**Course: Meanstack\_May Batch(1&2)**

**Portions: Html, CSS ,Bootstrap, Git n Github, java script**

**Faculty: Sajay**

1. Define HTML?  Explain the layout of HTML?

Ans: HTML stands for Hyper Text Markup Language. It is the standard language used for creating webpages. It consists of different elements and tags. There are different elements in the webpages such as headers, footers, navigation bars, body sections and so on. The HTML element is basically defined by a start tag, some content and a closing tag. Some HTML elements have no content. These elements are called empty elements. Empty elements do not have a closing tag. All tags don’t have a closing tag. These elements instruct the browser how to display the content on the screen.

HTML Layout:

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

</body>

</html>

The <!DOCTYPE html> declaration defines that this document is an HTML5 document. The <html> element is the root element of an HTML page. The <head> element contains meta information about the HTML page. The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab). The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

2.       Define the terms:

 HTML element:

An HTML element is defined by a start tag, some content, and an end tag. It is basically everything from the start tag to the end tag. Ex:

<h1>H1 Heading</h1>

<p>paragraph.</p>

Some HTML elements such as <br> and <hr> have no content. These elements are called empty elements. Empty elements do not have an end tag. HTML elements can be nested i.e. elements can contain other elements. All HTML documents consist of nested HTML elements.

HTML Tags:

HTML attributes: HTML attributes provide additional information about HTML elements. Attributes provide additional information about elements. Attributes are always specified in the start tag. Attributes usually come in name/value pairs like: name="value". Ex- width, height, src, href, style, etc.

HTML iframe: An HTML iframe is used to display a web page within a web page. The HTML <iframe> tag specifies an inline frame. An inline frame is used to embed another document within the current HTML document. An iframe can be used as the target frame for a link.

3.       HTML 5 VS. HTML?

|  |  |
| --- | --- |
| **HTML** | **HTML 5** |
| It didn’t support audio and video without the use of flash player support. | It supports audio and video controls with the use of <audio> and <video> tags. |
| It uses cookies to store temporary data. | It uses SQL databases and application cache to store offline data. |
| Does not allow JavaScript to run in browser. | Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5. |
| Vector graphics is possible in HTML with the help of various technologies such as VML, Silver-light, Flash, etc. | Vector graphics is additionally an integral a part of HTML5 like SVG and canvas. |
| It does not allow drag and drop effects. | It allows drag and drop effects. |
| Not possible to draw shapes like circle, rectangle, triangle etc. | HTML5 allows to draw shapes like circle, rectangle, triangle etc. |
| Older version of HTML are less mobile-friendly. | HTML5 language is more mobile-friendly. |

4.       What is SVG in HTML?

SVG stands for Scalable Vector Graphics. It is used to define graphics for the Web. The HTML <svg> element is a container for SVG graphics. SVG has several methods for drawing paths, boxes, circles, text, and graphic images. SVG is basically a language for describing 2D graphics in XML. In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

5.       Define CSS. What are its components and types?

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes. The style definitions are normally saved in external .css files. With an external stylesheet file, we can change the look of an entire website by changing just one file.

6.       Differentiate CSS box model and flex box with examples.

CSS box model:

The box model relies on margins, borders, and padding to define the spacial relationships between UI elements. Using things like float and clear, these "boxes" of content can be aligned both vertically and horizontally on the page. It has been around for a long time and is the most widely supported by browsers. It relies on a janky combination of floats and clears to make things work. To achieve a truly responsive design we have to supplement our layout with more media queries using the box model.

Flexbox:

CSS flexbox mainly emphasizes responsive design. Using flexbox, we can specify how elements should grow or shrink to fit different screen sizes. Parent flex container divs contain flex items in either a vertical or horizontal direction. Flexbox provides a simpler syntax for controlling layout. We don't have to rely on a janky combination of floats and clears or media queries to make things responsive. By specifying a flex direction, flexbox makes it easy to render components like navigation menus and side bars. Flexbox gets complicated with more complex layouts and is limited to only one direction i.e., either horizontal or vertical direction.

