**Javascript - Day -2 : Introduction to Browser & web**

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

**Ans:**

|  |  |
| --- | --- |
| **HTTP 1.1** | **HTTP 2** |
| * + Some methods and response codes are added.   + **“keep alive”** becomes officially supported. “host” header is supported for virtual domain.   + Syntax and semantics are separated.   + Programs like gzip have long been used to compress the data sent in HTTP messages, especially to decrease the size of CSS and JavaScript files.   + In order to solve this bottleneck, HTTP/2 uses HPACK compression to shrink the size of headers, a topic discussed further in the next section. | * + Support of parallel request transmission by “stream”(addressing HOL blocking issue of HTTP requests).   + Addition of flow-control function in units of “stream”.   + Addition of prioritization function in units of “stream”.   + Addition of PUSH function (send related file without request).   + One of the themes that has come up again and again in HTTP/2 is its ability to use the binary framing layer to exhibit greater control over finer detail.   + Additionally, HPACK can keep track of previously conveyed metadata fields and further compress them according to a dynamically altered index shared between the client and the server. |

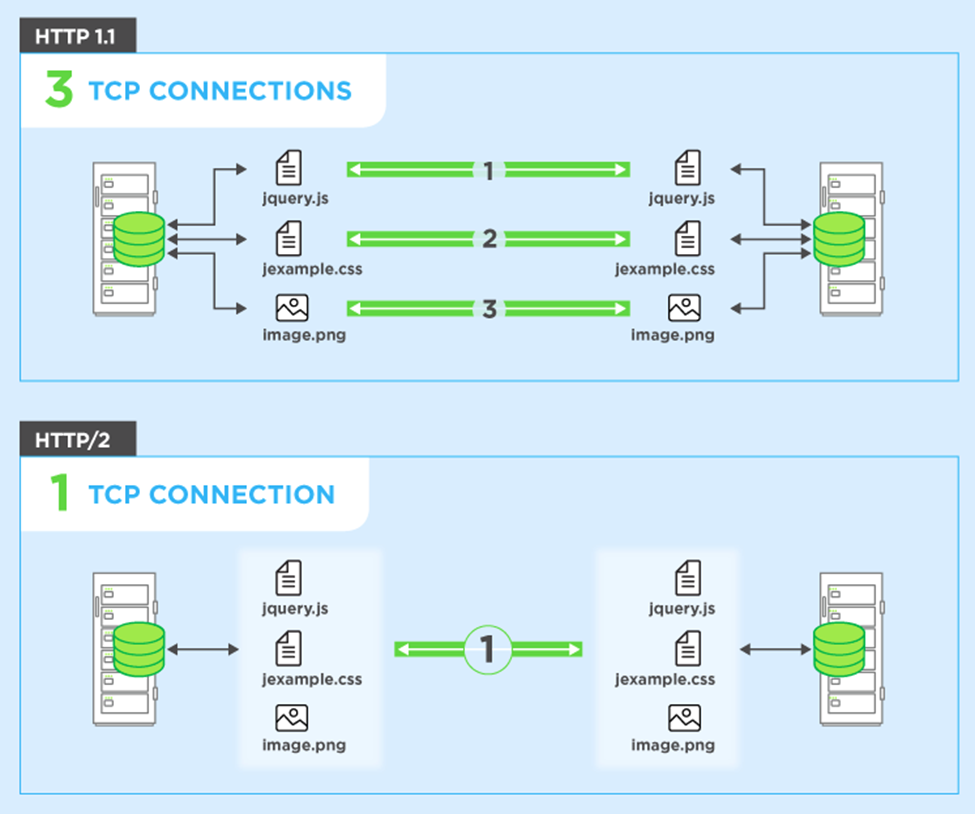
***Example image of Difference between HTTP1.1 vs HTTP2***

Fig1.1

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

**Ans:**

**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.
* For Eg. If your object is a student, it will have properties like name, age, address, id, etc and methods like updateAddress, updateNam, etc.