1) $ id - Display user id
$ pwd - current working directory
$ cd - change directory
$ ls - list files and directory
$ echo - prints the given Statement
$ mkdir - creates a new directory
$ rmdir - removes the directory
$ cat >> filename - creates and opens the file
$ cat filename - view contents in the file
$ mkdir - creates a new directory
$ rmdir - removes a directory
$ mv - rename files
$ ls -f - Distinguish files and directories
$ ls -f -r - Shows files that contained in a directory
$ ls -l - long listing
$ ls -sail - long list with inode and size
$ touch - creates an empty file
$ cp -i - copy with permission
$ ln -s - creates a soft link
$ tree - Shows the file organisation in professional manner.
$ grep -c - number of lines that matches the given string
$ grep -l - displays the files that contains the given string.
$ history - last tried commands will appear
$ man - ls - Description of commands
$ clear - clears the terminal
$ bc - basic calculator
$ cmp - compare the files
$ kill - Stops all the process

sort file - display content in alphabetical order
sort -n - display in numerical order
sort -g - according to decimal value
sort -c - checks if output is sorted or not
sort -d - arrange in dictionary order
sort -r - sort in reversing order.
tar - archiving a file.
tar - A - concadentate (Add new tar file to existing tar)
tar -c - Create tar files.
tar -d - differentiate between two files
tar -r - add new files (append)
tar -t - list all the archived files
tar -x - esctract tar files
tar -C - changes the directory
tar -f - list the archived files.
tar j - compress directly

$ printenv - display all the environment variables
$ env - display all the environment variables
$ set - Displays all the global, local and user defined variables
$ unset - unassign a variable as global
$ esport - Assign a variable as global
$ ps - display lot of information about process
$ chmod - change the file permissions.

2)

* HTML Stands for hyper text Markup Language.
* HTML is the standard markup language for creating web pages
* HTML describes the Structure of a web page
* <!DOCTYPE html> declaration defines that this document is an html5 document.
* HTML Headings

<h1> to <h6>
<h1> This is the heading </h1>
<h6> This is the heading </h6>

* Paragraph

* Links
<a>
 link

* Images
<img src = " image.jpg " >

* Break
<b>

* Style Attributes
 This is red
<body Style = " background-color : powderblue; ">
<h1 Style = " font-family : verdana; "> Hello </h1>
 Hello

<h1 Style : "text-align : center;"> paragraph </h1>

* Html formatting Elements

<b> - bold text
<Strong> - Important text
<i> - Italic text
 - Emphasized text
<mark> - Marked text
<Small> - Smaller text
<del> - deleted text
<ins> - inserted text
<Sub> - subscript text
<Sup> - Superscript text

* Abbreviations

 The <abbr title = "world health organisation">WHO </abbr> was founded in 1948

* comment

<!-- write your comments here -->

* colours

Tomato	Gray
orange	SlateBlue
DodgerBlue	Violet
MediumSeaGreen	LightGray.

* Border

<h1 Style = " border : 2px solid Tomato ; "> Hello </h1>

CSS

* CSS Stands for cascading style sheet
* CSS is the language we use to style a web page.
* CSS is used to define styles for your webpage, including the design, layout and variations in display for different device and screen size.

* There are three ways of inserting a style sheet:

  i) External CSS
  ii) Internal CSS
  iii) Inline CSS

i) External CSS

* with an external style sheet, you can change the look of an entire website by changing just one file.

```
<!DOCTYPE html>
<html>
<head>
<link rel = "stylesheet" href = "mystyle.css">
</head>
<body>
<h1> This is a heading </h1>
<p> This is a paragraph </p>
</body>
</html>
```

mystyle.css

```
body {
  background-color: lightblue;
}
h1 {
  color : navy;
  margin-left : 10px;
}
```

ii) Internal css

* An Internal style sheet may be used if one single HTML page has a unique style.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: linen;
}
h1 {
  color: purple;
  margin-left: 40px;
}
</style>
</head>
<body>
<h1> This is a heading </h1>
<p> This is a para </p>
</body>
</html>
```

iii) Inline

* An Inline style May be used to apply a unique style for a Single element

```
<!DOCTYPE html>
<html>
<body>
<h1 style = "color : blue ; text-align : center;">Heading </h1>
<p style = "color : red;"> This is a para </p>
</body>
</html>
```

3) Navigation bar

nav = main class to identify the navigation
nav-item = to enter the items
nav-link - to add the link to navigation
nav-link active/disabled = on mode/off mode

```
<nav class = "navbar navbar-expand-sm bg-light">
    <div class = "container-fluid">
    <ul class = "navbar-nav">
    <li class = "nav-item">
    <a class = "nav-link" href = "#">link1</a>
    </li>
    <li class = "nav item">
    <a class = "nav-link" href = "#">link 2</a>
    </li>
    </ul>
    </div>
</nav>
```

carousel:

carousel is a slideshow for cycling through elements

* carasel - creates a carousel
* carousel-indicator - add indicators for the carousel
* carousel-inner - adds slides to the carousel
* carousel-item - specifies the content of each slide
* carousell-control-prev - adds a previous button
* carousel-control-next - adds a next button
* slide - adds as css transition and animation

```
< div id = "demo" class = "carousel slide" data-bs-ride = "carousel">
  < div class = "carousel-inner">
  < div class = "carousel-item active">
  < img src = "Parota.jpg" class = "d-block w-100"> </div>
  < div class = "carousel-item">
  < img src = "Biriyani.jpg" class = "d-block w-100"> </div>
  < div class = "carousel-item">
    < img src = "Fried rice.jpg" class = "d-block w-100"> </div>
  </div>
```

Dropdown

A dropdown menu is a toggleable menu that allows the user to choose one value from a predefined list.

```
< div class = "dropdown">
< button type = "button" class = "btn btn-primary dropdown-toggle"
data-bs-toggle = "dropdown"> Dropdown button </button>
< ul class = "dropdown-menu">
< li > < a class = "dropdown-item" href = "#"> Link 1 </a></li>
< li > < a class = "dropdown-item" href = "#"> Link 2 </a></li>
</ul>
</div>
```

4) Steps to push file to remote repository

* Creates a new repository by filling up the required details.
* open a git bash
* Move to the 'specific path in your local computer by cd pathname.

* To locate the file 'pathname' that is
  C:/users/lenovo/Downloads/Face detect - Master
  needs to be given.
* use git init to initialize the repository
* use git add in your bash to add all the files to the given folder.
* use git status in your bash to view all the files in which are going to be staged to the first commit.

* give git commit -m "your Message"
  which adds changes to local repository.

* copy remote repository's URL from Github
* git remote add origin "url"
* git push -u origin Master is used for pushing local content to Github.
* Fill in your Github username and password
* you can finally see the file hosted on Github.

Creating branch and Merging of branch

* Main is the defaut branch in GIT, under 'main' branch we can create many sub branches

* Create a new branch using branch commands and Make it active by using checkout, commands.

* Create a new html file using cat command and save it as myfirstlhtml.html with some html content.

* Then git commit, the actions, if any authentication problem in commit, then use the below commands to set the global user

$ git config - global user.email "you @ exp.com"
$ git config - global user name "name"

* After commiting, checkout to the main branch and give $ git merge day.

* Now the files committed in day branch is merged with main branch.

* TO make this reflect in github, the use $git push

* Finally, refresh the Studentlib repository created in github.com, now you can observe the file myfirst.html created in subbranch 'day' is Merged with main branch & under main branch myfirst.html will be present in online after pushing.