1. **What is JavaScript?**

JavaScript is a scripting language. It is different from Java language. It is object-based, lightweight, cross-platform translated language. It is widely used for client-side validation. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

1. **What is DOM? What is the use of document object?**

DOM is an abbreviation for Document Object Model. The HTML document is represented by a document object. It can be used to access and modify HTML content.

1. **What is the difference between == and ===?**

The == operator only checks for equality, whereas the === operator checks for equality and data type, i.e., a value must be of the same type.

1. **What is the difference between client-side and server-side JavaScript?**

Client-side JavaScript consists of the basic language and predefined objects required to run JavaScript in a browser. Client-side JavaScript is directly embedded in HTML pages. At runtime, the browser interprets this script.

Server-side JavaScript also resembles client-side JavaScript. It contains relevant JavaScript that will be executed on a server. Only after compilation is the server-side JavaScript deployed.

1. **What does a strict mode do in JavaScript?**

Silent errors are generated by the JavaScript strict mode. It includes the "use strict"; expression, which enables the strict mode. Only the first statement in a script or function can contain this expression. For instance,

"use strict";

x=10;

console.log(x);

It will show “ReferenceError: x is not defined**”** because of “use strict”

1. **What is JavaScript Hoisting?**

It is a JavaScript's default behaviour, which is used to move all the declarations at the top of the scope before the execution of code. It can be applied to functions as well as on variables. It allows the JavaScript to use the component before its declaration. It does not apply to scripts that run in strict mode.

1. **What’s variable scope in JavaScript?**

Scope in JavaScript refers to the current context of code, which determines the accessibility of variables to JavaScript. The two types of scope are local and global:

* Global variables are those declared outside of a block
* Local variables are those declared inside of a block

1. **List some of the advantages and disadvantages of JavaScript.**

**Some advantages of JavaScript are:**

* Server interaction is less
* Feedback to the visitors is immediate
* Interactivity is high
* Interfaces are richer

**Some Disadvantages of JavaScript are:**

* No support for multithreading
* No support for multiprocessing
* Reading and writing of files is not allowed
* No support for networking applications.

1. **What do you understand by Callback and Callback hell in JavaScript?**

**Callback:** It is used to handle the execution of function after the completion of the execution of another function. A callback would be helpful in working with events. In the callback, a function can be passed as an argument to another function. It is a great way when we are dealing with basic cases such as minimal asynchronous operations.

**Callback hell:** When we develop a web application that includes a lot of code, then working with callback is messy. This excessive Callback nesting is often referred to as Callback hell.

1. **What’s back-end developers’ job focus?**

A backend developer's job entails focusing on website architecture, scripting, and writing code that communicates between the website’s database and the user's browser. Data consistency and integrity must be ensured.

1. **What’s backend developers’ task?**

Back-end developers are required to have technical expertise, analytical thinking, and excellent collaboration skills. As a back-end web developer, you should be able to work independently to design the web infrastructure.

Here’s what many back-end developers do on a day-to-day basis:

* **Build and maintain websites:**A back-end developer’s main responsibility is to use various tools, frameworks, and languages to determine how best to develop intuitive, user-friendly prototypes and turn them into websites. This requires an understanding of cross-platform functionality and compatibility.
* **Write high-quality code:**To produce sustainable web applications, developers must write clean and easily maintainable code.
* **Perform quality assurance (QA) testing:**Create and oversee testing schedules to optimize user interface and experience, ensuring optimal display on various browsers and devices.
* **Assess efficiency and speed:**Once a website is up and running, and during updates and edits, developers need to assess its performance and scalability, adjusting code as necessary.
* **Troubleshoot and debug:**Be able to troubleshoot issues and resolve them, while communicating them to project managers, stakeholders, and QA teams.
* **Train and support:**Maintain workflows with client teams to ensure ongoing support, along with leading training and mentorship for junior developers.

1. **Define Software Design**

Software design refers to the process of creating a specification of software artifact which will help to developers to implement the software. It is about designing individual modules/components means it defines what is module is performing, the classes, functions and their usages, etc. It is considered one of the initial phases of Software Development Life Cycle (SDLC). Software design process establishes a best possible design plan to implement the system. It mainly focuses more on internal components of the system and their interaction with each other.

