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

|  |  |
| --- | --- |
| * **HTTP 1.1** * It works on the textual format. * There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. * It uses requests resource Inlining for use getting multiple pages * It compresses data by itself. * **pipelining (the second request is sent before the response to the first is adequately served)** * **content negotiation (an exchange between client and server to determine the media type, it also provides the provision to serve different versions of a resource at the same URI)** * **cache control (used to specify caching policies in both requests and responses)** * The client (browser) has to send a request to the server using the method (GET/POST). * Server responds with the requested resource, for example – image, alongside the status of what it did to the client’s request * he flow control mechanism in HTTP/1.1 relies on the basic TCP connection. In beginning itself, both the machines set their buffer sizes automatically. If the receiver’s buffer is full, it shares the receive window details, telling how much available space is left. The receiver acknowledges the same and sends an opening signal. | * **HTTP.2** * It works on the binary protocol * It allows multiplexing so one TCP connection is required for multiple requests. * It uses PUSH frame by server that collects all multiple pages * It uses HPACK for data compression. * **HTTP/2 uses HPACK header compression algorithm that is resilient to attacks like CRIME and utilizes static Huffman encoding** * **Open the web developer tool on the web browser (like Firefox).** * **Under the network tab, select any of the resources and check the version number under the headers tab.** * All modern browsers support HTTP/2 over HTTPS with the SSL certificate installation. To open HTTPS capable invisible proxy ports on every relevant port, OWASP ZAP or its alternatives could be used. * The use of the HPACK algorithm enables HTTP/2 to overcome the common [API security](https://www.wallarm.com/what/api-security-tutorial) threats. * This protocol has commands in binary format and compresses the HTTP header metadata to protect sensitive data shared between both machines. |

**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.

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