



Savitribai Phule Pune University

S.Y.B. C. A. (Science)

SEMESTER III

Lab Course

BCA - 236

Computer Networks

&

Web Programming Laboratory

From the Chairman's Desk

It gives me a great pleasure to present this workbook prepared by the Board of studies in Computer Applications.

The workbook has been prepared with the objectives of bringing uniformity in implementation of lab assignments across all affiliated colleges, act as a ready reference for both fast and slow learners and facilitate continuous assessment using clearly defined rubrics.

The workbook provides, for each of the assignments, the aims, pre-requisites, related theoretical concepts with suitable examples wherever necessary, guidelines for the faculty/lab administrator, instructions for the students to perform assignments and a set of exercises divided into three sets.

I am thankful to the Chairman of this course and the entire team of editors. I am also thankful to the reviewers and members of BOS, Mr. Rahul Patil and Mr. Arun Gangarde. I thank all members of BOS and everyone who have contributed directly or indirectly for the preparation of the workbook.

Constructive criticism is welcome and to be communicated to the Chairman of the Course and overall coordinator Mr. Rahul Patil. Affiliated colleges are requested to collect feedbacks from the students for the further improvements.

I am thankful to Hon. Vice Chancellor of Savitribai Phule Pune University Prof. Dr. Nitin Karmalkar and the Dean of Faculty of Science and Technology Prof. Dr. M G Chaskar for their support and guidance.

Prof. Dr. S S Sane

Chairman, BOS in Computer Applications

SPPU, Pune

EDITORS:

- Priyamveda U. Patil
- Mrs. Veena Kiran Gandhi
- Sneha Ganesh Chavan
- Aparna Tushar Gohad
- Preeti Gawade
- Mohsin Tamboli

Compiled By:

**Prof. Kishor Dhane (Chairman, Computer Networks & Web Programming Laboratory)
VPASC,Baramati**

Reviewed By:

1. Prof. Arun Gangarde
New Arts, Commerce and Science College, Ahmednagar
BOS,BCA(Science)
2. Prof. Rahul Patil
KTHM College, Nashik.
BOS, BCA (Science)

Assignment Completion Sheet
Lab Course BCA 236: Computer Networks & Web Programming
Laboratory

Sr. No	Assignment Name	Date of Completion	Signature of Teacher
1.	Networking Commands		
2.	LAN Environment & Network Topology		
3.	Basic HTML Tags		
4.	GIT & GITHUB		
5.	List & Tables		
6.	Frames & Forms		
7.	CSS		
8.	Basics of JavaScript		
9.	Functions in JavaScript		
10.	Validation Using JavaScript & Event Handling		
11.	Designing HTML Screens		
12.	File Uploads, Field & Form Validation		
13.	Cookies		
14.	XML		

This is to certify that Mr/Ms. _____

of S.Y.B.C.A (Science) has successfully completed the Laboratory work for Lab Course BCA 236: Computer Networks & Web Programming Laboratory and has scored _____ marks out of 15 for Internal Evaluation.

Date: _____

Instructor

H.O.D / Coordinator

Internal Examiner

External Examiner

Assignment No 1 : Using networking commands and Study of network devices.

Author: Aparna TusharGohad

Allotted Slot: 1

Aim: To study different types of networking commands and different types of devices used in networking.

Pre-requisite: Basic knowledge of networking

The student should read following topics before starting exercise.

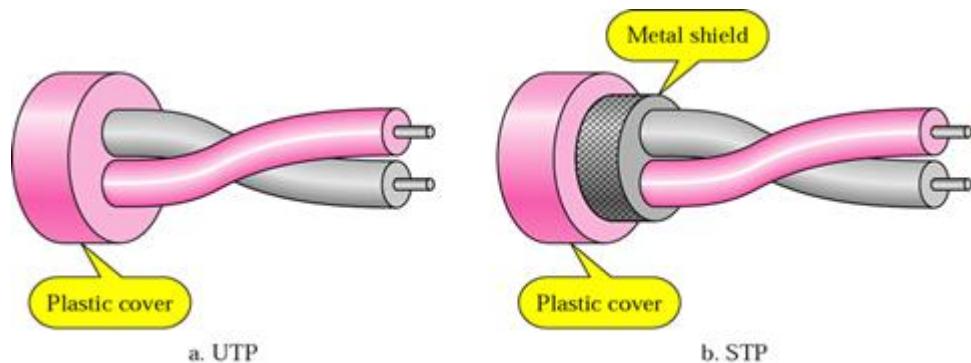
❖ Networking Commands:

Sr No	Command	Description	Example
1	Ping	PING (Packet INternet Groper) command is to test connectivity between two nodes. Ping use ICMP (Internet Control Message Protocol) to communicate to other devices. It checks if a remote host is up, or that network interfaces can be reached. You can ping host name of ip address.	[root@localhost]#ping 192.168.26.100 [root@localhost]#ping www.google.com
2	hostname	A hostname is a name that is given to a computer that attached to the network that uniquely identifies over a network and thus allows it to be accessed without using its IP address. Options: i – display ip address of hostname I - establishes all configured network interfaces and shows all network addresses host	[root@localhost]#hostname [root@localhost]# hostname i [root@localhost]# hostname I
3	traceroute	This command is used to get the route of a packet, i.e it is used to determine the path along which a packet travels. It also returns the number of hops taken by the packet to reach the destination.	root@localhost]#traceroute google.com
4	Netstat	Network Statistics is the command that is used to display routing table, connection information, the status of ports, etc.	root@localhost]#netstat
5	Who	It displays following information for each user currently logged in to the system: <ul style="list-style-type: none">• Login name of the users• Terminal line numbers• Login time of the users in to system• Remote host name of the user	root@localhost]#who
6	nmap	It produce information about the given host.	root@localhost]#nmap 192.168.20.4
7	nslookup	This command queries the DNS in order to get IP address or the domain name from DNS record.	root@localhost]#nslookupfacebook.com

❖ Types of Cables

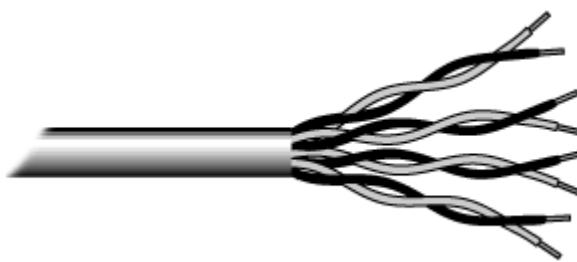
1. **Twisted pair cable:** It is a type of ordinary wiring which connects home and many business computers to the telephone company. It is made by putting two separate insulated wires together in a twisted pattern and running them parallel to each other, which helps to reduce crosstalk or electromagnetic induction between pairs of wires. Twisted pair cable is suitable for transferring balanced differential signals. The method of transmitting signals dates back to the early days of the telegraph and radio. The advantages of improved signal-to-noise ratio, crosstalk, and ground bounce that balanced signal transmission brings are particularly valuable in wide bandwidth and high fidelity systems.

According to whether the cable has a shielding layer, there are two common types of twisted pair cables—shielded twisted pair (STP) cable and unshielded twisted pair (UTP) cable. STP cable is available for Token Ring networks, while the UTP cable is more suitable for Ethernet networks. The most common UTP cable types applied in Ethernet network are cat5e, cat6a, and cat7 cables.



a) Unshielded twisted pair (UTP):

The quality of UTP may vary from telephone-grade wire to extremely high-speed cable. The cable has four pairs of wires inside the jacket. Each pair is twisted with a different number of twists per inch to help eliminate interference from adjacent pairs and other electrical devices.



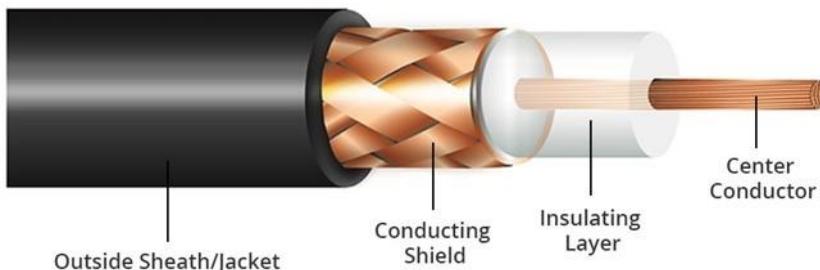
b) Shielded Twisted pair(STP):

It is a special kind of copper telephone wiring used in business installations. An external shield which functions as a ground is added to the normal twisted pair telephone wires. Shielded cables can also help in expanding the distance between the cables.

