1. Write a blog on Difference between HTTP1.1 vs HTTP2

Ans :

HTTP 1.1 vs HTTP 2

| **HTTP 1.1** | **HTTP 2** |
| --- | --- |
| * HTTP/1.1 relies on multiple connections, leading to increased overhead and slower performance. | * HTTP/2 enables concurrent data exchange over a single connection, slashing latency. |
| * HTTP/1.1 sends headers in plaintext, consuming unnecessary bandwidth with each request/response. | * HTTP/2 compresses headers, minimizing overhead and boosting efficiency. |
| * HTTP/1.1 lacks this feature, resulting in additional round trips and slower loading times. | * HTTP/2 facilitates server push, proactively delivering resources to the client's cache before being requested. |
| * HTTP/1.1's text-based protocol is more prone to parsing errors and less secure than HTTP/2's binary protocol. | * HTTP/2 utilizes a binary framing layer, streamlining parsing and reducing errors compared to HTTP/1.1's textual format. |
| * HTTP/1.1 doesn't have built-in flow control mechanisms, leading to potential congestion and inefficient data exchange. | * HTTP/2 introduces stream-level flow control, optimizing data transmission and resource utilization between client and server. |

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

Object Internal Representation in JavaScript

**Objects :**

* In JavaScript, objects are fundamental data structures representing key-value pairs.
* They can store various data types, including primitive values, functions, and even other objects.

For example,

| const person = {  name: "John",  age: 30,  isAdmin: false,  greet: function() {  console.log("Hello!");  }  }; |
| --- |

**Internal Representation:**

* Internally, objects are stored as collections of properties, where each property consists of a key-value pair.
* These properties are stored in an internal data structure called the object's property map.

**Property Access:**

* Properties can be accessed using dot notation or bracket notation.

For example,

| console.log(person.name); // Dot notation  console.log(person["age"]); // Bracket notation |
| --- |

**Adding and Deleting Properties:**

* Properties can be dynamically added or deleted from objects.

For example,

| person.email = "john@example.com"; // Adding a property  delete person.age; // Deleting a property |
| --- |

**Object Methods:**

* JavaScript provides built-in methods for working with objects, such as Object.keys(), Object.values(), and Object.entries() for iterating over object properties.

For example,

| const keys = Object.keys(person);  console.log(keys); // Output: ["name", "isAdmin", "greet", "email"] |
| --- |

Understanding how objects are internally represented in JavaScript is crucial for efficient manipulation and optimization of code.