WEEK 1. ASSIGNMENT.

1.When you type a URL into the browser it makes a DNS query to figure out which ip address is associated with that Hostname.

Here DNS (DOMAIN NETWORK SYSTEM) is like the phonebook of the internet it maps the human readable URL's or hostnames (Eg:www.akash.com) to IPaddress.

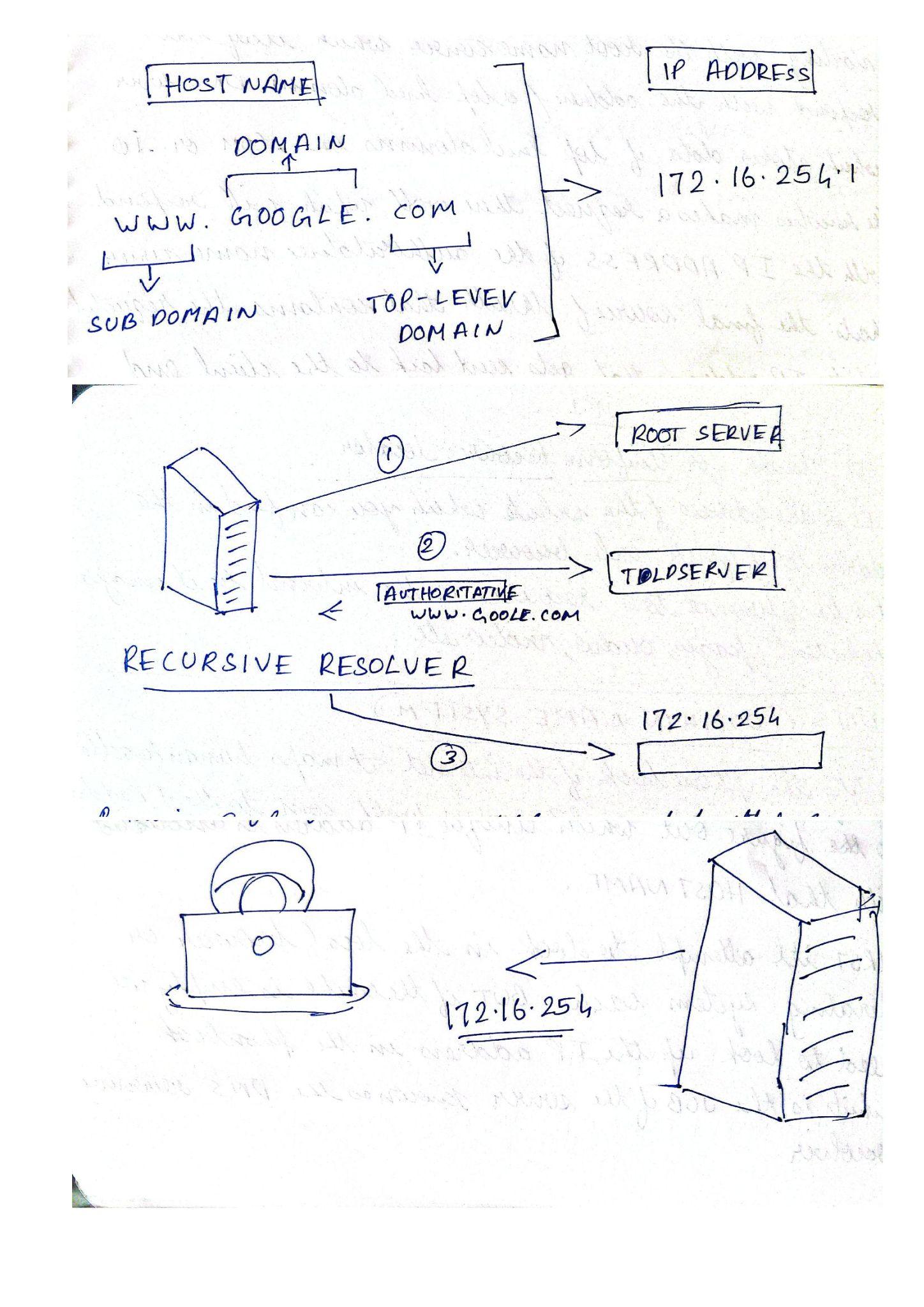
First itl attempt to look in the local browser or operating system cache but if it is empty we need to look up the IP address in the Phonebook which is the job of the server known as the DNS recursive resolver.

