1. **Difference between HTTP1.1 vs HTTP2:**

HTTP stands for hypertext transfer protocol & it is used in client-server communication. By using HTTP user sends the request to the server & the server sends the response to the user. There are several stages of development of HTTP but we will focus mainly on HTTP/1.1 which was created in 1997 & the new one is HTTP/2 which was created in 2015.

**HTTP/1.1:** For better understanding, let’s assume the situation when you make a request to the server for the GUVI.html page & server responds to you as a resource GUVI.html page. before sending the request and the response there is a TCP connection established between client & server. again you make a request to the server for image img.jpg & the server gives a response as an image img.jpg. the connection was not lost here after the first request because we add a keep-alive header which is the part of the request so there is an open connection between the server & client. there is a persistent connection which means several requests & responses are merged in a single connection. These are the drawbacks that lead to the creation of HTTP/2: The first problem is HTTP/1.1 transfer all the requests & responses in the plain text message form. The second one is head of line blocking in which TCP connection is blocked all other requests until the response does not receive. all the information related to the header file is repeated in every request.

**HTTP/2:** HTTP/2 was developed over the SPDY protocol. HTTP/2 works on the binary framing layer instead of textual that converts all the messages in binary format. it works on fully multiplexed that is one TCP connection is used for multiple requests. HTTP/2 uses HPACK which is used to split data from header. it compresses the header. The server sends all the other files like CSS & JS without the request of the client using the PUSH frame.

****Difference between HTTP/1.1 and HTTP/2 are:****

| **HTTP/1.1** | **HTTP/2** |
| --- | --- |
| It works on the textual format. | It works on the binary protocol. |
| There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. | It allows multiplexing so one TCP connection is required for multiple requests. |
| It uses requests resource Inlining for use getting multiple pages | It uses PUSH frame by server that collects all multiple pages |
| It compresses data by itself. | It uses HPACK for data compression. |

1. ****Multiplexing****: HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. In contrast, 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/2 does this by splitting data into binary-code messages and numbering these messages so that the client knows which stream each binary message belongs to.
2. ****Server push****: Typically, a server only serves content to a client device if the client asks for it. However, this approach is not always practical for modern webpages, which often involve several dozen separate resources that the client must request. HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. The server also sends a message letting the client know what pushed content to expect – like if Bob had sent Alice a Table of Contents of his novel before sending the whole thing.
3. ****Header compression****: Small files load more quickly than large ones. To speed up web performance, both HTTP/1.1 and HTTP/2 compress HTTP messages to make them smaller. However, HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets. This eliminates a few bytes from every HTTP packet. Given the volume of HTTP packets involved in loading even a single webpage, those bytes add up quickly, resulting in faster loading.
4. ****Prioritization:**** Prioritization refers to the order in which pieces of content are loaded. Prioritization affects a webpage's load time. For example, certain resources, like large JavaScript files, may block the rest of the page from loading if they have to load first. More of the page can load at once if these render-blocking resources load last. In HTTP/2, developers have hands-on, detailed control over prioritization. This allows them to maximize perceived and actual page load speed to a degree that was not possible in HTTP/1.1. HTTP/2 offers a feature called weighted prioritization. This allows developers to decide which page resources will load first, every time.
5. **Javascript Objects:**

## Methods are ****actions**** that can be performed on objects.

## Object properties can be both primitive values, other objects, and functions.

An ****object method**** is an object property containing a ****function definition****.

With JavaScript, you can define and create your own objects.

There are different ways to create new objects:

* Create a single object, using an object literal.
* Create a single object, with the keyword new.
* Define an object constructor, and then create objects of the constructed type.
* Create an object using Object.create().

## Using an Object Literal

This is the easiest way to create a JavaScript Object.Using an object literal, you both define and create an object in one statement.An object literal is a list of name:value pairs (like age:50) inside curly braces {}.

Let’s have an example of my favorite merc 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 soo 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.

 (I’m just trying to roll a picture in your mind. The car will not start / stop by means of code. I assume you’d get it).

You can also look into many examples & operations **[here](https://www.w3schools.com/js/js_object_definition.asp" \l ":~:text=Ruby and Perl-,Object Methods,property containing a function definition.&text=JavaScript objects are containers for named values, called properties and methods." \t "https://www.linkedin.com/pulse/_blank)**.

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

Hope this will be useful.