**Meta tags:-**

Meta tags provide metadata about the HTML document. They are placed within the **<head>** element of an HTML document and are not displayed on the web page itself. Meta tags are used to provide information such as character encoding, viewport settings, keywords and description for search engines,

Examples:

1) Below example deals with the viewport responsiveness of browser by taking browser viewport widths

# <meta name="viewport" content="width=device-width, initial-scale=1.0">

2) Provides a brief description of the page's content to the search engine

# <meta name="description" content="A brief description of the page">

# <meta name="keywords" content="courses, placement, keyword3">

3) Reloads the page after a specified time interval.

# <meta http-equiv="refresh" content="5">

**Comments:-**

**Documentation**: Comments are used to document the purpose or functionality of code

Documentation gives the proper opinion or info about the code to the developer or team so that code readability became easier

**Prevents the code execution**: Developers use comments for debugging purposes, such as temporarily disabling or isolating sections of code to identify issues.

**Favicon:-**

HTML Favicon is a small icon that represents a website and appears in the browser’s tab or bookmark bar. It is defined in [HTML](https://www.geeksforgeeks.org/html-introduction/) using the [<link> tag](https://www.geeksforgeeks.org/html-link-tag/) with the rel attribute set to “icon.”

# <link rel="icon" type="image/x-icon" href="url">

**Git (global information tracker):-**

Git is a distributed version control system (VCS) that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during development.

Git can acts as a medium between development environment to storing platform

Git features **Open Source and Community Support, Branching and Merging, Distributed Version Control, Staging Area** by using git we can do pull request and push the code

**Github:-**

Github is a platform where we can store, share, track and modify

By using github pages we can deploy the project. It is developed by microsoft

Github features **Open Source and Community Support, Collaboration Tools, Code Review, Code Hosting and Sharing, Security Features and Continuous Integration and Deployment (CI/CD)** by using github.

**Three components of Git**

The three areas of Git refer to the three main components where Git manages and stores data internally.

**Working Directory**: The working directory is a directory on your system where you’ll manipulate files.

**Staging Area**: Staging area acts as a medium between your working directory and repository.

**Repository**: It is referred as git respository where you will track and commit changes

**Git installation**

**Download Git Installer**: Visit the official Git website at <https://git-scm.com/> and navigate to the download page.

**Verify Installation**: After the installation is complete, open a terminal or command prompt and run “**git --version**”

**Setting Configuration: Set config using below commands**

**git config user.name "Your Name"**

**git config user.email "your@email.com"**

**Command lines of Git**

‘’**cd**” It's a command-line used to change the directory

“**ls**” It's a command-line used to list the files

“**cd..**” To move up one level

“**git version**” used to check the version also gives you the whether it is installed or not

‘**’git init**’’ Initializes a new Git repository in the current directory.

“**git status**” displays the status

“**git clone url**” clones a remote repository into your local machine.

“**git add file.name**” used to add changes to the staging area.

“**git commit**” commit stages changes to the local repository with a message’

“**git push –u origin branchname**” it pushes local commits to the remote repository.

“**git branch**” it displays the branches.

“**git checkout –b branchname**” used to switches the branches in working directory.

The **git config** command in Git is used to set or get configuration options for Git.

**git config user.name "Your Name"**

**git config user.email "your@email.com"**

How to upload files from our working directory to remote repository

**Process-1**

In Github

1. create repository in github with your project name
2. add the description of project
3. select the security of file (public or private)
4. Add a read.me file for the documentation of your project
5. After creating the repository, click on the code and copy the https url

In your working directory

1. Select a file open git bash terminal
2. Enter **git init**
3. Enter **git clone “url”**………enter the url which is created in github
4. Check the status by using **git status**
5. Change the directory using **cd foldname**
6. Copy the files into **foldname**
7. Check the status using git status
8. Add the files using “**git add filename**”
9. Commit the message using “**git commit –m “meaningfull message**””
10. Push the code using “**git push –u origin main**”

**Process-2**

In Github

1. create repository in github with your project name
2. add the description of project
3. select the security of file (public or private)
4. create the repository
5. below command lines is displayed
6. open your working directory

In your working directory

1. open the source code
2. open the terminal
3. paste the below commands one by one

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin git@github.com:saiteja-yernagula/b-n-.git

git push -u origin main

git is user-friendly software so it can display the help or suggestions , if it still remains unresolved go through below websites for reference.

