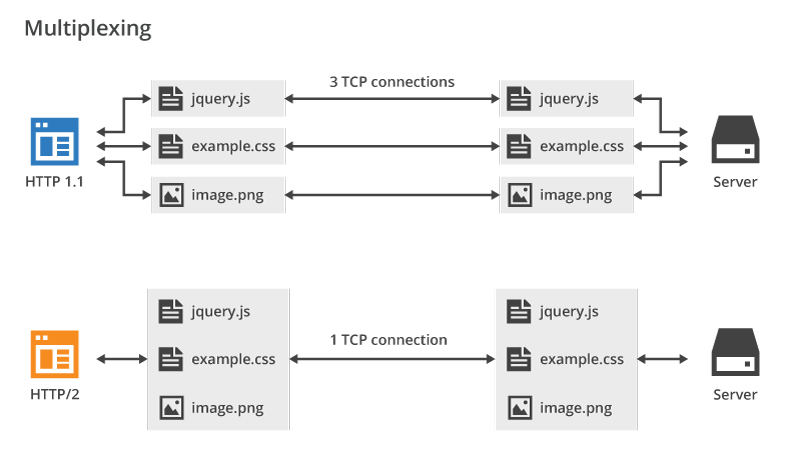
HTTP1.1 VS HTTP2.0

HTTP stands for hypertext transfer protocol & it is used in client-server communication. By using HTTP user sends the request to the server & the server sends the response to the user.

**Difference between HTTP/1.1 and HTTP/2 are:**

| **HTTP/1.1** | **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 In lining 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 vs. HTTP/2 Protocol**

HTTP/2 improved on HTTP/1.1 in a number of ways that allowed for speedier content delivery and improved user experience, including

* **Binary protocols** – Binary protocols consume less bandwidth, are more efficiently parsed and are less error-prone than the textual protocols. Additionally, they can better handle elements such as whitespace, capitalization and line endings.
* **Multiplexing** – HTTP/2 is multiplexed, i.e., it can initiate multiple requests in parallel over a single TCP connection. As a result, web pages containing several elements are delivered over one TCP connection. These capabilities solve the head-of-line blocking problem in HTTP/1.1, in which a packet at the front of the line blocks others from being transmitted.
* **Header compression** – HTTP/2 uses header compression to reduce the overhead caused by TCP’s [slow-start](https://en.wikipedia.org/wiki/TCP_congestion_control#Slow_start) mechanism.
* **Server push** – HTTP/2 servers push likely-to-be-used resources into a browser’s cache, even before they’re requested. This allows browsers to display content without additional request cycles.
* **Increased security** – Web browsers only support HTTP/2 via encrypted connections, increasing user and application security.

Objects & its internal representation in JavaScript

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.

**JavaScript Objects**

* Booleans can be objects (if defined with the new keyword)
* Numbers can be objects (if defined with the new keyword)
* Strings can be objects (if defined with the new keyword)
* Dates are always objects.
* Maths are always objects.
* Regular expressions are always objects.
* Arrays are always objects.

A javaScript object is an entity having state and behavior (properties and method). For example: car, pen, bike, chair, glass, keyboard, monitor etc.

JavaScript is an object-based language. Everything is an object in JavaScript.

JavaScript is template based not class based. Here, we don't create class to get the object. But, we direct create objects.

JavaScript Object by object literal

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

object={property1:value1,property2:value2.....propertyN:valueN}

As you can see, property and value is separated by : (colon).

Let’s see the simple example of creating object in JavaScript.

1. **<script>**
2. emp={id:102,name:"Shyam Kumar",salary:40000}
3. document.write(emp.id+" "+emp.name+" "+emp.salary);
4. **</script>**

Output is 102 shyam kumar 40000