**11.05.2021**

**TASK 01**

**Question 1. Difference between HTTP1.1 vs HTTP2**

|  |  |
| --- | --- |
| **HTTP 1.1** | **HTTP 2** |
| HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. | HTTP/2 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. And does this by splitting data into binary-code messages and numbering these messages so that the client knows which stream each binary message belongs to. |
| a server only serves content to a client device if the client asks for it. However, this approach is not always practical for modern webpages, which often involve several dozen separate resources that the client must request | HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. The server also sends a message letting the client know what pushed content to expect – like if Bob had sent Alice a Table of Contents of his novel before sending the whole thing. |

**Question 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
* HTTP/1.0 – Building Extensibility
* HTTP/1.1 – Standadized Protocol

**Even if 1.1 protocol revised its two versions**

* + **Using HTTP for Secure Tranmission**
  + **Using HTTP for complex applications**
* HTTP/2 – A protocol for greater performances

**Question 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. | Node doesn't have a predefined "window" object cause it doesn't have a window to draw anything. |
| "location" is another predefined object in browsers, that has all the information about the url we have loaded. | "location" object is related to a particular url; that means it is for page specific. So, node doesn't require that. |
| "document", which is also another predefined global variable in browsers, has the html which is rendered. | Of course Node doesn't have "document" object also, cause it never have to render anything in a page. |
| Browsers may have an object named "global", but it will be the exact one as "window". | Node has "global", which is a predefined global object. It contains several functions that are not available in browsers, cause they are needed for server side works only. |
| Browsers don't have "require" predefined. You may include it in your app for asynchronous file loading. | "require" object is predefined in Node which is used to include modules in the app. |
| Moduling is not mandatory in client side JavaScript, i.e. in browsers. | In Node everything is a module. You must keep your code inside a module. |
| Browsers are not headless. | Node is headless. |
| Browsers processes response objects. | Node processes request object. |

**Question 4. what happens when you type a URL in the address bar in the 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 a sync requests as needed.