIVE_INFINITY, divided by any negative value except NEGATIVE_INFINITY, is POSITIVE_INFINITY.
6. NEGATIVE_INFINITY, divided by any positive value except POSITIVE_INFINITY, is NEGATIVE_INFINITY.
7. NEGATIVE_INFINITY, divided by either NEGATIVE_INFINITY or POSITIVE_INFINITY, is NaN.
8. Any number divided by NEGATIVE_INFINITY is zero.

#7 Question: Explain function hoisting in JavaScript?

JavaScript's default behavior that allows moving declarations to the top is called Hoisting. The 2 ways of creating functions in JavaScript are Function Declaration and Function Expression. Let's find out more about these:

Function Declaration

A function with the specific parameters is known as function declarations. To create a variable in JavaScript is called declarations.

e.g:

```
hoisted(); // logs "foo"

function hoisted() {

  console.log('foo');

}
```

Function Expression

When a function is created by using an expression it is called function expression.

e.g:

```
notHoisted(); // TypeError: notHoisted is not a function

var notHoisted = function() {

   console.log('bar');

};
```

#8 Question: What is the use of let & const?

In modern javascript let & const are different ways of creating variables.Earlier in javascript, we use the var keyword for creating variables. let & const keyword is introduced in version ES6 with the vision of creating two different types of variables in javascript one is immutable and other is mutable.
const: It is used to create an immutable variable. Immutable variables are variables whose value is never changed in the complete life cycle of the program.
let: let is used to create a mutable variable. Mutable variables are normal variables like var that can be changed any number of time.

#9 Question: Explain Arrow functions?

An arrow function is a consise and short way to write function expressions in Es6 or above.A rrow functions cannot be used as constructors and also does not supports this, arguments, super, or new.target keywords. It is best suited for non-method functions. In general an arrow function looks like `const function_name= ()=>{}`

```
const greet=()=>{console.log('hello');}
greet();
```

#10 Question: What are exports and imports?

Imports and exports help us to write modular javascript code. Using Imports and exports we can split our code into multiple files. Imports allow taking only some specific variables or methods of a file. We can import methods or variables that are exported by a module. See the below example for more detail.

```
//index.js

import name,age from './person';

console.log(name);
console.log(age);

//person.js

let name ='Sharad', occupation='developer', age =26;

export { name, age};
```

#11 Question: What is difference between module.exports and export?

The module is a plain JavaScript object with an exports property. Exports is a plain JavaScript variable that happens to be set to module.exports. At the end of your file, node.js will basically ‘return’ module.exports to the require function. A simplified way to view a JS file in Node could be this:

```
var module = { exports: {} };
var exports = module.exports;

// your code
return module.exports;
```

If you set a property on exports, like exports.a = 9;, that will set module.exports.a as well because objects are passed around as references in JavaScript, which means that if you set multiple variables to the same object, they are all the same object; so then exports and module.exports are the same objects.

But if you set exports to something new, it will no longer be set to module.exports, so exports and module.exports are no longer the same objects.

#12 Question: How to import all exports of a file as an object.

import * as object name from ‘./file.js’ is used to import all exported members as an object. You can simply access the exported variables or methods using dot (.) operator of the object.

Example:

```
objectname.member1;
objectname.member2;
objectname.memberfunc();
```

#13 Question: Explain “use strict” ?

“use strict” is a javascript directive that is introduced in Es5. The purpose of using “use strict” directive is to enforce the code is executed in strict mode. In strict mode we can’t use a variable without declaring it. “use strict” is ignored by earlier versions of Javascript..

#14 Question: In Javascript are calculations with fractional numbers guaranteed to be precise?

NO, calculations with fractional numbers are not guaranteed to be precise in Javascript

#15 Question: List the comparison operators supported by Javascript?

Javascript supports below comparison operators

- > Greater than
- < Less than
- <= Less than or equal to
- >= Greater than or equal to
- == Equal to
- != Not Equal to
- === Equal to with datatype check

#16 Question: What will happen if an infinite while loop is run in Javascript?

The program will crash the browser.

#17 Question: List HTML DOM mouse events?

HTML DOM mouse events

- onclick
- ondblclick
- mousemove

- mousedown
- mouseover
- mouseout
- mouseup

#18 Question: How to get the last index of a string in Javascript?

string.length-1 is used to get the last index of a string in Javascript

Example Usage:-

```
var myString="JavascriptQuestions";
console.log(myString.length-1);
```

#19 Question: Given the following line of code, what would be the output of the console?
var foo = new Array(3); console.log(foo);

Output: [undefined, undefined, undefined]

#20 Question: What is JavaScript Array's push(), pop(), shift() and unshift() methods?

1.1 The JavaScript Array.push() Method:

The JavaScript Array.push() method adds a new element to the end of the array. When doing so, the array's length property increases by one. After adding the new element to the end of the array, this method returns the new length of the array.

Example # 1

```
var days = ['mon','tues','wed'];

console.dir(days);

console.log( days.push('thurs') ); //4

console.dir(days);
```

1.2 The JavaScript Array.pop() Method:

The JavaScript Array.pop() method removes the last element from the end of the array. When doing so, the array's length property decreases by one. After removing the last element from the end of the array, this method returns the array element that was removed.

Example:

```
var days = ['mon','tues','wed'];

console.dir(days);

console.log( days.pop() ); //wed

console.dir(days);
```

1.3 The JavaScript Array.shift() Method:

The JavaScript Array.shift() method removes the first element from the beginning of the array. When doing so, the array's length property decreases by one. After removing the first element from the beginning of the array, this method returns the array element that was removed.

Example:

```
var days = ['mon','tues','wed'];

console.dir(days);

console.log( days.shift() ); //mon

console.dir(days);
```

1.4 The JavaScript Array.unshift() Method:

The JavaScript Array.unshift() method adds a new element to the beginning of the array. When doing so, the array's length property increases by one. After adding the new element to the beginning of the array, this method returns the new length of the array.

Example:

```
var days = ['mon','tues','wed'];

console.dir(days);

console.log( days.unshift('sun') ); //4

console.dir(days)
```

#21 Question: What are the primitive data types in JavaScript?

A primitive is a basic data type that’s not built out of other data types. It can only represent one single value. All primitives are built-in data types by necessity, (the compiler has to know about them,) but not all built-in data types are primitives.

In JavaScript there are 5 primitive data types are available they are **undefined**, **null**, **boolean**, **string** and **number** are available.Everything else in Javascript is an object.

#22 Question: Explain Event bubbling and Event Capturing in JavaScript?

Event Capture and Bubbling: In HTML DOM API there are two ways of event propagation and determines the order in which event will be received. The two ways are Event Bubbling and Event Capturing. The first method event bubbling directs the event to its intended target, and the second is called event capture in which the event goes down to the element.

#23 Question: What does the instanceof operator do?

In Javascript **instanceof** operator checks whether the object is an instance of a class or not:

Example Usage

```
Square.prototype = new Square();
console.log(sq instanceof Square); // true
```

#24 Question: What is Javascript BOM?

BOM stands for “Browser Object Modal” that allows Javascript to ‘talk’ to the browser, no standards, modern browsers implement similar BOMS – window, screen, location, history, navigator, timing, cookies.

#25 Question: What are different types of Popup boxes available in Javascript?

In Javascript there are 3 types of Popup Boxes are available, they are

- Alert
- Confirm
- Prompt

#26 Question: How can you create an array in Javascript?

There are 3 different ways to create an array in Javascript. They are

- By array literal
  usage:

  ```
   var myArray=[value1,value2...valueN];
  ```

- By creating instance of Array
  usage:

  ```
  var myArray=new Array();
  ```

- By using an Array constructor
  usage:

  ```
  var myArray=new Array('value1','value2',...,'valueN');
  ```

#27 Question: What is the 'Strict' mode in JavaScript and how can it be enabled?

Strict mode is a way to introduce better error-checking into your code. When you use strict mode, you cannot, for example, use implicitly declared variables, or assign a value to a read-only property, or add a property to an object that is not extensible.

You can enable strict mode by adding **"use strict"**; at the beginning of a file, a program, or a function. This kind of declaration is known as a directive prologue. The scope of a strict mode declaration depends on its context. If it is declared in a global context (outside the scope of a function), all the code in the program is in strict mode. If it is declared in a function, all the code in the function is in strict mode.

#28 Question: How to calculate Fibonacci numbers in JavaScript?

Calculating Fibonacci series in JavaScript
Fibonacci numbers are a sequence of numbers where each value is the sum of the previous two, starting with 0 and 1. The first few values are 0, 1, 1, 2, 3, 5, 8, 13 ,...,

```
function fib(n) {
        var a=0, b=1;
        for (var i=0; i < n; i++) {
                var temp = a+b;
                a = b;
                b = temp;
        }
        return a;
}
```

#29 Question: What is the difference between the substr() and substring() functions in JavaScript?

Difference between the substr() and substring() functions in JavaScript.

