1. When a user enters an URL in the browser, how does the browser fetch the desired result? Explain this with the below in mind and Demonstrate this by drawing a diagram for the same. (2-3hours)

a. What is the main functionality of the browser?

b. High Level Components of a browser.

c. Rendering engine and its use.

d. Parsers (HTML, CSS, etc.)

e. Script Processors

f. Tree construction

g. Order of script processing

h. Layout and Painting

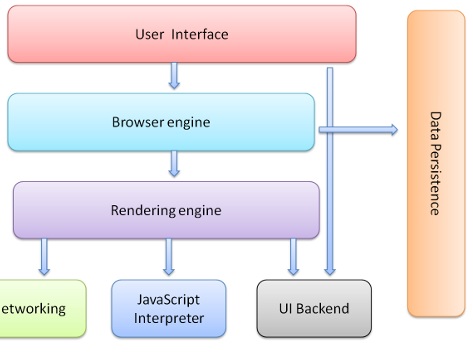
1. What is the main functionality of the browser?

Answer- A web Browser is primarily use to display the website or content which are created by using Hyper Text Markup Language (HTML) or you can say Browser convert hyper text transfer protocal webpages and website to human readable format.

In simple words, when you hit the url of a website in browser, browser load the HTML code which contains CSS , image, links etc. renders on the browser window.

1. High Level Components of a browser.

Answer- Below is the main components of browser.

1. User Interface- It gives some functionality to user to interact with the webpage or to interact with the internet which includes address bar, backward/ forward button, address bar, search box, menu and a window to display the website.
2. Browser Engine- Its Allow you search anything on internet.
3. Rendering Engine- It’s the responsible to parse the HTML and CSS and render the website on browser window.
4. Networking- Its actually responsible to deal with HTTP request.
5. UI backend- user=d for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific.
6. JavaScript Interpreter- Used to parse and execute JavaScript code.
7. Data Storage- It’s a persistence layer. The Browser may need to save all sorts of data locally, such as cookies. Browsers also support storage mechanisms such as localStorage, IndexedDB and filesystem.

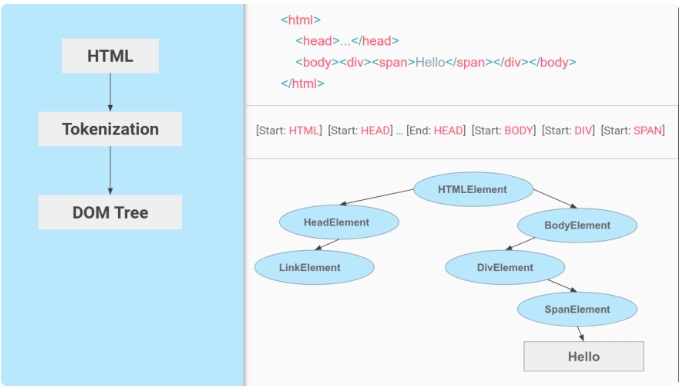
3. Rendering engine and its use

Answer- Main responsibility of rendering engine is to render the XML or HTML code which contains CSS, images, links etc. on the screen. Different browser use different rendering engine like chrome use WebKit, Firefox uses Gecko.

4. Parsers (HTML, CSS, etc.)

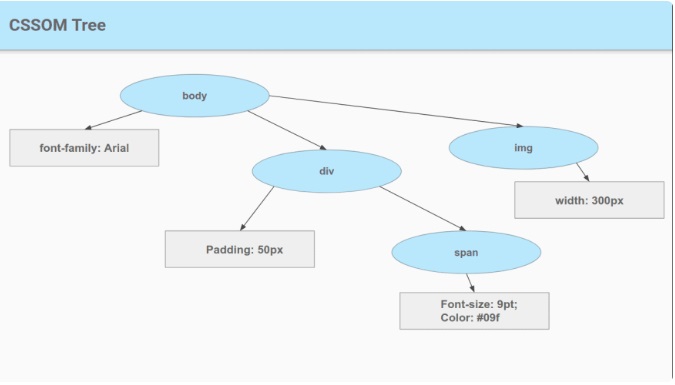
Answer- So we have HTML content at the beginning which goes through a process called tokenization, tokenization is a common process in almost every programming language where code is split into several tokens which are easier to understand while parsing. This is where the HTML's parser understands which is the start and which is the end of the tag, which tag it is and what is inside the tag.

Now we know, html tag starts at the top and then the head tag starts before the html ends so we can figure out that the head is inside html and create a tree out of it. Thus we then get something called a parse tree which eventually becomes a DOM tree as shown in the image



Just like HTML, CSS goes through a similar process where we have the CSS text and then the tokenization of CSS to eventually create something called a CSSOM or CSS Object Model.

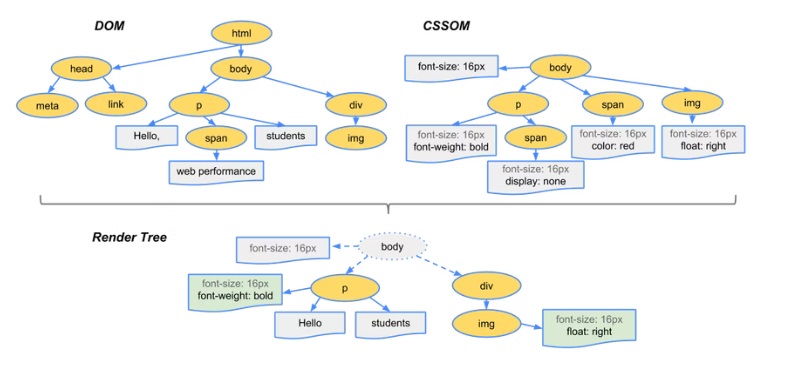
This is what a CSS Object Model looks like:



5. Tree Construction

Answer- To construct the tree, the browser roughly does the following:

1. Starting at the root of the DOM tree, traverse each visible node.
   * Some nodes are not visible (for example, script tags, meta tags, and so on), and are omitted since they are not reflected in the rendered output.
   * Some nodes are hidden via CSS and are also omitted from the render tree; for example, the span node---in the example above---is missing from the render tree because we have an explicit rule that sets the "display: none" property on it.
2. For each visible node, find the appropriate matching CSSOM rules and apply them.
3. Emit visible nodes with content and their computed styles.



6. Order of script processing

Answer- There are so many different ways to include JavaScript in a html page. Some are below

* inline code or loaded from external URI
* included in <head> or <body>
* having none, defer or async attribute (only external scripts)
* included in static source or added dynamically by other scripts (at different parse states, with different methods)

Layout and Painting

Answer- The layout is where the elements are marked on the screen. The layout includes all the calculations and mathematics behind an element's position so it takes all the properties related to the position (height, width, position, top left right bottom, etc) from The Render Tree and places the elements on the screen.

After Layout, a Paint happens. Paint takes properties like color, background-color, border-color, box-shadow, etc. to paint the screen with colors.

After the paint, we see the content on the screen and the first time we see something other than a white screen is called 'First Paint'. The term First Paint is used in performance reports to show how long your website took to show something on the screen.