**Difference Between HTTP 1.1 and HTTP 2**

|  |  |
| --- | --- |
| **HTTP /1.1** | **HTTP /2** |
| The first usable version of HTTP was created in 1997 | 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. |
| It works on the textual format. | It works on the binary protocol. |
| 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. |
| 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. |
| HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded | HTTP/2 is able to use a single TCP connection to send multiple streams of data at once |

1. objects and its internal representation in Javascript

**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.
* Objects are more complex and each object may contain any combination of these primitive data-types as well as reference data-types.  
  An object, is a reference data type. Variables that are assigned a reference value are given a reference or a pointer to that value.
* A JavaScript object has properties associated with it. A property of an object can be explained as a variable that is attached to the object. Object properties are basically the same as ordinary JavaScript variables, except for the attachment to objects. The properties of an object define the characteristics of the object. You access the properties of an object with a simple dot-notation.

objectName.propertyName

* Like all JavaScript variables, both the object name (which could be a normal variable) and property name are case sensitive. You can define a property by assigning it a value. For example, let’s create an object named my Car and give it properties named make, model, and year as follows:

var myCar = new Object();  
myCar.make = 'Toyoto';  
myCar.model = 'Innovo';  
myCar.year = 1996;