

# HTML

(Hyper Text Markup Language)

- HTML is used to design web page.
- Web Page is a Hyper Text Document that provide UI for interacting with resources.
- Hyper Text: The term Hyper is derived from Greek term which means “Beyond”
- Markup: It is general computer terminology derived from “Marking up”.
- Marking up is the process of preparing information to present according to requirement.
- Markup Language is a language used for Presentation.
- HTML is a presentation language.

- HTML is used to present information on browser.

## **FAQ: What is difference between programming language and markup language?**

- Programming language handles user interactions by dynamically accepting input from user, process the request and generate an output.
- Presentation language can used only for presenting content.

## **Evolution of Markup Language:**

- Internet start with a browser called “Mosaic”.
- The early markup language used for internet were “GML & SGML”

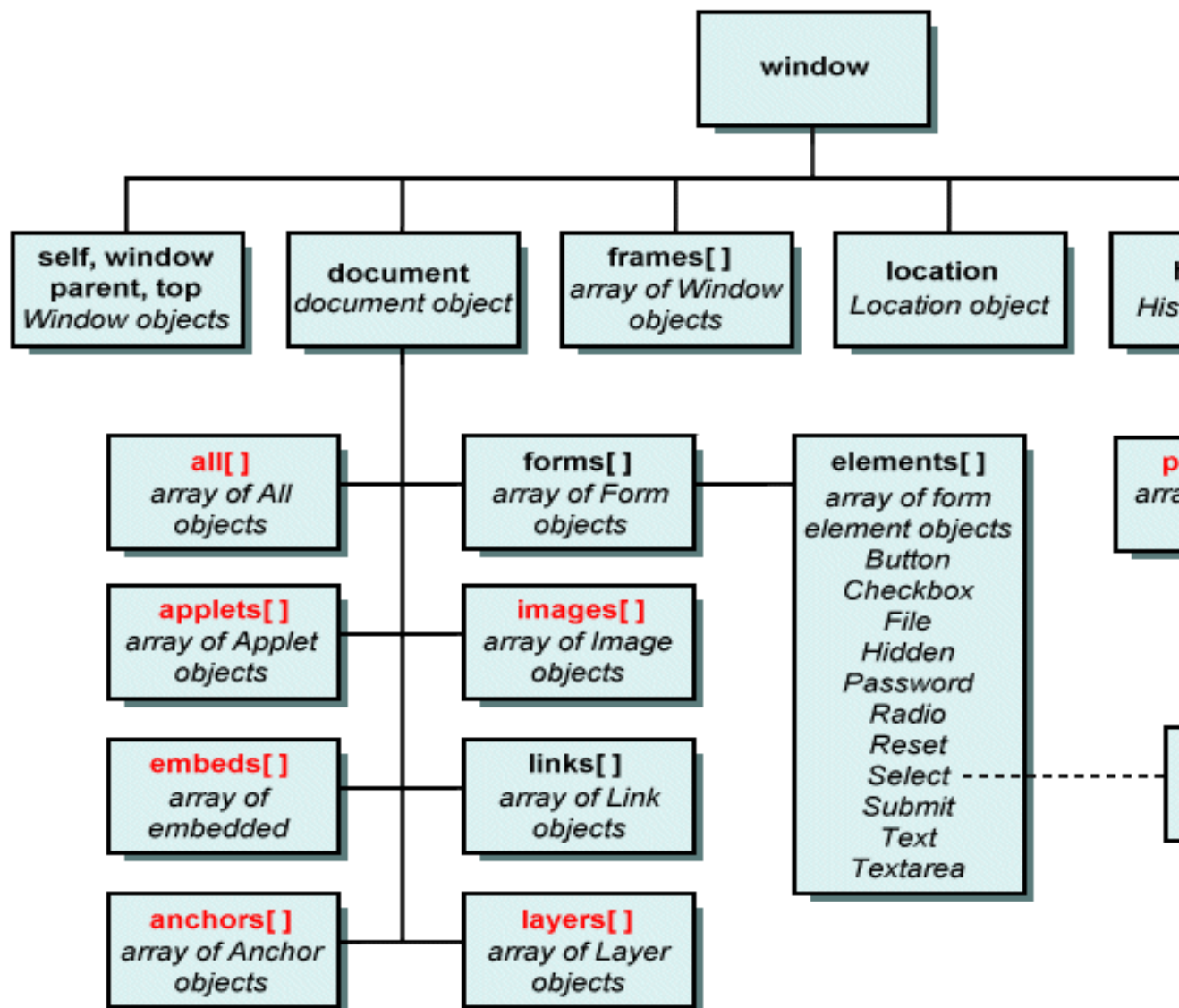
- Generic Markup Language and Standard Generic Markup Language.
- These languages were used for presentation on a browser called “Mosaic”.
- In the early 1990’s at CERN [Council for European Research and Nuclear] labs “**Tim Berners Lee**” introduced a language called “HTML”.
- Tim Berners Lee introduced the concept of Web. [Father of web]
- HTML is super set to GML and SGML.
- **1995** IETF [Internet Engineering Task Force] developed HTML version **HTML 2.0**

