**Welcome**

**Page 2 - About HTML**

**Page 3 - HTTP1.1 vs HTTP2**

**Page 4 – JavaScript Founder and objects**

**Page 5 to 8 – Web Browsers engines**

**Page 9 – Codekata practice files link**

**Page 9 to 12 – IP address and Mac address**

**HTTP stands for hypertext transfer protocol** and it is the basis for almost all web applications.

**Father of HTML**

Tim Berners-Lee

HTML / Inventor

**Sir Timothy John Berners-Lee, OM, KBE, FRS, FREng, FRSA, DFBCS, RDI (born 8 June 1955), also known as TimBL, is an English computer scientist best known as the inventor of the World Wide Web, the HTML markup language, the URL system, and HTTP.**

HTTP/2 is different from HTTP/1 in several ways, including supporting server push, multiplexing of requests, and header compression. HTTP/2 also uses binary encoding rather than text encoding, making it more efficient. A: Most modern web browsers, including Chrome, Firefox, and Safari, support HTTP/2

HTTP stands for hypertext transfer protocol & it is used in client-server communication. By using HTTP user sends the request to the server & the server sends the response to the user. There are several stages of development of HTTP but we will focus mainly on HTTP/1.1 which was created in 1997 & the new one is HTTP/2 which was created in 2015.

**About HTTP/1.1**

**For better understanding, let’s assume the situation when you make a request to the server for the geeksforgeeks.html page & server responds to you as a resource geeksforgeeks.html page. before sending the request and the response there is a TCP connection established between client & server. again you make a request to the server for image img.jpg & the server gives a response as an image img.jpg. the connection was not lost here after the first request because we add a keep-alive header which is the part of the request so there is an open connection between the server & client. there is a persistent connection which means several requests & responses are merged in a single connection. These are the drawbacks that lead to the creation of HTTP/2: The first problem is HTTP/1.1 transfer all the requests & responses in the plain text message form. The second one is head of line blocking in which TCP connection is blocked all other requests until the response does not receive. all the information related to the header file is repeated in every request.**

**About HTTP/2**

**HTTP/2 was developed over the SPDY protocol. HTTP/2 works on the binary framing layer instead of textual that converts all the messages in binary format. it works on fully multiplexed that is one TCP connection is used for multiple requests. HTTP/2 uses HPACK which is used to split data from header. it compresses the header. The server sends all the other files like CSS & JS without the request of the client using the PUSH frame.**

**Difference between HTTP1.1 vs HTTP2**

| **HTTP/1.1** | **HTTP/2** |
| --- | --- |
| It works on the textual format. | It works on the binary protocol. |
| There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. | It allows multiplexing so one TCP connection is required for multiple requests. |
| It uses requests resource Inlining for use getting multiple pages | It uses PUSH frame by server that collects all multiple pages |
| It compresses data by itself. | It uses HPACK for data compression. |
| png-transparent-computer-icons-security-hacker-penetration-test-icon-hacker-miscellaneous-silhouette-symbol-thumbnail.pngData’s are open  2960337-200.pngany one can see or access | 2960334-200.pngpng-transparent-computer-icons-security-hacker-penetration-test-icon-hacker-miscellaneous-silhouette-symbol-thumbnail.pngData’s are secured and encrypted and only authorized can access.  Third party and unauthorized persons can’t access. |

**About objects and its internal representation in JavaScript**

**JavaScript Founder**

****

Brendan Eich

JavaScript was invented by Brendan Eich in 1995. It was developed for Netscape 2, and became the ECMA-262 standard in 1997. After Netscape handed JavaScript over to ECMA, the Mozilla foundation continued to develop JavaScript for the Firefox browser.

**Objects in Javascript**

Objects, in JavaScript, are the most important data type and form the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data types **(Number, String, Boolean, null, undefined, and symbol)** in the sense that these primitive data types all store a single value each (depending on their types).

