**TASK-1**

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

-🡪HTTP1.1-In this process, a client sends a text-based request to a server by calling a method like GET or POST. In response, the server sends a resource like an HTML page back to the client. This request uses the GET method, which asks for data from the host server listed after Host.

HTTP2-HTTP/2 began as the SPDY protocol, developed primarily at Google with the intention of reducing web page load latency by using techniques such as compression, multiplexing, and prioritization.

🡪From a technical point of view, one of the most significant features that distinguishes HTTP/1.1 and HTTP/2 is the binary framing layer, which can be thought of as a part of the application layer in the internet protocol stack. As opposed to HTTP/1.1, which keeps all requests and responses in plain text format, HTTP/2 uses the binary framing layer to encapsulate all messages in binary format, while still maintaining HTTP semantics, such as verbs, methods, and headers.

🡪HTTP/1.1 transfers/delivers in plain-text messages, HTTP/2 encodes these into binary, allowing for significantly different delivery model possibilities.

🡪HTTP/2 establishes a single connection object between the two machines. Within this connection there are multiple streams of data. Each stream consists of multiple messages in the familiar request/response format. Finally, each of these messages split into smaller units called frames.

🡪**Main goals of developing HTTP/2 were-**

High-level compatibility with HTTP/1.1 — methods, status codes, URIs and header fields.

Page load speed improvements trough

Compression of request headers

Binary protocol

HTTP/2 Server Push

Request multiplexing over a single TCP connection

Request pipelining

HOL blocking (Head-of-line) — Package blocking

**2. HTTP version history**

The Hypertext Transfer Protocol (HTTP) is an [application layer](https://en.wikipedia.org/wiki/Application_layer) protocol for distributed, collaborative, [hypermedia](https://en.wikipedia.org/wiki/Hypermedia) information systems. HTTP is the foundation of data communication for the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), where [hypertext](https://en.wikipedia.org/wiki/Hypertext) documents include [hyperlinks](https://en.wikipedia.org/wiki/Hyperlink) to other resources that the user can easily access, for example by a [mouse](https://en.wikipedia.org/wiki/Computer_mouse) click or by tapping the screen in a web browser.

Development of HTTP was initiated by [Tim Berners-Lee](https://en.wikipedia.org/wiki/Tim_Berners-Lee) at [CERN](https://en.wikipedia.org/wiki/CERN) in 1989.The different versions of HTML are as below,

**HTML 1.0**

HTML 1.0 was the first release of HTML to the world. Not many people were involved in website creation at the time, and the language was very limiting. There really wasn’t much you could do with it bar getting some simple text onto the web.

**HTML 2.0**

HTML 2.0 included everything from the original 1.0 specifications but added a few new features to the mix. It was the standard for website design until January 1997 and defined many core HTML features for the first time.

**HTML 3.0**

More and more people were getting into the HTML game around now, and while the previous standards offered some decent abilities to webmasters (as they became known), they thirsted for more abilities and tags. They wanted to enhance the look of their sites.

This is where trouble started. A company called Netscape was the clear leader in the browser market at the time, with a browser called Netscape Navigator. To appease the cries of the HTML authors, they introduced new proprietary tags and attributes into their Netscape Navigator browser. These new abilities were called Netscape extension tags. This caused big problems as other browsers tried to replicate the effects of these tags so as not to be left behind but could not get their browsers to display things the same way. This meant that if you designed a page with Netscape ETs, the page would look bad in other browsers. This caused confusion and irritation for the markup pioneers.

**HTML 3.2**

The browser-specific tags kept coming, and it became increasingly apparent that a standard needed to be found. To this end, the [World Wide Web Consortium](http://www.w3.org/) was founded in 1994 to standardize the language and keep it evolving in the right direction. Their first work was code-named WILBUR, and later became known as  [HTML 3.2](http://www.w3.org/TR/REC-html32). This was a toned-down change to the existing standards, leaving many of the big steps forward for later versions. Most of the extensions tags that had been introduced by Netscape (and to a lesser-extent, Microsoft) did not make it into these new standards. It soon caught on and became the official standard in January ’97, and today practically all browsers support it fully.

