Browser

Users interact with browser to go get something out on the web with an URL (Uniform Resource Locator)

Url Syntax:

**Protocol/scheme://<Domain>:<Port>/<Path>?<Query String>#<FragmentId>**

- Protocol Specifies the protocol to be used for interacting with Internet.
- Protocol is followed by ://
- Domain part can use wither Doman name for DNS resolution or use IP address
- Port specifies which port the communication uses. Eg. By default HTTPS uses 443
- Which is followed by path of the resource
- Optional parameters includes
    - Query string
    - Fragment Id on a page

Note: The Url in location part of the browser is accessed to fetch the resourse by the user while the urls in HTML are used to fetch the resource automaticall by the browser

Browser issues the Hypertext protocol command to communicate with resource server

Url	Request →	Path Resolution

Content Area

Resource Resolution

Response
← { Content }

Browser   Server
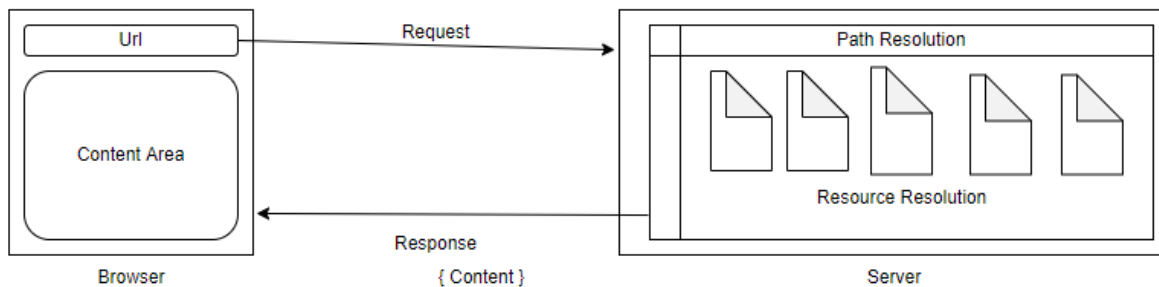
Once the browser receives the HTML file, the structure of the page is resolved and file content is resolved.
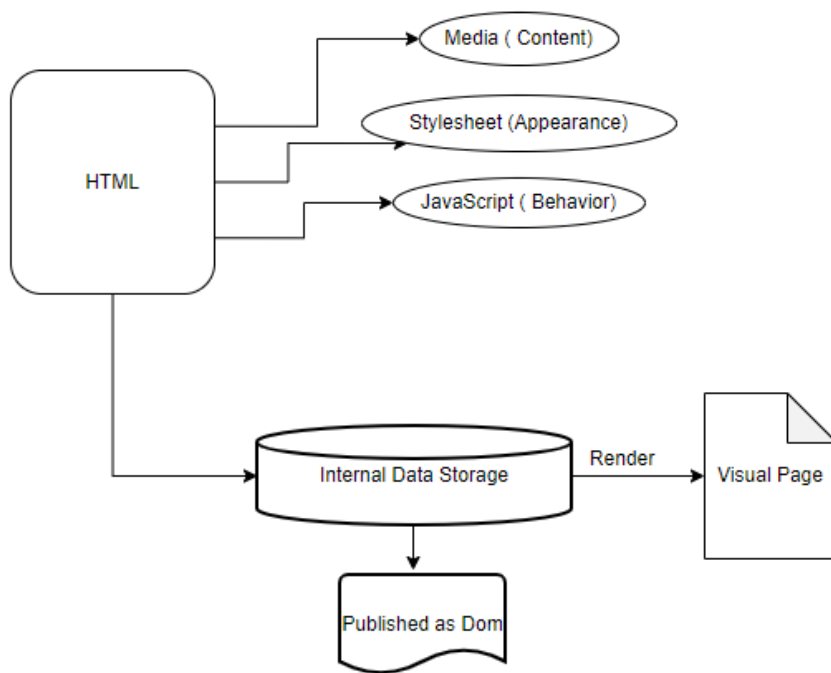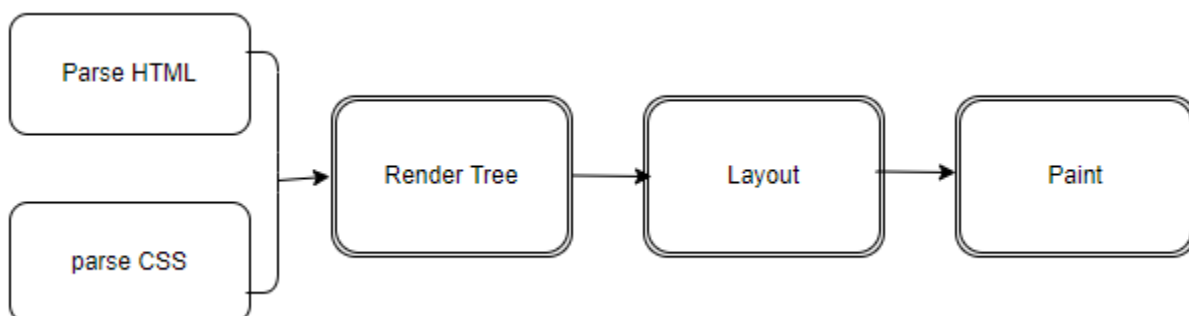
Browser includes Bindings, Rendering, Platform, JS VM modules which interact with each other, the following section focuses on the Rendering Part of browser.
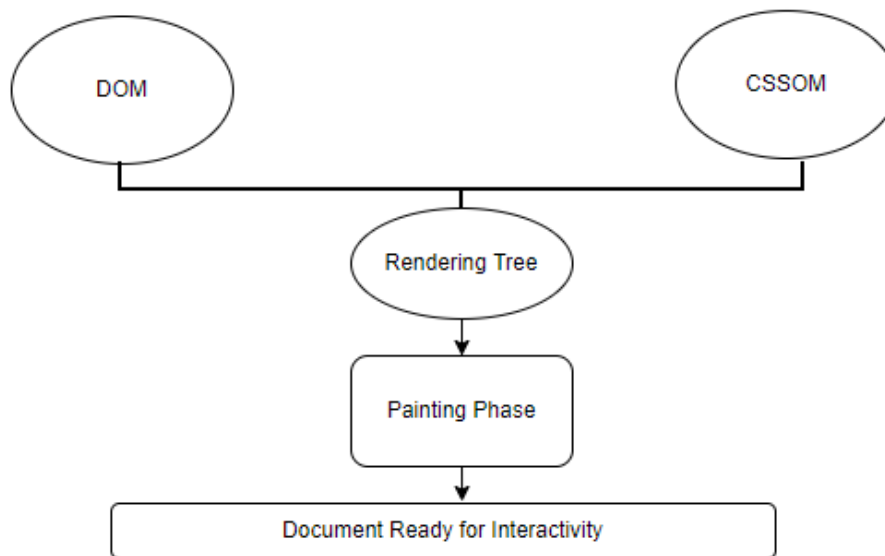
Rendering Engine