➤ Twisted pair Connector:

RJ-45 connector is used in twisted pair. This is a plastic connector that looks like a large telephone-style connector. A slot allows the RJ-45 to be inserted only one way. RJ stands for Registered Jack, implying that the connector follows a standard borrowed from the telephone industry. This standard designates which wire goes with each pin inside the connector.

2. **Coaxial cable:** Coaxial cable, or coax cable, is another type of copper cable which has an inner conductor surrounded by foam insulation, symmetrically wrapped by a woven braided metal shield, then covered by in a plastic jacket (as shown in the following image). This unique design allows coaxial cable runs to install next to metal objects such as gutters without the power losses that occur in other types of transmission lines. The coaxial cable acts as a high-frequency transmission cable made up of a single solid copper core and compared to twisted pair cable. It has 80 times or more transmission capability. This kind of cable is mainly adopted in feed lines connecting radio transmitters and receivers with their antennas, computer network connections, and distributing cable television signals.



Coaxial Cable Connector: The most common type of connector used with coaxial cables is the Bayonet-Neill-Concelman (BNC) connector. Different types of adapters are available for BNC connectors, including a T-connector, barrel connector, and terminator. Connectors on the cable are the weakest points in any network. To help avoid problems with your network, always use the BNC connectors that crimp, rather than screw, onto the cable.

3. **Fiber Optic Cable:** Fiber optic cabling consists of a center glass core surrounded by several layers of protective materials. It transmits light rather than electronic signals eliminating the problem of electrical interference. This makes it ideal for certain environments that contain a large amount of electrical interference. It has also made it the standard for connecting networks between buildings, due to its immunity to the effects of moisture and lighting.

Fiber optic cable has the ability to transmit signals over much longer distances than coaxial and twisted pair. It also has the capability to carry information at vastly greater speeds. This capacity broadens communication possibilities to include services such as video conferencing and interactive services. The cost of fiber optic cabling is comparable to copper cabling; however, it is more difficult to install and modify. 10BaseF refers to the specifications for fiber optic cable carrying Ethernet signals.



❖ Types of devices:

1. Repeater – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

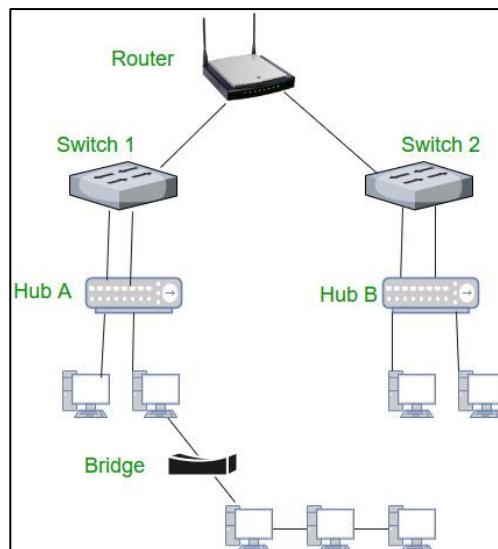
2. Hub – A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, collision domain of all hosts connected through Hub remains one. Also, they do not have intelligence to find out best path for data packets which leads to inefficiencies and wastage.

3. Bridge – A bridge operates at data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

4. Switch – A switch is a multiport bridge with a buffer and a design that can boost its efficiency (a large number of ports imply less traffic) and performance. A switch is a data link layer device. The switch can perform error checking before forwarding data, which makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only.

5. Routers – A router is a device like a switch that routes data packets based on their IP addresses. Router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divides broadcast domains of hosts connected through it.

6. Gateways - A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically work as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switch or router.



LAB Exercise:**SET A**

1. Use ping command to check connectivity of any website.
2. Use ping command to check connectivity of your departmental server by using ip address.
3. Write a command to get name of your computer.
4. Use traceroute command to trace the route of ant packet.
5. See the output of netstat command.
6. Write a command to display information of user.
7. Use nmap command to display information of any website using website name or ip address of site.
8. Use nslookup command to display ipaddress or DNS of website.

SET B

1. Check your computer lab setup. Check the types of cables and devices used.

Note: Instructors are expected to show types of cables and devices that are available in college lab.

Assignment Evaluation

0: Not Done [] 1: Incomplete [] 2: Late Complete []

3: Needs Improvement [] 4: Complete [] 5: Well done []

Signature of the Instructor: ----- **Date:** -----

Assignment No 2: Study of LAN Environment and Network topology

Author: Aparna Tushar Gohad

Allotted Slot: 1

Aim: To study commands to get information of IP address and MAC address and study of Network topologies.

Pre-requisite: Basic knowledge of networking

The student should read following topics before starting exercise.

❖ MAC Address(Physical address):

This address is included in the frame used by the data link layer. It is the lowest-level address. The size and format of these addresses vary depending on the network.

For example, Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC). Most local area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon.

e.g.

07:05:2A:01:2C:4b

❖ IP Address(Logical address):

Logical addresses are necessary for universal communications that are independent of underlying physical networks. Physical addresses are not adequate in an internetwork environment where different networks can have different address formats. A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network. The logical addresses are designed for this purpose. A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet. No two publicly addressed and visible hosts on the Internet can have the same IP address.

e.g.: 192.168.20.100

Command to get MAC address and IP address:

root@localhost]#ifconfig

Here is the output of ifconfig command.

```

himanshu@ansh:~$ ifconfig
enp3s0    Link encap:Ethernet HWaddr 70:4d:7b:70:d2:3e
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:73925 errors:0 dropped:0 overruns:0 frame:0
            TX packets:73925 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:7911049 (7.9 MB) TX bytes:7911049 (7.9 MB)

wlx18a6f713679b Link encap:Ethernet HWaddr 18:a6:f7:13:67:9b
          inet addr:192.168.2.6 Bcast:192.168.2.255 Mask:255.255.255.0
          inet6 addr: fe80::733f:7699:a8de:78ac/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:598724 errors:0 dropped:5949 overruns:0 frame:0
            TX packets:481412 errors:0 dropped:20 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:390451501 (390.4 MB) TX bytes:102506204 (102.5 MB)

```

Here enp3s0, lo and wlx18a6f713679b are the names of the active network interfaces on the system.

enp3s0 is the first Ethernet interface. (Additional Ethernet interfaces would be named eth1, eth2, etc.) This type of interface is usually a NIC connected to the network by a category 5 cable.

lo is the loopback interface. This is a special network interface that the system uses to communicate with itself.

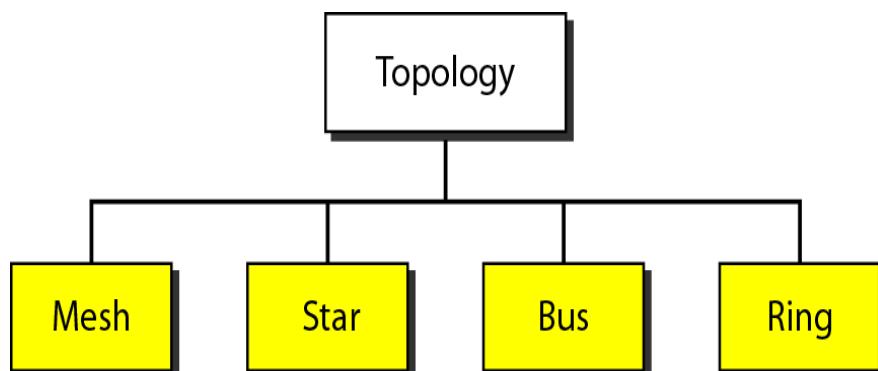
wlx18a6f713679b is the name of the first wireless network interface on the system. Additional wireless interfaces would be named wlan1, wlan2, etc

Note: *ifconfig can only assign a static IP address to a network interface. If you want to assign a dynamic IP address using DHCP, you should use the **dhclient** command.*

Options of ifconfig:

Option	Description	Example
-a	Display information for all network interfaces, even if they are down.	root@localhost]#ifconfig -a
-s	Display a short list in a format identical to the command "netstat -i".	root@localhost]#ifconfig -s
-v	Verbose mode; display additional information for certain error conditions.	root@localhost]#ifconfig -v

Types of different network topologies:



1. **Mesh:** Every link is dedicated Dedicated **point-to-point** to a central controller (Hub)
2. **Star:** No direct traffic between devices. The control acts as an exchange point-to-point link. The term dedicated means that the link carries traffic only between the two devices it connects.
3. **Bus:** It is multipoint. One long cable acts as a backbone used in the design of early LANs, and Ethernet LANs.
4. **Ring:** Each device has dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction from device to device until it reaches its destination.