**HTML 4.01**

HTML 4.0 was a large evolution of the HTML standards, and the last iteration of classic HTML. Early in development it had the code-name COUGAR. Most of the new functionality brought in this time is from the ill-fated HTML 3.0 spec, as well as a host of trimmings on old tags, a focus on internationalization, and support for HTML’s new supporting presentational language, [cascading stylesheets](https://www.yourhtmlsource.com/stylesheets/).

HTML 4.0 was recommended by the W3C in December ’97 and became the official standard in April 1998. Browser support was undertaken surprisingly earnestly by Microsoft in their [Internet Explorer](https://www.yourhtmlsource.com/starthere/browserreview.html#ie6) browser, and the market-leading IE5 (and current successor IE6) have excellent support for almost all of the new tags and attributes. In comparison, Netscape’s terribly flawed Navigator 4.7 was inept when it came to HTML 4.0 and even basic CSS. Modern browsers, however, are a vast improvement.

Once HTML 4.0 had been out for a little while, the documentation was revised and corrected in a few minor ways and was entitled HTML 4.01; the final version of the specification.

**HTML5**

After HTML 4.01 and XHTML 1.0, the guys who were in control of HTML’s direction got sidetracked working on a new proposal for XHTML 2. At the same time, clever web developers were innovating constantly, hacking new functionality into websites and browsers. The path that XHTML 2 was taking started to look both boring and [unrealistic](http://www.zeldman.com/daily/0103b.shtml#skyfall), and it became pretty clear that a new approach was needed.

It was around this time that a bunch of pragmatic web technology fans, browser programmers and specification writers started building something of their own, outside of the usual W3C procedures. They called themselves the Web Hypertext Application Technology Working Group ([WHATWG](http://www.whatwg.org/)), and developed a new spec. After some soul-searching, the W3C decided that HTML was still the future of the web. XHTML 2 was discontinued and HTML5 became the new specification that everyone’s effort should be poured into.

HTML5 is designed for the web, both now and in the future. This is the specification that we will be working with for the next decade at least, so the process of its development is relatively slow and considered. Many parts will be familiar, but there’s also plenty of [new elements, attributes and abilities](http://dev.w3.org/html5/html4-differences/) to get excited about.

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

**NodeJS :**

NodeJS is a cross-platform and opensource Javascript runtime environment that allows the javascript to be run on the server-side. Nodejs allows Javascript code to run outside the browser. Nodejs comes with a lot of modules and mostly used in web development.

**JavaScript :**

Javascript is a Scripting language. It is mostly abbreviated as JS. It can be said that Javascript is the updated version of the ECMA script. Javascript is a high-level programming language that uses the concept of Oops but it is based on prototype inheritance.

**Difference between Nodejs and JavaScript :**

| **S.No** | **Javascript** | **NodeJS** |
| --- | --- | --- |
| 1. | Javascript is a programming language that is used for writing scripts on the website. | NodeJS is a Javascript runtime environment. |
| 2. | Javascript can only be run in the browsers. | NodeJS code can be run outside the browser. |
| 3. | It is basically used on the client-side. | It is mostly used on the server-side. |
| 4. | Javascript is capable enough to add HTML and play with the DOM. | Nodejs does not have capability to add HTML tags. |
| 5. | Javascript can run in any browser engine as like JS core in safari and Spidermonkey in Firefox. | Nodejs can only run in V8 engine of google chrome. |
| 6. | Javascript is used in frontend development. | Nodejs is used in server-side development. |
| 7. | It is the upgraded version of ECMA script that uses Chrome’s V8 engine written in C++. | Nodejs is written in C, C++ and Javascript. |

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

**1. You enter the URL in the browser.**

Suppose you want to visit the website of AfterAcademy. So you type [afteracademy.com](https://afteracademy.com/) in the address bar of your browser. When you type any URL you basically want to reach the server where the website is hosted.

***2.* The browser looks for the IP address of the domain name in the DNS (Domain Name Server).**

DNS is a list of URLs and their corresponding IP address just like the telephone book has phone numbers corresponding to the names of the people. We can access the website directly by typing the IP address but imagine remembering a group of numbers to visit any website. So, we only remember the name of the website and the mapping of the name with the IP address is done by the DNS. The DNS checks at the following places for the IP address.