The rendering engine will start getting the contents of the requested document from the networking layer. This will usually be done in 8kB chunks.

High Level Flow of Rendering:

High Level Rendering & Painting

Rendering

Rendering includes the parsing of HTML & CSS i.e, translating a document into structure that a code can use by construction of Tree.
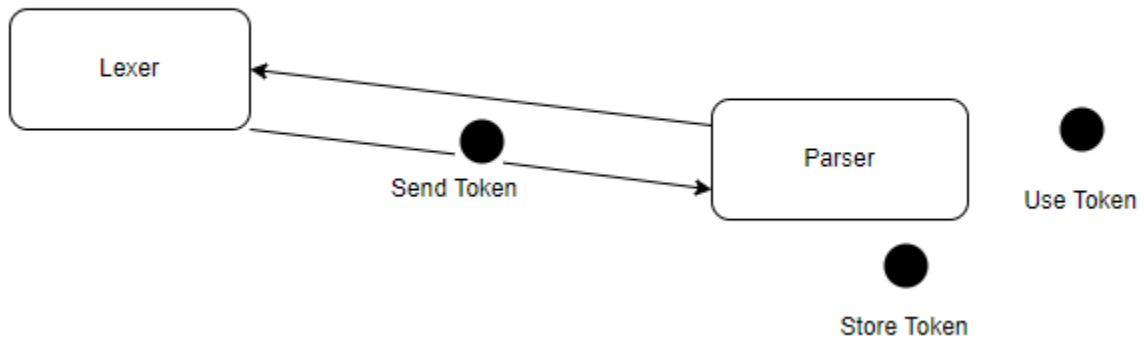
Parsing

