Question 1: Explain the use of JavaScript (or What you can do using a JavaScript)

Answer: JavaScript is a programming language used for website & software development. Here are the following uses of JavaScript.

i.	Used to provide interactivity in web pages.
ii.	Used to build web and mobile application.
iii.	With NodeJS, JavaScript is used to build server. Node is used fetch data and files from server with the help of JavaScript.
iv.	Use to develop desktop applications.
v.	Also used in cloud environments.

Question 2: What is the difference between client-side and server-side?

Answer: The difference between client-side and server-side is:

Client-side	Server-side
Client-side refers to the side, from where the user requests for any resource. Most of the time browser is the client.	Server-side refers to the system from where user request is served. These servers may be local or on network/remote.
Most of the time end-user interact with client-side with the help of interface of browser, Postman(rest-API) etc.	Server is kept hidden from user because it is used to just serve the request.
Examples: User types the URL in browser: https://learn.ineuron.ai/your-collections	Example: your collection will be fetched from server and presented to you if you are logged in.

Question 3: What is Nodejs?

Answer: Node.js is an open-source and cross-platform JavaScript runtime environment. Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. With the help of Nodejs we can use JavaScript to make server to fetch resources from system.

Question 4: Explain Scope in JavaScript

Answer: Scope refers to the area or place from where something can be accessed. In JavaScript there are some rules by which we can surround or limit the area for variables, functions and objects. For example: variable defined with "let" can accessed inside block where it was defined but not outside of block.

Question 5: JavaScript is asynchronous or synchronous.

Answer: JavaScript is a synchronous. But with the help of some programming techniques like async await or promises we can make it behave like synchronous.

Question 6: JavaScript is Single-threaded or multi-threaded.

Answer: JavaScript is single-threaded language. That just means that only one operation can be in progress at a time. But with the help of worker threads inside JS engine it can be make or behave like multithreaded so that it can serve multiple requests at same time.

Question 7: Explain DOM in your own word.

Answer: DOM is Document Object Model. The web page generally refers to Document & this document is made up of some HTML tags (objects). These HTML tags forms a tree like model which is called as DOM.
Example:

```
                        ┌──────────────┐
                        │   Document   │
                        └──────────────┘
                                │
                        ┌──────────────┐
                        │ Root element:│
                        │    <html>    │
                        └──────────────┘
                                │
            ┌───────────────────┴───────────────────┐
    ┌──────────────┐                         ┌──────────────┐
    │  Element:    │                         │  Element:    │
    │   <head>     │                         │   <body>     │
    └──────────────┘                         └──────────────┘
            │                                        │
    ┌──────────────┐   ┌──────────────┐   ┌──────────────┐   ┌──────────────┐
    │  Element:    │   │  Attribute:  │   │  Element:    │   │  Element:    │
    │   <title>    │───│    "href"    │   │    <a>       │   │    <h1>      │
    └──────────────┘   └──────────────┘   └──────────────┘   └──────────────┘
            │                                    │                   │
    ┌──────────────┐                     ┌──────────────┐   ┌──────────────┐
    │   Text:      │                     │   Text:      │   │   Text:      │
    │  "My title"  │                     │  "My link"   │   │ "My header"  │
    └──────────────┘                     └──────────────┘   └──────────────┘
```