**1) What is HTTP?**

[HTTP](https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/) stands for hypertext transfer protocol, and it is the basis for almost all web applications. More specifically, HTTP is the method computers and servers use to request and send information. For instance, when someone navigates to cloudflare.com on their laptop, their web browser sends an HTTP request to the Cloudflare servers for the content that appears on the page. Then, Cloudflare servers send HTTP responses with the text, images, and formatting that the browser displays to the user.

**What is HTTP/1.1?**The first usable version of HTTP was created in 1997. Because it went through several stages of development, this first version of HTTP was called HTTP/1.1. This version is still in use on the web.

**What is HTTP/2?**In 2015, a new version of HTTP called HTTP/2 was created. HTTP/2 solves several problems that the creators of HTTP/1.1 did not anticipate. In particular, HTTP/2 is much faster and more efficient than HTTP/1.1. One of the ways in which HTTP/2 is faster is in how it prioritizes content during the loading process.

**2) What are the differences between HTTP/2 and HTTP/1.1?**

**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](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) 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.

**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.

**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.

# **3) Objects and its internal representation in JavaScript**

JavaScript is designed on a simple object-based paradigm. An object is a collection of properties, and a property is an association between a name (or key) and a value. A property's value can be a function, in which case the property is known as a 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.

**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”, colour: “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”, colour: “White”, Fuel: Diesel, Weight: “850kg”, Mileage: “8Kmpl”, Rating: 4.5};

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.

**Syntax:** function () {return ignition. On}

Simple definition for Java Script Object methods is “Methods are actions that can be performed on objects”.

**Enumerating properties:**

There are three native ways to list/traverse object properties:

for...in loops. This method traverses all the enumerable string properties of an object as well as its prototype chain.

Object.keys(). This method returns an array with only the enumerable own string property names ("keys") in the object myObj, but not those in the prototype chain.

Object.getOwnPropertyNames(). This method returns an array containing all own string property names in the object myObj, regardless of if they are enumerable or not.