HTTP1.1 VS HTTP 2.0

|  |  |
| --- | --- |
| **HTTP 1.1** | **HTTP 2.0** |
| **1.HEAD OF LINE BLOCKING:**   * TCP connection is been blocked by the proceeding request until the server response to it, till then the following request has to wait * Loads a single request for every TCP connection. This can lead to network delays. | **NO MORE HEAD OF LINE BLOCKING**   * Introduces **multiplexing**, allowing multiple requests and responses to be sent over a single TCP connection simultaneously. This reduces latency. |
| **2.REDUNDANCY IN REQUEST HEADER**   * Since the http request is stateless, we are forced to send the same static header again and again * Resources are fetched incrementally, with multiple back-and-forth exchanges between the server and client. | * Adds **header compression**, which reduces the overhead of sending r header information with each request and response. * It allows to compress the http header by using **HPACK** * It Includes a **server push function**, enabling the server to proactively send additional resources (such as stylesheets, scripts, or images) to the client before they are explicitly requested |
| 3. Keeps all requests and responses in plain text format. | 3.It uses binary framing layer to encapsulate messages in binary format |

NOTE :

**HTTP 2.0** is faster, more reliable, and better suited for today’s web applications, especially those with rich media content

OBJECTS AND ITS INTERNAL REPRESENTAION

An object is a group of data that is stored of series of name value pairs encapsulated in on entity.

Let’s have an example of merc car and list out its properties:

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

Taking the above as reference and stress up on Objects, Object properties and Methods.

**1.OBJECT**

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

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

**Syntax:**

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

**2)Object Properties**

The name: values pairs are called **properties**.

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

|  |  |
| --- | --- |
| **Property** | **Property value** |
| Make | Mercedes |
| Model | C-class |
| Colour | White |
| Fuel | Diesel |
| Weight | 850kg |
| Mileage | 8kmpl |
| Rating | 4.5 |

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

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

* 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]

**3.Object Methods**

* An object method is an object property containing a function definition.
* 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, Java Script Object methods is “Methods are actions that can be performed on objects.”