So the recursive resolver makes multiple requests to the server starting with the root name server which will respond with the address of the top level domain like .com and .io .

The resolver makes a request which will respond with the IP ADDRESS of the authoritative name server that's the final source of truth that contains the content of the requested websites IP ADDRESS that gets back to the client with the content and is cached for future use.

URL stands for Uniform Resource locator. URL is the address of the website which you can find in the address bar of your web browser.

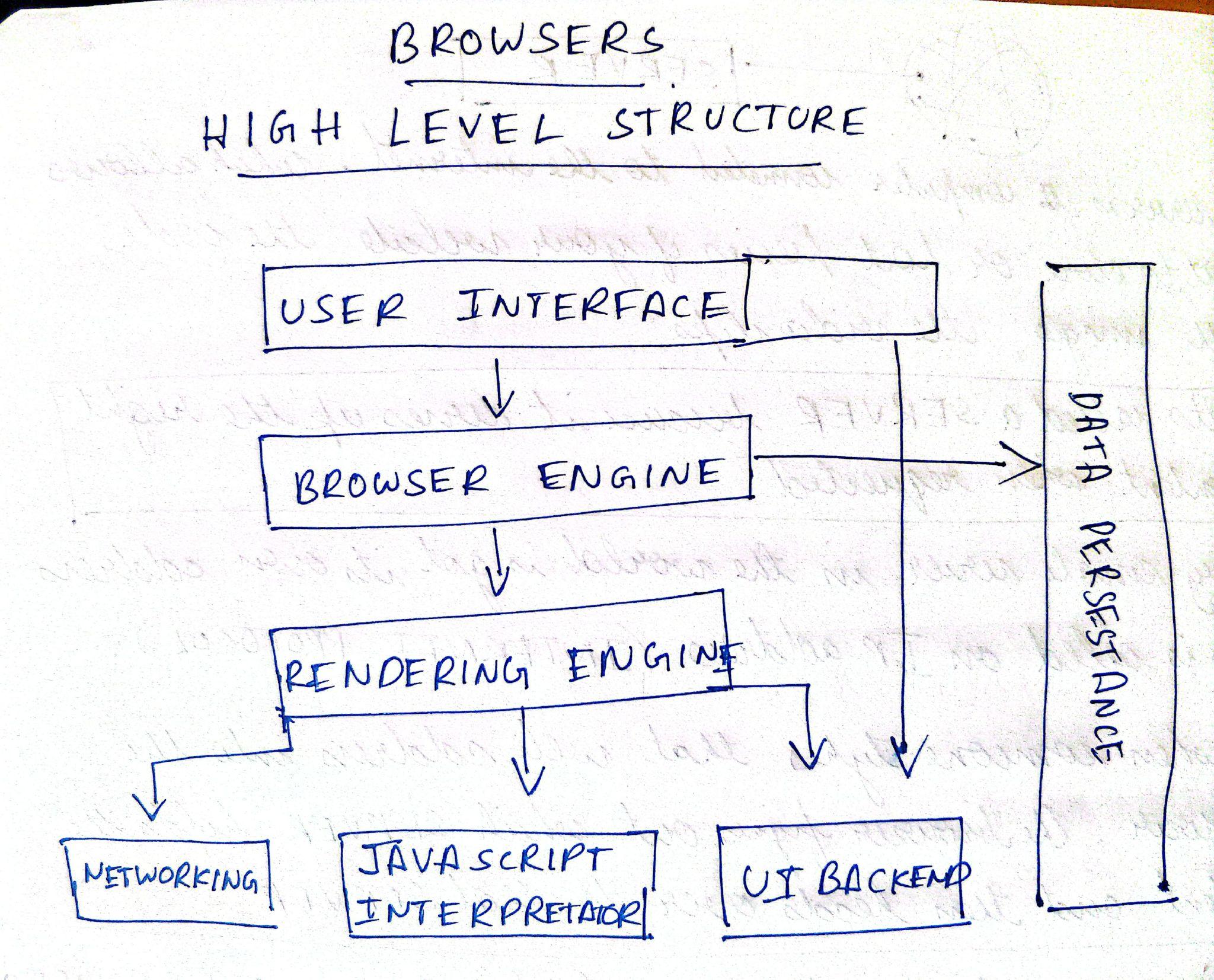
It is the reference to a resource on the internet be it images, hypertext pages , audio , video etc.