Parsing is based on the syntax rules that document obeys. Parsing can be separated into sub process

- Lexical analysis
  - Process of breaking the input into tokens.
  - Tokens are language vocabulary
    - Collection of valid building blocks
- Syntax analysis
  - Refers to applying of language syntax rules

Parsers are usually divided into two Components

- Lexers (Tokenizer)
  - Responsible for breaking the input into valid tokens
- Parser
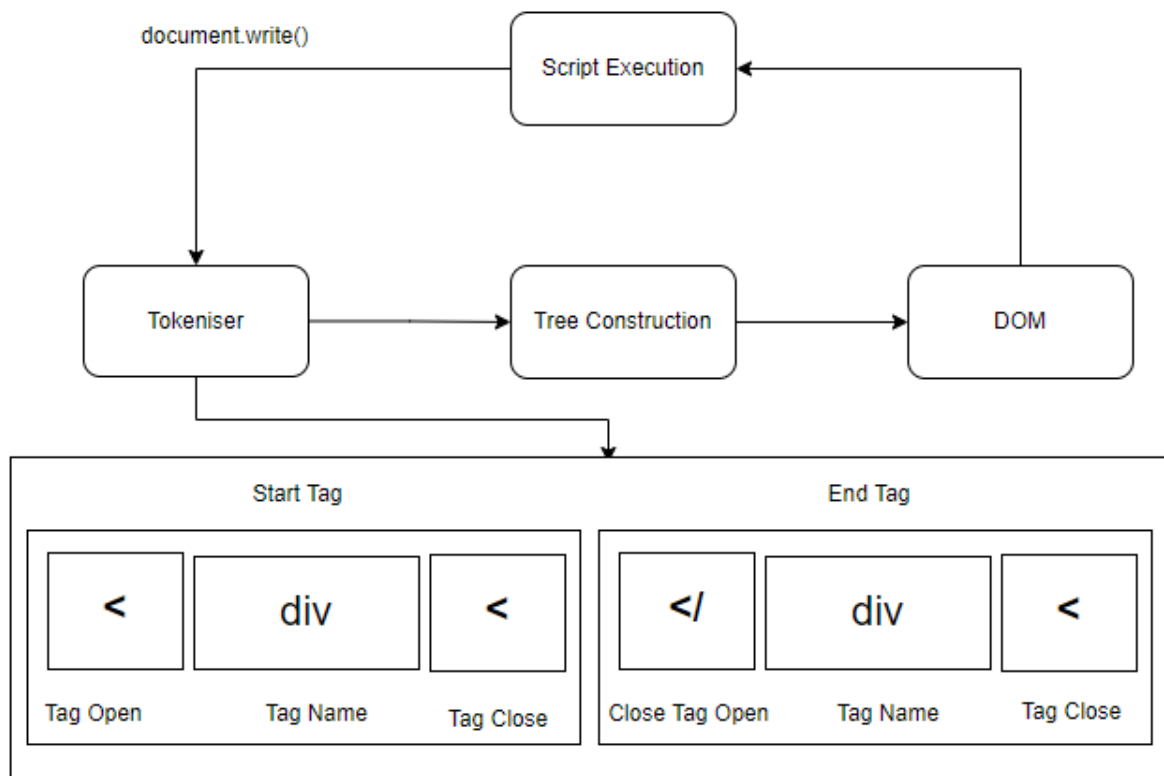 - Responsible for constructing the parse tree by analyzing the document structure according to syntax rules.

Types of Parsers
There are two types of parsers

- Top Down parser
- Bottom up parser

Parsing Process
- Iterative Process

Parse Tree:
- It is the Representation of HTML

HTML

  |-- Head

  |-- Body

    |-- Tags (eg. P)

      |-- Content (#Text)

    |--div

      |--Span

        |-- #Text

Dom Tree

HTMLhtmlElement

    |-- HTMLheadElement

    |-- HTMLbodyElement

      |--HTMLparagarphElement

        |-- Text

      |--HTMLdivElement

        |--HTMLSpanElement

          |--Text

The order of processing scripts and style sheets
1. Scripts

2. Speculative parsing
3. Style sheets

Layout

- Calculate position and size is call reflow or layout
- Computes geometry in one pass most of the cases
- It is a recursive process begins at root object