LAB Exercise:

SET A

1. Type a command to get MAC and IP address of your machine.
2. Try all options of the same command.

SET B

1. Identify the topology used in your computer lab with the help of instructor.
Draw that topology in your notebook.

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well done []

Signature of the Instructor: ----- **Date:** -----

Assignment No 3: Basic HTML Tags

Author: Priyamveda U. Patil

Allotted Slot :2

Topic:**Basic HTML Tags - headings, paragraphs, line break, colors, fonts,links, Images , etc**

Introduction :

I. What is HTML :

- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension
- An HTML file can be created using a simple text editor

If you are running Windows, start Notepad.

Sample code :

Step 1. Type in the following text:

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

Step 2. Save the file as "first.htm".

Step 3. Start your Internet browser. Select "Open" (or "Open Page") in the File menu of your browser. A dialog box will appear. Select "Browse" (or "Choose File") and locate the HTML file you just created - "first.htm" - select it and click "Open".

Step 4 .you will see an address in the dialog box, for example

"C:\MyDocuments\first.htm". Click OK, and the browser will display the page.

HTML Editors :Few Editors that can be used

Sublime Text 3

Pros

- Easily customizable
- Beginner-friendly
- Pleasant color schemes to choose from.

Cons

- Can't print documents or code
- No toolbar or dashboard available.

NotePad ++

Pros

- Distraction-free interface
- Auto-completion feature
- Plug-in options for extended functionalities.

Cons

- Can be difficult to get used to for beginners
- No support for Mac.

II. Basic Construction of an HTML Page :

These tags should be placed underneath each other **at the top of every HTML page** that you create.

<!DOCTYPE html> — This tag **specifies the language** you will write on the page. In this case, the language is HTML 5.

<html> — This tag signals that from here on we are going to write in HTML code.

<head> — This is where all the **metadata for the page** goes — stuff mostly meant for search engines and other computer programs.

<body> — This is where the **content of the page** goes.

Inside the **<head>** tag, there is one tag that is always included: **<title>**,

<title>This is where we insert the page name as it will appear at the top of the browser window or tab.

```
<html>
  <head>
    <title>This Is Your Title </title>
  </head>

  <body>
    <h1> This Is Your Header </h1>
    <p> This is your paragraph. </p>
  </body>
</html>
```

III. HTML documents :

- HTML documents are text files made up of HTML elements.
- HTML elements are defined using HTML tags.
- HTML Tags :
 - HTML tags are used to mark-up HTML elements
 - HTML tags are surrounded by the two characters < and >
 - The surrounding characters are called angle brackets
 - HTML tags normally come in pairs like and
 - The first tag in a pair is the start tag, the second tag is the end tag
 - The text between the start and end tags is the element content
 - HTML tags are not case sensitive, means the same as

Example of HTML element :

1) This text is bold

The HTML element starts with a start tag:

The content of the HTML element is: This text is bold

The HTML element ends with an end tag:

The purpose of the tag is to define an HTML element that should be displayed as bold.

2. <body>

This is my first homepage. This text is bold

</body>

This HTML element starts with the start tag <body>, and ends with the end tag </body>.

The purpose of the <body> tag is to define the HTML element that contains the body of the HTML document.

IV. HTML tags and attributes :

Basic HTML Tags :

- All HTML tags must enclosed within <> these brackets.
- Every tag in HTML perform different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)
- Syntax :
<tag> content </tag>

Tag Attributes :

- Tags can have attributes. Attributes can provide additional information about the HTML elements on your page.
- This tag defines the body element of your HTML page: <body>. With an added bgcolor attribute, you can tell the browser that the background color of your page should be red, like this: <body bgcolor="red">.

- Attribute values should always be enclosed in quotes. Double style quotes are the most common, but single style quotes are also allowed.
- In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes:
For example name='John "Master" Perera'

Basic HTML Tags

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

Headings :

- Headings are defined with the `<h1>` to `<h6>` tags.
- `<h1>` defines the largest heading And `<h6>` defines the smallest heading.
- `<h1>This is a heading</h1>`
- `<h2>This is a heading</h2>`
- `<h3>This is a heading</h3>`
- `<h4>This is a heading</h4>`
- `<h5>This is a heading</h5>`
- `<h6>This is a heading</h6>`
- ***HTML automatically adds an extra blank line before and after a heading.***

Paragraphs :

- Paragraphs are defined with the `<p>` tag.
- `<p>This is a paragraph</p>`
- `<p>This is another paragraph</p>`
- ***HTML automatically adds an extra blank line before and after a paragraph.***
- Preformatting Text in HTML Paragraphs.
- You can keep the formatting with the `<pre>` element. It preserves spaces and text precisely as you typed them.
The preformatted HTML paragraphs are displayed in a fixed-width font
- Example:

`<p>Pre formatting use :</p>`

`<pre>`

This is useful for preserving the format,

It displays text the way you type them.

`</pre>`

Line Breaks :

- The `
` tag is used when you want to end a line, but don't want to start a new paragraph.
- The `
` tag forces a line break wherever you place it.
- For example : `<p>This
 is a para
graph with line breaks</p>`

- The
 tag is an empty tag. It has no closing tag.
- We can use the
 element whenever you need to add an HTML new line
- to separate your paragraphs with a horizontal line instead of a simple HTML paragraph break, use the <hr> element

- For example:

```
<h1>S.Y.B.C.A</h1>
<h2>Web Technology </h2>
<p>It is a course designed for S.Y.B.C.A </p>
<hr>
<h2>CN</h2>
<p>Study the basic concepts of computer networks.</p>
```

Comments :

- The comment tag is used to insert a comment in the HTML source code.
- A comment will be ignored by the browser.
- You can use comments to explain your code, which can help you when you edit the source code at a later date.
- <!-- This is a comment -->
- Note that you need an exclamation point after the opening bracket, but not before the closing bracket.

Summarize above tags :

Tag Description	Tag Description
<html>	Defines an HTML document
<body>	Defines the document's body
<h1> to <h6>	Defines header 1 to header 6
<p>	Defines a paragraph
 	Inserts a single line break
<hr>	Defines a horizontal rule
<!-->	Defines a comment

Tag	Description	Attributes	Example
	Used to represent text in bold		This text will appear bold
<u></u>	To make text appear underlined		This is <u> underlined tag
<i></i>	To make text appear italic		This is <i> italicized </i>
<center></center>	Centers enclosed text		<center> Text is centered </center>
<big></big>	Sets the type one font larger than the surrounding text		<big> This will appear one big</big>
<small></small>	Sets the type one font smaller than the surrounding text		<small> This will appear one size small</small>
	Formats enclosed text superscript.		X^{<small> 2}</small>
	Formats enclosed text		H_{<small>2}</small>

	subscript		>O
<marquee></marqu	Creates a scrolling tex marquee area.	<p>1.align=top middle - Aligns the marquee with the top, middle or bottom of the neighboring text line.</p> <p>2.behavior=scroll alternate - Specifies how the marquee should behave. - Scroll is the default setting and means the text should start completely off one side and scroll all the way across and completely off, then start over again. - Slide stops the scrolling when the text touches the other margin. - Alternate means the text bounces back and forth within the marquee.</p> <p>3.bgcolor="#rrggbb - Sets the background color name.</p>	<marquee align=top behaviour =slide bgcolor="pink" direction=right height=20 hspace =5 > scrolling all the way from one end to other </marquee>

HTML Text Formatting :

HTML defines a lot of elements for formatting output, like bold or italic text.

Color :

HTML colors can be defined in a name, RGB, RGBA, HEX, HSL or HSLA value and applied to either the background or the text.

Defining Color:

There is no special HTML color tag, as design is not the main function of HTML. you need to use the style attribute in the opening tag you wish to add HTML color to.

You may use the color property to change the color of your text, or background-color to change the color of the background. Both of these properties take color names, RGB, RGBA, HEX, HSL or HSLA values.

Color: Text or Background :

The background-color property provides a color for the background of the text, but not for the whole document.

