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

|  |  |
| --- | --- |
| **HTTP/2** | **HTTP/1.1** |
| In 2015, a new version of HTTP called HTTP/2 was created. | The first usable version of HTTP was created in 1997 and was called HTTP/1.1. |
| It works on the binary protocol. | It works on the textual format. |
| **Multiplexing:** 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/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. |
| **Server push:** HTTP/2 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. | HTTP/1.1 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 allows to maximize perceived and actual page load speed. | HTTP/1.1 has limited to actual page load speed. |
| It uses HPACK for data compression. | It compresses data by itself. |

**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”.