ZEN CLASS

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

**HTTP 1.1:**

For better understanding, let’s assume the situation when you make a request to the server for the geeksforgeeks.html page & server responds to you as a resource geeksforgeeks.html page.

before sending the request and the response there is a TCP connection established between client & server. again you make a request to the server for image img.jpg & the server gives a response as an image img.jpg.

the connection was not lost here after the first request because we add a keep-alive header which is the part of the request so there is an open connection between the server & client. there is a persistent connection which means several requests & responses are merged in a single connection.

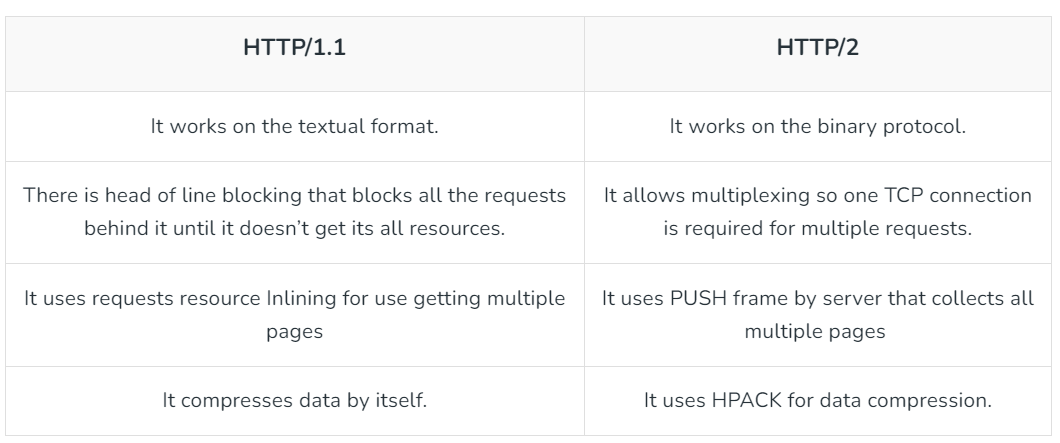
These are the drawbacks that lead to the creation of HTTP/2: The first problem is HTTP/1.1 transfer all the requests & responses in the plain text message form.

The second one is head of line blocking in which TCP connection is blocked all other requests until the response does not receive. all the information related to the header file is repeated in every request.

HTTP 2:

HTTP/2 was developed over the SPDY protocol. HTTP/2 works on the binary framing layer instead of textual that converts all the messages in binary format. it works on fully multiplexed that is one TCP connection is used for multiple requests.

HTTP/2 uses HPACK which is used to split data from header. it compresses the header. The server sends all the other files like CSS & JS without the request of the client using the PUSH frame.



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

**Introduction:**

JavaScript, the language that powers dynamic and interactive web applications, is built around the concept of objects. Objects are fundamental to the language, serving as a cornerstone for data manipulation and structuring. In this blog post, we’ll embark on a visual exploration of objects in JavaScript and unravel their internal representation.

**1.What is an Object in JavaScript?**

In JavaScript, an object is a complex data type that allows you to store and organize data in key-value pairs. Objects can represent real-world entities and are used to model and manipulate information efficiently.

**2. Creating Objects:**  
Objects in JavaScript can be created using object literals or the `Object` constructor. Let’s take a look at a simple object literal:

Var person = {

Name: “Tahir”;

Age: 25;

Is Employed: true;

}

**3. Internal Representation of Objects:**  
Internally, JavaScript engines represent objects using various data structures. One common representation is the hash table, where keys are hashed to optimize property access.

The hash table allows for quick lookup of properties, making object access efficient.

**4. Object Properties and Methods:**  
Objects can contain not only data properties but also methods, which are functions associated with the object. Let’s add a method to our `person` object:

**javascript code :  
*person.greet = function() {  
console.log(`Hello, I’m ${this.name}!`);  
};***

Now, our `person` object has a `greet` method that logs a greeting.

**5. Prototypes and Inheritance:**  
JavaScript is a prototype-based language, and objects can inherit properties and methods from other objects through their prototypes. This mechanism enables the creation of hierarchical structures.

In the JS code, you can see how objects and their prototypes are linked, forming a chain of inheritance.

**Conclusion:**

Objects lie at the heart of JavaScript, providing a powerful mechanism for organizing and manipulating data. The internal representation of objects, often implemented using hash tables, ensures efficient property access. As you dive deeper into JavaScript development, understanding how objects work and how they are internally represented will empower you to write more expressive and modular code. So, embrace the magic of objects, and let them be your companions in building dynamic and sophisticated web applications.