7.       What is cell spacing and cell padding?

Cell spacing:

Cell spacing specifies the space between cells i.e., it defines the whitespace between the edges of the adjacent cells. Cells pacing provides space outside the cells. Cell Spacing is used to set space between different table cells.

Cell Padding:

Cell padding specifies the space between the border of a table cell and its contents (i.e.) it defines the whitespace between the cell edge and the content of the cell. It is used to define the distance or space inside the cell. If we have some content inside a cell the cell will leave space on each side top, left, right, bottom that’s cellpadding. Cellpadding is an important feature to format and make table cells good. Default value of cellpadding is zero 0 which we can change by adding attribute cellpadding=”5px” in table tag.

8.       What is bootstrap?

Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website. It is a front-end framework used for easier and faster web development. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others. It can also use JavaScript plug-ins. It facilitates the user to create responsive designs. It is very easy to use. It is compatible on most of browsers. It facilitates users to develop a responsive website.

9.       Define the terms:

Bootstrap carousel:

The carousel is a slideshow for cycling through a series of content, built with CSS 3D transforms and a bit of JavaScript. It works with a series of images, text, or custom markup. It also includes support for previous/next controls and indicators. Carousels don’t automatically normalize slide dimensions. While carousels support previous/next controls and indicators, they’re not explicitly required.

Bootstrap grid system:

The Bootstrap Grid System is used for layout, specifically responsive layouts. Bootstrap’s grid system uses a series of containers, rows, and columns to layout and align content. It’s built with flexbox and is fully responsive. The grid is made up of groupings of rows & columns inside 1 or more containers.

Bootstrap navbar:

The navbar is one of the prominent features of Bootstrap sites. Navbars are responsive 'meta' components that serve as navigation headers for your application or site. Navbars collapse in mobile views and become horizontal as the available viewport width increases. At its core, the navbar includes styling for site names and basic navigation.

10.   What is GIT? Explain the Git architecture?

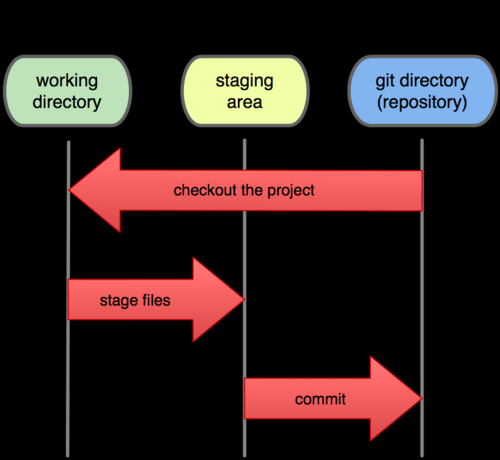
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is a mature, actively maintained open-source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel. A staggering number of software projects rely on Git for version control, including commercial projects as well as open source. In addition to being distributed, Git has been designed with performance, security and flexibility in mind.

GIT Architecture:

A Version Control System usually has three core functionalities, all of which Linus built into Git. It must be able store content, track changes to said content (all history including merge metadata), and optionally distribute the content and commit history with project collaborators. Git uses a Directed Acyclic Graph (DAG) for content storage as well as commit and merge histories. A DAG is a directed graph that has a finite number of vertices and edges (connections between vertices) that contain no cycles (acyclic). A DAG also must have topological ordering. This means that the vertices all have edges that are directed from earlier to later in the sequence.

Git also utilizes this Directed Acyclic Graph structure for content storage. Git is essentially a content-addressable filesystem made up of objects that form a hierarchy which mirrors the content’s filesystem tree. Git has three main primitive types it uses to represent content for a repository: trees, blobs, and commits. All content is essentially stored as either tree or blob objects. A blob is a file stored in the repository and a tree object references either subtrees or blobs. You can think of the blob as the file contents while the trees are like directories. A commit object, on the other hand, has three main attributes. It points to a tree which represents a top-level snapshot of the project at the time of the commit. It also contains references to the commits that came directly before it, a field for author of the commit and, optionally, a commit message.