1. **What’s the Difference between Software Design and Software Architecture?**

|  |  |
| --- | --- |
| **Software Design** | **Software Architecture** |
| Software design is about designing individual modules/components. | Software architecture is about the complete architecture of the overall system. |
| Software design defines the detailed properties. | Software architecture defines the fundamental properties. |
| In general, it refers to the process of creating a specification of software artifact which will help to developers to implement the software. | In general, it refers to the process of creating high level structure of a software system. |
| It helps to implement the software. | It helps to define the high-level infrastructure of the software. |
| It is considered as one initial phase of Software Development Cycle (SSDLC) and it gives detailed idea to developers to implement consistent software. | It is a plan which constrains software design to avoid known mistakes and it achieves one organizations business and technology strategy. |

1. **What’s debugging and when debugging is needed?**

* **Debugging:** Debugging is the process of finding and resolving defects or problems within a computer program that prevent correct operation of computer software or a system.
* **Need for debugging:** Once errors are known during a program code, it’s necessary to initial establish the precise program statements liable for the errors and so to repair them.

1. **What is dry code?**

There’s a principle in programming called DRY, or Don’t Repeat Yourself. It usually means refactoring code by taking something done several times and turning it into a loop or a function. DRY code is easy to change, because you only have to make any change in one place.

1. **What does MERN stands for?**

MERN is one of several variations of the MEAN stack (MongoDB Express Angular Node), where the traditional Angular.js front-end framework is replaced with React.js. Other variants include MEVN (MongoDB, Express, Vue, Node), and really any front-end JavaScript framework can work.

1. **What is Express JS?**

Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a “fast, unopinionated, minimalist web framework for Node.js,”. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

1. **What is node JS?**

Node.js is a server-side platform based on Chrome's JavaScript runtime that allows for the rapid development of fast and scalable network applications. It is a non-blocking, asynchronous I/O runtime based on Google's V8 JavaScript engine and the libuv library.

1. **What’s the difference between frontend and backend development?**

Frontend refers to the client-side of an application. It’s the part of a web application that users can see and interact with. On the other hand, Backend refers to the server-side of an application. It constitutes everything that happens behind the scenes.

1. **What is NPM?**

NPM stands for Node Package Manager. NPM is responsible for managing all the packages and modules for Node.js. It provides online repositories for node.js packages/modules. It also provides command line utility to install Node.js packages.

1. **What are modules in Node.js?**

Modules are like JavaScript libraries that can be used in a Node.jS application to include a set of functions. A module is included in a Node.JS application by using require () function with the parenthesis containing the name of the module.

1. **Why is Node.js being preferred over other backend technologies like Java and PHP?**

Node.js is really fast. Node Package Manager has over 50,000 bundles for at the developers' disposal. Perfect for data intensive, real-time web applications as Node.js never waits for an API to return data. Better synchronization of code between server and client due to same code base. Easy for web developers to start using Node.js in their projects as it is a JavaScript library.

1. **Which database is typically used with Node.js?**

MongoDB is the most popularly used database used with Nodejs, it is a NoSQL, cross-platform, document-oriented database that provides, high performance, high availability, and easy scalability.

1. **What do you understand by the term I/O?**

The term I/O stands for input and output. It is used to access anything outside of your application. The I/O describes any program, operation, or device that transfers data to or from a medium or another medium. This medium can be a physical device, network, or files within a system.

I/O is loaded into the machine memory to run the program once the application starts.

1. **How many types of API functions are available in Node.js?**

here are two types of API functions in Node.js:

Asynchronous, Non-blocking functions

Synchronous, Blocking functions

1. **In which types of applications is Node.js most frequently used?**

Node.js is most frequently and widely used in the following applications:

* Internet of Things
* Real-time collaboration tools
* Real-time chats
* Complex SPAs (Single-Page Applications)
* Streaming applications
* Microservices architecture etc.

1. **What are the pros and cons of Node.js?**

|  |  |
| --- | --- |
| Pros | Cons |
| Fast processing and an event-based model | Not suitable for heavy computational tasks |
| Uses JavaScript which is known by many developers | Using callback is complex since you end up with many nested callbacks |
| Node Package Manager has over 50,000 packages that provide functionality to an application | Dealing with relational databases is not a good option for Nodejs |
| Best suited for streaming huge  amounts of data and 1/0 intensive operations | Since NodeJS is single-threaded, CPU intensive tasks is not its strong suit |

1. **What is the command used to import external libraries?**

Command require is used for importing external libraries, for example, “var http=require (“http”)”. This will load the http library and the single exported object through the http variable

1. **What are the modules in Node.js? Which are the different modules used in Node.js?**

In Node.js applications, modules are like JavaScript libraries and include a set of functions. To include a module in a Node.js application, we must use the require () function with the parentheses containing the module's name.

