

1.What is client-side and server-side in web development, and what is the main difference between the two?

Answer: Client-side: Client-side refers to the part of the web application that runs on the user's device, typically a web browser. It consists of HTML, CSS and JavaScript code that is downloaded from the server and executed on the client's machine. The client-side code is responsible for the user interface, handling user interactions, and performing tasks directly on the user's device.

Server-side: Server side refers to the part of the web application that runs on the server. It consists of server-side programming language (such as PHP, Python, Ruby etc) and frameworks that handle the processing of data, logic and interactions with databases or other external services.

Main differences: The main differences between client-side and server-side lies in where the code is executed. Client side code runs on the user's device, while server-side code runs on the server. This difference has implications for their responsibilities and capabilities.

2.What is an HTTP request and what are the different types of HTTP requests?

Answer: An HTTP request is a message sent by a client to a server in the context of the Hypertext Transfer Protocol (HTTP). It is used to initiate a communication and retrieve or send information over the web. The different types of HTTP requests are

1.GET, 2. POST, 3.Delete, 4.PUT, 5.PATCH, 6. HEAD 7.OPTIONS, 8.TRACE.

3.What is JSON and what is it commonly used for in web development?

Ans: JSON (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate. It is based on a subset of the JavaScript programming language.

In web development, JSON is commonly used for the following purposes:

1.Data interchange, 2.AJAX requests. 3.Configuration files. 4.API 5.Storage and persistence.

4.What is a middleware in web development, and give an example of how it can be used?

Answer: In web development,middleware refers to software components or functions that sit between the application and the server.It acts as a bridge ,facilitating communication and handling various tasks during the processing of a request or response.

Lets consider an Express.js application, a popular web framework for Node.js.Express.js allows developers to define middleware functions using the `app.use()` method.Here's an example of a custom middleware function that logs incoming requests.

```
Const express = require('express')
Const app = express();

Const loggerMiddleWare = (req,res,next) => {

console.log('This is a middleware')

next()
}

app.use(loggerMiddleWare)
```

In this example ,we define a custom middleware function name `loggerMiddleWare`.We register this middleware globally ,meaning it will be executed for every incoming request before the corresponding route handler is invoked.

5.What is a controller in web development, and what is its role in the MVC architecture?

Answer: In web development, a controller is a component or module responsible for handling user requests,processing input and determining the appropriate response in the Model-View-Controller(MVC) architectural pattern.

In the MVC pattern, the controller acts as the intermediary between the user interface (view) and the data model. Its primary role is to receive user input from the view, interpret it, and update the model accordingly. It also handles the logic of how the model should be modified based on user actions and triggers updates to the view.