**JAVA SCRIPT DAY-1 INTODUCTION TO BROWSER AND WEB**

Write a blog on Difference between HTTP1.1 vs HTTP2

|  |  |
| --- | --- |
| HTTP1.1 | HTTP2 |
| * For every TCP connection there could be multiple requests and responses, and pipelining | * Uses multiplexing, where over a single TCP connection resources |
| * The client can request several resources from the server at once. | * It is delivered at interleaved and arrive at the client almost at the same time. |
| * It introduces a warning header field to carry additional information about the status of a message. | * Underlying semantics of HTTP such as headers, status codes remains the same. |
| * Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match and by using entity tags. | * HTTP/2 does not change much in terms of caching. With the server push feature if the client finds the resources are already present in the cache, it can cancel the pushed stream. |
| * Secure mechanism NTLM authentication is uded. | * For security concerns from previous versions will continue to be seen in HTTP/2. |

Write a blog about objects and its internal representation in Javascript

Objects, in JavaScript, is it’s most important data-type and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data-types like number, string, boolean, null, undefined and symbol in the sense that while these primitive data-types all store a single value each depending on their types.

Objects are complex and each object may contain a combination of primitive data-types as well as reference data-types. An object is a reference data type. Variables that are assigned a reference value are given a reference or a pointer to that value. That reference or pointer points to the location in memory where the object is stored. The variables don’t actually store the value.

A JavaScript object has properties associated with it. A property of an object can be explained as a variable that is attached to the object.

Syntax: objectName.propertyName

Unassigned properties of an object are undefined

myCar.color; //undefined

An object can be created with figure brackets {} with an optional list of properties. A property is a “key: value” pair, where a key is the property name value can be anything.

function greet(person: { name: string; age: number })

{

return "Hello " + person.name;

}