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

**Ans: -**

|  |  |
| --- | --- |
| HTTPS 1.1 | HTTP 2 |
| Connection Multiplexing operates on a request-response model, requiring multiple connections for parallel downloads. This can lead to increased latency and slower page loading times. | Introduces multiplexing, allowing multiple streams within a single connection. This minimizes latency, speeds up resource loading, and enhances overall performance. |
| Headers can be verbose, contributing to increased latency and larger data transfers. | Utilizes header compression to reduce redundancy and optimize data transfer. This results in more efficient communication between the client and server. |
| Does not have built-in mechanisms for prioritizing the loading of critical resources. | Introduces stream prioritization, allowing the server to prioritize the delivery of essential resources. This ensures faster loading of crucial elements, improving the user experience. |
| Relies on the client to request each resource individually, leading to multiple round-trip requests. | Features server push, allowing the server to proactively push relevant assets to the client without waiting for explicit requests. This reduces the need for multiple round-trip requests and further speeds up page load times. |
| Relies on a text-based protocol, which can be less efficient for processing and data compression. | Adopts a binary protocol, simplifying parsing and allowing for more efficient use of network resources. The binary format is designed for faster processing and improved data compression. |

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

**Ans: -**

**Objects:**

* The Building Blocks At its core, an object in JavaScript is a collection of key-value pairs, where each key is a string (or symbol) and each value can be any data type, including other objects.
* Objects provide a way to structure and organize data in a meaningful way, enabling developers to create, access, and modify properties dynamically.

**Internal Representation:**

* Objects as Maps Internally, JavaScript engines often represent objects as maps, using a data structure that allows for efficient key-based access.
* This enables quick retrieval of values associated with specific keys, making property access and manipulation fast and efficient.