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

**Answer:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Differentiator** | **HTTP/1.1** | **HTTP2** |
| 1 | **Year** | It was created in **1997** | It was created in **2015** |
| 2 | **Request Multiplexing** | It loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it | It use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resource. |
| 3 | **Header compression** | It compresses data by itself | It uses **HPACK** for data compression |
| 4 | **Server Push** | It uses requests resource Inlining for use getting multiple pages | It uses **PUSH** frame by server that collects all multiple pages. |
| 5 | **Delivery Models** | HTTP/1.1 sends messages as plain text | HTTP/2 encodes the binary data |

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

**Answer:**

* 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.

**Syntax:**

**var objectname={**

**Keyname : value**

**};**

**Ex:**

var person={

fname=”XXX”,

name=”YYY”,

age=25,

};

**creating instance of Object**

**Syntax:**

**var objectname=new Object();**

**Ex:**

var emp=new Object();

**Accessing the elements inside the Object.**

* We can access the elements inside the object by using the keyname.
* There are two types of accessing the objects.
* They are,

**1)Dot Method**

**2)Box Method.**

**Dot Method:**

**Syntax:**

**Objectname.keyname;**

**Ex:**

person.fname;

**Box Method:**

**Syntax:**

**objectname[“keyname”];**

**Ex:**

person[“fname”];

**Insertion :**

**Syntax:**

**objectname.keyname=value;**

**Ex:**

person.gender=”male”;

**Deletion:**

**Syntax:**

**delete Objectname.keyname;**

**Ex:**

delete person.age;

**Updation:**

* It will be applicable for the key which is already present.

**Syntax:**

**objectname.keyname=value;**

**Ex:**

person.gender=”female”;

**Enumerating Properties**

There are three native ways to list/traverse object properties:

* **For..in** loops. This method traverses all of the enumerable string properties of an object as well as its prototype chain.
* **Object.key(myObj).** This method returns an array with only the enumerable own string property names ("keys") in the object myObj, but not those in the prototype chain.
* **Object.getOwnPropertyName(myObj).** This method returns an array containing all the own string property names in the object myObj, regardless of if they are enumerable or not.