

1. Difference between HTTP1.1 vs HTTP2

HTTP/1.1	HTTP/2
Pipelining and Head-of-Line Blocking	Advantages of the Binary Framing Layer
Resource In-lining	Server Push
Relies on the transport layer to avoid buffer overflow, each new TCP connection requires a separate flow control mechanism.	Multiplexes streams within a single TCP connection, and will have to implement flow control in a different manner.

2. http version history

HTTP (HyperText Transfer Protocol) is the underlying protocol of the World Wide Web. Developed by Tim Berners-Lee and his team between 1989-1991, HTTP has seen many changes, keeping most of the simplicity and further shaping its flexibility.

HTTP has evolved from an early protocol to exchange files in a semi-trusted laboratory environment, to the modern maze of the Internet, now carrying images, videos in high resolution and 3D.

HTTP/0.9 – The one-line protocol

The initial version of HTTP had no version number; it has been later called 0.9 to differentiate it from the later versions. HTTP/0.9 is extremely simple: requests consist of a single line and start with the only possible method GET followed by the path to the resource.

HTTP/1.0 – Building extensibility

HTTP/0.9 was extremely limited and both browsers and servers quickly extended it to be more versatile:

- Versioning information is now sent within each request (HTTP/1.0 is appended to the GET line)
- A status code line is also sent at the beginning of the response, allowing the browser itself to understand the success or failure of the request and to adapt its behavior in consequence
- With the help of the new HTTP headers, the ability to transmit other documents than plain HTML files has been added.

HTTP/1.1 – The standardized protocol

HTTP/1.1 clarified ambiguities and introduced numerous improvements:

- A connection can be reused, saving the time to reopen it numerous times to display the resources embedded into the single original document retrieved.
- Pipelining has been added, allowing to send a second request before the answer for the first one is fully transmitted, lowering the latency of the communication.
- Additional cache control mechanisms have been introduced.

HTTP/2 – A protocol for greater performance

The HTTP/2 protocol has several prime differences from the HTTP/1.1 version:

- It is a binary protocol rather than text. It can no longer be read and created manually. Despite this hurdle, improved optimization techniques can now be implemented.

- It is a multiplexed protocol. Parallel requests can be handled over the same connection, removing the order, and blocking constraints of the HTTP/1.x protocol.
- It compresses headers. As these are often similar among a set of requests, this removes duplication and overhead of data transmitted.
- It allows a server to populate data in a client cache, in advance of it being required, through a mechanism called the server push.

HTTP/3 - HTTP over QUIC

Post-HTTP/2 evolution - The next major version of HTTP, HTTP/3, will use QUIC instead TCP/TLS for the transport layer portion.

QUIC is a general-purpose transport layer network protocol.

3. List 5 difference between Browser JS(console) vs Nodejs

Browser JS (Console)	Node JS
"window" is a predefined global object which has functions and attributes, that have to deal with window that has been drawn.	Does not have a predefined "window" object
"location" is predefined object in browsers, that has all the information about the url loaded.	Does not require "location" object
"document", is predefined global variable in browsers, has the html which is rendered.	"document" object not required as this never have to render anything in a page.
Can have an object named "global", but it will be the exact one as "window".	Node has a predefined global object "global". It contains several functions that are not available in browsers, as they are needed for server side works only.
Does not have "require" predefined – it can be included in app for asynchronous file loading.	"require" object is predefined in Node which is used to include modules in the app.
Moduling is not mandatory	All code must be kept inside a module.

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

The browser looks up the IP address for the domain name via DNS for the entered URL's web server name. The browser sends a HTTP request to the server. The server sends back a HTTP response. Browser parses the received response data and presents to the user.