|  |  |  |
| --- | --- | --- |
| **Key Point** | **HTTP/1.1** | **HTTP/2** |
| Year | 1991 | 1997 |
| Key Features | It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once.  HTTP2 Protocol | Uses multiplexing, where over a single TCP connection resources to be delivered are interleaved and arrive at the client almost at the same time  HTTP3 Protocol |
| Authentication Mechanism | It is relatively secure since it uses digest authentication, NTLM authentication. | Security concerns from previous versions will continue to be seen in HTTP/2. However, it is better equipped to deal with them due to new TLS features like connection error of type Inadequate\_Security. |
| Caching | Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match and by using entity tags. | HTTP/2 does not change much in terms of caching. With the server push feature if the client finds the resources are already present in the cache, it can cancel the pushed stream. |
| Web Traffic | HTTP/1.1 provides faster delivery of web pages and reduces web traffic as compared to HTTP/1.0. However, TCP starts slowly and with domain sharding (resources can be downloaded simultaneously by using multiple domains), connection reuse and pipelining, there is an increased risk of network congestion. | HTTP/2 utilizes multiplexing and server push to effectively reduce the page load time by a greater margin along with being less sensitive to network delays. |
| Protocol Type | Text based protocol that is in the readable form. | It is a binary protocol (HTTP requests are sent in the form of 0s and 1s). Needs to be converted back from binary in order to read it. |

2.HTTP version history

|  |  |
| --- | --- |
| **Year** | **HTTP Version** |
| 1991 | 0.9 |
| 1996 | 1.0 |
| 1997 | 1.1 |
| 2015 | [2.0](https://en.wikipedia.org/wiki/HTTP/2) |
| Draft (2020) | [3.0](https://en.wikipedia.org/wiki/HTTP/3) |

3. List 5 differences between Browser JS vs Node Js.

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Browser JS** | **Node Js** |
| 1. | JavaScript is a simple programming language which runs in any browser JavaScript Engine. | Node JS is an interpreter or running environment for a JavaScript programming language which holds a lot of excesses require libraries which can easily be accessed from JavaScript programming for better use. |
| 2. | JavaScript is normally used for any client-side activity for one web application. Activity can be addressing business validation or dynamic page display in some schedule time interval or basic Ajax call kind of task. Those are used on a maximum time for any web application. | Node JS mainly used for accessing or running any operating system for non-blocking operation. An operation like create or executing shell script, or getting some specific hardware related information on one call or installed certificate details in the system or lot of define task which are non-blocking on an operating system. |
| 3. | JavaScript running in any engine like Spider monkey (FireFox), JavaScript Core (Safari), V8 (Google Chrome). So JavaScript programming is very easy to write and put any running environment means proper browser. | Node JS only support V8 engine which googles chrome specific. But whether it support V8 engine, written JavaScript code can able to run in any environment. So there has no browser specific constraint on it. |
| 4. | JavaScript is normally following [Java Programming language](https://www.educba.com/java-programming-language-features/) standard. There may have some different way of writing code but at the same time, we can say it following Java Programming language standard. | node JS is [written in C++](https://www.educba.com/c-plus-plus-interview-questions/), and provide V8 engine base browser javascript running engine which helps us to run written javascript program in any browser environment. |
| 5 | For accessing any operating system specific non-blocking task JavaScript has some specific object but all of them are operating system specific. An example is ActiveX Control which is only running in Windows. | Node JS is given utility to run some operating system specific non-blocking tasks from any JavaScript programming. It doesn’t have any operating system specific constant. Node JS is very much familiar to create a specific binding with the file system, and also allowing developer for reading or sometimes write on disk. |

4. what happens when you type a URL in the address bar in the browser?

You enter a URL into a web browser

1. The browser looks up the IP address for the domain name via DNS
2. The browser sends a HTTP request to the server
3. The server sends back a HTTP response
4. The browser begins rendering the HTML
5. The browser sends requests for additional objects embedded in HTML (images, css, JavaScript) and repeats steps 3-5.
6. Once the page is loaded, the browser sends further async requests as needed.

That’s really it. Here’s a description in words for this site.

When you type “[https://example.com”](https://example.com) into your browser the first thing that happens is a Domain Name Server (DNS) matches “[example.com](http://wsvincent.com/)” to an IP address. Then the browser sends an HTTP request to the server and the server sends back an HTTP response. The browser begins rendering the HTML on the page while also requesting any additional resources such as CSS, JavaScript, images, etc. Each subsequent request completes a request/response cycle and is rendered in turn by the browser. Then once the page is loaded some sites (though not current site) will make further asynchronous requests.