The substr() function has the form substr(startIndex,length). It returns the substring from startIndex and returns ‘length’ number of characters.

```
var s = "hello";
( s.substr(1,4) == "hello" ) // true
```

The substring() function has the form substring(startIndex,endIndex). It returns the substring from startIndex up to endIndex – 1.

```
var s = "hello";
( s.substring(1,4) == "ell" ) // true
```

#30 Question: What is output of undefined * 2 in Javascript?

NAN is output of undefined * 2.

#31 Question: Give the basic examples for Array REORDERING METHODS and MANIPULATION METHODS.

REORDERING METHODS: reverse() and sort()
Example:

```
var arr = [true, 2, 3, "4"];
new_array = arr.reverse()
console.log(new_array);        // ["4", 3, 2, true]

new_array = arr.sort();
console.log(new_array);        // [2, 3, "4", true]

var arr = [1, 2, 3, 4, 10];
console.log(arr.sort());        // [1, 10, 2, 3, 4]
console.log(arr.sort((el1, el2) => el1 - el2)); // [1, 2, 3, 4, 10]
```

MANIPULATION METHODS: concat(),slice(),splice()
Example:

```
var arr = [true, 2, 3, "4"];
new_array = arr.concat(5, 6, 7)
console.log(new_array);        // [true, 2, 3, "4", 5, 6, 7]

new_array = arr.slice(0, 2);
console.log(new_array);        // [true, 2]

console.log(arr);              // arr not changed - [true, 2, 3, "4"]
arr.splice(0, 1); // I replace the first item with 1
console.log(arr);               // [1, 2, 3, "4"]
```

#32 Question: How to add/remove properties to object dynamically in Javascript?

You can add a property to an object using object.property_name =value, delete object.property_name is used to delete a property.

Example:

```
   let user = new Object();
   // adding a property
   user.name='Anil';
   user.age  =25;
   console.log(user);
   delete user.age;
   console.log(user);
```

#33 Question: How to convert Javascript date to ISO standard?

toISOString() method is used to convert javascript date to ISO standard. It converts JavaScript Date object into a string, using the ISO standard.

Usage:

```
var date = new Date();
var n = date.toISOString();
console.log(n);
// YYYY-MM-DDTHH:mm:ss.sssZ
```

#34 Question: How to get inner Html of an element in JavaScript?

InnerHTML property of HTML DOM is used to get inner Html of an element in JavaScript.

Example Usage:

This is inner Element

```
<script type="text/javascript">
		var  inner= document.getElementById("inner").innerHTML ;
		console.log(inner); // This is inner Element
		document.getElementById("inner").innerHTML = "Html changed!";
		var  inner= document.getElementById("inner").innerHTML ;
		console.log(inner); // Html changed!
</script>
```

#35 Question: How to clone an object in Javascript?

Object.assign() method is used for cloning an object in Javascript.Here is sample usage

```
    var x = {myProp: "value"};
    var y = Object.assign({}, x);
```

#36 Question: List different ways of empty an array in Javascript?

In Javascript, there are many ways to empty an array in Javascript, below we have listed 4 major

By assigning an empty array.

```
var arr1 =[1,4,5,6];
arr1=[];
```

By assigning array length to 0.

```
var arr2 =[1,4,5,6];
arr2.length=0;
```

By poping the elements of the array.

```
var arr2 =[1,4,5,6];
while(arr.length > 0) {
   arr.pop();
}
```

By using .splice() .

```
var arr =[1,4,5,6];
arr.splice(0,arr.length)
```

How to get an element by class in JavaScript ?

document.getElementsByClassName() method is used in Javascript to get an element with a class name.

#37 Question: Explain Typecasting in Javascript?

In Programming whenever we need to convert a variable from one data type to another Typecasting is used. In Javascript, we can do this via library functions. There are basically 3 typecasts are available in Javascript Programming, they are:

- Boolean(value): Casts the inputted value to a Boolean

- Number(value): Casts the inputted value to an Integer or Floating point Number.
- String(value) : Casts the inputted value value a string

#38 Question: How to encode and decode a URL in JavaScript?

encodeURI() function is used to encode an URL in Javascript.It takes a url string as parameter and return encoded string. Note: encodeURI() did not encode characters like **/ ? : @ & = + $ #**, if you have to encode these characters too please use encodeURIComponent(). Usage:

```
var uri = "my profile.php?name=sammer&occupation=pāntiNG";
var encoded_uri = encodeURI(uri);
```

decodeURI() function is used to decode an URL in Javascript.It takes a encoded url string as parameter and return decoded string. Usage:

```
var uri = "my profile.php?name=sammer&occupation=pāntiNG";
var encoded_uri = encodeURI(uri);
decodeURI(encoded_uri);
```

#39 Question: How to you change the title of the page by JavaScript?

You can change the title of a webpage using setting the title property of the document object.

Example usage

```
document.title="My New Title";
```

#42 Question: List some Unit Testing Frameworks JavaScript

Below is the list of few most Popular Javascript Unit Testing Frameworks:

- Unit.js
- Jasmine
- Karma
- Chai
- AVA
- Mocha
- JSUnit
- QUnit
- Jest

#43 Question: How to add a new property in existing function JavaScript?

It is easy to add a new property in existing function by just giving value to the existing function it. For example, let we have an existing object person, to give new property check the below code:

person.country= “India”;

The new property “country” has added to the object person.

#47 Question: Explain few difference between null, undefined or undeclared JavaScript variable?

Null is a value that can be assigned to a variable or an object.

Undefined means a variable has been declared but no value is assigned to it. This type of variable is declared itself to be undefined.

Undeclared means the variable has declared without any datatype.

Null, Undefined are primitive data types whereas Undeclared is not a primitive data type.

#50 Question: Explain spread operator in JavaScript?

The spread operator expands an expression in places where multiple argument/variables/elements are needed to present. It represents with three dots (…).

For example:

```
var mid = [3, 4];

var newarray = [1, 2, ...mid, 5, 6];

console.log(newarray);

// [1, 2, 3, 4, 5, 6]
```

In above example, instead of appending mid array, it rather expands in the newarray with the help of spread operator. This is how spread operator works in JavaScript.

#51 Question: How to remove duplicates from JavaScript Array?

There are many ways to remove duplicates from JavaScript array. These are described below with examples:

1. By using Set: It is the simplest approach to remove duplicates. Set is an inbuilt object to store unique values in an array. Here's how we use set:

```
function uniquearray(array) {
  let unique_array= Array.from(set(array))
```

```
    return unique_array;}
```

As in the above code, you created a set of an array which automatically eliminates the duplicate values.

2. By using Filter: Another approach to remove duplicates from an array is applying filter on an array. To call filter method, it requires three arguments: array, current element, index of current element. Here’s how we use filter:

```
function unque_array (arr){
 let unique_array = arr.filter(function(elem, index, self) {
 return index == self.indexOf(elem); }
return unique_array }
 console.log(unique_array(array_with_duplicates));
```

3. By using for loop: In this, we can use for loop to remove duplicates. In this we make an empty array in which those elements will be added from the duplicate array which are not present in this before. Thus, finally we will get an array which has unique elements. Code to implement this:

```
Array dups_names = ['Ron', 'Pal', 'Fred', 'Rongo', 'Ron'];
function dups_array(dups_names) {
 let unique = {};
 names.forEach(function(i) {
    If (!unique[i]) {
      unique[i] = true;    }
  });
return Object.keys(unique);}   // Ron, Pal, Fred, Rongo
Dups_array(names);
```

These are the main three methods used in JavaScript to get a unique array.

#53 Question: Explain Promise in JavaScript?

A promise is an object in JavaScript which is used to produce a value that may give result in the future. The value can be resolved value or it can be a reason which tells why the value is not resolved.

A promise can be of three states:

- Fulfilled: The operation is completed and the promise has a specific value.
- Rejected: The operation is failed and promise has a reason which shows why the operation failed.
- Pending: Th operation is not fulfilled or rejected, means it has not completed yet.

#54 Question: What is difference between Array.splice() and Array.slice() method in JavaScript?

- The array.slice() removes items from the array and then return those removed items as an array whereas array.slice() method is selected items from an array and then those elements as a new array object.
- The splice() method affects the original array whereas slice() method doesn’t affect the original array.
- Splice() method takes n number of arguments whereas slice() can take only two arguments.

Syntax of splice(): array.splice(index, howmany, item1, ..., itemX)

Syntax of slice(): array.slice(start, end)

#55 Question: Is JavaScript multi-threaded or single-threaded?

JavaScript is single-threaded.