Node.js has several modules which are used to provide the basic functionality needed for a web application. Following is a list of some of them:

|  |  |
| --- | --- |
| **Core Modules** | **Description** |
| HTTP: | The HTTP module includes classes, methods, and events to create a Node.js HTTP server. |
| util: | The util module includes utility functions required in the application and is very useful for developers. |
| url: | The url module is used to include the methods for URL parsing. |
| fs: | The fs module includes events, classes, and methods to handle the file I/O operations. |
| stream: | The stream module is used to include the methods to handle streaming data. |
| query string | The stream module is used to include the methods to handle streaming data. |
| zlib: | The zlib module is used to include the methods to compress or decompress the files used in an application. |

1. **What are buffers in Node.js?**

In general, a buffer is a temporary memory mainly used by the stream to hold on to some data until it is consumed. Buffers are used to represent a fixed-size chunk of memory allocated outside of the V8 JavaScript engine. It can't be resized. It is like an array of integers, which each represents a byte of data. It is implemented by the Node. js Buffer class. Buffers also support legacy encodings like ASCII, utf-8, etc.

1. **What is an asynchronous API?**

All the API's of Node.js library are asynchronous means non-blocking. A Node.js based server never waits for an API to return data. The Node.js server moves to the next API after calling it, and a notification mechanism of Events of Node.js responds to the server for the previous API call.

1. **What is error-first callback?**

Error-first callbacks are used to pass errors and data. If something goes wrong, the programmer has to check the first argument because it is always an error argument. Additional arguments are used to pass data.

fs.readFile(filePath, function(err, data) {

    if (err) {

      //handle the error

    }

    // use the data object

  });

1. **Define ECMAScript. What is ES6?**

The ECMA-262 standard defines the specification for creating a general-purpose scripting language.

ES6 was released in June 2015, marking the language's sixth edition. It was initially known as ES6 before being renamed ECMAScript 2015. Modules, iterators, classes, arrow functions, for...of loops, promises, and many other new features are included in this edition. Brendan Eich created it.

1. **What are the new features introduced in ES6?**

The new features that are introduced in ES6 are listed as follows:

* Let and const keywords.
* Default Parameters.
* Arrow functions.
* Template Literals.
* Object Literals.
* Rest and spread operators.
* Destructuring assignment.
* Modules, Classes, Generators, and iterators.
* Promises, and many more.

1. **Define let and const keywords.**

let: The variables declared using let keyword will be mutable, i.e., the values of the variable can be changed. It is similar to var keyword except that it provides block scoping.

const: The variables declared using the const keyword are immutable and block-scoped. The value of the variables cannot be changed or re-assigned if they are declared by using the const keyword.

1. **What do you mean by IIFE?**

IIFE stands for Immediately invoked function expressions. IIFE is a function in JavaScript that runs as soon as it is defined. It is also called as the Self-Executing Anonymous Function. It includes two major parts that are as follows:

* The first part is an anonymous function that has a lexical scope (static scope), which is enclosed within the Grouping operator ().
* The second part creates the IIFE by which the JavaScript engine will interpret the function directly.

1. **What is Map in ES6?**

Prior to ES6, we would frequently use an object to map keys and values. Map objects are a new collection type introduced in ES6. It stores key-value pairs where any type of value can be used as either a key or a value. A map object always remembers the order in which the keys were inserted. Maps are ordered, so they traverse the elements in the order in which they were inserted.

1. **Describe the Promises in ES6.**

ES6 promises are the easiest way to work with asynchronous programming in JavaScript. Asynchronous programming includes running of processes individually from the main thread, and it notifies the main thread when it gets complete. Prior to ES6, there is the use of Callbacks for performing asynchronous programming. Promises are used to overcome the problem of Callback hell.

1. **What’s ternary operation?**

The conditional (ternary) operator is the only JavaScript operator that takes three operands: a condition followed by a question mark (?), then an expression to execute if the condition is truthy followed by a colon (:), and finally the expression to execute if the condition is false. This operator is frequently used as an alternative to an if...else statement.

