Question 2: **Differences between HTTP 1.1 and HTTP 2**

|  |  |  |  |
| --- | --- | --- | --- |
| S. No | **Differentiator** | **HTTP 1.1** | **HTTP 2** |
| 1. | **Year** | 1991 | 2015 |
| 2. | **Key Features** | 1. **Connection Re-Use:**   For every TCP connection there could be multiple requests and responses.   1. **Pipelining:**   The client can request several resources from the server at once. | 1. **Multiplexing:**   Over a single TCP connection resources are interleaved and arrive at the client almost at the same time.   1. **Server Push:**   Allows the server to send data that the client will need but has not yet requested. |
| 3. | **Status Code** | It can define up to 24 status codes. Error reporting is quicker and efficient. | Status code and headers remains the same. |
| 4. | **Authentication Mechanism** | It is secure as it uses digest authentication, NLM authentication | New TLS features like connection error |
| 5. | **Caching** | Expands caching support by using additional headers like cache-control, conditional headers and entity tags. | No change in caching. With the server push feature if the client finds the resources are already present in the cache, it cancels the push stream. |
| 6. | **Web Traffic** | Increased risk of web congestion. | Reduces page load time. |

**Question 3: Objects and its internal representation in JavaScript**

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.

**Objects and its Properties:**

A property of an object can be explained as a variable that is attached to the object. We can define property by assigning a value. Unassigned properties of an object are undefined.

**Creating Objects in JavaScript:**

1. **By using object literal:**

Simply defining the property and value

Example: Let mobilephone= {name: “S60”, company: “Samsung”, RAM: “4GB”

1. **By using constructors:**

Constructor is a function and it allows to create multiple objects of same kind.

Example:

Function Mobile (name, company) {

This.name=name

This. maker=maker

1. **By using the keyword new**

Example: var person new Object ()

1. **Using object.create method:**

This method can be very useful, because it allows you to choose the prototype object for the object you want to create, without having to define a constructor function.