# Question: How to remove duplicate values from a JavaScript array?

We can use array.indexOf method to check a value exists or not. See below example to remove duplicate values.

```
let duplicates = ['delhi','kanpur','kanpur','goa','delhi','new york'];
```

```
function removeDuplicatesValues(arr){
   let unique_array = [];
   for(let i = 0;i < arr.length; i++){
      if(unique_array.indexOf(arr[i]) == -1){
         unique_array.push(arr[i])
      }
   }
   return unique_array
}

console.log(removeDuplicatesValues(duplicates));
```

# Question: Write a program to reverse a string in pure JavaScript?

There are many ways to reverse a string in JavaScript. These are:

Using in-built functions: the inbuilt function reverse() reverses the string directly. Here’ how:

```
str="jQuery";
str = str.split(""); //convert 'jQuery' to array
str = str.reverse(); //reverse 'jQuery' order
str = str.join(""); //then combines  the reverse order values.
alert(str);
```

First split the string to an array, then reverse an array and after that join the characters to form a string.

Using a loop: First, count a number of characters in a string, then apply a decrementing loop on an original string which starts from the last character and prints each character until count becomes zero.

# Question: Write program to remove duplicate in an array ?

There are multiple ways to remove duplicates from an array.

The simplest approach (in my opinion) is to use the Set object which lets you store **unique values** of any type. In other words, `Set` will automatically remove duplicates for us.

```
const names = ['John', 'Paul', 'George', 'Ringo', 'John'];

let unique = [...new Set(names)];
console.log(unique); // 'John', 'Paul', 'George', 'Ringo'
```

Another option is to use filter().

```
const names = ['John', 'Paul', 'George', 'Ringo', 'John'];

let x = (names) => names.filter((v,i) => names.indexOf(v) === i)
x(names); // 'John', 'Paul', 'George', 'Ringo'
```

And finally we can use `forEach()`.

```
const names = ['John', 'Paul', 'George', 'Ringo', 'John'];

function removeDups(names) {
  let unique = {};
  names.forEach(function(i) {
    if(!unique[i]) {
      unique[i] = true;
    }
  });
  return Object.keys(unique);
}

removeDups(names); // // 'John', 'Paul', 'George', 'Ringo'
```

# Question: What are the different types of errors available in JavaScript?

There are three types of errors available in JavaScript

- **Load time errors**: Errors which come up when loading a web page like improper syntax errors are known as Load-time errors and it generates the errors dynamically.
- **Run time errors**: Errors that come due to misuse of the command inside the HTML language.
- **Logical Errors**: These are the errors that occur due to the bad logic performed on a function which is having a different operation.

Recursion:

It is perfectly okay for a function to call itself, as long as it doesn't do it so often that it overflows the stack. A function that calls itself is called recursive.

Recursion allows some functions to be written in a different style. Take, for example, this alternative implementation of power:

```
function power(base, exponent) {

if (exponent == 0) {

return 1;

} else {

return base * power(base, exponent - 1);

}

}

console.log(power(2, 3));

// → 8
```

****
****

NodeJS

#1 Question: What is Node js ?

Node Js is one of the most popular and powerful server technologies today.
It allows you built the entire website only in one programming Language i.e Javascript. Node js is free and open source server technology that uses Javascript to create complete web software.It runs on various platforms like Windows, Linux, Unix, Mac OS X, etc.

#2 Question: Explain CLI.

CLI stands for Command Line Interface. It is a utility or program on your computer where users type commands to perform some action or run some script rather than clicking on the screen.
There are different types of command line interfaces depending on which operating system you are using. We have listed some of them below.

- Bash on Linux.
- Terminal of Mac.
- Command Prompt or Powershell on Windows
- Shell/ Command line/terminal on Unix and Ubuntu

#3 Question: In which Language Node Js is written ?

Node js is written in C, C++,JavaScript.It uses Google’s open source V8 Javascript Engine to convert Javascript code to C++.

#5 Question: Explain What is a Javascript Engine ?

A Javascript Engine is a program that converts code written in Javascript to something that computer processor understands.

#6 Question: Explain V8 Engine ?

V8 is Google’s open source high-performance JavaScript engine, written in C++ and used in Google Chrome, the open source browser from Google, and in Node.js, among others. It implements ECMAScript as specified in ECMA-262, and runs on Windows 7 or later, macOS 10.5+, and Linux systems that use IA-32, ARM, or MIPS processors. V8 can run standalone or can be embedded into any C++ application.

#7 Question: Explain ECMAScript ?

ECMAScript is the standard on which Javascript is based on. It was created to standardize Javascript. It is commonly used for client-side scripting on the World Wide Web and used by Node Js for writing server applications and services.

#8 Question: How can you check the installed version of Node Js ?

Use `node -v` command to check the installed version of Node Js.

#9 Question: Explain What is NPM ?

NPM stands for node package manager. It is default Package Manager for JavaScript programming language. NPM is used for installing/updating packages and modules of Javascript.

#10 Question: Explain Modules in Node Js ?

Modules are reusable block of code whose existence does not impact other code in any way. It is not supported by Javascript. Modules are introduced in ES6. Modules are important for Maintainability, Reusability, and Namespacing of Code.

#11 Question: What are CommonJs Modules ?

CommonJS Modules is the Standard how to code modules are structured. It specifies an ecosystem for JavaScript outside on the server or for native desktop applications.

#12 Question: For what require() is used in Node Js ?

require() is used to include modules from external files in Node Js. It is the easiest way to include a module in Node. Basically require is a function that takes a string parameter which contains the location of the file that you want to include. It reads the entire javascript file, executes the file, and then proceeds to return the exports object.
Syntax:

```
require('path');
```

#13 Question: Explain module.exports in Node Js ?

The method or variable defined in modules cannot be directly accessible by the outer world, that means you cannot call a module member from the external file. In order to access module member, we need to export the functions or variables of modules using module.exports method.

Syntax and usage:

```
// greet.js
var greet=function(){
console.log("hello World");
}
module.exports=greet;

//In app.js

var greet=require('./greet.js');
```

```
greet();
```

#14 Question: Is Node.js Single-threaded ?

Yes, Node Js is single threaded to perform asynchronous processing. Doing async processing on a single thread could provide more performance and scalability under typical web loads than the typical thread-based implementation.

#15 Question: What are events ?

An event is an action or occurrence recognized by software/app that is handled by event handler by writing a code that will be executed when the event fired.
Mouse move, Click, file copied or deleted are some examples of events.
In Node Js there are two types of events.
1)System Events: The event that comes from the C++ side.
2)Custom Events: Custom events are user-defined events.

#16 Question

Explain event loop in Node Js ?

In Node Js processes are single threaded, to supports concurrency it uses events and callbacks. An event loop is a mechanism that allows Node.js to perform non-blocking I/O operations.

#17 Question

How to create a simple server in Node js that returns Hello World ?

By writing following line of code, you can create a server in Node Js.

```
var http =require('http');
http.createServer(function(req,res){


res.writeHead(200,{'Content-Type':'text/plain'});
res.end('Hello World\n');


}).listen(1320,'127.0.0.3');
```

#22 Question

List some features of Express JS.

Some of the main features of Express JS are listed below: –

- It is used for setting up middlewares so as to provide a response to the HTTP or RESTful requests.
- With the help of express JS, the routing table can be defined for performing various HTTP operations.
- It is also used for dynamically rendering HTML pages which are based on passing arguments to the templates.
- It provides each and every feature which is provided by core Node JS.
- The performance of Express JS is adequate due to the presence of a thin layer prepared by the Express JS.
- It is used for organizing the web applications into the MVC architecture.
- Everything from routes to rendering view and performing HTTP requests can be managed by Express JS.

#23 Question: Write the steps for setting up an Express JS application.

Following are the steps used to set up an express JS application: –

1. A folder with the same name as the project name is created.
2. A file named package.json is created inside the folder created.
3. "npm install" command is run on the command prompt. It installs all the libraries present in package.json.
4. A file named server.js is created.
5. "Router" file is created inside the package which consists of a folder named index.js.
6. "App" is created inside the package which has the index.html file.

This way, an express JS application is set up.

#24 Question: What do you mean by Express JS?

Express JS is an application framework which is light-weighted node JS. A number of flexible, useful and important features are provided by this JavaScript framework for the development of mobile as well as web applications with the help of node JS.

#26 Question: What is the use of Express JS?

Express.js is a lightweight web application which helps in organizing the web application into MVC architecture on the server side.

#27 Question: What function are arguments available to Express JS route handlers?

The arguments which are available to an Express JS route handler-function are-

- Req – the request object
- Res – the response object
- Next (optional) – a function which is used to pass control to one of the subsequent route handlers.

The third argument is optional and may be omitted, but in some cases, it is useful where there is a chain of handlers and control can be passed to one of the subsequent route handlers skipping the current one.

#28 Question: How to config properties in Express JS?

In Express JS, there are two ways for configuring the properties:

- With process.ENV:
- A file with the name ".env" is to be created inside the project folder.
- All the properties are to be added in the ".env" file.
- Any of the properties can be used in server.js.
- With require JS:
- A file with the name "config.json" is to be created in the config folder inside the project folder.
- The config properties are to be added in the config.json file.
- Now, require should be used to access the config.json file.

#31 Question: Which template engine is supported by express JS?

