1. **Difference between HTTP1.1 and HTTP2**

* Mutliplexing:

HTTP1.1 loads resources synchronously like one after the other, so this inturn acts as a bottleneck cause if one resource cannot be loaded, it blocks all the resources behind or after the current resource.

Whereas on the other hand HTTP2 uses a single TCP connection to send multiple streams of data at once so that no resource blocks the other resources. This is achieved by splitting the data into binary-code and numbering these messages so that the client knows which stream each binary message belongs.

* Server Push:

In HTTP1.1 Server serves content to the client only when the clients request for the resources. But this approach is not effective or practical for modern web pages, which have a humongous amount of resources that client need to request.

HTTP2 solves this by allowing the server to push content to the client before the client ask for it. The server also sends a message letting the client know what pushed content to expect.

* Header Comparison:

Small files load more faster than the large ones. To improve the speed and performance of the web pages both HTTP1.1 and HTTP2 compress HTTP messages to make them smaller, but HTTP2 uses a more advanced compression method called HPACK and results in fast loading.

In a conclusion the advantage of HTTP2 over HTTP1.1 are loading multiple resources at a time in a parallel way, pushing the content to client without waiting for individual resource requests and finally making use of header compression technique to improve the performance.

1. **List 5 differences between Browser JS(console) vs Nodejs?**

|  |  |
| --- | --- |
| **NODE** | **BROWSER** |
| Node doesn’t have a predefined “window” object cause it doesn’t have a window to draw anything. | “Window” is a predefined global object which has functions and attributes, that have to deal with window that has |
| Node doesn’t have “document” object also, cause it never have to render anything in a page. | “document”, which is also another predefined global variable in browsers, has the html which is rendered. |
| In Node everything is a module. You must keep your code inside a module. | Moduling is not mandatory in client side Javascript, i.e. in browsers. |
| “require” object is predefined in Node which is used to include modules in the app. | Browsers doesn’t have “require” predefined. You may include it in your app for asynchronous file loading. |
| Node processes request object. | Browser processes response objects. |
| Javascript moves so fast, this means that you can write all the modern ES6-7-8-9 Javascript that your Node.js version supports. | Browsers are bit slow and sometimes on the web, you are stuck with using older JS/ECMAScript releases. Babel is used to transform code to ES-5 compatible before shipping it to the browser. |