|  |  |
| --- | --- |
| **HTTP1.1** | **HTTP2** |
| 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. |
| HTTP1.1 uses plaintext protocol. | HTTP2 is more secure as it uses binary protocol instead of plaintext |

objects and its internal representation in Javascript:

Objects are important data types in javascript. Objects are different than primitive datatypes. 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 in JavaScript may be defined as an unordered collection of related data.

Two types of object notation in Java script:

1.Json- Java script object notation.

2.oops object notation

Json- Java script object notation: Json uses text format of object notation (i.e key:value pair)

Example:

Student={

{Name: name of student;

Roll.no: roll number of student;

Section: Section of student in school}

{Name: name of student;

Roll.no: roll number of student;

Section: Section of student in school}

}

This json object groups the unordered collection of data related to two students.

2.oops object notation:

Objects using oops introduced in ECMAScript15. It holds related data based on class.

Example:

class Employee

  {

    constructor(id,name)

    {

      this.id=id;

      this.name=name;

    }

    detail()

    {

  document.writeln(this.id+" "+this.name+"**<br>**")

    }

  }

var e1=new Employee(101,"Martin Roy");

var e2=new Employee(102,"Duke William");

e1.detail();

e2.detail();

it holds data of employees in object format. Data of an employee can be accessed using employee object reference.