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

**Answer:**

|  |  |  |
| --- | --- | --- |
| **Differentiator** | **HTTP/1.1** | **HTTP/2** |
| **Year** | **1997** | **2015** |
| **Key Features** | It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once. However, pipelining was hard to implement due to issues such as head-of-line blocking and was not a feasible solution. HTTP2 Protocol. | Uses multiplexing, where over a single TCP connection resources to be delivered are interleaved and arrive at the client almost at the same time. It is done using streams which can be prioritized, can have dependencies and individual flow control. It also provides a feature called server push that allows the server to send data that the client will need but has not yet requested. HTTP3 Protocol |
| **Status Code** | Introduces a warning header field to carry additional information about the status of a message. Can define 24 status codes, error reporting is quicker and more efficient. | Underlying semantics of HTTP such as headers, status codes remains the same. |
| **Authentication Mechanism** | It is relatively secure since it uses digest authentication, NTLM authentication. | Security concerns from previous versions will continue to be seen in HTTP/2. However, it is better equipped to deal with them due to new TLS features like connection error of type Inadequate Security. |
| **Caching** | 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. |
| **Web Traffic** | HTTP/1.1 provides faster delivery of web pages and reduces web traffic as compared to HTTP/1.0. However, TCP starts slowly and with domain shading (resources can be downloaded simultaneously by using multiple domains), connection reuse and pipelining, there is an increased risk of network congestion. | HTTP/2 utilizes multiplexing and server push to effectively reduce the page load time by a greater margin along with being less sensitive to network delays. |

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

**Answer: Objects and Its Internal Representation in JavaScript**

In simple terms. “A JavaScript object is a collection of named values having state and behavior (properties and method)”.

For example: Person, car, pen, bike, Personal Computer, Washing Machine etc.

Take the case of cars.

All cars have the same properties, but the property values differ from car to car. All cars have the same methods, but the methods are performed at different times.



Let’s have an example of my favorite Mercedes car and list out its properties (Features):

1. Make: Mercedes
2. Model: C-Class
3. Color: White
4. Fuel: Diesel
5. Weight: 850kg
6. Mileage: 8Kmpl
7. Rating: 4.5

Taking the above as reference, I'll stress up on objects, Object properties and Methods.

## ****1)Objects:****

The following code assigns a **simple value** (Mercedes) to a **variable** named car:

var car = "Mercedes";

Objects are variables too. But objects can contain many values.

The following code assigns **many values** (Mercedes, C-class, White and so on) to a **variable** named Car:

var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”, Mileage: “8Kmpl”, Rating: 4.5};

The values are written as **name: value** pairs (name and value separated by a colon).

Syntax:

var <object-name> = {key1: value1, key2: value2,... keyN: valueN};

So, conclusion and definition for JS objects is “JavaScript objects are containers for named values”.

## ****2) Object Properties****

The name: values pairs (in JavaScript objects) are called **properties**.

var car = {Make: “Mercedes”, Model: “C-Class”, Color: “White”, Fuel: Diesel, Weight: “850kg”, Mileage: “8Kmpl”, Rating: 4.5};

From the above snippet, let’s have a look what falls under property and property value:



The object properties can be different primitive values, other objects and functions.

Properties can usually be changed, added, and deleted, but some are read only.

**The syntax for adding a property to an object is :**

ObjectName.ObjectProperty = propertyValue;

**The syntax for deleting a property from an object is:**

delete ObjectName.ObjectProperty;

**The syntax to access a property from an object is:**

objectName.property        // Car.Make

//or

objectName["property”]    // Car["Make"]

//or

objectName[expression]   // x = "Make"; Car[x]

So, Conclusion and simple definition for Java Script properties is “Properties are the values associated with a JavaScript object”.

## ****3) Object Methods****

An object method is an object property containing a function definition. i.e.,

Let’s assume to start the car there will be a mechanical functionality.

function(){return ignition.on}

and so similar is to stop/brake/headlights on & off, etc.

So, Conclusion and simple definition for Java Script Object methods is “Methods are actions that can be performed on objects.”.