If you wish to change the HTML color for the whole page, you should use the bgcolor attribute in the opening tag of the <body> section:

Example :

```
<body bgcolor="blue">  
<h2 style="color: "red">  
I am using colors to this text and to the whole document  
</h2>
```

Ways to Define Color :

Name

The color name depicts the specific name for the HTML color. There are 140 color names supported in CSS

Example :

- 1 . <h2 style="color: pink;">
Illustration of using color name to color text
</h2>

2. <h2 style="background-color: steelblue;">
Illustration of using color name to color background
</h2>

3. <h2 style="background-color: brown; color: bisque;">
Illustration of using a color name to assign a color to background and text also
</h2>

RGB and RGBA Values

The RGB value defines HTML color by mixing red, green, and blue values. The first number describes the red color input, the second – the green color input, and the third one – the blue color input.

The value of each color can vary from 0 to 255. For example, to get the same HTML red you saw in previous section, we would have to use RGB(255,0,171)

Example :

1. <h2 style="color: rgb(212, 136, 229);">
Use of RGB codes to color text
</h2>

2. <h2 style="background-color: rgb(235, 255, 138);">
Use of RGB codes to color background
</h2>

3. <h2 style="color: rgb(255, 236, 139); background-color: rgb(143, 188, 143);">
Use of RGB codes to color background and text
</h2>

RGBA values are very similar, they have one more value. The additional fourth value A stands for alpha and represents the opacity. It can be defined in a number from 0 (not transparent) to 1 (completely transparent):

Example:

1. <h2 style="color: rgba(212, 166, 239, 0.35);">
 Use of RGBA
</h2>
2. <h2 style="color: rgba(222, 106, 248, 0.75);">
 Use of RGBA
</h2>

HEX Value

HEX color value works pretty similarly to RGB .

To describe the intensity of the color Using HEX, the code contains both numbers from 0 to 9 and letters from A to F. The first two symbols determine red intensity, the two in the middle - green intensity, and the last two - blue intensity.

For example, to get a simple HTML blue, we would use the code #0000fe:

1. <h2 style="color: #FC9CF9;">
 Use of Hex code to color text
</h2>
2. <h2 style="background-color: #FFEC8B;">
 Use of Hex code to color background
</h2>
3. <h2 style="color: #B0E0E6; background-color: #ACAFFF;">
 Use of Hex code to color and text
</h2>

HSL and HSLA Values:

In HTML, colors can also be defined in HSL values. HSL stands for hue, saturation and lightness:

Hue is defined in degrees from 0 to 360. On a color wheel, red is around 0/360, green is at 120 and blue is at 240.

Saturation is defined in percentages from 0 (black and white) to 100 (full color).

Lightness is defined in percentages from 0 (black) to 100 (white).

For example, to color the background in HTML blue, you could use hsl(240, 100%, 50%):

Example:

- 1 . <h2 style="color: hsl(217, 97%, 57%);">
 Use of HSL codes to color text
</h2>
2. <h2 style="background-color: hsl(218, 77%, 88%);">
 Use of HSL codes to color background
</h2>
3. <h2 style="color: hsl(38, 95%, 25%); background-color: hsl(38, 77%, 88%);">
 Use of HSL codes to color background and text
</h2>

V. HTML Attributes

- All HTML elements can have attributes
- Attributes provide additional information about elements
- Attributes are always specified in the start tag
- Attributes usually come in name/value pairs like: name="value"

href Attribute

The `<a>` tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to:

Example

```
<a href="https://www.google.com">Search Engine</a>
```

The src Attribute

The `` tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed:

Example

```

```

The width and height Attributes

The `` tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels):

Example

```

```

The alt Attribute

The required alt attribute for the `` tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.

Example

```

```

The style Attribute

The style attribute is used to add styles to an element, such as color, font, size, and more.

Example

```
<p style="color:blue;">This is a blue paragraph.</p>
```

The lang Attribute

You should always include the lang attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

The following example specifies English as the language:

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

The title Attribute

The title attribute defines some extra information about an element.

The value of the title attribute will be displayed as a tooltip when you mouse over the element:

Example<p title="I'm a tooltip">This is a paragraph.</p>

•Summarization of Tags used to add Hyperlinks in html document are given in the foll. table.

Sr	Tag	Description	Attributes	Example
1	<A>	Adds an anchor or hyperlink	href= “url” specifies the url of the target page.	<html><body>Click here to search</body></html>

Solved Assignments.

I. Check the output of the following :

1. <Html>
 <Head>
 <Title>The First Page</title>
 </head>
 <Body>
 Hello World
 </body>
</html>

2. <!DOCTYPE html>
 <html>
 <body>
 <h1>This is heading 1</h1>
 <h2>This is heading 2</h2>
 <h3>This is heading 3</h3>
 <h4>This is heading 4</h4>
 <h5>This is heading 5</h5>
 <h6>This is heading 6</h6>
 </body>
</html>

Explanation of example:

- The <!DOCTYPE html> declaration defines this document to be HTML5.
- The <html> element is the root element of an HTML page.
- The <head> element contains meta information about the document.
- The <title> element specifies a title for the document.
- The <body> element contains the visible page content.

3.

```
<!DOCTYPE html>
<html>

<head>
<meta charset="utf-8">
<title>coffee restaurant </title>
</head>

<body>
Coffee Restaurant
The Restaurant
The Coffee Restaurant offers casual lunch and dinner fare in a hip atmosphere. The menu changes regularly to highlight the freshest ingredients.

Catering
You have fun... we'll do the cooking. Coffee Restaurant can handle events from snacks for club to elegant corporates..
Location and Hours New Highway,Mumbai ; Monday through Thursday 11am to 9pm, Friday and Saturday, 11am to midnight
</body>
</html>
```

Explanation:

Create the document head that contains the title for the page. Insert `<head>` and `</head>` tags before the content. Within the head element, add information about the character encoding `<meta charset="utf-8">`, and the title, “Black Goose Bistro”, surrounded by opening and closing `<title>` tags

4.<!DOCTYPE html>

```
<html>
<head>
    <title>Display_Paragraph</title>
</head>
<body>
    <pre>
        This paragraph has multiple
        lines. But it is displayed
        as it is unlike the paragraph
        tag.
    </pre>

    <pre>
        This    paragraph has multiple
        spaces. But    it is displayed
        as it is unlike the paragraph
        tag.
    </pre>
</body>
</html>
```

5.

```
<!-- Write HTML code here -->
<head>
<title>Wel-Come All</title>
<style type="text/css">
    h1{
        color:#0FFF0;
        background-color: hsl(200, 50%, 20%);
        color: hsl(200, 20%, 90%);

    }

    h4{
        color:rgb(0, 255, 0);
        background-color: hsl(150, 20%, 40%);
        color: hsl(360, 30%, 90%);
    }
    li{
        color:rgba(11, 99, 150, 1);
        background-color: hsl(250, 45%, 60%);
        color: hsl(175, 35%, 87%);
    }
</style>
</head>
<body>
<h1>Wel-Come All</h1>
<h4>Subjects for SY</h4>
<ul>
<li>Data Structure</li>
<li>Computer Networks</li>
<li>Web Tecnology</li>
</ul>
</body>
</html>
```

3.Create an html page with following specifications

- a. Title should be about myCity
- b. Place your City name at the top of the page in large text and in blue color
- c. Add names of landmarks in your city each in a different color, style and typeface
- d One of the landmark, your college name should be blinking
- e. Add scrolling text with a message of your choice.
- f . Add some image at the bottom

Solution :

```
<HTML>
<HEAD><TITLE><CENTER>My CITY</CENTER></TITLE></HEAD>
<BODY BGCOLOR="PINK">
<FONT SIZE="7" FACE="ARIAL" COLOR="ORANGE"><CENTER>PUNE,
```

```
<SMALL>MAHARASHTRA</SMALL></FONT><BR>
<BODY BGCOLOR="RED"><CENTER>
<font size="4" face="arial" color="pink"><B>Landmarks</B></font><BR>
<font size="4" face="arial" color="yellow"><B>Gateway of India</B></font><BR>
<font size="4" face="arial" color="red"><B>Elephanta Caves</B></font><BR>
<font size="4" face="arial" color="blue"><B>Mahalaxmi mandir </B></font><BR>
<BODY BGCOLOR="PINK"><CENTER>
<MARQUEE BEHAVIOUR="SLIDE"> Wonderful Place To Visit</MARQUEE>
<MARQUEE BEHAVIOUR="SLIDE">Mumbai Is best City </MARQUEE>
<IMG SRC="b.jpg" WIDTH="400" HEIGHT="400" ALT="IMAGE CANNOT BE DISPLAYED">
</BODY>
</HTML>
```

LAB Exercise:

Set A

1. Create an html page with all the different text styles (bold, italic and underlined) and its combinations on separate lines. State style of each line in its text.
2. Create an html page containing the polynomial expression as follows:

$$A_0 + A_1 X + A_2 X^2 + A_3 X^3$$

3. Write a HTML script to display following screen in figure 1.

Set B

1. Create an html page with red background with a message “warning” in large size blinking.

Add scrolling text “read the message” below it.

