***Day1 : Full Stack Web Development Training Notes***  21/10/2023

World Wide Web Consortium (**W3C**) Decides the Version of HTML,CSS JS

HTML - HyperText Markup Language – HTML5

* HyperText - Text Linking the sections/ pages
* Markup language – To define Skeleton or structure for the web page
  + < > </> -- tags
  + HTML you can use only predefined tags
    - Body
    - Head
    - Label
    - P
    - Form
    - Div
    - Select 🡪 option
    - Span
    - Etc

**HTML Tag classification :**

**1.Document Structure:**

<html>: The root element that encapsulates the entire HTML document.

<head>: Contains meta-information about the document, such as the title, character encoding, and linked stylesheets.

<title>: Specifies the title of the HTML document displayed in the browser's title bar or tab.

<meta>: Provides metadata about the document, including character encoding and author information.

<link>: Establishes links to external resources like stylesheets and icons.

<style>: Used for defining inline or embedded CSS styles.

<script>: Embeds or links to JavaScript code for client-side scripting.

**2.Document Content:**

**<body>:** Encloses the visible content of the document, such as text, images, and multimedia elements.

**Headings (<h1> to <h6>):** Define hierarchical headings for content sections.

**Paragraphs (<p>):** Used to structure and format text**.**

**Lists (<ul>, <ol>, <li>):** Create unordered (bulleted) and ordered (numbered) lists**.**

**Anchor (<a>):** Creates hyperlinks to other web pages or resources.

**Images (<img>):** Embeds images within the document**.**

**Multimedia (<audio>, <video>):** Embeds audio and video content.

**Forms (<form>):** Contains form elements for user input.

**3.Text Formatting:**

**<strong> and <em>:** Emphasize text by making it bold or italic.

**<b>** and **<i>:** Define bold and italic text, which may not carry the same semantic weight as <strong> and **<em>.**

**<u>:** Underlines text.

**<s>:** Represents strikethrough text.

**<mark>:** Highlights or marks a portion of text.

**<sup>** and **<sub>:** Create superscript and subscript text, respectively.

**4. Division and Layout:**

**<div>:** A generic container for grouping and styling content.

**<span>:** Used for inline styling and grouping of elements.

**<table>, <tr>, <td>:** Create tabular data structures.

**<hr>:** Inserts a horizontal rule or line to separate content.

**<section>, <article>, <aside>:** Elements introduced in HTML5 for semantic structuring of content within a document.

**<header>, <footer>, <nav>, <main>:** More HTML5 semantic elements used for defining document sections.

5. **Interactive Elements:**

**<button>:** Represents a clickable button element.

**<input>:** Provides various input types for forms, such as text fields, radio buttons, and checkboxes.

**<select>:** Creates dropdown lists for form input.

**<textarea>:** Defines a multi-line text input field.

**<label>:** Labels form elements for better accessibility.

**<iframe>:** Embeds an external web page or document within the current page.

**<canvas>**<**svg>:** Used for drawing graphics and animations.

**HTML4 VS HTML5 Key difference**

HTML4 and HTML5 are two different versions of the Hypertext Markup Language, used for creating and structuring web content. HTML5 is the newer and more advanced version, and it introduced several significant improvements and new features over HTML4. Here's a comparison of HTML4 and HTML5:

**1.Doctype Declaration:**

HTML4: It requires a strict or transitional Doctype declaration, which specifies the version of HTML being used.

HTML5: It uses a simplified <!DOCTYPE html> declaration, making it easier to get started with.

**2.New Structural Elements:**

HTML4: It has limited structural elements like headings, paragraphs, lists, and tables.

HTML5: It introduces a set of new structural elements, such as <header>, <nav>, <section>, <article>, <aside>, and <footer, which make it easier to define the structure of a web page.

**4.Multimedia Support:**

HTML4: It has limited support for multimedia, primarily relying on plugins like Flash for video and audio.

HTML5: It includes native support for embedding multimedia elements like **<video> and <audio,** reducing the reliance on third-party plugins.

**5.Form Enhancements:**

HTML4: It provides basic form elements like text fields, radio buttons, and checkboxes.

HTML5: It introduces new form input types, attributes, and elements, including date pickers, email inputs, and <datalist>, which makes form creation more user-friendly.

**6.Canvas and SVG:**

HTML4: It lacks support for creating graphics and animations within the browser.

HTML5: It introduces the <canvas> element for 2D graphics and animation and also includes native support for Scalable Vector Graphics (SVG).

**7.Geolocation:**

HTML4: It doesn't have built-in support for determining a user's location.

HTML5: It introduces the Geolocation API, which allows web applications to access a user's geographical location, enabling location-based services.

**Offline Storage:**

HTML4: It lacks native client-side storage options.

HTML5: It includes the Web Storage API, which provides local storage (localStorage) and session storage (sessionStorage) for storing data on the client side.

**Improved Semantics:**

HTML5: It introduces more semantic elements and attributes, making it easier for search engines and assistive technologies to understand the content.

**Compatibility and Modernization:**

HTML5 is designed to be backward compatible with HTML4, so existing HTML4 content can generally be used within an HTML5 document. However, some older features may not be supported in modern HTML5 browsers.

**Adoption and Browser Support:**

HTML5 is widely adopted, and modern web development is based on HTML5 standards. All major web browsers support HTML5.

Day2 : Full Stack Web Development Training Notes 22/10/2023

1. Create new repository
2. git clone <repo\_url> into a particular directory where we want to keep
3. git branch -a : to display all the local and remove branches available right now
4. git checkout -b <name of the branch>
5. work on feature or development activity and will create files and folder
6. git status – to check the changes in git repo
   1. if working tree is clean then perform step9
7. git add <filnamewithextension> git add \* /git add .to add all files and folders
8. git commit -m “commit message”
9. git pull origin <branchname> e.g. git pull origin develop
   1. if already uptodate perform step 10
10. gill push –set-upstream origin feature1
    1. will be created in repo to be compared with other branches and to be merged

git branching strategies