- **1997** W3C [World Wide Web Consortium] developed **HTML 3.2** [Jan-1997]
- **1997** December W3C developed **HTML 4.0**
- **2004** WHATWG [Web Hypertext Application Technology Work Group] started contributing to HTML along with W3C.
- **2014** W3C developed **HTML 5**. Supported by 2 groups: W3C & WHATWG

### **What HTML Comprises of?**

- HTML is a markup language that comprises of elements.
- HTML presents everything by using elements.

- Elements are presented in a hierarchy called “DOM” [Document Object Model]



- HTML elements are classified into 5 groups
  - Normal Elements

- Void Elements
- RC Data Elements
- Raw Text Elements
- Foreign Element

## **How website starts?**

- Every website by default starts with a page called “index.html”
- If there is no “index.html” then you have to manually request any page.
- Index.html is the start-up page for website.

Ex:

- Go to your project in VS code
- Add a new file by name “index.html”
- Write some text:

“Welcome to HTML”

- Request your website from browser

<http://localhost/fullstackweb>

- This will automatically load index.html
- If you want to manually access any another page, then you can request directly in URL.

<http://localhost/fullstackweb/home.html>

## **What should be the extension for Static page?**

- You can define static page with extension
  - .html
  - .htm

## **What is difference between HTML & Htm?**

- Technically both are same.
- Always recommended to define extension “.html”
- “.htm” is the extension often given by tools that can publish webpages.

Ex:

- Open Ms-Word
- Write some text in a file.
- Save as type “Web Page” with name “welcome”
- It will save the file by name “welcome.htm”

