|  |  |
| --- | --- |
| Http 1.1 | Http 1.2 |
| Slower than Http 1.2 | Faster than Http 1.1 |
| Do not use prioritization | Uses prioritization to load a web page and so effective. |
| It loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it | It is able to use a single [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) connection to send multiple streams of data at once so that no one resource blocks any other resource.  (Multiplexing) |
| Will not do server push. Will provide content only if client looks. | Will do server push. |
| Compresses Http messages but not advanced. | Advanced compression called Hpack that eliminates redundant information. |

Difference between Http 1.1 and Http 1.2

Http version history

A simple protocol to exchange documents is the Hypertext transfer protocol. The protocol used in the earlier version were simple and sometimes one line. The first protocol didn’t have any number and later to identify it was named as http 0.9.It started with the only possible method ‘get’.Later, servers and browsers quickly extended them to be more versatile and hence 1.1 and further versions were created.

Difference between node js and browser js console

* Node.js apps brings the comfort of programming everything - the frontend and the backend - in a single language.
* In the browser, most of the time what you are doing is interacting with the DOM, or other Web Platform APIs like Cookies. Those do not exist in Node.js, of course. You don't have the document, window and all the other objects that are provided by the browser.
* Another big difference is that in Node.js you control the environment. Unless you are building an open source application that anyone can deploy anywhere, you know which version of Node.js you will run the application on. Compared to the browser environment, where you don't get the luxury to choose what browser your visitors will use, this is very convenient
* Another difference is that Node.js uses the Common JS module system, while in the browser we are starting to see the ES Modules standard being implemented.
* In the browser, we don't have all the APIs that Node.js provides through its modules, like the filesystem access functionality.

When you type URL in the address bar of the browser, The browser checks the cache for a DNS record to find the corresponding IP address. First, it checks the browser cache, then OS cache, then router cache and ISP cache.  If the requested URL is not in the cache, ISP’s DNS server initiates a DNS query to find the IP address of the server that hosts the URL. The browser initiates a TCP connection with the server. The browser sends an HTTP request to the webserver. The server handles the request and sends back a response. The server send out the HTML content.