2. Create a HTML 5 page with following specifications

The Background colour should be green.

The text colour should be red.

The heading should be large in size as ‘ My Assignment’.

Display a horizontal line after the heading.

Display your name in Bold, address in Italics and year as S.Y

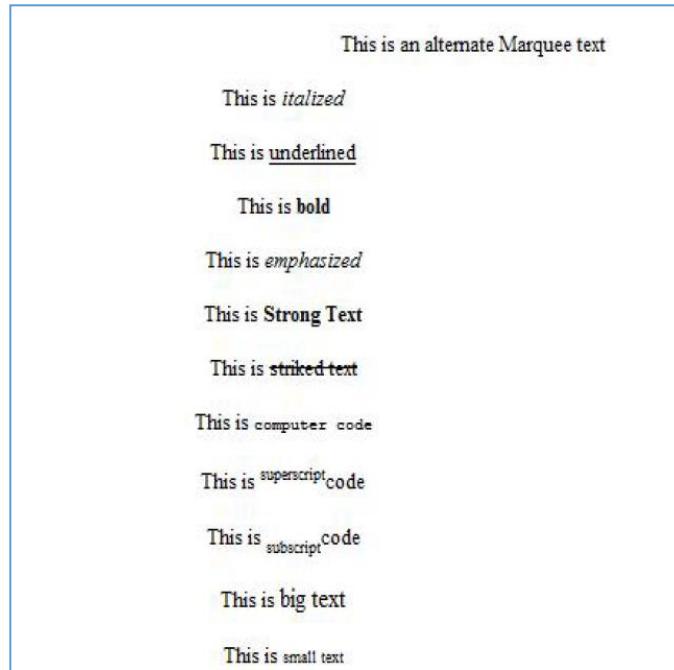


Figure 1

Set C :

1. Write a HTML script to display following screen

To illustrate link Tags

Text as a link/hyperlink to another page : [Click here!!!](#)

Image as a link/hyperlink :

See also Chapter 8 (link within a page)

Chapter 1
This chapter explains Pointers

Chapter 2
This chapter explains variables

Chapter 3
This chapter explains operator

Chapter 4
This chapter explains structure

2. Create a Web Site with following specifications

- a) Title should be about MY CITY
- b) Put image of your city map in the background
- c) Place popular college name of your city at the bottom in smaller size
- d) Add names of historical places in a different color, style and typeface
- e) Add scrolling text with a message of your choice
- f) Add photo of historical place at the top

LAB Exercise Assignment on Hyper-Link.

Use an image as hyperlink. We just need to use an image inside hyperlink at the place of text as shown below –

```
<!DOCTYPE html>
<html>

<head>
<title>Image Hyperlink Example</title>
</head>

<body>
<p>Click following link</p>
<a href="google.com" target="_self">

</a>
</body>

</html>
```

The target attribute specifies where to open the linked document:

Attribute	Description
_blank	Opens the linked document in a new window or tab
_self	Opens the linked document in the same frame as it was clicked (this is default)
_parent	Opens the linked document in the parent frame
_top	Opens the linked document in the full body of the window
_blank	Opens the linked document in a new window or tab

Set A:

1. Create an HTML page giving information about celebrating Dussehra in your country.

Use various tags in HTML to give it a pleasant look. It should be having following links:

- a. clip of Dussehra celebration
- b. Information why it is celebrated
- c. When it is celebrated
- d. What you do on this day

2. Create an html5 page with following specifications

- a. Title should be about MYCOLLEGE
- b. Put the windows logo image in the background
- c. Place your college name at the top of the page in large text followed by address in smaller size
- d. Add names of courses offered each in a different color, style, and typeface
- e. Add scrolling text with message of your choice
- f. Add college image at the bottom

Set B:

1. From Set A Example 1.

- a. Insert an Image of Dushera festival
- b. Create a hyperlink on that image .
- c. on clicking the image the information of the festival should be shown

2. From Set A Example 2.

- a. Insert the college logo
- b. Create a hyperlink on that image .
- c. on clicking the image the information of the college should be shown .

Set C:

1. From Assignment 1 , Set C, Program 2

- a. Insert the image for each historical place
- b. Create a hyperlink on all images .
- c. on clicking the image the information of those places.

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well done []

Signature of the Instructor: ----- **Date:** -----

Assignment No.4: Introduction to Git & GitHub

Author : Sneha Ganesh Chavan

Allotted Slot: 1

Perquisites:

- a) Terminal commands of Linux like: ls, mkdir, rm etc.
- b) Version Control System.

This assignment will be divided into two parts Part I will be about Git and Part II about GitHub.

PART-I

Before we get started with what is git and GitHub we need to understand the sheer importance of it and why is it widely used in the IT industry extremely.

To begin with, I would like to introduce you to the **Version Control System (VCS)**. It tracks the history of changes as people and teams collaborate on projects together. As the project evolves, teams can run tests, fix bugs, and contribute new code with the confidence that any version can be recovered at any time. Developers can review project history to find out:

- Which changes were made?
- Who made the changes?
- When were the changes made?
- Why were the changes needed?

Why GitHub?

Apart from definition, I would like to take you through real-world scenarios to understand this in a very Crisp way. In a company, you aren't the only person who works on a project but the entire team does. The work is divided between all the team members. Now if you and one of your colleagues are working on the same web page and didn't communicate with each other about it there will be clashes. To avoid that git helps you to know who is working on which part of the code as well as when you make numerous changes to your code. You submit the changes once to git and next time you think I need to make more changes to it.

Then you change it and again submit it. You end up doing this 6 times more. So, there will be a total of 8 files which you have submitted serving the same purpose but some modification done to the file. Each file has a different modification. Then you end up thinking what I did the 3rd time was the perfect code. The best part is you can get the 3rd modified file and restore it. That is about Version Control System.

Developers work in every time zone. With a Distributed VCS like Git, collaboration can happen any time while maintaining source code integrity. Using branches, developers can safely propose changes to production code.

GitHub has your various versions of your files that are stored on the cloud. There are many VCS but why we choose just GitHub? Because it is open source as well as the above benefits listed above.

Git is a remote repository which is only limited to your personal computer or laptop.

GitHub hosts your remote repository online on web page so others can see your work.

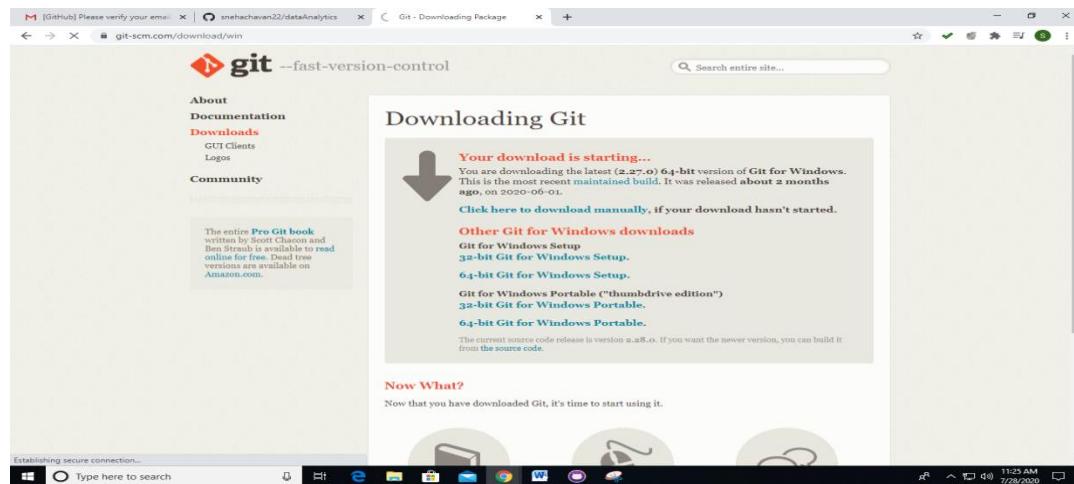
Both of them are collaborated. Git commands are used to upload the project on the GitHub

