**Exercise 1.1 : Working of a Browser**

Solution :

Web Browser allows exchange of information(i.e. visiting websites , online interaction, watching videos etc ) over the Internet through the HTTP(Hyper Text Transfer Protocol). As the name suggests , it describes protocols for transferring HTML( Hyper Text Markup Language) over the internet. Whenever a user(or client side) types in the URL on the search bar and requests for information , the browser retrieves the information from the server and displays the website through the help of different browser components.

Before diving into how the retrieval process comes to fruition, it's imperative to get a clear idea of what exactly a website is. Website is a collection of files - i.e. HTML,CSS , Javascript etc. These files are stored in an external computer - which is called server. A website has a name - which in technical jargon is defined as domain. So for instance, for www.google.com - the domain name would be **google**.

Every domain name has a corresponding IP address(Example : 203.0.113.0) associated with it. For us humans it's easier to remember the name of a website rather than its equivalent number format. Computer devices however, don't work with domain names while trying to locate each other on the internet. Hence they use IP address to locate the servers on which the website eventually resides.

It's also crucial to know what a DNS(Domain Name System) is before taking a deep dive into how a website gets retrieved through the internet. DNS helps facilitate communication over the internet between the client side(users/program) and the server). DNS, in nutshell converts the domain name into its numerical equivalent format i.e. the IP address and helps in finding out where the website eventually resides.

Following steps involved in retrieving a website’s address. :

1. Whenever the user enters the website on the address bar, a DNS server goes to work to find the address where the website is located. The different servers involved in this process are :
   1. DNS Resolver : Receives the request to resolve the domain name(i.e. website name ) with the IP address.
   2. DNS root server : Receives a request from the DNS resolver and returns the address of the TLD ( Top Level Domain ) server. A top level domain is the equivalent .com or .in part of the URL the users type into the address bar.
   3. TLD server : The DNS resolver will interact with this server, which in turn will direct the DNS resolver to Authoritative Server where the site is actually returned.
   4. Authoritative Name Server : The DNS resolver finally queries the authoritative name server - which in turn returns the actual IP address of the website.

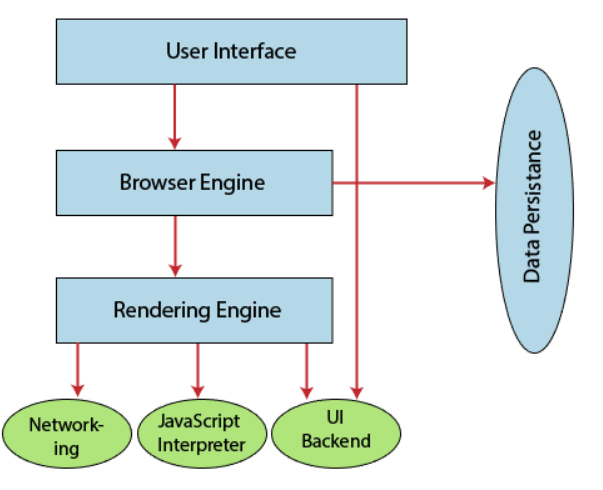
**HTTP/HTTPS** :

1. A visit to a web page begins with a URL(Uniform Resource Locator ) that points that web page. Example https://www.google.com.
2. The https at the start is a networking scheme **HTTP** describes how a browser makes request to the server to receive the website. Once all the protocols and standards are met , the client side i.e. the browser receives the webpage from the server.

**Browser Components :**

Before getting into what happens once the website is retrieved in the form of files, it's important to see what are the different components in the Browser :

1. User Interface(UI) : Includes the address bar , settings , back/forward button and other things that make up the GUI of the browser.
2. Browser Engine : It parses the HTML & CSS. Parsing involves breaking down of HTML file into the DOM tree structure. It does the same with CSS as well - the final end product is a CSS object Model( CSSOM).
3. Rendering Engine : Rendering engine is responsible for displaying the website on the page. The CSSOM and DOM trees created in the parsing step are combined into a render tree -which is then used to create the layout of every visible element.
4. Java script Interpretor : Used to interpret Javascript code and execute them. Furthermore, it may be used to cache a retrieved document to reduce network traffic.
5. Data Storage : This is a persistence layer- meaning it stores the data long after the process which brought forward data is completed. Thus it's job is to persist i.e. store the data locally such as cookies, cache etc.
6. Networking : It retreives the URL's by using internet protocol like HTTP or FTP. For making different network calls such as HTTP/ HTTPS.
7. UI Backend : Generates a specific interface which is not platform specific.