Express JS supports any template engine that conforms to the (path, locals, callback) signature.

#32 Question: How can plain HTML be rendered in express JS?

There's no need to render HTML with the res.render () function. If there's a specific file, then you should use the res.sendFile () function. If any assets are being served from a dictionary, then express.static () middleware function needs to be used.

#33 Question: Why to use Express.js?

Below are the few reasons why to use Express with Node.js

- Express js is built on top of Node.js. It is the perfect framework for ultra-fast Input / Output.
- Cross Platform
- Support MVC Design pattern
- Support of NoSQL databases out of the box.

- Multiple templating engine support i.e. Jade or EJS which reduces the amount of HTML code you have to write for a page.
- Support Middleware, basic web-server creation, and easy routing tools.

#34 Question: Explain the difference between readFile and createReadStream in Node.js?

- readFile load the whole file which you had marked to read whereas createReadStream reads the complete file in the parts of the size you have declared.
- The client will receive the data faster in the case of createReadStream in contrast with readFile.
- In readFile, a file will first completely read by memory and then transfers to a client but in later option, a file will be read by memory in a part which is sent to clients and the process continue until all the parts finish.

#35 Question: List types of Http requests?

Http defines a set of request methods to perform the desired actions. These request methods are:

1. **GET**: The GET method asked for the representation of the specifies resource. This request used to retrieve the data.
2. **POST**: The POST technique is utilized to present an element to the predetermined resource, generally causing a change in state or reactions on the server.
3. **HEAD**: The HEAD method is similar to the GET method but asks for the response without the response body.
4. **PUT**: This method is used to substitute all current representations with the payload.
5. **DELETE**: It is used to delete the predetermined resource.
6. **CONNECT**: This request is used to settle the TCP/IP tunnel to the server by the target resource
7. **OPTION**: This method is used to give back the HTTP strategies to communicate with the target resource.
8. **TRACE**: This method echoes the message which tells the customer how much progressions have been made by an intermediate server.
9. **PATCH**: The PATCH method gives partial modifications to a resource.

#37 Question: How can you set default node version using nvm?

Run below command on the terminal to set default node version along multiple installed versions of node. You can list all install versions of the node by running `nvm ls`

```
nvm alias default v7.3.0
```

#38 Question: How to generate unique UUIDs/ guid in Node Js

Use node-uuid package to generate unique UUIDs/ guid in Node Js. Below code demonstrates how to generate it.

```
var uuid = require('node-uuid');

// Generate a v1 (time-based) id
uuid.v1();

// Generate a v4 (random) id
uuid.v4();
```

#40 Question: List the types of application you can build using Node Js ?

Using Node Js you can build applications like:

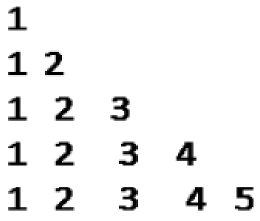
- Internet of Things
- Real-Time Chats Applications
- Complex Single-Page Applications
- Real-Time Collaboration Tools
- Streaming apps
- Microservices / API’s

#43 Question: Write a program to Print 0 to N element in pyramid shape?

```
function generatePyramid() {

  var totalNumberofRows = 5;

  var output="";

  for (var i = 1; i <= totalNumberofRows; i++) {

    for (var j = 1; j <= i; j++) {

      output+=j + "     ";

    }

    console.log(output);

    output="";
```

```
   }

}

generatePyramid();
```

Output:



```
1
1 2
1  2   3
1  2    3   4
1  2    3    4   5
```

ECMAScript 6 (ES-6)

#1 Question

What is ES6?

Es6 or ECMASCRIPT 2015 is sixth major release of ECMAScript language which comes with a lot of new features and syntax for writing web applications in Javascript. As currently, not all browsers support ES6, they support pre-versions of ES6. So to write web applications in ES6 that will support all Browsers we needed tools like Babel and Webpack.

#2 Question

List some new features of ES6

New Features in ES6.

1. Support for constants (also known as "immutable variables")
2. Block-Scope support for both variables, constants, functions
3. Arrow Functions
4. Extended Parameter Handling
5. Template Literals
6. Extended Literals
7. Enhanced Regular Expression
8. Enhanced Object Properties
9. Destructuring Assignment
10. Modules, Classes, Iterators, Generators
11. Support for Map/Set & WeakMap/WeakSet
12. Promises, Meta-Programming ,Internationalization & Localization

#3 Question

What is Babel?

Babel is one of the most popular JavaScript transpilers and becomes the industry standard. It allows us to write ES6 code and convert it back in pre-Es6 JavaScript that browser supports.

For example look the below code snippet.

In ES6 (ECMASCRIPT 2015)

```
const PI = 3.141593;
PI > 3.0 ;
export{PI};
```

In ES5 after conversion

```
"use strict";

Object.defineProperty(exports, "__esModule", {
  value: true
});
var PI = 3.141593;
PI > 3.0;
exports.PI = PI;
```

#4 Question

List steps to install Babel?

Installation: In order to install Babel, you require node.js and NPM. Make sure Node.js is installed on your server.

To check node installed or not run below commands on your terminal.

```
node -v
npm -v
```

Installing Babel : We can install Babel CLI locally by running below command on terminal.

```
npm install --save-dev babel-cli
```

#5 Question

What is Webpack?

Webpack allows you to run an environment that hosts babel. Webpack is opensource javascript module bundler which takes modules with dependencies and generates static assets representing those modules.

#6 Question

List benefits of using Webpack?

Benefits of using Webpack.

1. It bundles your multiple modules and packs it into a single .js file.
2. It comes with integrated dev server. A small express app for local development. You simply include one Javascript tag pointed to the server, like localhost:8080/assets/bundle.js, and get live code updating and asset management for free.

#7 Question

Explain Constants in Es6?

Constants also are known as immutable variables are a special type of variables whose content is not changed. In Es6 a constant is defined using const keyword. Constants in Es6 enable protection to overwrite a variable value, improve performance and helps programmers to write readable and cleaner code.

Example

In Es6

```
const WEBSITE_URL = "http://www.abc.com";
WEBSITE_URL="new url"; // generate an error;
console.log(WEBSITE_URL);
```

In prior version of Es6

```
//  and only in global context and not in a block scope

Object.defineProperty(typeof global === "object" ? global : window, "WEBSITE_URL", {
value: "http://www.abc.com", enumerable: true,
writable:     false,
configurable: false
});

console.log(WEBSITE_URL);
```

#8 Question

What are template literals in Es6?

Template literals are the string with embedded code and variables inside. Template literal allows concatenation and interpolation in much more comprehensive and clear in comparison with prior versions of ECMAScript.

Let see an example of concatenating a string in JavaScript.

```
var a="Hello";
var b="John";
var c = a+ " " + b;
Console.log(c); //outputs Hello John;
```

In ES6 concatenation and interpolation is done by backtick “ in a single line. To interpolate a variable simply put in to {} braces forwarded by $ sign.>/p>

// In ES6

```
let a="Hello";
let b="John";
let c=`${a} ${b}`;
console.log(c); //outputs Hello John;
```

#9 Question

What is Spread Operator in ES6?

Spread Operator provides a new way to manipulate array and objects in Es6.A Spread operator is represented by … followed by the variable name.

Example :

```
let a =[7,8,9];
let b=[1,2,3,...a,10];
console.log(b); // [1,2,3,7,8,9,10]
```

So spread operator spreads the contents of variable a and concatenates it in b.

Another Example

```
function print(...z){
        console.log(z);
}
```

```
print(1,2,3,4);//[1,2,3,4]
```

#10 Question

Explain Destructuring Assignment in ES6?

Destructing assignment in another improvement in Es6. It allows us to extract data from array and objects into separate variables.

Example

```
let full_name =['John','Deo'];


let [first_name,last_name]=full_name;


console.log(first_name,last_name);
// outputs John Deo
```

Another example

```
let c=[100,200,330,400];


let [a,...b]=c;


console.log(a,b);


// outputs 100 [200, 330, 400]
```

#11 Question

How to create a Javascript class in ES6?

In Es6 you can create a class using the Class keyword.Below is sample javascript class in ES6.

```
class User{
   constructor(name,age) {
      this.name  = name;
      this.age = age;
   }

   getData() {
      console.log(this.name + " is " + this.age + " years old !");
   }
}
```

```
var user = new User("foo", 7);
s1.getData();
```

****

React JS

#1 Question: What is ReactJS?

React js is javascript based UI Library developed at Facebook, to create an interactive, stateful & reusable UI components. It is one of the most popular javascript frameworks that is created for handling the presentation layer for the web and mobile apps.

#2 Question: List some advantages of ReactJS ?

