Day 1 – Assignment

1. **Write a blog on the difference between HTTP 1.1 and HTTP 2**

HTTP is an abbreviation for Hyper Text Transfer Control. It is the primary protocol used to send data between a browser and a website.

There are various versions of HTTP: HTTP 0.9, HTTP 1.0, HTTP 1.1, HTTP 2, HTTP 3

HTTPS 1.1 is built on HTTPS 1.0.

Its attributes are HOST, USER AGENT, ACCEPT-LANGUAGE, ACCEPT-ENCODING, CONTENT-TYPE, DATA, LAST-MODIFIED, SERVER, VARY and more.

Some of the key features are Persistent connections and pipelining.

HTTP 2 is a major revision of the HTTP 1.1, it has many improved features that lacked in the previous version such as: Binary protocol, Header compression, Steam prioritization, Client Cache Mechanism and Multiplexing.

The key differences between these two are:

|  |  |
| --- | --- |
| HTTP 1.1 | HTTP 2 |
| Works on textual format. | Works on 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 enough for multiple requests. |
| It uses requests resource inlining for getting multiple pages. | It used PUSH frame by server that collects multiple pages. |
| It compresses data by itself. | It uses HPACK for data compression. |

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

Objects can hold many values in form of key:value pair. These keys can be variables or functions and are called properties and methods. They are a combination of primitive data types. They are of reference data type. That reference points to the location in memory where the object is stored.

Objects are the building blocks of Javascript. They are briefly defined as a collection of unordered related data.

The properties of an object can be explained as a variable that is attached to the object. Object properties are basically the same as ordinary Javascript variables, except for attachment to objects. We can access the properties of an object with a dot notation.

Example: objectName.propertyName; car.Model, car.Color

The easiest way to create a javascript object is object literal.

Let car = {name: ‘Aston Martin’, color: ‘Black’, Engine: ‘v10’ };

Internation Representation of Objects:

Javascript engine utilizes various data structures and mechanisms to represent objects efficiently. One common approach is using hash tables or dictionaries to store properties and their corresponding values. When you create an object literal like the above example or any like “{}”, Javascript’s engine allocates memory for it and employs a hash table to manage its properties. This mechanism allows for fast property access regardless of the size of the object.

Javascript also supports prototypal inheritance allowing objects to inherit properties and method from other objects. This is implemented through a mechanism called the prototype chain.

Javascript manages memory dynamically through a process called garbage collection. When an object is no longer referenced by any part of the program, it becomes eligible for garbage collection and the memory it occupies is reclaimed.