1. Difference between HTTP1.1 and HTTP2

|  |  |
| --- | --- |
| **HTTP1.1** | **HTTP2** |
| * HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. | * HTTP/2 is able to use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resource. |
| * 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. |
| * **HTTP/1.1 is a well-established and widely supported protocol**that is simple to understand and implement. | * **HTTP/2 is a more efficient and secure protocol that offers better performance and reduced latency.** |

1. Objects and its internal representation in java script

* **A Javascript object is a collection of Named values**
* Objects are written with the const keyword
* Example:

const person = {

firstName: "John",

lastName: "Doe",

age: 50,

eyeColor: "blue"

};

* Objects in JavaScript may be defined as an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.
* The property of an object is accessed with a simple dot-notation:

**objectName.propertyName**

* Like all Javascript variables both the object name and property name are case sensitive
* For example, let’s create an object named myCar and give it properties named make, model, and year as follows:

var myCar = new Object();  
myCar.make = 'Ford';  
myCar.model = 'Mustang';  
myCar.year = 1969;

* Unassigned properties of an object are undefined

myCar.color; //Undefined

* Properties of JavaScript objects can also be accessed or set using a bracket notation=. Objects are sometimes called associative arrays, since each property is associated with a string value that can be used to access it. So, for example, you could access the properties of the myCar object as follows:

myCar['make'] = 'Ford';  
myCar['model'] = 'Mustang';  
myCar['year'] = 1969;