**ASSIGNMENT DAY – 1**

**1.  Difference between HTTP1.1 vs HTTP2**

**HTTP** stands for **H**yper **T**ext **T**ransfer **P**rotocol

Communication between client computers and web servers is done by sending **HTTP requests** and receiving **HTTP Responses**

HTTP is a stateless application-level protocol and it requires a reliable network transport connection to exchange data between client and server.

**Difference between HTTP1.1 vs HTTP2**

In HTTP implementations TCP/IP connections are used using well-known. In HTTP/2 a TCP/IP connection + multiple protocol channels are used.

HTTP2 is much faster and more reliable than HTTP1.1 . HTTP1 loads a single request for every TCP connection, while HTTP2 avoids network delay by using multiplexing.

|  |  |
| --- | --- |
| HTTP1.1 | HTTP2 |
| 1. The client (browser) has to send a request to the server using the method (GET/POST). It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once. However, pipelining was hard to implement due to issues such as head-of-line blocking and was not a feasible solution. 2. Protocol Http 1.1 is much more secured than Http 1.0 because it uses digest authentication and NTLM authentication. 3. In this version, SSL or secure sockets layer is not required but recommended. Digest authentication is an improvement over HTTP 1.0 which is now being used in HTTP 1.1. Moreover, HTTPS uses SSL/TLS for secure encrypted communication. 4. It provides faster delivery of web pages and reduces web traffic if you compare it to Http 1.0. However, there is an increased risk of network congestion. 5. Some of the optimizations used in the HTTP 1.1 version are sprinting, inlining, domain sharding, and concatenating. 6. This protocol introduces a warning header field to carry additional information about the status of a message. It can define 24 status codes, error reporting is quicker and more efficient. | 1. Uses multiplexing, where over a single TCP connection resource to be delivered are interleaved and arrive at the client almost at the same time. It is done using prioritized streams and can have dependencies and individual flow control. It also provides a feature called server push that allows the server to send data that the client will need but has not yet requested. 2. The security concern in the HTTP 2 version is also good and almost the same as HTTP 1.1. Rather HTTP 2 is better equipped to deal with security threats because of the new features it brings. For example, new TLS features like connection errors of type inadequate security. 3. In the HTTP 2 protocol, security is not at all recommended. It is because the security is encrypted since all almost all clients demand traffic to be encrypted. It also has minimum standards and minimum key size for encryption. 4. The HTTP 2 version utilizes multiplexing and server pushes to effectively reduce the page load time by a greater margin along with being sensitive to network delays. 5. This protocol version removes the need for unnecessary optimization hacks. 6. It brings the fundamental semantics of HTTP like headers, and status code remains the same. |

2. **Objects And Its Internal Representation In JavaScript**

Objects, in JavaScript, is it’s the 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 these primitive data-types all store a single value each (depending on their 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.

Otherway around, 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.  
An object can be created with figure brackets {} with an optional list of properties. A property is a “key: value” pair, where a key is the property name value can be anything.

Every object has some property associated with some value. These values can be accessed using these properties associated with them.

var myCar = new Object();

myCar.make = 'Suzuki';

myCar.model = 'Altros';

myCar.year = 1978;

myCar.wheels = 2;

After creating myCar object, the value inside the object can be accessed using keys.

i.e.

myCar.year

Output: 1978

These values can be accessed using brackets notation also.

myCar.year

Output: 1978