"BY THE WAY GIT AND GITHUB ARE NOT SAME"

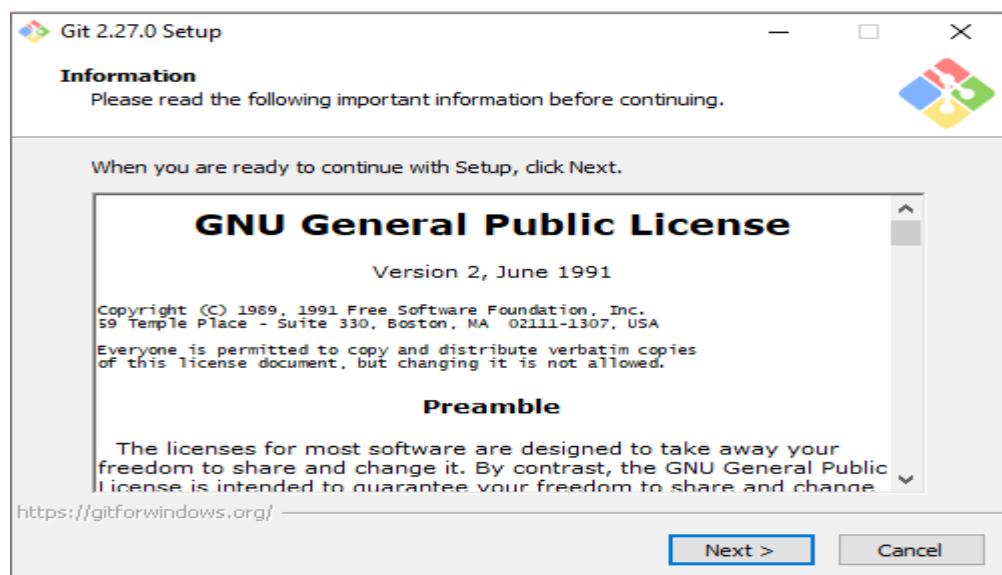
Installation of Git:

Lets go ahead and see how the installation is supposed to be done, it would be based on windows10 64bit, but there would be youtube videos for unix, macbook and windows7.

- 1) Open Browser and then type this url: <https://git-scm.com/downloads>
This below page would open.

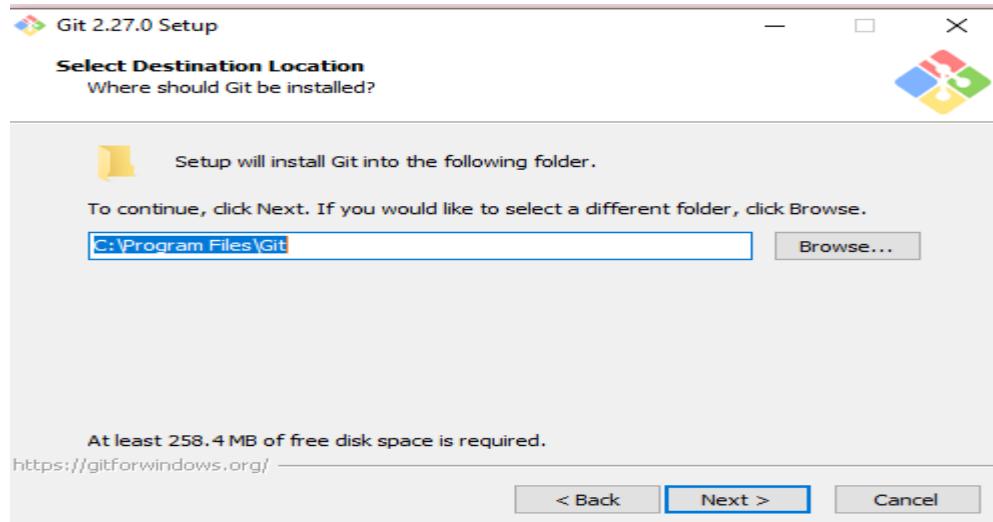


- 2) According to your system Configuration select any one.
Once it gets downloaded double click on it and it will give a pop like:

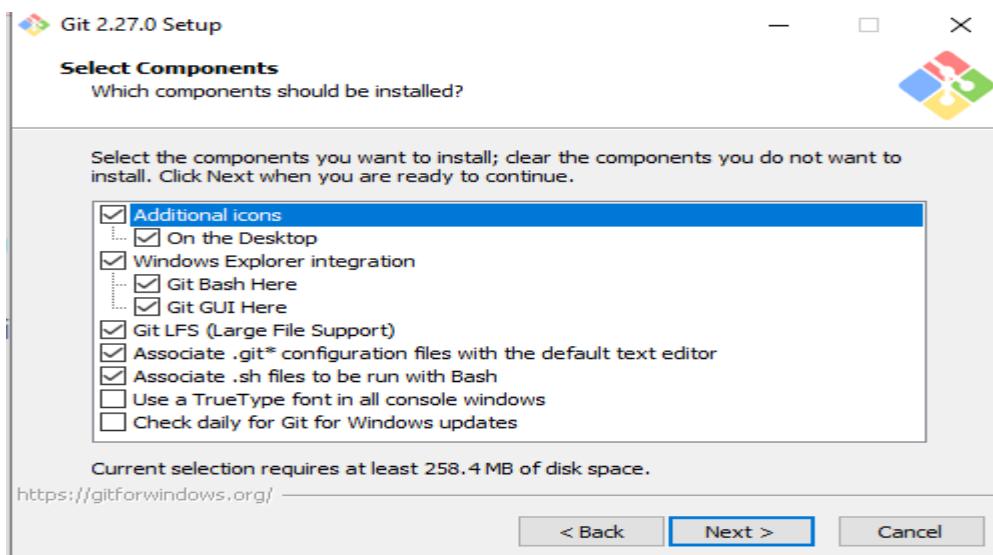


Click on next.

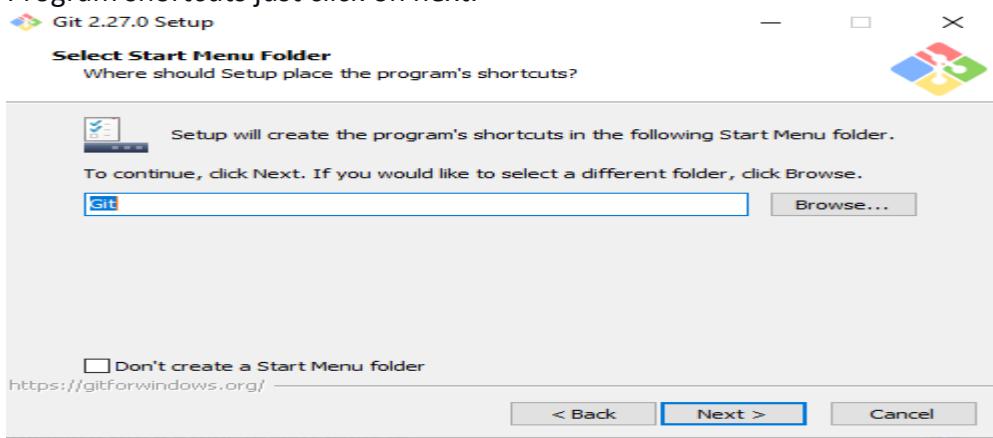
- 3) Path in which git will be stored. Keep it default if you don't want to change it.
Click on next after that.



- 4) Select the required fields and then click on next.