Java script code sample:

let company = {

name : "GUVI GEEK NETWORK PVT LTD",

location : "IIT MADRAS PARK CHENNAI",

established : "2014",

founder : "Mr. ARUN PRAKASH",

cofounder1 : "Late Mrs. SRIDEVI ARUN PRKASH",

cofounder2: "SP BALA MURUGAN",

special : "Teaching in six regional languages like Tamil, Malayalam, Hindi and more",

displayinfo : function(){

console.log(`${company.name} was established

in ${company.established} at ${company.location}

founder ${company.founder} and

co-founders ${company.cofounder1} and ${company.cofounder2}

speciality is GUVI offered ${company.special}`);

}

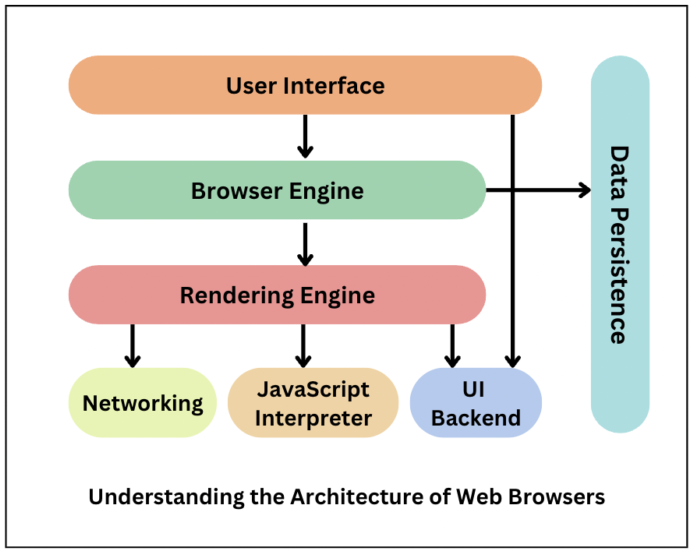
} company.displayinfo();

### Understanding the Architecture of Web Browsers

Browsers are built of FrontEnd and BackEnd. While the FrontEnd ensures how the webpage appears on the browser, the backend handles the requests and is the carrier of information. Its different components work in coordination to deliver a seamless web experience.

Each browser is made up of 7 different components:

1. User Interface
2. Browser Engine
3. Rendering Engine
4. Networking
5. JavaScript Interpreter
6. UI Backend
7. Data Persistence



### Components of Web Browser

Web Browsers consist of 7 different components listed below:

#### 1. User Interface

This component allows end-users to interact with all visual elements available on the web page. The visual elements include the **address bar, home button, next button,** and all other elements that fetch and display the web page requested by the end-user.

#### 2. Browser Engine

It is a core component of every web browser. The browser engine functions as an intermediary or a bridge between the user interface and the rendering engine. It queries and handles the rendering engine as per the inputs received from the user interface.

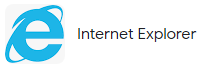
The performance and features of a browser engine can greatly impact the user experience of a web browser. A fast and efficient browser engine can help web pages load quickly and smoothly, while a slower or less capable engine may struggle to render complex pages or provide a smooth browsing experience.

#### 3. Rendering Engine

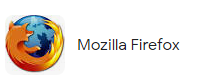
As the name suggests, this component is responsible for rendering a specific web page requested by the user on their screen. It interprets HTML and XML documents along with images that are styled or formatted using CSS, and a final layout is generated, which is displayed on the user interface.

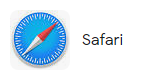
**Note**: Every browser has its own unique rendering engine. Rendering engines might also differ for different browser versions. The list below mentions browser engines used by a few common browsers:

1. Google Chrome and Opera v.15+: **Blink**
2. Internet Explorer: **Trident**



1. Mozilla Firefox: **Gecko**

****

1.  iOS and Safari: **WebKit**

#### 4.Networking

This component is responsible for managing network calls using standard protocols like HTTP or FTP. It also looks after security issues associated with internet communication.

