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

**HTTP 1.1:**

**T**ransfer all the requests & responses in the plain text message form. The second one is head of line blocking in which TCP connection is blocked all other requests until the response does not receive. all the information related to the header file is repeated in every request.

* Introduced in 1997.
* Internet landscape was constantly changing with websites becoming more dynamic&heavy.
* Feature like CORS, keep-alive (most important one) was introduce in this update.
* HOL-Head of line blocking.
* Repeatition of header data.
* More focus on gzip, minifying css/js, caching ect.

|  |  |
| --- | --- |
| **HTTP/1.1** | **HTTP/2** |
| the uses works on the textual format. | It works on the binary protocol. |
| There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. | 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. |

**HTTP2.O:**

HTTP/2 was developed over the SPDY protocol. HTTP/2 works on the binary framing layer instead of textual that converts all the messages in binary format. it works on fully multiplexed that is one TCP connection is used for multiple requests. HTTP/2 uses HPACK which is used to split data from header. it compresses the header. The server sends all the other files like CSS & JS without the request of the client using the PUSH frame.

* Introduced in 2015.
* HTTP 2 is an HTTP 1.1 connection with some additional features.
* One secured TCP connection is setup in which HTTP requestion are transferred streams .
* HPACK: Header data is separate from request data and can be zipped.
* HPACK also enables reuse of header data which is repeated in every request.
* HPACK reduces HTTP request size.
* Push frames enable us to send mandatory resource in advance along with a HTTP response.
* Push frames should be used with care as this can lead to increase in size of the HTTP response.

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

Objects, in JavaScript, is it’s 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.

**Object:**

In JavaScript, an object is a standalone entity, with properties and type. Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc. The same way, JavaScript objects can have properties, which define their characteristics.

**Creating Objects in JavaScript:**

* By object literal
* By creating instance of Object directly (using new keyword)

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.