2. The Main functionality of browser is to present the web resource you choose , by requesting it from the server and displaying it on the browser window.When the browser fetches the data from an internet connected server it uses a piece of software called the rendering engine to translate the HTML,CSS,JS code to create what we see, hear and experience on the internet.

3. High level structure of Browser

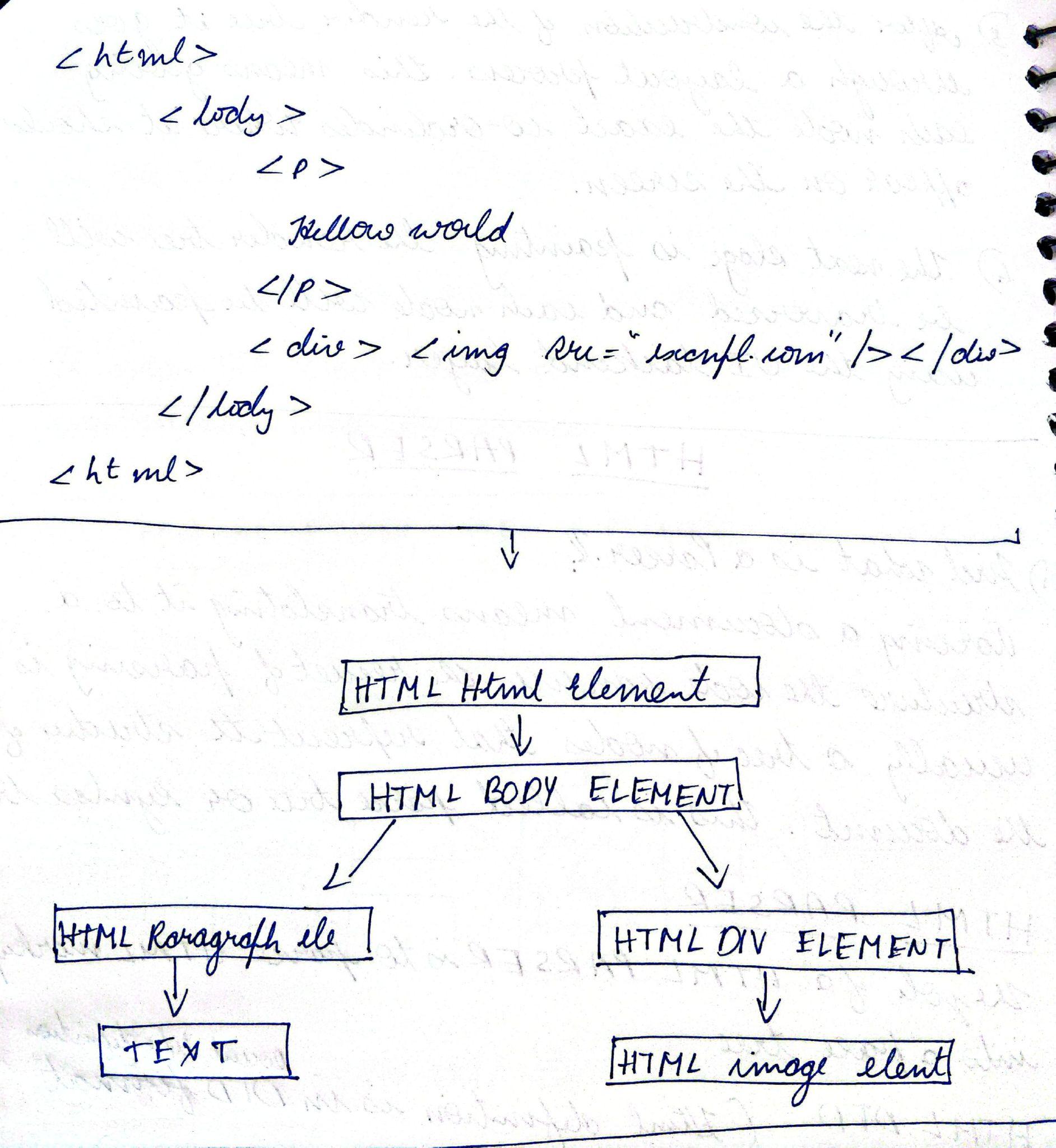
1. The User Interface - it is where the user interacts with the browser, it includes the address bar, back and next button, stop and refresh, home button, bookmark option etc.
2. The Browser Engine- works as a bridge between the user interface and the rendering engine. According to the input from the UI it queries and gives the result to the rendering engine.
3. Rendering Engine - is responsible for rendering the requested web page on the browser window or screen. It is a piece if software that is used to translate the HTML data into text and images
4. Networking - is where the network calls like HTTP requests happen, it handles all the aspects of internet communication and security. It also manages cache memory of the retrieved data so the network traffic can be reduced.
5. Javascript Interpreter - Reads and executes the javascript code embedded in the website the interpreted results are sent to the rendering engine for display.
6. UI backend - used for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific. Underneath it uses operating system user interface methods.
7. Data persistence/storage - supports storage mechanisms such as local storage, indexDB, web SQL and file system. It is a small DB created in the local drive of the computer where the browser is installed. It manages user data such as cache, cookies,bookmarks and preferences.



