1. **Explain the use of JavaScript?**

* JavaScript is a light-weight object-oriented programming language that is used by several websites for scripting the webpages.
* JavaScript enables dynamic interactivity on websites when it is applied to an HTML document.
* It helps the users to build modern web applications to interact directly without reloading the page every time.
* JavaScript is commonly used to dynamically modify HTML and CSS to update a user interface by the DOM API.
* It is mainly used in web applications.

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

**Client Side :**

Client-side code is code that is run on the user's computer. when a web page is viewed, the page's client-side code is downloaded, then run and displayed by the browser. In this module we are explicitly talking about client-side JavaScript.

**Server Side :**

Server-side code on the other hand is run on the server, then its results are downloaded and displayed in the browser. Examples of popular server-side web languages include PHP, Python, Ruby, ASP.NET and JavaScript. JavaScript can also be used as a server-side language, for example in the popular Node.js environment

1. **What is Nodejs?**

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project.

Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.

A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

1. **Explain Scope in JavaScript?**

The context in which values and expressions are visible or can be referenced. If a variable or other expression is not in the current scope, then it is unavailable for use. Scopes can also be layered in a hierarchy, so that child scopes have access to parent scopes, but not vice versa.

A function serves as a closure in JavaScript, and thus creates a scope, so that (for example) a variable defined exclusively within the function cannot be accessed from outside the function or within other functions.

**For Example :**

function data(){

var x = “X declared inside a function”;

console.log(“Inside a Function”);

console.log(x);

}

console.log(x)

1. **JavaScript is asynchronous or synchronous.**

JavaScript is a synchronous, blocking, single-threaded language. That just means that only one operation can be in progress at a time.

* The JavaScript engine uses the stack data structure to keep track of currently executed functions. The stack is called the function execution stack.
* The function execution stack executes the functions sequentially, line-by-line, one-by-one.
* The browser/web APIs use callback functions to complete the tasks when an asynchronous operation/delay is done. The callback function is placed in the callback queue.
* The promise executor functions are placed in the job queue.
* For each loop of the event loop, one task queue is completed out of the callback queue.
* Once that task is complete, the event loop visits the job queue. It completes all the micro-tasks in the job queue before it looks for the next thing.
* If both the queues get entries at the same point in time, the job queue gets preference over the callback queue.

1. **JavaScript is Single-threaded or Multi-threaded.**

JavaScript is a single-threaded language, which means it has only one call stack that is used to execute the program. The call stack is the same as the stack data structure. As we know stacks are FILO that is First In Last Out. Similarly, within the call stack, whenever a line of code gets inside the call stack it gets executed and move out of the stack. In this way, JavaScript is a single-thread language because of only one call stack.

JavaScript is a single-threaded language because while running code on a single thread, it can be really easy to implement as we don’t have to deal with the complicated scenarios that arise in the multi-threaded environment like deadlock.

1. **Explain DOM in your own word.**

The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects, that way, programming languages can interact with the page.

A web page is a document that can be either displayed in the browser window or as the HTML source. In both cases, it is the same document but the Document Object Model (DOM) representation allows it to be manipulated. As an object-oriented representation of the web page, it can be modified with a scripting language such as JavaScript.