1. Write a blog on Difference between HTTP 1.1 vs HTTP2

|  |  |
| --- | --- |
| HTTP1.1 | HTTP2 |
| * Uses a text based protocol. * Only allows one request, and response to be sent at a time. * Does not support server push. * Does not support multiplexing * 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 In lining for use getting multiple pages * It compresses data by itself. | * Uses a binary protocol * Allows multiple requests,and response to be sent over a single connection. * Supports server push. * Support multiplexing and also 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. |
| **BENEFITS:**   * Persistent connection * Pipelining * Caching * **Chunked transfer encoding.** | **BENEFITS:**   * Implementation complexity * Server resource consumption * Security concerns * Performance limitation |

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

* Objects are important data types in javascript. Objects are different than primitive datatypes (i.e. number, string, boolean, etc.). Primitive data types contain one value but Objects can hold many values in form of Key: value pair. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.
* 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.
* In JavaScript, objects are collections of key-value pairs, where keys are strings (or symbols) and values can be of any data type, including other objects. Objects are used to represent real-world entities, data structures, and more complex data types.
* Example:

var person=

{name:”Mega”,age:23,gender:”female”,education=”be cse”};

* Internally, JavaScript engines use various data structures to represent objects efficiently. One common approach is using a hash table or a similar structure to store the object’s properties and their corresponding values. This allows for fast access and manipulation of properties.
* Example:

Internal Representation:

* {name:”Mega”,age:23,gender:”female”,education=”be cse”};