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

HTTP/1.1:

* The first usable version of HTTP was created in 1997, it was called as HTTP/1.1
* HTTP/1.1 was quickly adopted by both browsers and end-users.
* HTTP/1.1 were additional cache controls, Content Negotiation that allowed for different languages, content encodings, and types.
* HTTP/1.1 introduced standardization, new features, and improved the efficiency of the protocol through better Caching, encoding.
* This saves in terms of connection overhead, in particular, speeding up media-rich documents.

HTTP/2:

* HTTP/2 is able to use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resource.
* HTTP/2 does this by splitting data into binary-code messages and numbering these messages so that the client knows which stream each binary message belongs to.
* HTTP/2 solves problem by allowing a server to "push" content to a client before the client asks for it.
* HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets.
* Given the volume of HTTP packets involved in loading even a single webpage, those bytes add up quickly, resulting in faster loading.

1. Write a blog about Objects and it’s internal representation in JavaScript

Objects:

* 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.
* The reference or pointer points to the location in memory where the object is stored. The variables don’t actually store the value.

Objects internal representation in JavaScript:

* 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 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.
* You access the properties of an object with a simple dot-notation:

objectName.propertyName