| 1. **DIFFERENCE BTW**   **HTTP/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. |

**2)Objects and its internal representation in Javascript**

Object:

In JavaScript, an object is a standalone entity, with properties and type. Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc. The same way, JavaScript objects can have properties, which define their characteristics.

**Creating Objects in JavaScript:**

1. By object literal
2. By creating instance of Object directly (using new keyword)

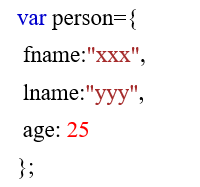
**By object literal:**

The syntax of creating object using object literal is given below:



Property and value is separated by colon(:).

**Example:**



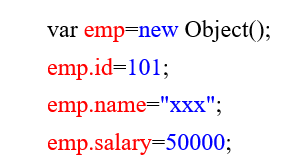
**By creating instance of Object directly (using new keyword):**

The syntax of creating object directly is given below:



Here, **new keyword** is used to create object.

**Example:**



**Accessing JavaScript Objects:**

The syntax for accessing the property of an object is

*objectName.property*

or

*objectName*[“*property*”]

Accessing ‘fname’ from example 1 using dot operator,



Accessing ‘name’ form example 2 using [],

