***Q.1.What is the main functionality of the browser?***

***Ans-->***

***A web browser’s primary function is to render HTML,***

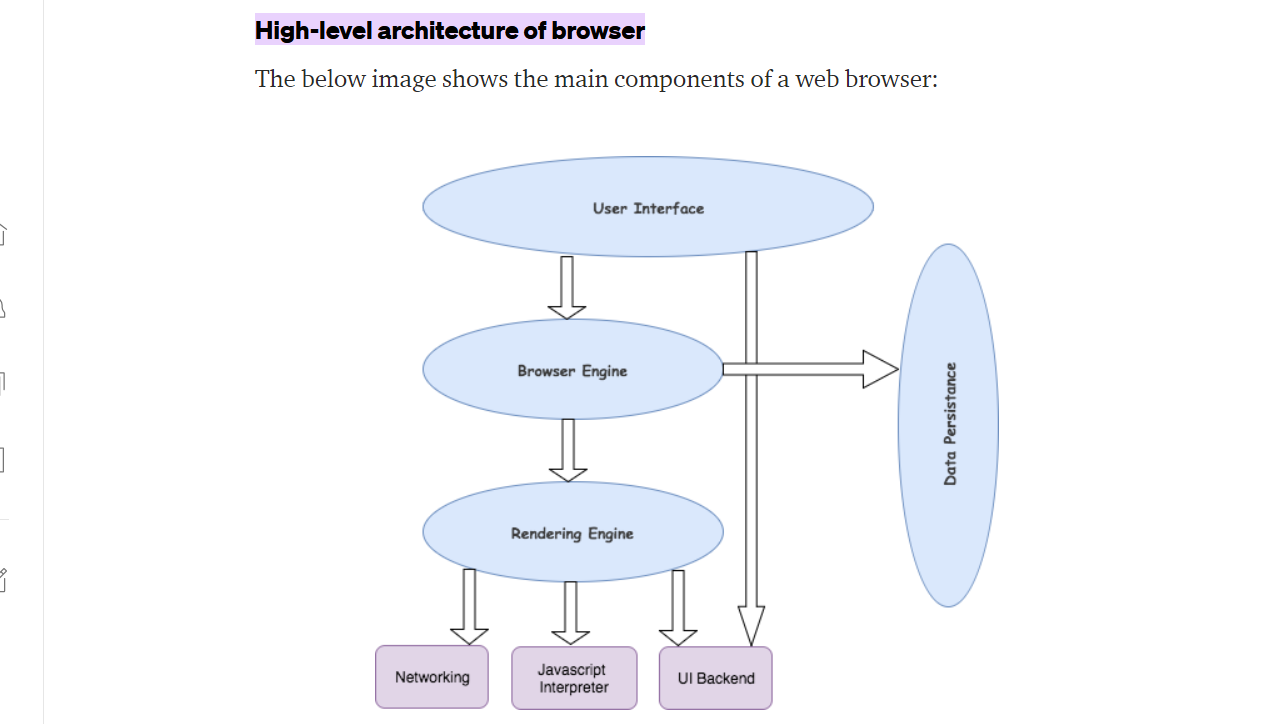
***the code that is used to design or “mark up” web pages.***

***When a browser loads a web page, it processes the HTML,***

***which may contain text, links, and references to images***

***and other items like CSS and JavaScript functions.***

***Q.2 High Level Components Of the browser***

******

1. ***The User Interface: The user interface is the space where User interacts with the browser. It includes the address bar, back and next buttons, home button, refresh and stop, bookmark option, etc. Every other part, except the window where requested web page is displayed, comes under it.***
2. ***The Browser Engine: The browser engine works as a bridge between the User interface and the rendering engine. According to the inputs from various user interfaces, it queries and manipulates the rendering engine.***
3. ***The Rendering Engine: The rendering engine, as the name suggests is responsible for rendering the requested web page on the browser screen. The rendering engine interprets the HTML, XML documents and images that are formatted using CSS and generates the layout that is displayed in the User Interface. However, using plugins or extensions, it can display other types data also. Different browsers user different rendering engines:  
   \* Internet Explorer: Trident  
   \* Firefox & other Mozilla browsers: Gecko  
   \* Chrome & Opera 15+: Blink  
   \* Chrome (iPhone) & Safari: Webkit***
4. ***Networking: Component of the browser which retrieves the URLs using the common internet protocols of HTTP or FTP. The networking component handles all aspects of Internet communication and security. The network component may implement a cache of retrieved documents in order to reduce network traffic.***
5. ***JavaScript Interpreter: It is the component of the browser which interprets and executes the javascript code embedded in a website. The interpreted results are sent to the rendering engine for display. If the script is external then first the resource is fetched from the network. Parser keeps on hold until the script is executed.***
6. ***UI Backend: UI backend is used for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific. It underneath uses operating system user interface methods.***
7. ***Data Persistence/Storage: This is a persistence layer. Browsers support storage mechanisms such as localStorage, IndexedDB, WebSQL and FileSystem. It is a small database created on the local drive of the computer where the browser is installed. It manages user data such as cache, cookies, bookmarks and preferences.***