#### 5. JavaScript Interpreter

As the name suggests, it is responsible for parsing and executing the JavaScript code embedded in a website. Once the interpreted results are generated, they are forwarded to the rendering engine for display on the user interface.

#### 6. UI Backend

This component uses the user interface methods of the underlying operating system. It is mainly used for drawing basic widgets (windows and combo boxes).

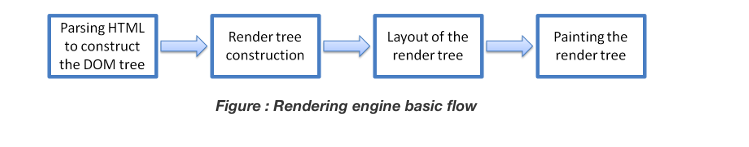
#### 7. Data Storage/Persistence

It is a persistent layer. A web browser needs to store various types of data locally, for example, cookies. As a result, browsers must be compatible with data storage mechanisms such as WebSQL, IndexedDB, FileSystem, etc.

Now that we are aware of the key components involved in building a web browser let’s dig deep into the role of the rendering engine.

### Role of Rendering Engine

Once a user requests a particular document, the rendering engine starts fetching the content of the requested document. This is done via the networking layer. The rendering engine starts receiving the content of that specific document in chunks of 8 KBs from the networking layer. After this, the basic flow of the rendering engine begins.



The four basic steps include:

1. The requested HTML page is parsed in chunks, including the external CSS files and in style elements, by the rendering engine. The HTML elements are then converted into DOM nodes to form a **“content tree” or “DOM tree.”**
2. Simultaneously, the browser also creates a **render tree.**This tree includes both the styling information as well as the visual instructions that define the order in which the elements will be displayed. The render tree ensures that the content is displayed in the desired order.
3. Further, the render tree goes through the **layout process.** When a render tree is created, the position or size values are not assigned. The entire process of calculating values for evaluating the desired position is called a layout process. In this process, every node is assigned the exact coordinates. This ensures that every node appears at an accurate position on the screen.
4. The final step is to paint the screen, wherein the render tree is traversed, and the renderer’s **paint()** method is invoked, which paints each node on the screen using the UI backend layer.

As discussed earlier, **every browser has its own unique rendering engine.**So naturally, every browser has its own way of interpreting web pages on a user’s screen. Here’s where a challenge arises for web developers regarding the cross-browser compatibility of their website.

This is where cross-browser testing comes into the picture.

Cross browser testing is a quality assurance method used to verify the consistency of web applications in terms of functionality and design across multiple browsers. These tests enable QA teams to explore any issues by conducting a responsive test, that may occur when their website is accessed via different browsers or browser versions.

Codekata practice

Link attached:

Please double click and it will open automatically in browser then you can inspect and console



Please open this file in visual studio code application

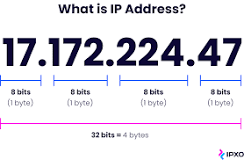


Now you can see the codekata practice JS codes

**About IP address, port, HTTP methods, MAC address**

If we want to call some one in crowd, we should call with particular name or identity.

Here in networks having lots of things inside, so if we want to call or search about guvi home page we should type guvi in browser search bar, so suddenly network work browser search the guvi with particular reference via numbers. This is called IP address (8bits)



An Internet Protocol address (IP address) is a numerical label such as 192.0.2.1 that is connected to a computer network that uses the Internet Protocol for communication. An IP address serves two main functions: network interface identification, and location addressing.