## **What are Elements and Tags?**

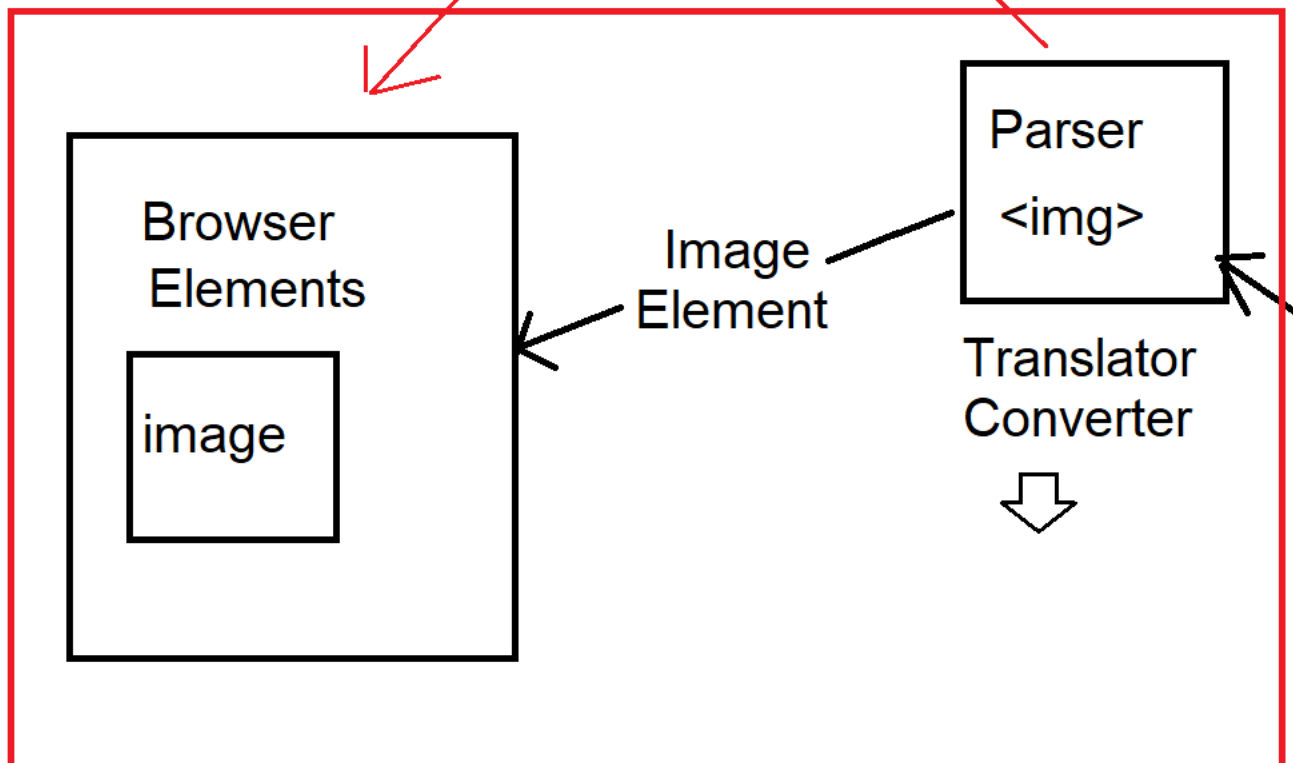


- HTML is a collection of elements arranged in a hierarchy called DOM.
- Browser can understand only Elements.
- The presentation in HTML is defined by using Elements, like Images, Anchor, Form etc.
- Elements are designed by using 2 techniques
  - Dynamically
  - Statically
- Statically elements are presented by using “tag”
- “Tag” is used to build and present elements statically.

<http://localhost/fullstackweb/index.html>



Browser loads an event called  
"DOMContentLoaded"



- You can also present element dynamically.
- You can build element dynamically

## Types of Elements in HTML

- HTML elements are classified into following types
  - Normal Element
  - Void Element
  - RC Data Element
  - Raw Text Element
  - Foreign Element

Element Type	Description
Normal Element	<ul style="list-style-type: none"><li>- A normal element directly returns some presentation directly on call back.</li><li>- It starts returning the presentation and will never stop.</li><li>- Usually it stops only at the end of</li></ul>

	<p>document.</p> <ul style="list-style-type: none"> <li>- These types of elements are designed by using Start Tag <code>&lt;tagname&gt;</code> End Tag <code>&lt;/tagname&gt;</code></li> </ul> <p>Ex: <code>&lt;b&gt; &lt;/b&gt;</code></p>
Void Element	<ul style="list-style-type: none"> <li>- Terms void defines no-return value.</li> <li>- Void element will not return anything directly on call back.</li> <li>- You have to define what to return by using attributes.</li> <li>- It will return only the content that you asked to. And stops automatically.</li> </ul>

	<ul style="list-style-type: none"><li>- These elements don't require an end tag.</li><li>- They are self-ending elements</li></ul> <p>Ex: &lt;img&gt;</p>
--	---