***Q.3 Rendering Engines And Its Use***

***A rendering engine is software that draws text and images on the screen. The engine draws structured text from a document (often***[***HTML***](https://developer.mozilla.org/en-US/docs/Glossary/HTML)***), and formats it properly based on the given style declarations (often given in***[***CSS***](https://developer.mozilla.org/en-US/docs/Glossary/CSS)***). Examples of layout engines:***[***Blink***](https://developer.mozilla.org/en-US/docs/Glossary/Blink)***,***[***Gecko***](https://developer.mozilla.org/en-US/docs/Glossary/Gecko)***, EdgeHTML,***[***WebKit***](https://developer.mozilla.org/en-US/docs/Glossary/WebKit)***.***

***Q.4 Parsers(HTML,CSS)***

***Parsing means analyzing and converting a program into an internal format that a runtime environment can actually run, for example the***[***JavaScript***](https://developer.mozilla.org/en-US/docs/Glossary/JavaScript)***engine inside browsers.***

***The***[***browser parses HTML***](https://developer.mozilla.org/en-US/docs/Learn/HTML)***into a***[***DOM***](https://developer.mozilla.org/en-US/docs/Glossary/DOM)***tree. HTML parsing involves***[***tokenization***](https://developer.mozilla.org/en-US/docs/Web/API/DOMTokenList)***and tree construction. HTML tokens include start and end tags, as well as attribute names and values. If the document is well-formed, parsing it is straightforward and faster. The parser parses tokenized input into the document, building up the document tree.***

***When the HTML parser finds non-blocking resources, such as an image, the browser will request those resources and continue parsing. Parsing can continue when a CSS file is encountered, but <script> tags—particularly those without an***[***async***](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function)***or defer attribute—blocks rendering, and pauses parsing of HTML.***

***Q.5 Script Processors***

***The script processor uses the***[***script cache***](https://www.elastic.co/guide/en/elasticsearch/reference/current/scripts-and-search-speed.html)***to avoid recompiling the script for each incoming document. To improve performance, ensure the script cache is properly sized before using a script processor in production.***

***Q.6 Tree Construction***

***The site tree is a diagram or map used to organize the content, the pages of the site and the links that connect them. It ensures that visitors can quickly get to the page that interests them and not get lost. Otherwise, they will not hesitate to look for the information elsewhere.***

***Q.7 Order Of Script Processing***

***Executable objects go through four execution stages. The second one is the generation stage. Scripts are generated during this stage. The time at which the script is generated depends on object attributes. The order in which scripts are processed in an object depends on which Process pages the scripts are on.***

***This page includes the following:***

* [***Execution Stages***](https://docs.automic.com/documentation/webhelp/english/AA/12.3/DOCU/12.3/Automic%20Automation%20Guides/Content/Script/Writing/Script_Processing_Order.htm#link1)
* [***Time of Processing***](https://docs.automic.com/documentation/webhelp/english/AA/12.3/DOCU/12.3/Automic%20Automation%20Guides/Content/Script/Writing/Script_Processing_Order.htm#link2)
* [***Order of Processing***](https://docs.automic.com/documentation/webhelp/english/AA/12.3/DOCU/12.3/Automic%20Automation%20Guides/Content/Script/Writing/Script_Processing_Order.htm#link3)
* [***Processing In Scripts***](https://docs.automic.com/documentation/webhelp/english/AA/12.3/DOCU/12.3/Automic%20Automation%20Guides/Content/Script/Writing/Script_Processing_Order.htm#link4)

***Q.8 Layout And Painting***

***A website layout is the arrangement of all visual elements on a webpage. Through the intentional positioning of page elements, we can control the relationship between them to better guide the user experience. Paint actually creates the picture of the layout that needs to be rendered.***