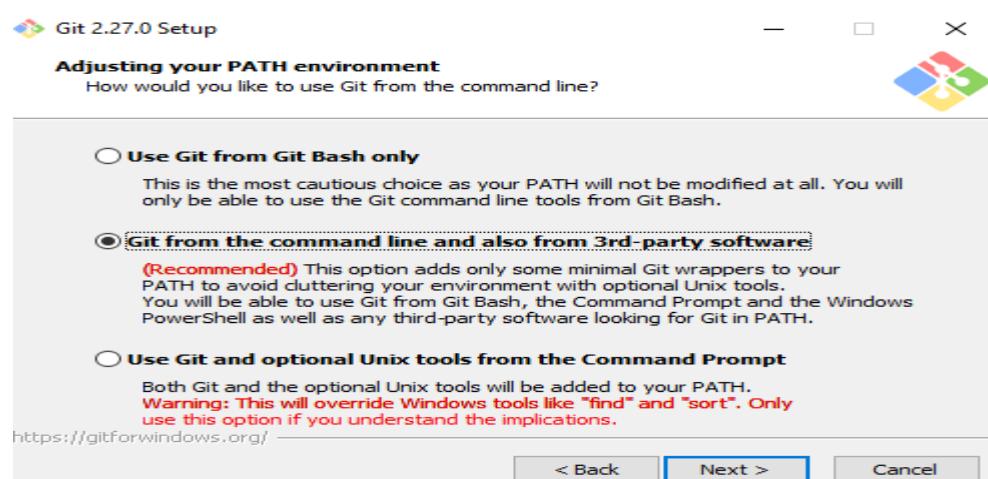
- 5) Program shortcuts just click on next.



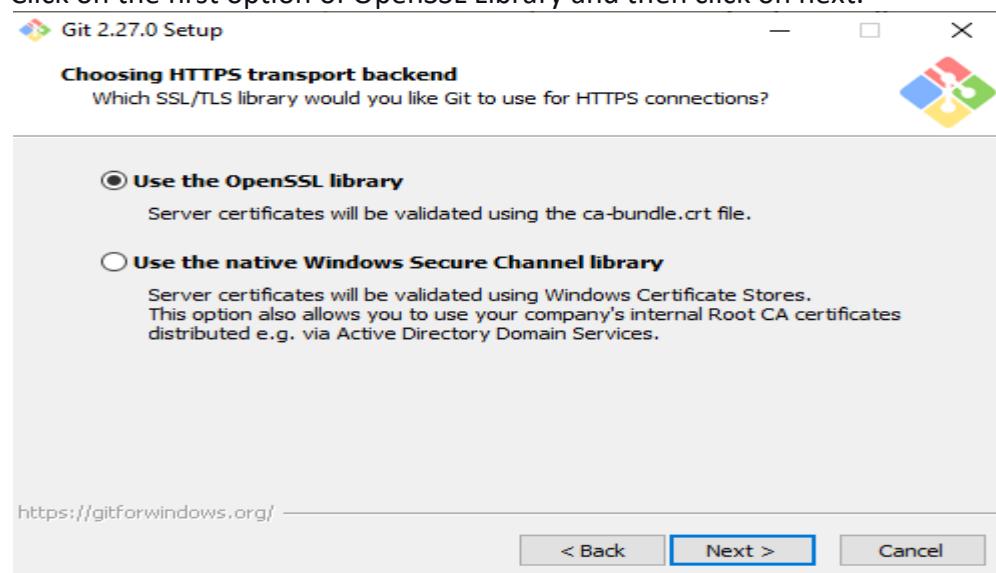
- 6) There will be multiple options given to you on for your preference if not vim editor you can choose any other editor from the drop down and then click on next.



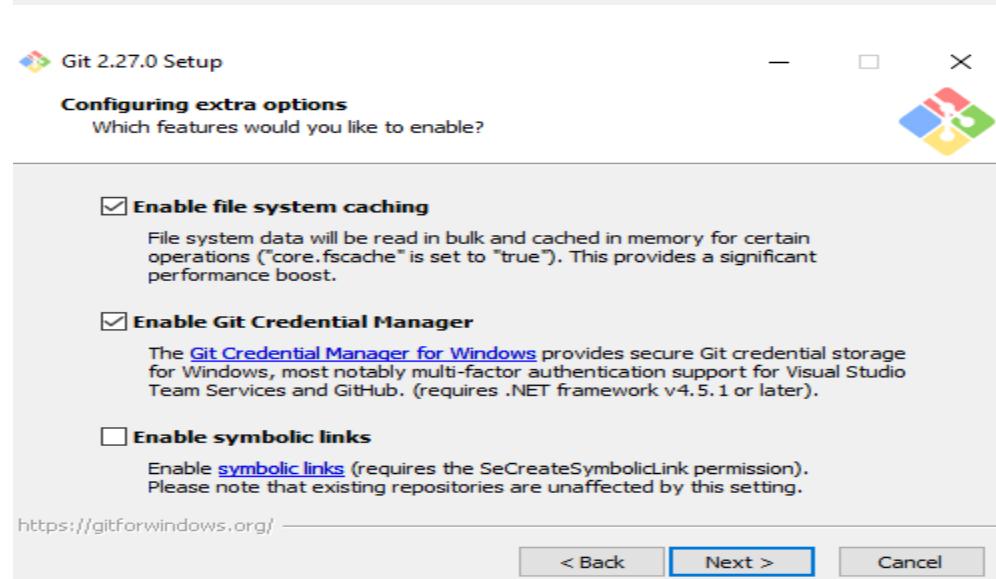
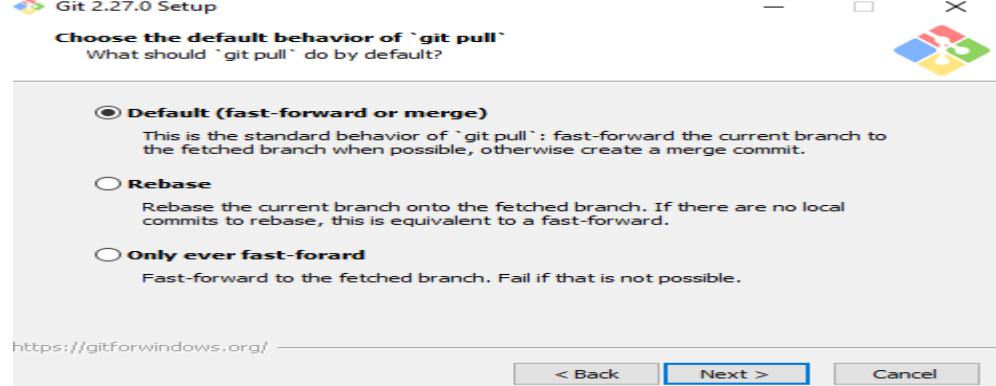
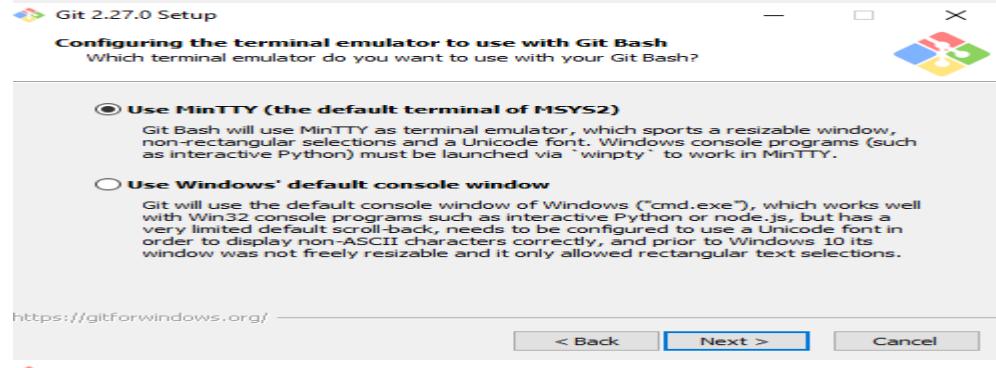
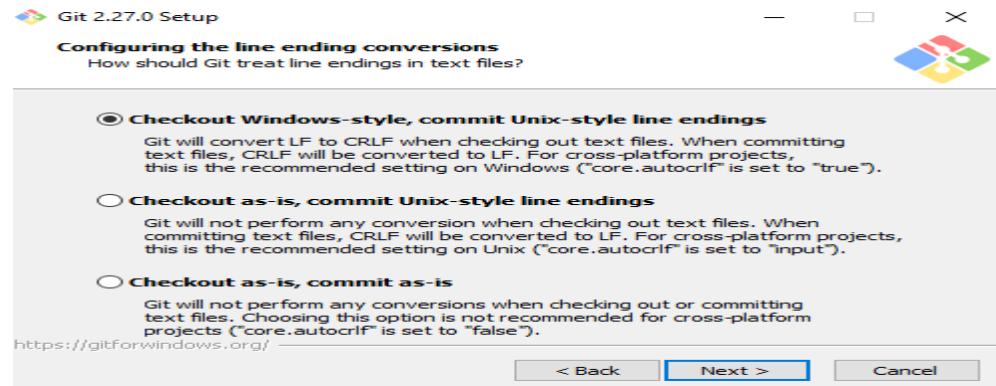
- 7) Instead of manually going and setting the path git gives you an option to set the path in the installation itself so select second option and click on next.



