Activities

JavaScript - Day -1 : Introduction to Browser & web

18/02/2023 - Saturday - 12:00 PM : 2:30 PM

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

HTTP (Hypertext Transfer Protocol) is the foundation of how data is transmitted over the internet. If you're into web development, you know that HTTP/1.1 has been the go-to protocol for years. But, there's a new kid on the block, HTTP/2, that's been designed to replace HTTP/1.1. So, what's the difference between these two protocols?

Firstly, let's talk about performance. HTTP/1.1 was great for the early days of the internet, but now that websites are more complex, it's just not good enough. With its "head of line blocking" approach, HTTP/1.1 can take forever to process requests and responses, especially for sites with tons of resources. HTTP/2, on the other hand, uses "multiplexing" which allows for multiple requests and responses to be sent simultaneously. This makes for much faster loading times, even for websites with tons of resources.

Secondly, HTTP/2 is better at compression. While HTTP/1.1 uses GZIP compression, which can slow down the server and the client, HTTP/2 uses a new algorithm called HPACK which is more efficient and compresses headers.

Thirdly, HTTP/2 has a feature called server push, which means that the server can send additional resources to the client before the client even requests them. This reduces the number of round trips needed to load a webpage, which is great for performance.

Finally, HTTP/2 is more secure than HTTP/1.1. It requires the use of TLS encryption, which is more secure than the SSL encryption used in HTTP/1.1. This makes HTTP/2 less susceptible to attacks like man-in-the-middle attacks.

In conclusion, if you're into web development, you should consider using HTTP/2. It's faster, more efficient, and more secure than HTTP/1.1. But, keep in mind that not all servers and clients support HTTP/2 yet, so make sure to check with your hosting provider before making the switch.

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

JavaScript is an awesome programming language used to build websites and web applications. Objects are an essential data type used to organize and store data in JavaScript. Understanding the internal representation of objects in JavaScript is important for writing efficient code that performs well.

When we talk about objects in JavaScript, we refer to a collection of key-value pairs. Keys are strings that identify properties of the object, and values can be of any data type, including other objects. Accessing keys is done using either dot notation or bracket notation.

Under the hood, objects in JavaScript are implemented using hash tables, which is a data structure that allows for quick access to values. In a hash table, keys are hashed, generating an index into an array that stores values.

When you create a new object in JavaScript, the engine allocates memory for the object and sets up a hash table to store its properties. Each property is stored as a key-value pair in the hash table. The keys are hashed to generate an index into the hash table, and the values are stored at that index. Accessing a property of an object requires the engine to calculate the hash of the key, look up the index in the hash table, and retrieve the value.

It's important to keep in mind that the order of properties in an object is not guaranteed in JavaScript. The order in which properties are added to an object may not be the order in which they are stored in the hash table. If the order of properties is essential, then an array should be used instead.

Internal properties of an object, such as the object's prototype and its class, are used by the JavaScript engine to manage the object. These internal properties are not directly accessible from JavaScript code.

To conclude, objects in JavaScript are implemented using hash tables, which allows for quick access to values. Keys are hashed, generating an index into an array that stores values. By understanding the internal representation of objects in JavaScript, you can write more efficient and effective code, which is critical in developing complex web applications.