**where to resolve issues by researching**

* stack overflow
* medium
* docs
* chatgpt

**How to deploy project in github-pages:-**

deploy the source code into the github repository

1. **Configure GitHub Pages:**

* Go to your repository on GitHub.
* Click on the "Settings" tab.
* Scroll down to the "GitHub Pages" section.
* Under "Source", select the branch that contains your website files (usually main or master).
* Click "Save".

1. **Access Your Website:**

* Once GitHub Pages finishes deploying your site, you can access it at https://<username>.github.io (replace <username> with your GitHub username).

1. **Updating Your Website**

Whenever you make changes to your website files, make sure to use add, commit and push those changes to GitHub. GitHub Pages will automatically update your site after you push changes to the configured branch.

**Cascading Style Sheets**

Cascading means:- Flowing one from another

What Css does:- simply flowing different styles to the html element.

**Css in official terms**

Cascading Style Sheets, is a style sheet language used to describe the presentation of a document written in HTML. CSS describes how elements should be rendered on screen. It enables web designers to control the layout, colors, fonts, and other visual aspects of multiple web pages all at once.

**How to insert css into html (3 ways)**

* Inline styling
* Internal styling
* External styling

**Inline Styling**: In inline styling CSS styles directly apply to the individual HTML elements using the style attribute.

<h1 **style="color:blue;text-align:center;"**>This is a heading</h1>

**Internal Styling**: Internal styling in CSS refers to the method of embedding CSS rules directly within the HTML document using the <style> tag.

<!DOCTYPE html>  
<html>  
<head>  
**<style>**  
body {  
  background-color: linen;  
}  
  
h1 {  
  color: maroon;  
  margin-left: 40px;  
}  
**</style>**  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
</body>  
</html>

**External Styling**: External CSS is used to style multiple HTML pages with a single style sheet. External CSS contains a separate CSS file with a **.css**extension.

**How to link external stylesheet into html document**

<head>  
<link rel="stylesheet" href="mystyle.css">  
</head>

**How to create external stylesheet**

Create a file in a folder with .css extension and write the css code.

# body {   background-color: lightblue; } h1 {   color: navy;   margin-left: 20px; }

Css have selectors , property and values

1. Selectors are used to point outs the html elements.
2. Each CSS property has a specific name and value that determine how an element is styled.



**Types of selectors**

1. **Simple selectors**
2. **Combinator selectors**
3. **Pseudo class selectors**
4. **Pseudo element selectors**
5. **Attribute selectors**

**Simple selectors**:- points out the elements based on Element name, Id, Class name

Based on name: it selects all h1 elements in a document

**h1** { color: red; }

Based on ID: Id selector uses the id attribute of an HTML element to select a specific element.

**#id1**{ color: red; }

<p **id="id1"**>Hello World!</p>

Based on class name: The class selector selects HTML elements with a specific class attribute.wc can give multiple classes in a classname.

**.classname**{ color: red; }

<p **class="classname"**>Hello World!</p>

**Universal selector**: universal selector (\*) selects all HTML elements on the page

\* { text-align: center; color: blue; }

**Grouping selectors:** grouping selectors allow you to apply the same styles to multiple selectors.

**h1, h2, h3** { font-family: Arial, sans-serif; color: #333; }

## **!important** in css: it is used to give a particular style rule more weight or importance than others. When you add !important to a CSS property.

p {  
  background-color: red !important;  
}

**Boxmodel:-** The CSS (Cascading Style Sheets) box model is a fundamental concept that describes the layout and design of elements on a web page. It consists of four main components: content, padding, border, and margin.