Dirty bit System

- It makes sure that browser don't need to full layout on every iteration

Global & Incremental Layout

- Layout triggered on Entire render tree is global Layout
- Dirty renders layouts are incremental

Layout Process

1. Parent renderer determines its own width
2. Parent goes over children
    a. Place the child renderer
    b. Calls child layout if neded
3. Parent uses children's sizing
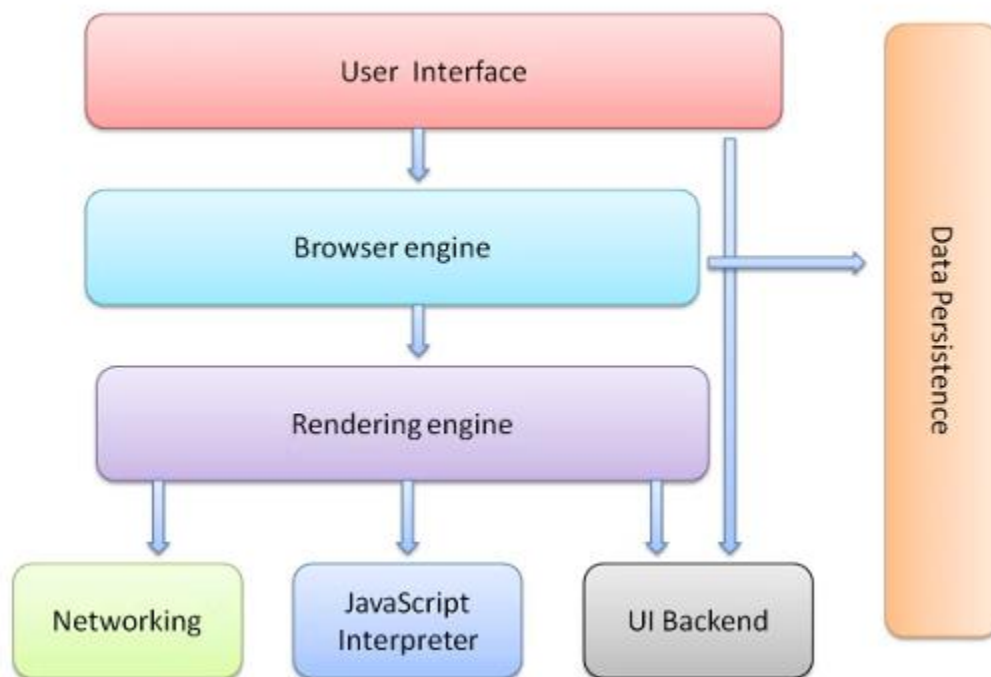4. Sets dirty bit to false

Painting

In the painting stage, the render tree is traversed and the renderer's "paint()" method is called to display content on the screen. Painting uses the UI infrastructure component.

- Its uses global & incremental painting

Painting Order

1. Background color
2. Background image
3. Border
4. Children
5. Outline

Browser Components

User Interface:

- Includes the visual elements for interaction

Browser Engine

- Direct Actions between user interface and rendering engine

Rendering Engine

- Responsible for visual representation of webpage
- Constructs the page by applying the specifications in HTML & CSS
  - HTML defines the Markup
  - CSS defines the style
- Can render Images , video, SVG, Audio and by default HTML & XML are rendered
- Eg. Chrome and Opera uses Blink Engine, Firefox uses Gecko, Safari uses Webkit
  Network Layer
  - Responsible for resolving the resource over the internet.
  Javascript Interpretor
  - Interactive logics and functionality
  Eg. 1. Chakra by Microsoft edge

2. Spidermonkey by firefox

3. V8 by google

Data Storage

- Local Storage
- Cookies
- Session Storage
- Web SQL

OSI Model

The Open Systems Interconnection (OSI) model describes seven layers that computer systems use to communicate over a network.

Network layer of browser is a part of OSI Model; the network layer has two main functions. One is breaking up segments into network packets, and reassembling the packets on the receiving end. The other is routing packets by discovering the best path across a physical network. The network layer uses network addresses (typically Internet Protocol addresses) to route packets to a destination node.

References:

https://web.dev/howbrowserswork/

https://www.imperva.com/learn/application-security/osi-model/