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

**HTTP:**

* HTTP stands for **H**yper **T**ext **T**ransfer **P**rotocol.
* This is an application protocol that is currently the foundation of **data communication for the World Wide Web.**
* HTTP allow communication **between server and client.**
* It is sending HTTP Requests and receiving HTTP Responses

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

|  |  |
| --- | --- |
| **HTTP 1.1**  The standardized protocol | **HTTP 2**  The SPDY(Speedy) Protocol |
| * HTTP/1.1 Developed by**Timothy Berners-Lee** in 1989. | * HTTP/2 was developed by Internet Engineering Task Force (**IETF**) in 1997. |
| * HTTP 1.1 much faster than HTTP 1.0 but **slower than HTTP2**. | * HTTP2 is **much faster and more reliable**. |
| * HTTP 1.1 can use **multiple TCP** **connection** between the same client and the server. | * HTTP 2 is **multiplexing** multiple requests over a **single TCP connection** |
| * HTTP1.1 is **cache support** which is big **risk for security** of data passing from server to client. | * HTTP 2 is **more secure** as it uses binary protocol instead of plain text. |
| * HTTP/1.1 uses **gzip compression**. | * HTTP/2 instead, uses **HPACK**which is more powerful and secure than gzip |
| * HTTP 1.1 which keeps all request and response in **plain text format**. | * HTTP 2 is **binary format**, instead of textual. |
| * In HTTP/1.1, this was easy, as**head-of-line blocking**made it simple to load various assets in the correct order. | * In HTTP 2 **resolves the head-of-line blocking issue** in HTTP/1.1 by ensuring that no message has to wait for another to finish |

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

**OBJECT:**

* **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.**
* It is a common practice to declare objects with the **const**keyword.
* JavaScript is an **object-based language**. Everything is an object in JavaScript
* Booleans can be objects (if defined with the new keyword)
* Numbers can be objects (if defined with the new keyword)
* Strings can be objects (if defined with the new keyword)
* Dates are always objects
* Maths are always objects
* Regular expressions are always objects
* Arrays are always objects
* Functions are always objects
* Objects are always objects
* An **object** created inside curly **braces** {}. The following example creates a new **JavaScript object** with four properties:

Example:

const student = {

Id: “14009”,

Name:”Praneeta.B”,

Class:”UKG”

Section:”C”

}

* The syntax for accessing the property of an object is:
* Objectname.property // student.id
* Objectname[“property”] // student[“id”]

**Internal representation of Object;**

In javascript object can be represent internally as key and value, array or array of object.

Syntax:

const objectname={};/create an empty object

Objectname[key]=value;//assign a value of key inside the object

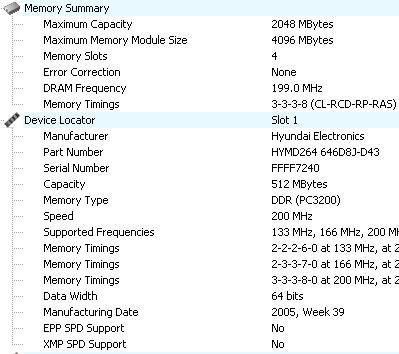
Eg

const frequency={};

frequency["1"]=2;

frequency ["2"]=5;

console.log(frequency);



* The Above image has a collection of data that need to create an object like below

const taskip = [{"Memory Summary":[{"Maximum Capacity":"2048 MBytes",

"Maximum Memory Module Size":"4096 MBytes",

"Memory Slots":4,

"Error Correction":"None",

"DRAM Frequency":"199.0 MHz",

"Memory Timings":"3-3-3-8 (CL-RCD-RP-RAS)" }]}

,

{"Device Locator":[{"Name":"Slot1",

"Manufacturer":"Hyundai Electronics",

"Part Number":"HYMD264 646D8J-D43",

"Serial Number":"FFFF7240",

"Capacity":"512 MBytes",

"Memory Type":"DDR (PC3200)",

"Speed":"200 MHZ",

"Supported Frequencies":"133 MHz, 166 MHz, 200 M",

"Memory Timings":["2-2-2-6-0 at 133 MHz, at","2-3-3-7-0 at 166 MHz, at 2","3-3-3-8-0 at 200 MHz, at 2"],

"Data Width":"64 bits",

"Manufacturing Date":"2005, Week 39",

"EPP SPD Support":"No",

"XMP SPD Support":"No"}]

}];

**Input:**

console.log(taskip[1]["Device Locator"][0]["Name"]);

**Output:**

Slot1

**Input:**

console.log(taskip[1]["Device Locator"][0]["Memory Timings"][0]);

**output:**

2-2-2-6-0 at 133 MHz,at