Q1. Write a blog on the Difference between HTTP1.1 vs HTTP2.

Ans:-

* **HTTP:-** The Hypertext Transfer Protocol (HTTP) is an application-layer protocol for sending hypermedia content such as HTML. It was created for communication between web browsers and web servers, but it may also be used for other applications. HTTP follows a traditional client-server approach, with a client initiating a connection to make a request and then waiting for a response. HTTP is a stateless protocol, which means that the server does not save any data (state) between requests.
* **HTTP1.1:-** HTTP protocol was developed in 1989 as the common language that enables client and server machines’ interaction. Keep in mind that this is not a one-time process. Such requests and responses need to be transferred between both these machines until the client receives all the resources, essential to load a web page on the end-users (you’re) screen. This request-response exchange can be regarded as an IP stack being handled by the transfer layer and networking layers before finally reaching the application layer. Now, let’s see how HTTP/2 handles the same scenario.
* **HTTP2.0:-** HTTP2.0 was released at Google as a significant improvement of its predecessor. It was initially modeled after the SPDY protocol and went through significant changes to include features like multiplexing, header compression, and stream prioritization to minimize page load latency. After its release, Google announced that it would not provide support for SPDY in favor of HTTP2.0. HTTP 2.0 was released at Google as a significant improvement of its predecessor. It was initially modeled after the SPDY protocol and went through significant changes to include features like multiplexing, header compression, and stream prioritization to minimize page load latency. After its release, Google announced that it would not provide support for SPDY in favor of HTTP2.0.

**Q2.** Write a blog about objects and their internal representation in Java script.

* Objects, in JavaScript, is its most important data type and form 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 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, etc, and methods like updateAddress, updateNam, 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.
* 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.
* 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.