4. Rendering Engine is responsible for rendering the requested web page on the browser window or screen. It is a piece of software that is used to translate the HTML data into text and images. Different browsers use different rendering engines CHROME uses Blink,

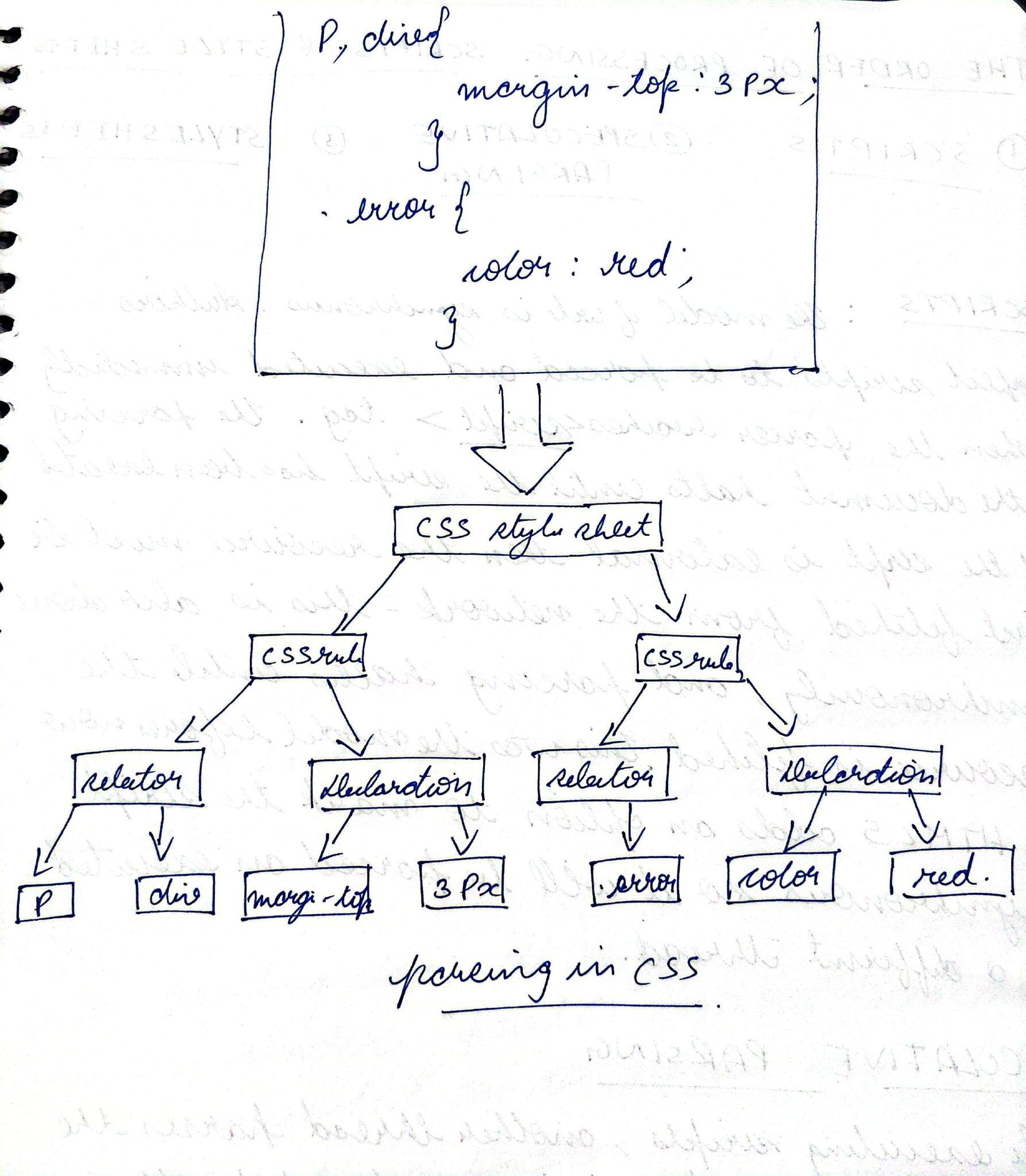
Firefox uses GECKO

5. Parsing a document means translating it to a structure the code can use. The result of parsing is usually a tree of nodes that represent the structure of the document; this is called parse tree or syntax tree.



HTML PARSER is to parse the HTML markup to a parse tree.

CSS Parser - the CSS file is passed into a style sheet object each object contains CSS rules. The CSS rule object contains selector and declaration objects and their objects corresponding to CSS grammar.



6) The order of processing script and style sheets.

Scripts

The model of the web is synchronous. Authors expect scripts to be parsed and executed immediately when the parser reaches a <script> tag. The parsing of the document halts until the script has been executed. If the script is external then the resource must first be fetched from the network - this is also done synchronously, and parsing halts until the resource is fetched. This was the model for many years and is also specified in HTML4 and 5 specifications. Authors can add the "defer" attribute to a script, in which case it will not halt document parsing and will execute after the document is parsed. HTML5 adds an option to mark the script as asynchronous so it will be parsed and executed by a different thread.

Speculative parsing

While executing scripts, another thread parses the rest of the document and finds out what other resources need to be loaded from the network and loads them. In this way, resources can be loaded on parallel connections and overall speed is improved. Note: the speculative parser only parses references to external resources like external scripts, style sheets and images: it doesn't modify the DOM tree - that is left to the main parser.

Style sheets

Style sheets on the other hand have a different model. Conceptually it seems that since style sheets don't change the DOM tree, there is no reason to wait for them and stop the document parsing. There is an issue, though, of scripts asking for style information during the document parsing stage. If the style is not loaded and parsed yet, the script will get wrong answers and apparently this caused lots of problems. It seems to be an edge case but is quite common. Firefox blocks all scripts when there is a style sheet that is still being loaded and parsed. WebKit blocks scripts only when they try to access certain style properties that may be affected by unloaded style sheets.

7) LAYOUT - When the render is created and added to the tree it does not have a position or size calculating these values and providing the coordinates for each node as to where they have to be rendered on the window is called layout or reflow.

PAINTING - In the painting stage, the render tree is traversed and the render “paint()” method is called to display content on the screen. Painting uses UI infrastructure components.