Browser maintains a cache of the DNS records for some fixed amount of time. It is the first place to run a DNS query.

1. Check OS Cache: If the browser doesn't contain the cache then it requests to the underlying Operating System as the OS also maintains a cache of the DNS records.
2. Router Cache: If your computer doesn't have the cache, then it searches the routers as routers also have the cache of the DNS records.
3. ISP(Internet Service Provider) Cache: If the IP address is not found at the above three places then it is searched at the cache that ISP maintains of the DNS records. If not found here also, then ISP’s DNS recursive search is done. In "DNS recursive search", a DNS server initiates a DNS query that communicates with several other DNS servers to find the IP address.

So, the domain name which you entered gets converted into a DNS number. Suppose the above-entered domain name [afteracademy.com](https://afteracademy.com/) has an IP address 100.95.224.1. So, if we type [https://100.95.224.1](https://100.95.224.1/) in the browser we can reach the website.

#### **3. The Browser initiates a TCP connection with the server.**

When the browser receives the IP address, it will build a connection between the browser and the server using the internet protocol. The most common protocol used is TCP protocol. The connection is established using a three-way handshake. It is a three-step process.

1. Step 1 (SYN): As the client wants to establish a connection so it sends an SYN(Synchronize Sequence Number) to the server which informs the server that the client wants to start a communication.
2. Step 2 (SYN + ACK): If the server is ready to accept connections and has open ports then it acknowledges the packet sent by the server with the SYN-ACK packet.
3. Step 3 (ACK): In the last step, the client acknowledges the response of the server by sending an ACK packet. Hence, a reliable connection is established and data transmission can start now.

#### **4. The browser sends an HTTP request to the server.**

The browser sends a GET request to the server asking for [afteracademy.com](https://afteracademy.com/) webpage. It will also send the cookies that the browser has for this domain. Cookies are designed for websites to remember stateful information (items in the shopping cart or wishlist for a website like Amazon) or to record the user’s browsing history etc. It also has additional information like request header fields(User-Agent) for that allows the client to pass information about the request, and about the client itself, to the server. Other header fields like the Accept-Language header tells the server which language the client is able to understand. All these header fields are added together to form an HTTP request.

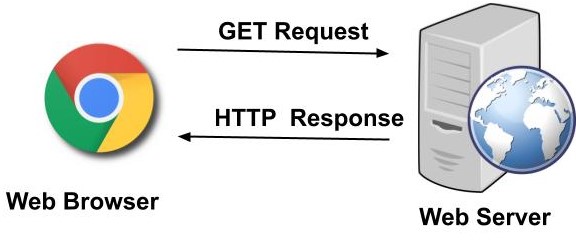
Eg-GET /abc.htm HTTP/1.1

#### **5. The server handles the incoming request and sends an HTTP response.**

The server handles the HTTP request and sends a response. The first line is called the status line. A Status-Line consists of the protocol version(e.g HTTP/1.1) followed by numeric status code(e.g 200)and its associated textual phrase(e.g OK). The status code is important as it contains the status of the response.

1. 1xx: Informational: It means the request was received and the process is continuing.
2. 2xx: Success: It means the action was successful.
3. 3xx: Redirection: It means further action must be taken in order to complete the request. It may redirect the client to some other URL.
4. 4xx: Client Error: It means some sort of error in the client’s part.
5. 5xx: Server Error: It means there is some error on the server-side.

It also contains response header fields like Server, Location, etc. These header fields give information about the server. A Content-Length header is a number denoting the exact byte length of the HTTP body. All these headers along with some additional information are added to form an HTTP response.



#### **6. The browser displays the HTML content.**

Now the browser gets the response and the HTML web page is rendered in phases. First, it gets the HTML structure and then it sends multiple GET requests to get the embedded links, images, CSS, javascript files, etc and other stuff. The web page will be rendered and in this case, the [afteracademy](https://afteracademy.com/" \t "_blank)web page will be displayed.