All the computers of the world on the Internet network communicate with each other with underground or underwater cables or wirelessly. If I want to download a file from the internet or load a web page or literally do anything related to the internet, my computer must have an address so that other computers can find and locate mine in order to deliver that particular file or webpage that I am requesting. In technical terms, that address is called **IP Address or**

**Internet Protocol Address**.

Let us understand it with another example, like if someone wants to send you a mail then he/she must have your home address. Similarly, your computer too needs an address so that other computers on the internet can communicate with each other without the confusion of delivering information to someone else’s computer. And that is why each computer in this world has a unique IP Address. Or in other words, an IP address is a unique address that is used to identify computers or nodes on the internet. This address is just a string of numbers written in a certain format. It is generally expressed in a set of numbers for example 192.155.12.1. Here each number in the set is from 0 to 255 range. Or we can say that a full IP address ranges from 0.0.0.0 to 255.255.255.255. And these IP addresses are assigned by IANA (known as Internet Corporation For Internet Assigned Numbers Authority).

But what is Internet protocol? This is just a set of rules that makes the internet work. You are able to read this article because your computer or phone has a unique address where the page that you requested has been delivered successfully.

**Working of IP addresses**

The working of IP addresses is similar to other languages. It can also use some set of rules to send information. Using these protocols we can easily send, and receive data or files to the connected devices. There are several steps behind the scenes. Let us look at them

* Your device directly requests your Internet Service Provider which then grants your device access to the web.
* And an IP Address is assigned to your device from the given range available.
* Your internet activity goes through your service provider, and they route it back to you, using your IP address.
* Your IP address can change. For example, turning your router on or off can change your IP Address.
* When you are out from your home location your home IP address doesn’t accompany you. It changes as you change the network of your device.

### Types of IP Address

IP Address is of two types:

**1. IPv4:**Internet Protocol version 4. It consists of 4 numbers separated by the dots. Each number can be from 0-255 in decimal numbers. But computers do not understand decimal numbers, they instead change them to binary numbers which are only 0 and 1. Therefore, in binary, this (0-255) range can be written as (00000000 – 11111111). Since each number N can be represented by a group of 8-digit binary digits. So, a whole IPv4 binary address can be represented by 32-bits of binary digits. In IPv4, a unique sequence of bits is assigned to a computer, so a total of (2^32) devices approximately = 4,294,967,296 can be assigned with IPv4.

IPv4 can be written as:

**189.123.123.90**

**Classes of IPv4 Address:** There are around 4.3 billion IPv4 addresses and managing all those addresses without any scheme is next to impossible. Let’s understand it with a simple example. If you have to find a word from a language dictionary, how long will it take? Usually, you will take less than 5 minutes to find that word. You are able to do this because words in the dictionary are organized in alphabetical order. If you have to find out the same word from a dictionary that doesn’t use any sequence or order to organize the words, it will take an eternity to find the word. If a dictionary with one billion words without order can be so disastrous, then you can imagine the pain behind finding an address from 4.3 billion addresses. For easier management and assignment IP addresses are organized in numeric order and divided into the following 5 classes :

| **IP Class** | **Address Range** | **Maximum number of networks** |
| --- | --- | --- |
| Class A | 1-126 | 126 (27-2) |
| Class B | 128-191 | 16384 |
| Class C | 192-223 | 2097152 |
| Class D | 224-239 | Reserve for multitasking |
| Class E | 240-254 | Reserved for Research and development |

The 0.0.0.0 is a Non routable address is that indicates an invalid or inapplicable end-user address.

A loopback address is a distinct reserved IP address range that starts from 127.0.0.0 ends at 127.255.255.255 though 127.255.255.255 is the broadcast address for 127.0.0.0/8. The loopback addresses are built into the IP domain system, enabling devices to transmit and receive the data packets. The loopback address 127.0.0.1 is generally known as local host.

**2. IPv6:**But, there is a problem with the IPv4 address. With IPv4, we can connect only

1. Starting from Layer 2 in the network stack MAC address designates the physical address of the device which is a 6 byte address. ...
2. At the layer 3 we have IP address, it designates the logical address of the device. ...
3. Port address is at layer 4 (transport layer : TCP/UDP)

-------------------------------------------END----------------------------------------