All of these object primitives are referenced by a 40-digit SHA hash. Two identical objects will have the same hash and different objects will have different hashes. By using the SHA hash as a reference identity, Git can calculate diffing efficiently. In order to safeguard against data corruption, one can recalculate an object’s hash to easily identify corruption or data loss. Git also uses a DAG to track the history of changes to the content. As stated above, each commit object contains metadata about its ancestors where a commit can have any number of parent commits. Git’s usage of DAGs to store content and keep track of commit and merge histories allows it to maintain full branching capability as the history of a file is linked all the way back up its directory structure to the root directory and a commit object.



11. Difference between

1. GIT and GitHub

|  |  |
| --- | --- |
| **Git** | **GitHub** |
| Git is a software. | GitHub is a service. |
| Git is a command-line tool | GitHub is a graphical user interface |
| Git is installed locally on the system | GitHub is hosted on the web |
| Git is maintained by linux. | GitHub is maintained by microsoft. |
| Git is focused on version control and code sharing. | GitHub is focused on centralized source code hosting. |
| Git is a version control system to manage source code history. | GitHub is a hosting service for Git repositories. |
| Git has no user management feature. | GitHub has built-in user management feature. |

1. git pull and git fetch?

Git fetch is the command that tells your local git to retrieve the latest meta-data info from the original yet doesn’t do any file transferring. It’s more like just checking to see if there are any changes available. We use git fetch to know the changes done in the remote repo/branch since the last pull. This is useful to allow for checking before doing an actual pull, which could change files in your current branch and working copy.

The git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content. Merging remote upstream changes into your local repository is a common task in Git-based collaboration work flows. The git pull command is actually a combination of two other commands, git fetch followed by git merge. In the first stage of operation git pull will execute a git fetch scoped to the local branch that HEAD is pointed at. Once the content is downloaded, git pull will enter a merge workflow. A new merge commit will be-created and HEAD updated to point at the new commit.

12. What is JavaScript? List its features.

JavaScript is the programming language of the web. It is used in front-end as well as back-end. It is basically used to program the behaviour of the web pages. JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Features of JS:

1. Validating User’s Input
2. Simple Client-side Calculations
3. Greater Control
4. Handling Dates and Time
5. Platform Independent
6. Generating HTML Content
7. Detecting the User’s Browser and OS

13. What is the difference between Local storage & Session storage?

Local Storage:

The localStorage read-only property of the window interface allows users to access a Storage object for the Document's origin. The stored data is saved across browser sessions. localStorage is similar to sessionStorage, except that while localStorage data has no expiration time, sessionStorage data gets cleared when the page session ends. In all current browsers, localStorage seems to return a different object for each file URL. In other words, each file URL seems to have its own unique local-storage area. It is possible that browsers may change their file URL handling for localStorage at any time. In fact some browsers have changed their handling for it over time. The keys and the values stored with localStorage are always in the UTF-16 DOMString format, which uses two bytes per character. As with objects, integer keys are automatically converted to strings.

Session Storage:

Whenever a document is loaded in a particular tab in the browser, a unique page session gets created and assigned to that particular tab. That page session is valid only for that particular tab. A page session lasts as long as the tab or the browser is open, and survives over page reloads and restores. Opening a page in a new tab or window creates a new session with the value of the top-level browsing context, which differs from how session cookies work. Opening multiple tabs/windows with the same URL creates sessionStorage for each tab/window. Duplicating a tab copies the tab's sessionStorage into the new tab. Closing a tab/window ends the session and clears objects in sessionStorage. Data stored in sessionStorage is specific to the protocol of the page. In particular, data stored by a script on a site accessed with HTTP is put in a different sessionStorage object from the same site accessed with HTTPS.