Advantages of React Js

- **React.js is extremely efficient:** React.js creates its own virtual DOM where your components actually live. This approach gives you enormous flexibility and amazing gain in performance. React.js also calculates what are the changes needed to be made in DOM. This process of React.js avoids expensive DOM operations and make updates in a very client manner.
- **It makes writing Javascript easier:** React.js uses a special syntax called JSX, which allows you to mix HTML with Javascript. The user can drop a bit of HTML in the render function without having to concatenate strings, this is another fantastic thing. React.js turns those bits of HTML into functions with a special JSXTransformer.
- **It gives you out-of-the-box developer tools:** When you start your journey with React.js, do not forget to install official React.js chrome extension. It makes debugging your application much easier. After you install the extension, you will have a direct look into the virtual DOM as if you were browsing a regular DOM tree in the elements panel. Isn’t it pretty amazing!
- **It’s awesome for SEO:** One of the biggest problems with other Javascript frameworks is that they do not search engine friendly. Though there have been some improvements in this area, search engines generally have trouble reading Javascript heavy applications. React.js stands out from the crowd because you can run React.js on the server, and the virtual DOM will be rendered to the browser as a regular web page.
- **UI Test Cases:** It is extremely easy to write UI test cases because the virtual DOM system implemented entirely in JS

- **#3 Question:** What are Components in ReactJS?
- React Components let you split the UI into independent, reusable pieces, and think about each piece in isolation.
- Conceptually, components are like JavaScript functions. They accept arbitrary inputs (called “props”) and return React elements describing what should appear on the screen.
- Below is ample component written in ES6 class to display a welcome message on the screen.
- `class Welcome extends React.Component {`
- `  render() {`
- `    return <h1>Hello, {this.props.name} </h1>;`

```
•      }
•    }
•
•    const element = <Welcome name="Sara" />;
•    ReactDOM.render(
•      element,
•      document.getElementById('root')
•    );
```

#4 Question: What is JSX?

JSX is an XML/HTML-like syntax used by React that extends ECMAScript so that XML/HTML-like text can co-exist with JavaScript/React code. The syntax is intended to be used by preprocessors (i.e., transpilers like Babel) to transform HTML-like text found in JavaScript files into standard JavaScript objects that a JavaScript engine will parse.

Basically, by using JSX you can write concise HTML/XML-like structures (e.g., DOM like tree structures) in the same file as you write JavaScript code, then Babel will transform these expressions into actual JavaScript code. Unlike the past, instead of putting JavaScript into HTML, JSX allows us to put HTML into JavaScript.

By using JSX one can write the following JSX/JavaScript code:

```
var nav = (
   <ul id="nav">
     <li><a href="#">Home</li>
     <li><a href="#">About
<li><a href=”#”>Clients

<li><a href=”#”>Contact Us

</ul> );
```

And Babel will transform it into this:

```
var nav = React.createElement(
  "ul",
  { id: "nav" },
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "Home"
    )
  ),
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "About"
    )
  ),
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "Clients"
    )
  ),
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "Contact Us"
    )
  )
);
```

#5 Question: Explain life Cycle of React JS Component ?

React JS Component Lifecycle
Each component has several “lifecycle methods” that you can override to run code at particular times in the process. Methods prefixed with will are called right before something happens, and methods prefixed with did are called right after something happens.

Mounting

These methods are called when an instance of a component is being created and inserted into the DOM:

- constructor()
- componentWillMount()
- render()
- componentDidMount()

Updating

An update can be caused by changes to props or state. These methods are called when a component is being re-

rendered:

- componentWillReceiveProps()
- shouldComponentUpdate()
- componentWillUpdate()
- render()
- componentDidUpdate()

Unmounting

This method is called when a component is being removed from the DOM:

- componentWillUnmount()

Other APIs

Each component also provides some other APIs:

- setState()
- forceUpdate()

Class Properties

- defaultProps
- displayName

Instance Properties

- props
- state

#6 Question: List some features of ReactJS?

Undoubtedly today React is among of one the best JavaScript UI frameworks. It comes with a lot of features that helps programmers to create beautiful application easily, we have listed some of them below.

- It’s Adaptability
- Free and Open Source
- Decorators from ES7
- Server-side Communication
- Asynchronous Functions & Generators
- Flux Library
- Destructuring Assignments
- Usefulness of JSX

#7 Question: How to use Events in ReactJS?

Using Events is React js is very similar to handling event on DOM elements.The difference is only in syntax like.

1. The name of event is React js is always in camelCase.
2. With JSX you pass a function as the event handler, rather than a string.

Lets understand by an example:

```
// In HTML
<button onclick="activateAccount()">
  Activate Account
</button>
//In React
<button onClick={activateAccount}>
  Activate Account
</button>
```

#8 Question: What is flux in JavaScript?

Flux is an application architecture for creating data layers in JavaScript applications.It was designed at Facebook along with the React view library.Flux is not a framework or a library. It is simply a new kind of architecture that complements React and the concept of Unidirectional Data Flow.

#9 Question: What are refs in React?When to use it.

In React ref is used to store the reference of element or component returned by the component render() configuration function.Refs should be avoided in most cases, however, they can be useful when we need DOM measurements or to add methods to the components.

Refs can be used in the following cases

- Managing focus, text selection, or media playback.
- Triggering imperative animations.
- Integrating with third-party DOM libraries.

#10 Question: What are stateless components in React?

Stateless components are components that don’t have any state. When something is stateless, it calculates its internal state but it never directly mutates it.For creating a stateless components No class and this keyword is needed.You can create a stateless components using plain functions or Es6 arrow function. Below is an example of stateless component in react.

```
//In Es6
const Pane = (props) =>
{props.children}

;
```

```
//In Es5
const Username = ({ username }) =>
The logged in user is: {username}
```

#11 Question: What is the difference between State and props in ReactJs?

Props are shorthand for properties.they are very similar to an argument is passed to a pure javascript function. Props of the component are passed from parent component which invokes component. During a component’s life cycle props should not change consider them as immutable.In React all props can be accessible with this.props.

```
import React from 'react';
class Welcome extends React.Component {
  render() {
    return <h1>Hello {this.props.name}</h1>;
  }
}
const element = ;
```

State are used to create dynamic and interactive components in React.State is heart of react component that makes it alive and determines how a component renders & behaves.

```
// simple state example
import React from 'react';
class Button extends React.Component {
  constructor() {
    super();
    this.state = {
      count: 0,
    };
  }

  updateCount() {
    this.setState((prevState, props) => {
      return { count: prevState.count + 1 }
    });
  }

  render() {
    return (<button
              onClick={() => this.updateCount()}
            >
              Clicked {this.state.count} times
            </button>);
  }
}

export default Button;
```

#12 Question: What are Synthetic events?

SyntheticEvent is a cross-browser wrapper around browser’s native event. In React all of your event handlers will be passed instances of SyntheticEvent.The synthetic event works the same way as the event system of browsers, the only difference is that the same code will work across all browsers.

Below is a simple example of how to listen for a click event in react

```
import React, { Component } from 'react';

class ShowAlert extends Component {

  showAlert() {
    alert("Im an alert");
  }

  render() {
    return (
      <button onClick={this.showAlert}>show alert </button>
    );
  }
}

export default ShowAlert;
```

#13 Question: What is the difference between Dom and virtual Dom in React js?

DOM is the acronym for Document Object Model. Dom is also called HTML DOM as it is an abstraction of structured code called HTML for web developers. Dom and HTML code are interrelated as the elements of HTML are known as nodes of DOM. It defines a structure where users can create, alter, modify documents and the content present in it. So while HTML is a text, DOM is an in-memory representation of this text.

Virtual DOM is an abstraction of abstraction as it has been derived from HTML DOM. It is a representation of DOM objects like a lightweight copy. The virtual DOM was not invented by React, it is only used and provided for free.

#14 Question: Enlist the advantages and disadvantages of React js?

React.js is used by web developers for creating large web pages without reloading the entire page. It uses the data and can be changed over time.

The following are the advantages of using React.js-

1- React makes Search engine optimization (SEO) easy.

2- It is very efficient as it ensures readability and easy maintenance.

3- It gives extraordinary developer tools to web developers and makes Java coding easier for them.

4- UI test cases.

The following are the disadvantages of React-

1- Some major configurations are required for integrating React js with traditional MVC framework such as substituting erb with React js.

2- It is a steep learning process for people who are new to web development world.

#15 Question

What are the controlled components and uncontrolled components in React?

Controlled component is more advisable to use as it is easier to implement forms in it. In this, form data are handled by React components. A controlled input accepts values as props and callbacks to change that value.

The **uncontrolled component** is a substitute for controlled components. Here form data is handled by DOM itself. In uncomfortable components, the ref can be used to get the form values from DOM.

#16 Question: Explain the difference between functional and class components.

Functional components are those components that returns React elements as a result. These are just simple old JavaScript functions. React 0.14 has given a new shortcut for creating simple stateless components that are known as functional components. These components make use of easy JavaScript functions.

Class components – most of the tech-savvy people are more familiar with class components as they have been around the corner for a longer time. These components make use of plain old java objects for creating pages, mixins, etc in an identical way. Using React's create a class factory method, a literal is passed in defining the methods of a new component.

