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

Ans:

Key Characteristics of HTTP/1.1:

1. **Connection Multiplexing:**
   * HTTP/1.1 relies on a single connection per request. This means that if a browser needs multiple resources (CSS files, images, scripts), it has to open multiple connections, resulting in potential bottlenecks.
2. **Header Overhead:**
   * Each request and response in HTTP/1.1 carries its set of headers. This redundancy contributes to increased data overhead, impacting the overall performance, especially on high-latency connections.
3. **No Built-in Compression:**
   * HTTP/1.1 does not provide native support for header compression, leading to larger payload sizes and slower loading times.

HTTP/2 Advancements:

HTTP/2 was introduced to address the shortcomings of HTTP/1.1 and to enhance the web browsing experience. It introduces several features aimed at improving efficiency, reducing latency, and accommodating modern web applications.

Key Characteristics of HTTP/2:

1. **Multiplexing:**
   * Perhaps the most significant improvement in HTTP/2 is the introduction of multiplexing. Multiple requests and responses can be sent and received concurrently over a single connection, eliminating the need for multiple connections and significantly reducing latency.
2. **Header Compression:**
   * HTTP/2 uses a more efficient header compression mechanism called HPACK. This reduces redundant header information, minimizing data overhead and leading to faster loading times.
3. **Binary Protocol:**
   * Unlike the text-based protocol of HTTP/1.1, HTTP/2 is a binary protocol. This makes it more compact and efficient for data transfer.
4. **Server Push:**
   * HTTP/2 enables servers to push resources to the client before they are explicitly requested. This preemptive approach can improve page load times by reducing the number of round-trips needed.

Conclusion:

The transition from HTTP/1.1 to HTTP/2 represents a significant leap in web protocol technology. HTTP/2's focus on multiplexing, header compression, and other optimizations addresses the challenges posed by its predecessor. As web developers and businesses embrace HTTP/2, users can expect faster, more efficient, and responsive web experiences.

In conclusion, while HTTP/1.1 has served the web well, HTTP/2 emerges as a more robust and capable successor, paving the way for a faster and more optimized internet. The adoption of HTTP/2 is a testament to the continuous efforts to enhance the performance and efficiency of the web for users worldwide.

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

Ans:

**1. Properties and Methods:**

In JavaScript, objects can have properties and methods. Properties are essentially variables attached to the object, while methods are functions associated with the object. Both properties and methods contribute to the internal structure of an object.

**2. Prototypes and Inheritance:**

JavaScript is a prototype-based language, and objects can inherit properties and methods from other objects through prototypes. This is a key aspect of the internal representation of objects and is fundamental to JavaScript's approach to inheritance.

**3. Dynamic Nature:**

One of the distinctive features of JavaScript objects is their dynamic nature. Properties can be added or removed from objects at runtime, providing flexibility to adapt to changing requirements.

**4. Object Descriptors:**

Each property of an object in JavaScript is associated with an object descriptor. The descriptor contains information about the property, such as whether it is writable, enumerable, or configurable. This information is crucial for understanding and manipulating object properties.

## Conclusion:

JavaScript objects play a central role in the language, offering a flexible and powerful way to structure and organize data. Understanding the internal representation of objects, including properties, methods, prototypes, and descriptors, empowers developers to write efficient and maintainable code.

As you delve deeper into JavaScript development, mastering the intricacies of objects and their internal representation will enhance your ability to design robust and scalable applications. Objects are not just entities in JavaScript; they are the backbone of dynamic and expressive programming in the world of web development.