1. W**rite a blog on Difference between HTTP1.1 vs HTTP2.**

**What is HTTP?**

HTTP stands for hypertext transfer protocol, and it is the basis for almost all web applications. More specifically, HTTP is the method computers and servers use to request and send information.

**What is HTTP 1.1?**

The first usable version of HTTP was created in 1997. Because it went through several stages of development, this first version of HTTP was called HTTP/1.1. This version is still in use on the web.

**What is HTTP 2?**

In 2015, a new version of HTTP called HTTP/2 was created. HTTP/2 solves several problems that the creators of HTTP/1.1 did not anticipate. In particular, HTTP/2 is much faster and more efficient than HTTP/1.1. One of the ways in which HTTP/2 is faster is in how it prioritizes content during the loading process.

**Difference between HTTP 1.1 and HTTP 2:**

|  |  |
| --- | --- |
| **HTTP 1.1** | **HTTP 2** |
| It works on the textual format. | It works on the 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 required for multiple requests. |
| It uses requests resource Inlining for use getting multiple pages. | It uses PUSH frame by server that collects all 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.**

**Introduction:**

JavaScript, as a versatile and dynamic programming language, owes much of its power to its robust object-oriented paradigm. Objects are the building blocks of JavaScript applications, providing a flexible and intuitive way to structure and manipulate data. In this blog, we'll embark on a journey to uncover the intricacies of objects and explore how they are internally represented in JavaScript.

**The Essence of Objects in JavaScript:**

At its core, an object in JavaScript is a collection of key-value pairs, where each key (also known as a property) is associated with a value. This fundamental structure allows developers to represent and organize complex data in a comprehensible manner.

1. Creating Objects: Objects can be created in various ways, whether through literal notation, constructor functions, or class syntax in modern JavaScript. Regardless of the method, the result is an entity with properties and methods that can be accessed and manipulated.
2. Properties and Methods: Properties are the characteristics or attributes of an object, while methods are functions associated with the object. This distinction allows developers to encapsulate related functionalities within the object, promoting modularity and code organization.

**Understanding how objects are represented in memory sheds light on their performance characteristics and aids in writing efficient code.**

1. Reference vs. Value: In JavaScript, objects are reference types, while primitive types (e.g., numbers, strings) are value types. This means that when an object is assigned to a variable, the variable holds a reference to the object, not the actual object itself.

2. Memory Management: JavaScript uses automatic memory management through garbage collection. Objects that are no longer reachable are automatically identified and removed from memory, preventing memory leaks.

**Conclusion:**

Objects lie at the heart of JavaScript, offering a versatile and expressive means of representing and organizing data. The language's internal representation of objects, including prototypes, property descriptors, and hidden classes, plays a crucial role in how JavaScript efficiently manages and accesses object properties. Armed with this knowledge, developers can navigate the intricacies of objects and unleash the full potential of JavaScript in their applications.