#17 Question: What do you understand by mixin or higher order components in React?

Higher order components (HOC) is a function that takes component as well as returns a component. It is a modern technique in React that reuses the component logic. However, Higher order components are not a part of React API, per se. These are patterns that emerge from React's compositional nature. In other words, HOC's are functions that loop over and applies a function to every element in an array.

#18 Question: How is flux different from redux?

Difference Between FLUX and Redux

FLUX	REDUX
Flux is a container for application state and logic that are registered to callbacks.	Redux is a container for JavaScript apps.
It is an observer pattern that is modified to fit React.	It offers interesting features such as writing applications, testing in different environments such as a server, client, etc.
It is a fancy name given to observer pattern and Facebook has developed tools to aid the implementation of these patterns.	In redux, actions can be functions or promises.
Flux supports actions that are simple JavaScript objects.	Redux is the first choice for web developers because it offers live code editing.

#19 Question: How is React different from angular and VUE js?

Angular Js – developed by Google, angular is a typescript based JavaScript application framework. It is also known as Super-heroic JavaScript MVW Framework. It was developed with the motive to encounter the challenges of creating single page applications. There are several versions of angular such as Angular 2+, Angular 2 or ng2. Angular is the rewritten, mostly incompatible successor to AngularJS which means AngularJS is the oldest framework.

React– React was developed by Facebook in March 2013. It is a JavaScript library that is used for building user interfaces. React creates large web applications and also provides speed, scalability, and simplicity.

Vue Js- Launched in February 2014, Vue is the most famous and rapidly growing framework in JS field. Vue is an intuitive, fast and composable MVVM for building interactive interfaces. It is extremely adaptable and several JavaScript libraries make use of this. Vue is also a web application framework that helps in making advanced single page applications.

#20 Question: What is the use of arrow function in React?

Arrow functions are extremely important for React operations. It prevents this bugs. Arrow functions make it easier to predict the behavior of this bugs when passed as callbacks. They don't redefine the value of this within their function body. Hence, prevents bugs caused by the use of this within callbacks.

#21 Question: What are refs in React?

Refs are used for managing focus, selecting text and triggering animations. It also integrates with third-party libraries. Refs help in getting the reference to a DOM (Document Object Model) node. They return the node that we are referencing. Whenever we need DOM measurements, we can use refs.

#22 Question: What is the purpose of render() function in React?

render() function is used to update the UI. For this, you have to create a new element and send it to ReactDOM.render(). React elements are immutable and once you create an element, you cannot change its attributes. Thus, elements are like a single frame and it depicts the UI at some point. ReactDOM.render() controls the content of the container node you pass and if there is any DOM element already present in it then it would be replaced when first called.

Why to use React js?

- Reusable components, using mixins(that allow overlap only on lifecycle methods, and have a predictable execution order).
- Hierarchical components(Components that appear within another component in the mock should appear as a child in the hierarchy).
- Uses inheritance

- High Cohesion, Loose coupling
- Virtual, shadow DOM
- It's SEO friendly - React can run on the server and virtual DOM render data and return the regular web page.
- Easy but Powerful javascript library
- Tools & Extensions

****

REACT NATIVE

#1 Question: Explain React Native?

React Native is mobile app development platform that allow you to create android and ios apps only using Javascript. React Native uses the same fundamental UI building blocks as regular iOS and Android apps so applications created using React Native are indistinguishable from an app built using Objective-C or Java.

#2 Question: Can we combine native ios or android code in React Native.

Yes, we can. React Native smoothly combines the components written in Objective-C, Java, or Swift.

#3 Question: Who uses React Native?

Thousands of apps are created using React Native. Here is the list of some popular companies who are using React Native.

- Facebook
- Facebook Ads Manager
- Instagram
- F8
- Airbnb
- Skype
- Tesla

#4 Question: List Step to Create and start a React Native App?

Run the following command to Create and start a React Native App.

- Step 1: npm install -g create-react-native-app // Installs create native app
- Step 2: create-react-native-app AwesomeProject // Create a project named AwesomeProject
- Step 3: cd AwesomeProject
- Step 4: npm start

#5 Question: What does StyleSheet.create do?

StyleSheet.create method ensures that values are immutable and opaque, they are also only created once.

#6 Question: For what XHR Module is used in React Native?

XHR Module is used for implementation of XMLHttpRequest to post data on the server.

#7 Question: List some core components of React Native?

Text, Image, View, TextInput, ListView are some core components of React Native.

#8 Question: How Virtual DOM works in React Native?

React creates an in-memory data structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently. This allows developers to write code as if the entire page is rendered on each change while the React libraries only render subcomponents that actually change.

****

MONGO-DB

#1 Question: What do you know about MongoDB?

MongoDB is a cross-platform document-oriented database program which is open source and free in nature. It can also be classified as NoSQL database program. It was developed by MongoDB Inc. JSON-like documents with schema are used by MongoDB. Field, range queries, and regular expressions are also supported by MongoDB. The queries used by this contain user-defined JavaScript functions. It provides high availability with the replica sets. Fields can be indexed with primary and secondary indices. Sharding is used by MongoDB for scaling horizontally which determines how the data will be distributed. It can also be used as a file system with load balancing. It can also be used for batch processing of data.

#2 Question: List the important features of MongoDB.

The important features of MongoDB are listed below: –

1. **Aggregation framework**– it uses aggregation framework for the batch processing of data and aggregation operations. It should be used in an effective manner.
2. **The usage of BSON format in MongoDB**– it uses the binary-encoded serialization of JSON-like documents. The data-types like- date and binary are supported in the format.
3. **The sharing feature of MongoDB**– using this feature, MongoDB supports the distribution of data across multiple machines. It supports deployment with large sets of data.
4. **The Ad hoc queries in MongoDB**– it supports field, range queries, and regular expressions.

5. **Collections**– MongoDB supports fixed size collection which can also be termed as capped collections.

#3 Question: Which all languages can be used with MongoDB?

Here goes a list of the languages which can be used with MongoDB: –

- C
- C++
- C#
- Java
- Node.js
- Perl
- PHP
- Python
- Ruby
- Scala
- Go
- Erlang.

Currently, MongoDB provides driver support for the languages listed above. It can be used easily with any of these languages.

#4 Question: What is the use of a namespace in MongoDB?

In MongoDB, a namespace is a combination of the database name and collection or index name. It is a canonical name for an index or a collection in MongoDB. A namespace consists of all the documents in MongoDB. The maximum length of a collection of namespaces is called the namespace length. It consists of the database name, a dot operator (.), and the name of the collection.

Syntax- <database>.<collection>

The namespace contains all the helper classes which can be used to construct various options which are to be used in the drivers. A namespace can also be termed as the concatenation of the collection name and the database name.

#5 Question: What do you mean by a replica set in MongoDB?

A group of mongo instances which is able to host the same data set is known as a replica set in MongoDB. A replica set consists of a primary node and a secondary node too. With the help of a replica set, all the data from primary node to the secondary node replicates. Replication is a process of synchronizing the data. Replication provides redundancy and it also increases the availability of data with the help of multiple copies of data on the different database server. It also protects the database from the loss of a single server.

#6 Question: What should all points be taken into consideration while creating a schema in MongoDB?

Following are the points which should be taken into consideration when you create a schema in MongoDB: –

- Your schema should be designed according to your requirements.
- If the objects are to be used together, you should combine them into one document. Otherwise, you should use them separately.
- Joins should be performed while writing and not while reading.
- If use cases are used more frequently, then you should optimize your schema.
- Complex aggregation should be done in the schema.

#7 Question: What do you understand by a profiler with reference to MongoDB?

A database profiler is something which collects fine-grained data about write operations of MongoDB, the cursors, and the database commands. Profiling can be enabled on a per database or pre instance basis. When profiling is enabled, the profiling level also gets configurable. By default, the profiler will be off. All the data collected by the system.profiler collection is written by the database profiler to it. This collection is a capped collection. 0, 1, and 2, these are the three levels available in the database profiler. Profiling in a database can be enabled from the mongo shell or using a profile command through the driver. While enabling profiling, profiling level is also set. The profiler is used to record data in the system.profile collection.

#8 Question: Write the syntax for creating a collection and dropping a collection in MongoDB.

Creation of the collection is the basic step in MongoDB. Here is the syntax for creating a collection in MongoDB-

```
Db.createCollection (name, options)
```

Dropping a collection basically means deleting a collection. Following is the syntax for dropping a collection in MongoDB-

```
Db.collection.drop()
```

#9 Question

What are the different types of NoSQL databases? Give some examples.

There are 4 basic types of NoSQL database. They are as follows: –

- Key value store NoSQL database
- Document store NoSQL database
- Column store NoSQL database
- Graph-based NoSQL database

The various examples of NoSQL database are- MongoDB, Cassandra, CouchDB, Hypertable, Redis, Riak, Neo4j, HBase, Couchbase, MemcacheDB, Voldemort, RevenDB, etc.

