1. HTTP 1.1 VS HTTP 2

Answer:

HTTP/1.1 and HTTP/2 are different versions of the HTTP protocol used for communication between web browsers and web servers. Here are some key differences between the two:

1. Multiplexing

HTTP/1.1 uses a single connection for each request, which means that requests have to wait for previous responses to be completed before they can be sent. In contrast, HTTP/2 supports multiplexing, allowing multiple requests and responses to be processed simultaneously over a single connection, reducing latency and improving performance.

1. Header Compression

HTTP/1.1 sends headers as plain text with each request and response, leading to increased overhead and larger data transfers. HTTP/2 uses header compression techniques like Huffman coding, which significantly reduces the size of headers and improves efficiency.

3. Server Push

HTTP/2 introduces server push, where the server can proactively send resources to the client before they are explicitly requested. This can help optimize page loading times and improve overall performance.

4. Binary Protocol

HTTP/1.1 uses a text-based protocol, which can be inefficient when parsing and generating messages. HTTP/2 uses a binary protocol, making it more compact and easier for machines to process.

5. Security

Both HTTP/1.1 and HTTP/2 can be used over encrypted connections (HTTPS), but HTTP/2 encourages the use of encryption by making it mandatory in some modern browsers.

Overall, HTTP/2 offers significant improvements over HTTP/1.1 in terms of performance, efficiency, and security, making it the preferred choice for modern web applications

2. object and its internal representation in Javascipt

Answer:

Objects, in JavaScript, is it’s most important data-type and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data-types(Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive data-types all store a single value each (depending on their types).

Objects are more complex and each object may contain any combination of these primitive data-types as well as reference data-types.  
An object, is a reference data type. Variables that are assigned a reference value are given a reference or a pointer to that value. That reference or pointer points to the location in memory where the object is stored. The variables don’t actually store the value.

Loosely speaking, objects in JavaScript may be defined as an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

For Eg. If your object is a student, it will have properties like name, age, address, id, and methods like

3. IP ADDRESS ,

4. PORT ,

5. HTTP Method ,

6. MAC ADDRESS

Answer:

3. An IP address, short for Internet Protocol address, is an identifying number for network [hardware](https://www.lifewire.com/computer-hardware-2625895) connected to a network. Having an [IP address](https://www.lifewire.com/what-is-the-ip-address-of-facebook-818152) allows a device to communicate with other devices over an IP-based network like the internet.

|  |  |
| --- | --- |
| 1 | **GET**  The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data. |
| 2 | **HEAD**  Same as GET, but transfers the status line and header section only. |
| 3 | **POST**  A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms. |
| 4 | **PUT**  Replaces all current representations of the target resource with the uploaded content. |
| 5 | **DELETE**  Removes all current representations of the target resource given by a URI. |
| 6 | **CONNECT**  Establishes a tunnel to the server identified by a given URI. |
| 7 | **OPTIONS**  Describes the communication options for the target resource. |
| 8 | **TRACE**  Performs a message loop-back test along the path to the target resource. |

4. A port is a point on a computer where information exchange between multiple programs and the internet to devices or other computers takes place. To ensure consistency and simplify programming processes, ports are assigned port numbers. This, in conjunction with an IP address, forms vital information that each internet service provider (ISP) uses to fulfill requests.

5.

6.

MAC addresses are associated with specific devices and assigned to them by the manufacturer. Wi-Fi, Bluetooth, and Ethernet connections all use MAC addresses. MAC addresses work with the card in your device that lets it connect wirelessly to the internet, called a Network Interface Controller (NIC).