| **HTTP/1.1** | **HTTP/2** |
| --- | --- |
| HTTP/1.1 uses textual format to transfer the data | HTTP/2 works on binary format |
| If the response is not accepted, then all the requests are stopped | It allows multiplexing so one TCP connection is required for multiple requests. |
| It uses requests resource Inlining for use getting multiple pages | It uses PUSH frame by server that collects all multiple pages |
| It compresses data by itself. | It uses HPACK for data compression. |

1)

Main Advantage:

* One of the primary advantages of using the HTTP/2 protocol is that we only need one server connection to load a website were as with the HTTP/1.1 protocol, you may need multiple requests or connections to load a site, which can slow the site down.
* But when we use HTTP/2 with a single TCP connection or server push for the entire website session it reduces time that the data must take to set up multiple TCP connections. Reducing those trips speeds up the connection, page load times and makes it more efficient to use.

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

\* As compared to primitive datatypes, objects are different from it were it stores data as “Key:Value” form.

\*In primitive datatype like string, Boolean, Number store only single data type but were as objects store different data types.

\*In primitive data types of the values are called as variables, were as in objects it is called as functions and methods.

Input:

let student={

    name:"Balamurugan A",

    rollNum:"BME20",

};

console.log(student.name);

console.log(student.rollNum);

Output:

Balamurugan A

BME20

\*The internal representation of objects in JavaScript involves sophisticated mechanisms for managing property storage and access. By understanding these concepts, you can write more efficient and performant JavaScript code.

\*This involves being mindful of object shapes, minimizing property additions and deletions, and taking advantage of optimizations provided by modern JavaScript engines.