#10 Question: What are the differences between MongoDB and MySQL?

The various differences between MongoDB and MySQL are listed below:

- **In terms of data representation**– in MySQL, we represent data in the form of tables and rows. Whereas, in MongoDB data is represented as collections of JSON documents.
- **Querying**– in SQL, we put together a string in the query language which is then parsed by the database system. Whereas, in MongoDB, object querying is used.
- **Relationships**– in MySQL, the relational database is only the Join operation which allows us to perform queries across multiple tables. Whereas MongoDB does not support join operation but can support multi-dimensional data types such as arrays, etc. in this, embedding is a process in which we place one document inside the other.
- **Transactions**– MySQL supports atomic transactions which are the ability to contain multiple operations within a transaction. Whereas, MongoDB does not support transactions.
- **Schema definition**– in MySQL, you need to define your tables and columns before storing anything. Whereas, in MongoDB, you don't need to define a schema.

#14 Question: What is the "__v" field in Mongoose?

When the mongoose is created at the first time the version key is a property set on every document. The value of this key comprises of the internal revision of the document. It is understood that the name of this document is configurable. The default key is __v

****

JSON

#1 Question: What are different ways to create objects?

You can create Object by

- object literals
- Object.create
- constructors

#2 Question: What is the default value of a constructor's prototype?

A plain, empty object that derives from Object.prototype is the default value of a constructor's prototype

#3 Question: List some benefits of JSON over XML?

- It is faster and lighter than XML as on the wire data format
- XML data is typeless while JSON objects are typed

- JSON types: Number, Array, Boolean, String
- XML data are all string
- Data is readily available as JSON object is in your JavaScript
- Fetching values is as simple as reading from an object property in your JavaScript code

#4 Question: What is the difference between JSON and JSONP?

- **JSON:** JSON is a simple data format used for communication medium between different systems
- **JSONP:** It is a methodology for using that format with cross-domain ajax requests while not being affected by same origin policy issue.

#5 Question: Who is the Father of JSON and What is the scripting language JSON is based on?

Douglas Crockford called as the Father of JSON. JSON is based on **ECMAScript**.

#6 Question: What is JSON-RPC? List some Features of JSON-RPC-Java

JSON-RPC: JSON-RPC is a simple remote procedure call protocol similar to XML-RPC although it uses the lightweight JSON format instead of XML.

JSON-RPC-Java is a Java implementation of the JSON-RPC protocol.Below is list of some of its features

- Dynamically call server-side Java methods from JavaScript DHTML web applications. No Page reloading.
- Asynchronous communications.
- Transparently maps Java objects to JavaScript objects.
- Lightweight protocol similar to XML-RPC although much faster.
- Leverages J2EE security model with session specific exporting of objects.
- Supports Internet Explorer, Mozilla, Firefox, Safari, Opera, and Konqueror.

#7 Question: What are natively supported JSON types?

Following data types are natively supported in JSON.

- **Numbers**: Integer, float or Double
- **String**: string of Unicode characters, must be rapped into double quotes “”
- **Boolean**: True or false
- **Array**: ordered list of 0 or more values
- **Objects** : An unordered collection key/ value pairs
- **Null**: An Empty value

#8 Question: What is BSON?

BSON is the superset of JSON, which used by MongoDB.BSON supports the embedding of documents and arrays within other documents and arrays. BSON also contains extensions that allow representation of data types that are not part of the JSON spec.

#9 Question: How to convert an Object into JSON? What is the full syntax of JSON.stringify?

JSON.stringify method is used to convert an Javascript Object into JSON.
Syntax:

```
let json = JSON.stringify(value[, replacer, space])
```

#10 Question: What JS-specific properties are skipped by JSON.stringify method?

Following JS-specific properties are skipped by JSON.stringify method

- Function properties (methods).
- Symbolic properties.
- Properties that store undefined.

#11 Question: What is JSON? For what is used for?

JSON (JavaScript Object Notation) is a data storage and communication format based on key-value pair of JavaScript object literals. It is a lightweight text-based open standard designed for human-readable data interchange which is derived from the JavaScript programming language for representing simple data structures and associative arrays, called objects.

In JSON

- all property names are surrounded by double quotes.
- values are restricted to simple data: no function calls, variables, comments, or computations.

JSON is used for communication between javascript and serverside technologies.

#12 Question: How to convert Javascript objects into JSON?

JSON.stringify(value); is used to convert Javascript objects into JSON.

Example Usage:

```
  var obj={"website":"Onlineinterviewquestions"};
  JSON.stringify(obj); // '{"website":"Onlineinterviewquestions"}'
```

### . #13 Question

List types Natively supported by JSON?

JSON supports Objects, Arrays, Primitives (strings, numbers, boolean values (true/false), null) data types.

#14 Question: What does Object.create do?

Object.create creates a new object with the specified prototype object and properties.

#15 Question: What does hasOwnProperty method do?

It returns **true** if the property was set on an actual object rather than inherited.

#16 Question: What does $.parseJSON() do ?

$.parseJSON() takes a well-formed **JSON** string and returns the resulting JavaScript value.

#17 Question: How do you decode a JSON string?

Use JSON.parse method to decode a JSON string into a Javascript object.

#18 Question: How to delete an index from JSON Obj?

Deleting an Element from JSON Obj

```
var exjson = {'key':'value'};
delete exjson['key'];
```

****

AJAX

#1 Question: Brief about Ajax.

Ajax is considered to be a developer’s best friend because it offers several benefits to him such as updating the web page without reloading the entire page, requesting data from the server and also sending data to the server. AJAX is the acronym for Asynchronous JavaScript and XML. It is a new technique to communicate to and from a server/ database without completely refreshing the page. It creates faster, interactive, better web applications with help of CSS, XML, HTML, and JavaScript.

#2 Question: Differentiate between Synchronous and Asynchronous Ajax requests.

Synchronous Ajax requests: In this, the script stops and waits for the server to reply before continuing. In the web application world, one has to happen after the other, i.e. the interaction between the customer and the server is synchronous. Synchronous is not recommended as it blocks/hangs the page until the response is received from the server.

Asynchronous Ajax requests handle the reply as and when it comes and allows the page to continue to be processed. Under Asynchronous, if there is any problem in the request it can be modified and recovered. The request doesn’t block the client as the browser is responsive. The user can perform other operations as well.

#3 Question: List some advantages and disadvantages of using Ajax.

Ajax is a very easy concept if one has a sound knowledge of JavaScript. It uses JavaScript functions to call methods from a web service. It has certain advantages-

- **Speed**- Ajax reduces the server traffic and also the time consumption on the server and client side.
- Ajax is very responsive and fast, data can be transferred at a time.
- XMLHttpRequest plays a significant role in Ajax. It is a special JavaScript object that calls asynchronous HTTP request to the server for transferring data.
- One of the biggest advantages of using Ajax as forms are common elements in the web page. Ajax gives options for validation and much more.
- One doesn't have to completely reload the page.

There are some disadvantages attached to Ajax. They are-

- Search engines would not be able to index Ajax applications so Ajax maybe a mistake.
- Anyone can have access to the code of Ajax and can view source it.
- ActiveX requests are enabled only in internet explorer and other new browsers.

#4 Question: What are different readyState in Ajax.

There are total 5 ready state in Ajax:

Value	State	Description
0	UNSET	Client has been created. Open() not yet called.
1	OPENED	Open() called.
2	HEADRERS_RECIEVED	Send() called and headers are available.
3	LOADING	Downloading: responseText holds partial data.
4	DONE	The operation is complete.

#5 Question: What is XMLHttpRequest object in Ajax? How can you XMLHttpRequest Object?

The XMLHttpRequest objects are used to exchange data with a server. It is an API whose methods transfer between a web browser and a web server. In XHR, it's not necessary that data have to be in form of XML. It can be JSON or HTML. XHR can be used with protocols other than HTTP.

XMLHttpRequest is any developers kit because it has the option to update the page without reloading the entire page. You can request data from the server and also send data to the server in the background.

#6 Question: How to cancel the current request in Ajax?

Current request in AJAX is cancelled when the user performs an action which sets of an Ajax request. This can be depicted with the help of auto-complete functionality for a search box where users can be helped with related search terms based on their current input, by making an AJAX request each time they pass a key in search field. The user types faster than the Ajax request and you would want to abort any non-finished requests, before starting the next one.

#7 Question: How to send an Ajax request in JavaScript?

HTTP Requests are created with the help of XMLHttpRequest objects. It facilitates the transfer of data between client and server which happens via request and response. In XMLHttpRequest, you can perform the same function plus you can grab data from URL without having to refresh the page. AJAX lets you perform actions without reloading the entire page. The following steps tell how to call AJAX-

- Create an XMLHttpRequest object.
- Open the request with open method.
- Now, send the request with the send method.

```
var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
   document.getElementById("demo").innerHTML = this.responseText;
  }
 };
 xhttp.open("GET", "get_results.php", true);
 xhttp.send();
```

#8 Question: Explain Fetch API in JavaScript.