14. What is 'this' keyword in JavaScript?

The JavaScript this keyword refers to the object it belongs to. It has different values depending on where it is used. In a method, this refers to the owner object. Alone, this refers to the global object. In a function, this refers to the global object. In a function, in strict mode, this is undefined. In an event, this refers to the element that received the event. Methods like call(), and apply() can refer this to any object. In an object method, this refers to the owner of the method. When used alone, the owner is the Global object, so this refers to the Global object. In strict mode, when used alone, this also refers to the Global object.

15. What are all types of Pop-up boxes available in JavaScript?

In Javascript, popup boxes are used to display the message or notification to the user. There are three types of pop up boxes in JavaScript namely Alert Box, Confirm Box and Prompt Box.

* Alert Box: It is used when a warning message is needed to be produced. When the alert box is displayed to the user, the user needs to press ok and proceed.

Syntax: alert("your Alert here")

* Prompt Box: It is a type of pop up box which is used to get the user input for further use. After entering the required details user have to click ok to proceed next stage else by pressing the cancel button user returning the null value.

Syntax: prompt("your Prompt here")

* Confirm Box: It is a type of pop up box which is used to get the authorization or permission from the user. The user has to press the ok or cancel button to proceed.

Syntax: confirm("your query here")

16. What is the difference between === operator and == operator? Explain with an example.

The ‘==’ operator tests for abstract equality i.e., it does the necessary type conversions before doing the equality comparison.

But the ‘===’ operator tests for strict equality i.e. it will not do the type conversion hence if the two values are not of the same type, when compared, it will return false.

Ex: console.log((9 == "9"); This will return true value as the datatypes are not considered.

console.log(9 === ”9”); This will return false value as the datatypes are different.

17. Explain what is pop () and push () method in JavaScript?

Pop () – This method is used to remove an item from the end of an array. pop() returns the removed item. If the array is empty, the pop() method returns undefined.

Push () - The push() method allows you to add one or more elements to the end of the array. The push() method returns the value of the length property that specifies the number of elements in the array. The push() method also allows you to add multiple items to the end of the array at a time.

18. Explain try n catch concept in JavaScript using examples.

The role of try/catch is as follows:

* try: The code block that follows this statement will contain the code to try and see whether it gives an error
* catch: The block that follows this statement will see if the code following the try statement gave an error, and decides what to do with it

A piece of code may result in an error for some reason, therefore, we employ methods to handle these errors so that the program shows us what the exact error is and we may sort out what to do about the error, avoiding a program crash.

19. Explain error and exception handling with examples.

An exception signifies the presence of an abnormal condition which requires special operable techniques. In programming terms, an exception is the anomalous code that breaks the normal flow of the code. Such exceptions require specialized programming constructs for its execution. In programming, exception handling is a process or method used for handling the abnormal statements in the code and executing them. It also enables to handle the flow control of the code/program. For handling the code, various handlers are used that process the exception and execute the code. A throw statement is used to raise an exception. It means when an abnormal condition occurs, an exception is thrown using throw.

There are following statements that handle if any exception occurs:

* throw statements
* try…catch statements
* try…catch…finally statements

20. Write a program to reverse a string.

In separate file.

21. Write a JavaScript program to find the Armstrong numbers of 3 digits.

In separate file.

22. Write a JavaScript program to construct the following pattern, using a nested for loop.

In separate file.

\*

\* \*

\* \* \*

\* \* \* \*

23. Write a JavaScript program which computes the average mark of a student (5 subjects) and assign corresponding grades.

In separate file.

24. What is DOM?

The DOM or Document Object Model is a programming interface for HTML and XML documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects. That way, programming languages can connect to the page. It is the data representation of the objects that comprise the structure and content of a document on the web. The modern DOM is built using multiple APIs that work together. The core DOM defines the objects that fundamentally describe a document and the objects within it. This is expanded upon as needed by other APIs that add new features and capabilities.

25. Design a webpage of your interest with technologies that you have learned.

Github link: https://shobhit210.github.io/JS-Assignment-1/

In separate file.