Syntax: condition ? exprIfTrue : exprIfFalse

1. **What do you understand by NoSQL databases? Is MongoDB a NoSQL database?**

At the present time, the internet is loaded with big data, big users, big complexity etc. and also becoming more complex day by day. NoSQL is answer of all these problems, It is not a traditional database management system, not even a relational database management system (RDBMS). NoSQL stands for "Not Only SQL". NoSQL is a type of database that can handle and sort all type of unstructured, messy and complicated data. It is just a new way to think about the database.

Yes. MongoDB is a NoSQL database.

1. **Is MongoDB better than other SQL databases? If yes then how?**

MongoDB is better than other SQL databases because it allows a highly flexible and scalable document structure.

For example:

* One data document in MongoDB can have five columns and the other one in the same collection can have ten columns.
* MongoDB database are faster than SQL databases due to efficient indexing and storage techniques.

1. **Explain the structure of ObjectID in MongoDB.**

ObjectID is a 12-byte BSON type. These are:

* 4bytes value representing seconds
* 3 bytes machine identifier
* 2 bytes process id
* 3 bytes counter

1. **What’s JSON? What type of data are supported by JSON?**

JSON stands for JavaScript object notation. JSON has been derived from JavaScript, where JavaScript is a programming language. It was originally created to hold the structured data that could be used in JavaScript. JSON became so popular that it is used for data for all kinds of applications. It is the most popular way of sending the data for Web APIs.

Basic data types supported by JSON are:

* Strings: Characters that are enclosed in single or double quotation marks.
* Number: A number could be integer or decimal, positive or negative.
* Booleans: The Boolean value could be either true or false without any quotation marks.
* Null: Here, null means nothing without any quotation marks.

1. **Why do we use JSON?**

Since JSON is an easy-to-use, lightweight language data interchange format in comparison to other available options, it can be used for API integration. Following are the advantages of JSON:

* Less Verbose: In contrast to XML, JSON follows a compact style to improve its users' readability. While working with a complex system, JSON tends to make substantial enhancements.
* Faster: The JSON parsing process is faster than that of the XML because the DOM manipulation library in XML requires extra memory for handling large XML files. However, JSON requires less data that ultimately results in reducing the cost and increasing the parsing speed.
* Readable: The JSON structure is easily readable and straightforward. Regardless of the programming language that you are using, you can easily map the domain objects.
* Structured Data: In JSON, a map data structure is used, whereas XML follows a tree structure. The key-value pairs limit the task but facilitate the predictive and easily understandable model.

1. **Do the MongoDB databases have schema?**

Yes. MongoDB databases have dynamic schema. There is no need to define the structure to create collections.

1. **What is CRUD in MongoDB?**

MongoDB supports following CRUD operations:

* Create
* Read
* Update
* Delete

1. **What is a storage engine in MongoDB? Which are the storage engines used by MongoDB?**

A storage engine is the part of a database that is used to manage how data is stored on disk.

For example: one storage engine might offer better performance for read-heavy workloads, and another might support a higher-throughput for write operations.

MMAPv1 and WiredTiger are two storage engines used by MongoDB.

1. **Which architectural designs are mostly used to design applications?**

In software design, we use the following architectural design patterns:

* Model View Controller
* Master-Slave Pattern
* Layered Pattern
* Model View Presenter
* Monolithic Architecture
* Event-Driven Architecture Pattern

1. **What is MVC?**

MVC stands for Model View Controller. It splits an application into three logical components i.e., Model, View, and Controller. It separates the business-specific logic (Model component) from the presentation layer (View component) from each other.

1. **What is RESTful API?**

The term REST stands for Representational State Transfer. It is an architectural style that is used to create Web Services. It uses HTTP requests to access and use the data. We can create, update, read, and delete data. An API (Application Program Interface) for a website is the code that allows two software programs to communicate with each other. It allows us to write requesting services from an operating system or other application.

1. **What is middleware in Node.js?**

The middleware in node.js is a function that will have all the access for requesting an object, responding to an object, and moving to the next middleware function in the application request-response cycle. This function can be used for modifying the req and res objects for tasks like adding response headers, parsing requesting bodies, and so on.

1. **Why is middleware needed?**

Middleware is required for helping developers in building applications more effectively and efficiently. The middleware will act as a connection between data, applications, and the users. When you are an organization with a multi-cloud environment, then middleware will make it more cost-effective for the development and running of the application at scale.