Fetch API uses to request and response objects that can be used in future whenever needed. It provides an interface for fetching resources. Fetch API uses promises that enable cleaner API. Cache API or other similar things handles the request and responses that might require you to generate your own responses programmatically.

Thus, you can now crack your interview and work as a developer in one of your dream companies by preparing from these set of question and answers to be technically sound.

#9 Question: Explain limitations of Ajax.

1) Browser Integration

The dynamically created page does not register itself with the browser history engine, so triggering the "Back" function of the users' browser might not bring the desired result.

2) Response-time Concerns

Network latency – or the interval between the user request and server response – need to be considered carefully during Ajax development.

3) Search Engine Optimization (SEO)

Websites that use Ajax to load data which should be indexed by search engines must be careful to provide equivalent Sitemaps data at a public, linked URL that the search engine can read, as search engines do not generally execute the JavaScript code required for Ajax functionality.

4) Reliance on JavaScript

Ajax relies on JavaScript, which is often implemented differently by different browsers or versions of a particular browser. Because of this, sites that use JavaScript may need to be tested in multiple browsers to check for compatibility issues.

#10 Question: Explain what is polling in AJAX.

The Process of retrieving data from a server to obtain near-live data regularly is called **AJAX polling**.

****

jQuery

#1 Question: What is jQuery?

jQuery is a lightweight JavaScript library which gives a quick and simple method for HTML DOM traversing and manipulation, its event handling, its client-side animations, and so on. One of the best features of jQuery is that jQuery supports an efficient way to implement AJAX applications because of its lightweight nature and make normalize and efficient web programs.

#2 Question: How is body onload() function is different from document.ready() function used in jQuery?

Document.ready() function is different from body onload() function because off 2 reasons.

- We can have more than one document.ready() function in a page where we can have only one onload function.
- Document.ready() function is called as soon as DOM is loaded where body.onload() function is called when everything gets loaded on the page that includes DOM, images and all associated resources of the page.

#3 Question: What are features of JQuery or what can be done using JQuery?

Features of Jquery

- One can easily provide effects and can do animations.
- Applying / Changing CSS.
- Cool plugins.
- Ajax support
- DOM selection events
- Event Handling

#4 Question: What are the different type of selectors in Jquery?

There are 3 types of selectors in Jquery

- CSS Selector
- XPath Selector
- Custom Selector

#5 Question: What are the advantages of JQuery ?

There are many advantages of JQuery. Some of them are :

- It is more like a JavaScript enhancement so there is no overhead in learning a new syntax.
- It has the ability to keep the code simple, readable, clear and reusable.
- Cross-browser support (IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+)
- It would eradicate the requirement for writing complex loops and DOM scripting library calls.
- Event detection and handling.
- Tons of plug-ins for all kind of needs.

#6 Question: Why jQuery?

jQuery is very compact and well-written JavaScript code that increases the productivity of the developer by enabling them to achieve critical UI functionality by writing very less amount of code.

It helps to

- Improve the performance of the application
- Develop most browser compatible web page
- Implement UI related critical functionality without writing hundreds of lines of codes
- Fast
- Extensible – jQuery can be extended to implement customized behavior

Other advantages of jQuery are

- No need to learn fresh new syntax’s to use jQuery, knowing simple JavaScript syntax is enough
- Simple and Cleaner code, no need to write several lines of codes to achieve complex functionality.

#7 Question: What is jQuery Selectors? Give some examples

- jQuery Selectors are used to select one or a group of HTML elements from your web page.
- jQuery support all the CSS selectors as well as many additional custom selectors.
- jQuery selectors always start with dollar sign and parentheses: $()

There are three building blocks to select the elements in a web document.

1. Select elements by tag name

Example

$(div)

It will select all the div elements in the document.

2. Select elements by ID

Example

$(“#abc”)

It will select single element that has an ID of abc.

3. Select elements by Class

Example

$(“.xyzClass”)

It will select all the elements having class xyzClass.

#8 Question: Explain width() vs css(‘width’)

In jQuery, there is two way to change the width of an element. One way is using .css(‘width’) and other way is using .width().

For example

```
$('#mydiv').css('width','300px');
```

```
$('#mydiv').width(100);
```

- The difference in .css('width') and .width() is the data type of value we specify or return from the both functions.
- In .css('width') we have to add "px" in the width value while in .width() we don't have to add.
- When you want to get the width of "mydiv" element then .css('width') will return '300px' while .width() will return only integer value 300.

#9 Question: Explain bind() vs live() vs delegate() methods.

The bind() method will not attach events to those elements which are added after DOM is loaded while **live()** and **delegate()** methods attach events to the future elements also.

The difference between **live()** and **delegate()** methods is **live()** function will not work in chaining. It will work only on an selector or an element while **delegate()** method can work in chaining.

For example

```
$(document).ready(function(){
$("#myTable").find("tr").live("click",function(){
alert($(this).text());
});
});
```

Above code will not work using live() method. But using delegate() method we can accomplish this.

```
$(document).ready(function(){
$("#dvContainer")children("table").delegate("tr","click",function(){
alert($(this).text());
});
});
```

#10 Question: What is the use of param() method.

The param() method is used to represent an array or an object in serialize manner.

While making an ajax request we can use these serialize values in the query strings of URL.

Syntax:

```
$.param(object | array, boolValue)
```

"object | array" specifies an array or an object to be serialized.

"boolValue" specifies whether to use the traditional style of param serialization or not.

Example

```
personObj=new Object();
empObject.name="Ravi";
empObject.age="28";
empObject.dept="IT";
$("#clickme").click(function(){
$("span").text($.param(empObject));
});
```

It will set the text of span to “name=Ravi&age=28&dep=IT”

#11 Question: What is the difference between jquery.size() and jquery.length?

jQuery **.size()** method returns number of element in the object. But it is not preferred to use the size() method as jQuery provide **.length** property and which does the same thing. But the .length property is preferred because it does not have the overhead of a function call.

#12 Question: How to read, write and delete cookies in jQuery ?

To deal with cookies in jQuery we have to use the Dough cookie plugin.

Dough is easy to use and having powerful features.

- **Create cookie:**
  $.dough(“cookie_name”, “cookie_value”);
- **Read Cookie:**
  $.dough(“cookie_name”);
- **Delete cookie:**
  $.dough(“cookie_name”, “remove”);

#13 Question: What is difference between $(this) and this in jQuery ?

```
$(document).ready(function(){
$('#clickme').click(function(){
 alert($(this).text());
 alert(this.innerText);
});
});
```

this and **$(this)** references the same element but the difference is that “this” is used in traditional way but when “this” is used with $() then it becomes a jQuery object on which we can use the functions of jQuery.?

In the example given, when only “this” keyword is used then we can use the jQuery text() function to get the text of the element, because it is not jQuery object. Once the “this” keyword is wrapped in $() then we can use the jQuery function text() to get the text of the element.

#14 Question: What are the various ajax functions ?

Ajax allows the user to exchange data with a server and update parts of a page without reloading the entire page. Some of the functions of ajax are as follows:

$.ajax(): This is considered to be the most low level and basic of functions. It is used to send requests . This function can be performed without a selector.

$.ajaxSetup(): This function is used to define and set the options for various ajax calls.

For example.

```
$.ajaxSetup({
"type":"POST",
"url":"ajax.php",
"success":function(data){
$("#bar")
.css("background","yellow")
.html(data);
}
});
```

Shorthand ajax methods: They comprise of simply the wrapper function that call $.ajax() with certain parameters already set.

$.getJSON(): this is a special type of shorthand function which is used to accept the url to which the requests are sent. Also optional data and optional callback functions are possible in such functions.

#15 Question: Explain .empty() vs .remove() vs .detach().

- **.empty()** method is used to remove all the child elements from matched elements.
- **.remove()** method is used to remove all the matched element. This method will remove all the jQuery data associated with the matched element.
- **.detach()** method is same as .remove() method except that the .detach() method doesn’t remove jQuery data associated with the matched elements.

.remove() is faster than .empty() or .detach() method.

Syntax:

```
$(selector).empty();
$(selector).remove();
$(selector).detach();
```

#16 Question: How can events be prevented from stopping to work after an ajax request?

There are two ways to handle this issue:

Use of event delegation: The event delegation technique works on principle by exploiting the event bubbling. It uses event bubbling to capture the events on elements which are present anywhere in the domain object model. In jquery the user can make use of the live and die methods for the events delegation which contains a subset of event types.

For example. handling even delegation, handling of clicks on any <a> element:

```
$('#mydiv').click(function(e){
if( $(e.target).is('a') )
fn.call(e.target,e);
});
$('#mydiv').load('my.html')
```

Event rebinding usage: When this method is used it requires the user to call the bind method and the added new elements.

```
$('a').click(fn);
$('#mydiv').load('my.html',function(){
$('#mydiv a').click(fn);
});
```