

1. Difference between HTTP/2 and HTTP/1.1

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.

HTTP is a top-level application protocol that exchanges information between a client computer and a local or remote web server. In this process, a client sends a text-based request to a server by calling a method like GET or POST. In response, the server sends a resource like an HTML page back to the client.

For example, let's say you are visiting a website at the domain www.example.com. When you navigate to this URL, the web browser on your computer sends an HTTP request in the form of a text-based message.

This request uses the GET method, which asks for data from the host server listed after Host:. In response to this request, the [example.com](http://www.example.com) web server returns an HTML page to the requesting client, in addition to any images, stylesheets, or other resources called for in the HTML.

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.

2. 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

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

//or

objectName["property"]

//or

objectName[expression]

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

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.”

Create JavaScript Object with Object Literal

One of easiest way to create a javascript object is object literal, simply define the property and values inside curly braces as shown below

```
let bike = {name: 'SuperSport', maker:'Ducati', engine:'937cc'};
```

Create JavaScript Object with Constructor

Constructor is nothing but a function and with help of new keyword, constructor function allows to create multiple objects of same flavor as shown below

```
function Vehicle(name, maker) {  
    this.name = name;  
    this.maker = maker;  
}  
let car1 = new Vehicle('Fiesta', 'Ford');  
let car2 = new Vehicle('Santa Fe', 'Hyundai')  
console.log(car1.name); //Output: Fiesta  
console.log(car2.name); //Output: Santa Fe
```

Using the JavaScript Keyword new

The following example also creates a new JavaScript object with four properties:

```
var person = new Object();  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```

Using the Object.create method

Objects can also be created using the Object.create() method. This method can be very useful, because it allows you to choose the prototype object for the object you want to create, without having to define a constructor function.

```
// Animal properties and method encapsulation
```

```
var Animal = {  
  type: 'Invertebrates', // Default value of properties  
  displayType: function() { // Method which will display type of Animal  
    console.log(this.type);  
  }  
};  
// Create new animal type called animal1  
var animal1 = Object.create(Animal);  
animal1.displayType(); // Output:Invertebrates  
// Create new animal type called Fishes  
var fish = Object.create(Animal);  
fish.type = 'Fishes';  
fish.displayType();  
// Output:Fishes
```