Task-1

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

* **HTTP** - **Hypertext Transfer Protocol**
* HTTP is the entire backbone of the world wide web.
* HTTP is a TCP/IP based communication protocol.
* Define how files are to be transferred between clients and servers on the world wide web.

|  |  |
| --- | --- |
| **HTTP/1.1** | **HTTP/2** |
| 1. HTTP/1.1 loads the website slowly. | 1. HTTP/2 loads the website fast. |
| 1. HTTP/1.1, the first standardizes version of HTTP, was introduced in 1997. | 1. Google introduced HTTP/2 in 2015 which is based on the SPDY protocol. |
| 1. HTTP/1.1 it requires performance optimization to load the website effectively. | 1. HTTP/2 does not require any additional optimization. |
| 1. It present significant performance optimizations (over HTTP/0.9 and HTTP/1.0) | 1. It supported multiplexing (multiple request/response sent and received asynchronously over a single TCP connection) |
| 1. HTTP/1.1 loads resources one after the other, so if one resource cannot be load, it blocks all the other resources behind it. | 1. HTTP/2 is able to use a single TCP connection to send multiple streams of data at once, so that no one resource blocks any other resource HTTP/1. |
| 1. HTTP/1.1 introduced chunked transfer encoding to allow content on persistent connection to be streamed rather than buffered. | 1. HTTP/2 no longer supports HTTP/1.1’s chunked transfer encoding mechanism, as it provide its own, more efficient, mechanism for data streaming. |

* HTTP/1 is less secured when compared with HTTP/2 and HTTP/2 is more secure when compared with HTTP/1.1
* HTTP/2 is binary, whereas HTTP1 is textual.
* HTTP/2 is fully multiplexed, instead of ordered and blocking.
* HTTP/2 can, therefore, use one connection for parallelism.
* HTTP/2 uses header compression to reduce overhead.
* HTTP/2 allows servers to “push” response proactively into client caches.
* HTTP/2 is secured by default.
* vers to “push” response proactively into client caches

**HTTP2 is much faster and more reliable than HTTP1**. HTTP1 loads a single request for every TCP connection, while HTTP2 avoids network delay by using multiplexing. HTTP is a network delay sensitive protocol in the sense that if there is less network delay, then the page loads faster.

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

Objects are important data types in javascript. Objects are different than primitive datatypes (i.e. number, string, boolean, etc.). Primitive data types contain one value but Objects can hold many values in form of Key: value pair. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

Every object has some property associated with some value. These values can be accessed using these properties associated with them.

var myCar = new Object();

myCar.make = 'Suzuki';

myCar.model = 'Altros';

myCar.year = 1978;

myCar.wheels = 2;

After creating myCar object, the value inside the object can be accessed using keys.

i.e.

myCar.year

Output: 1978

These values can be accessed using brackets notation also.

myCar.year

Output: 1978

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.

Otherway around, objects in JavaScript may be defined as an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.  
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.