JavaScript - Day -1: Introduction to Browser & web

1. Difference between HTTP1.1 vs HTTP2.

HTTP/1.1 and HTTP/2 are two different versions of the Hypertext Transfer Protocol, which is the protocol used for transferring data over the World Wide Web.

HTTP/1.1 uses a **text-based protocol**, which is human-readable but less efficient for machines to process.
HTTP/2 uses **a binary protocol,** which is more efficient for both machines and browsers to parse. This can lead to faster and more reliable communication.

HTTP/1.1, each request and **response is handled sequentially.** This means that if you have multiple resources to request (e.g., images, stylesheets, scripts) on a web page, they are fetched one at a time, leading to potential bottlenecks and slower page load times.
HTTP/2 introduces multiplexing, which **allows multiple requests and responses to be sent** and received in parallel over a single TCP connection. This greatly improves the efficiency and speed of data transfer, reducing latency and improving page load times.

HTTP/1.1: Each HTTP request and response includes **headers that can be quite large, especially for complex web applications.** In HTTP/1.1, these headers are not compressed, which can lead to increased overhead and slower data transfer.
HTTP/2: HTTP/2 **employs header compression techniques (HPACK) to reduce the size of headers**, making more efficient use of network resources and improving performance.

HTTP/1.1: In HTTP/1.1, the client initiates each request, and the server responds with the requested resources. There is **no mechanism for the server to push additional resources** to the client without a specific request.
HTTP/2: HTTP/2 introduces **server push, where the server can proactively send additional resources** (e.g., images, scripts) to the client before they are requested. This reduces the need for extra round trips and can speed up page loading significantly.

HTTP/1.1: HTTP/1.1 **is fully backward compatible with HTTP/1.0,** which means that older clients and servers can still communicate with newer ones, albeit without some of the performance benefits of HTTP/1.1.
HTTP/2: While HTTP/2 **is backward compatible with HTTP/1.1,** it requires more sophisticated negotiation and handling, and some older servers and proxies may not fully support it.

JavaScript - Day -1: Introduction to Browser & web

2. Write a blog about objects and its internal representation in JavaScript.

Objects, in JavaScript, is its 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 E.g. If your **object is a student**, it will have **properties** like **name, age, address**, id, etc and **methods** like **update Address, update Nam**, etc.

A JavaScript object has properties associated with it. A property of an object can be explained as a variable that is attached to the object. Object properties are basically the **same as ordinary JavaScript variables**, except for the attachment to objects. The properties of an **object define the characteristics** of the object. You access the properties of an object with a simple dot-notation:

Like all JavaScript variables, both the object name (which could be a normal variable) and property name are **case sensitive**. You can define a property by assigning it a value.
For example, let's create an object named myCar and give it properties named make, model, and year as follows:

```
varmyCar = new Object();

myCar.make = 'Ford';

myCar.model = 'Mustang';

myCar.year = 1969;
```

Properties of JavaScript objects can also be accessed or set using a **bracket notation** (for more details see property accessors). Objects are sometimes called ***associative arrays*****,** since each

property is associated with a string value that can be used to access it. So, for example, you could access the properties of the myCar object as follows:

```
myCar['make'] = 'Ford';

myCar['model'] = 'Mustang';

myCar['year'] = 1969;
```

Internal Representation

- **Memory allocation** - allocates memory to store that object.
- **Property storage** - Each property's name (key) is stored as a string
- **Property access** - JavaScript internally performs a lookup to find the associated value within the object's memory.
- **Object prototype** - prototype is another object that the current object inherits properties and methods from.
- **Reference - based variables** - Variables in JavaScript do not directly store objects; instead, they hold references to the memory locations where objects reside

JavaScript - Day -1: Introduction to Browser & web

3. codekata practice – **Practice done with input output concept**
4. Read about IP address, port, HTTP methods, MAC address

   Links used for reference are listed below:-

   IP Address

   - https://www.geeksforgeeks.org/what-is-an-ip-address
   - https://www.kaspersky.com/resource-center/definitions/what-is-an-ip-address
   - https://www.techtarget.com/whatis/definition/IP-address-Internet-Protocol-Address#:~:text=An%20Internet%20Protocol%20(IP)%20address,for%20communicating%20across%20the%20internet.

   Port

   - https://www.tutorialspoint.com/what-is-network-port
   - https://www.geeksforgeeks.org/introduction-of-ports-in-computers/

   HTTP methods

   - https://www.geeksforgeeks.org/different-kinds-of-http-requests/
   - https://www.w3schools.com/tags/ref_httpmethods.asp
   - https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods

   MAC address

   - https://www.geeksforgeeks.org/mac-address-in-computer-network/
   - https://www.techtarget.com/searchnetworking/definition/MAC-address
   - https://www.javatpoint.com/what-is-mac-address