**Parsing**:

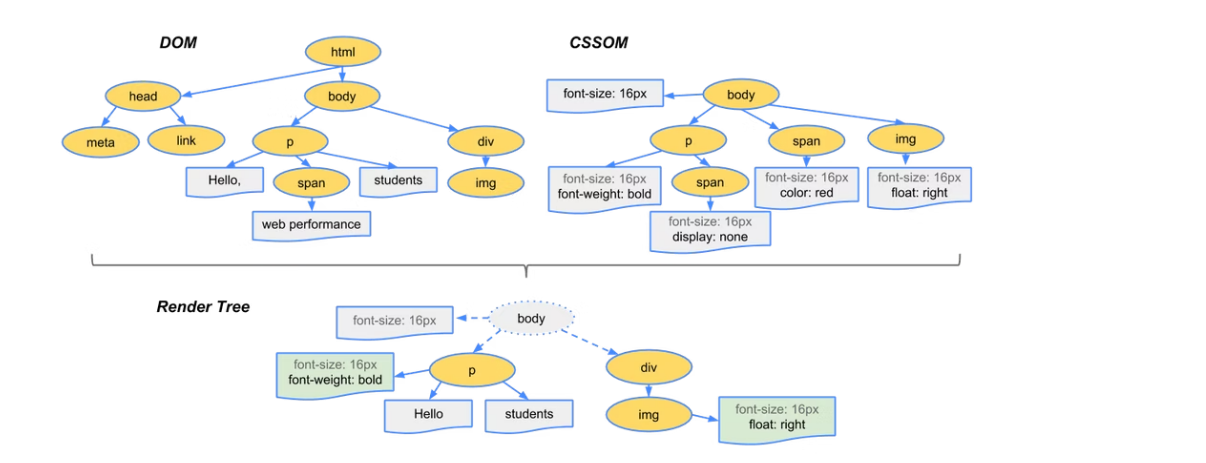
Once the browser retrieves the website in the form of files i.e. HTML , CSS , Javascript etc it begins the parsing process. Parsing involves creating of DOM & CSSCOM model. To say, it breaks down the HTML and CSS elements into a heirarchical model or a tree like structure as below. It's important to note that both the DOM & CSSOM are independent objects capturing different aspects of the document : DOM captures the content while CSSOM captures the style rules which needs to be applied to the DOM.

**Rendering** :

Render Engine is involved in combining the DOM & CSSOM to form a render tree- to display the website on the screen. Rendering happens in the following steps :

1. Traverse through each visible node of the DOM tree. It omits the tags(or nodes) which are not to be reflected in the output i.e. head tag, meta tag.
2. Same process above through the CSSOM tree structure.
3. For each visible node in the DOM, find its matching CSSOM rules and apply them.

Final output of the render engine is a render that contains the content and style information of all the visible content on the screen.



With the render tree in it's place follows the Layout Stage.

**Layout :**

Layout is the process where elements are placed on the screen. The layout is a computationally intensive process as it involves calculations about an element position behind the screen.

The web's layout model defines that one element can affect others i.e. width of the <p> can affect its child element and so on.

HTML layout helps in designing well-structured and responsive web pages. As mentioned above , layout in nutshell involves placing HTML elements on the screen.

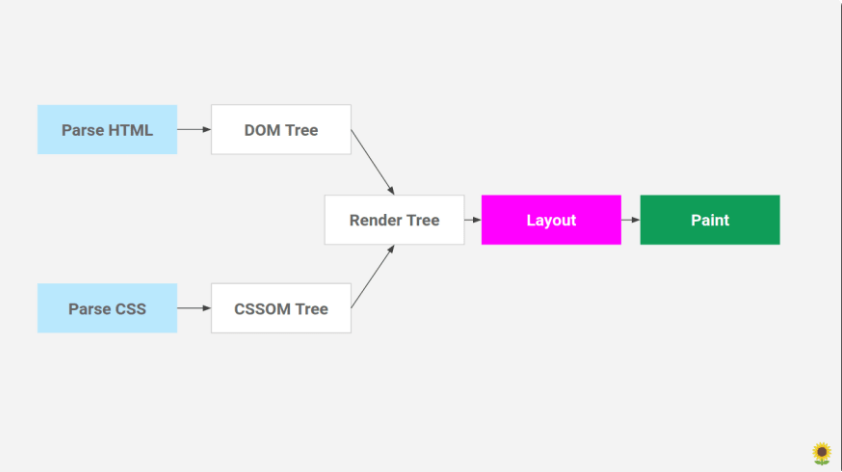
**Painting** :

Post Layout - the paint happens. Paint takes properties such as background-colour ,border-colour , colour to paint the screen with colors.

Painting is of 2 types :

1. Global - the entire tree gets painted. For instance - painting the entire background of the webpage.
2. Incremental - Only selective nodes of the tree gets painted in a way that doesn't affect the entire tree. For instance - painting a div element.

Below is the sequence of action leading to website getting uploaded on the web-browser :



**Processing Scripts :**

Scripts i.e. Javascript are parsed and executed immediately when the parser reacher the <script> tag. The parsing of the document halts until the script is executed. Hence it is recommended to place <script> tag towards the end of body so the parsing gets completed.

HTML5 adds an option to mark the script as asynchronous so that it gets parsed separately.