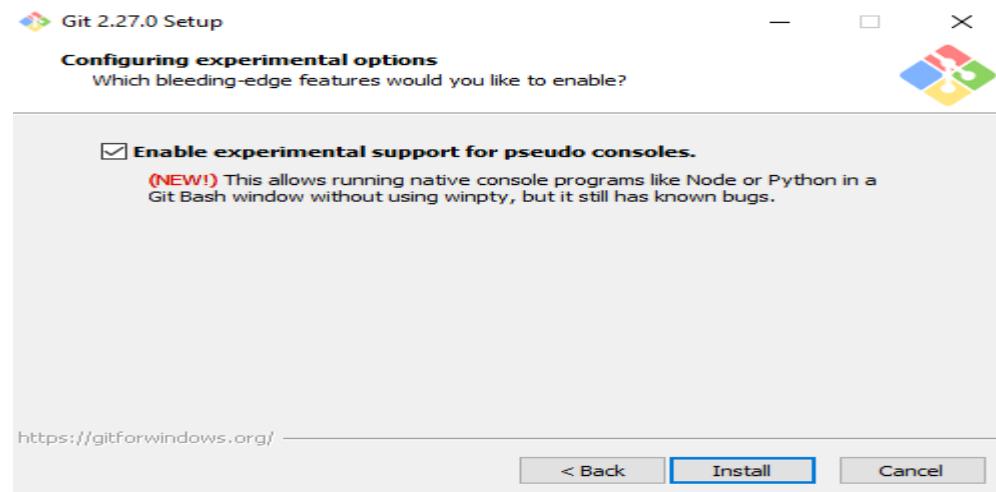
- 8) Click on the first option of OpenSSL Library and then click on next.



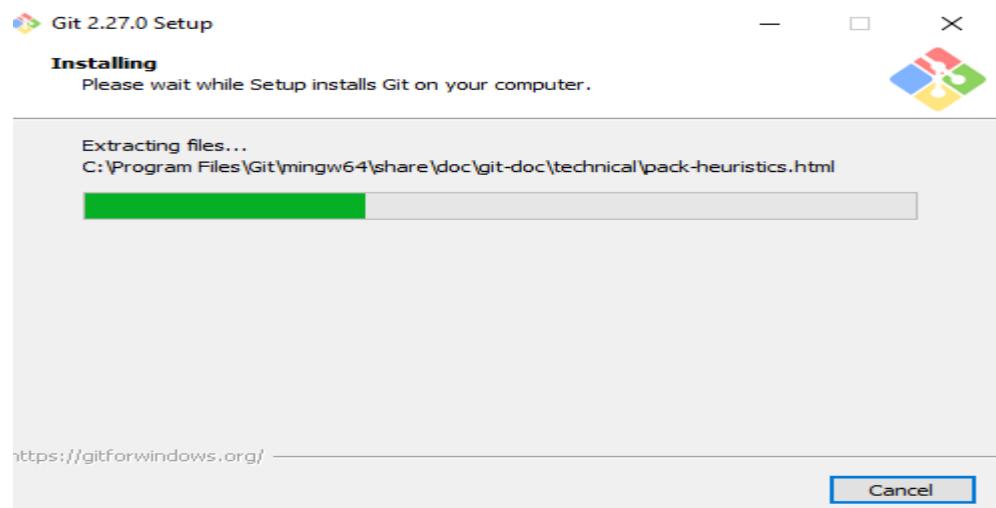
- 9) For the next all pictures you just have to select whatever given and click on next.



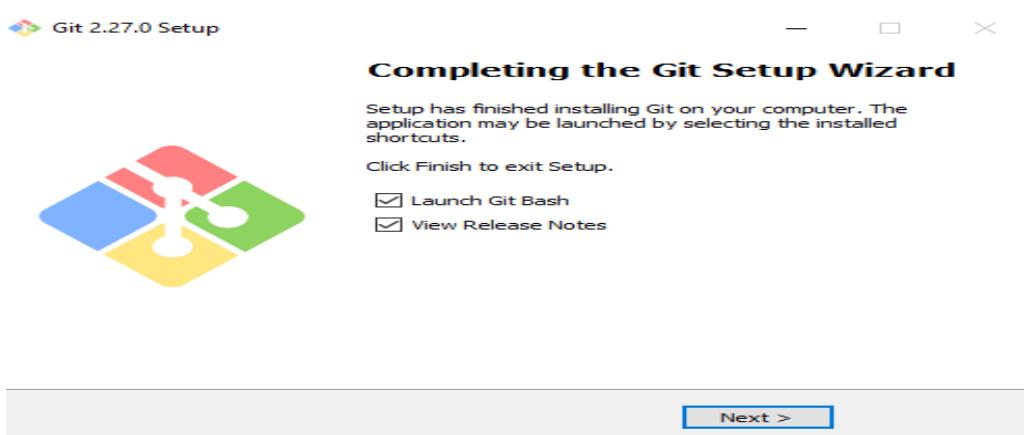
10) Click on install after selecting enable experimental support for pseudo consoles.



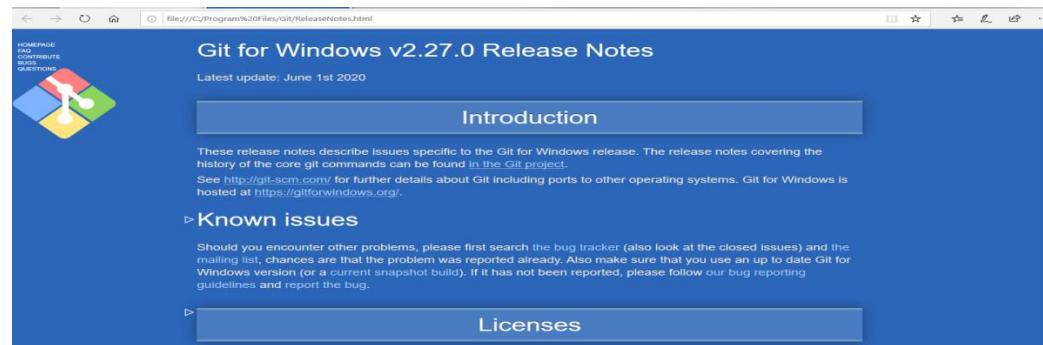
11) Once you click on install it will start installing on your pc/ laptop.



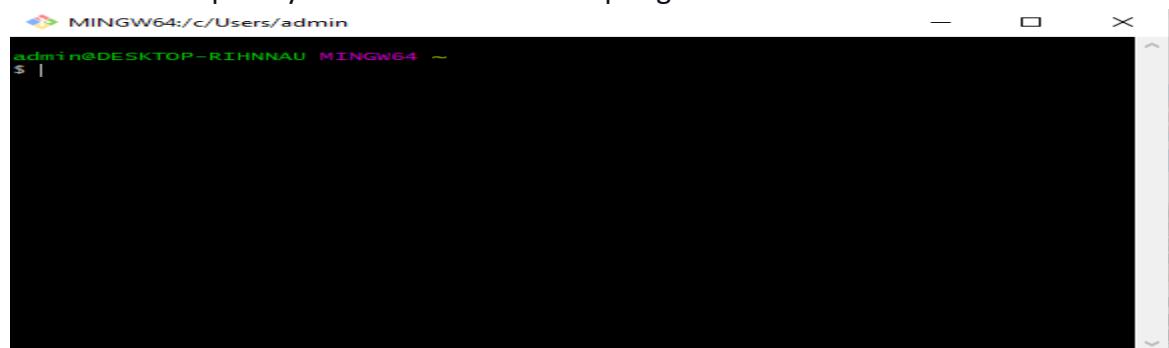
12) When done installing this pop will appear. Select both options and click on next.



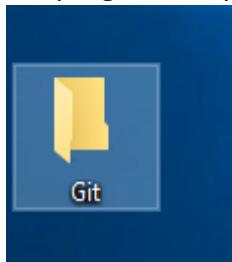
13) It will open Release Notes related to Git and you can go through the details of git.



14) It will parallelly open Git Bash Terminal as well. It is a command prompt like shell Which acts completely like linux terminal accepting commands like ls.

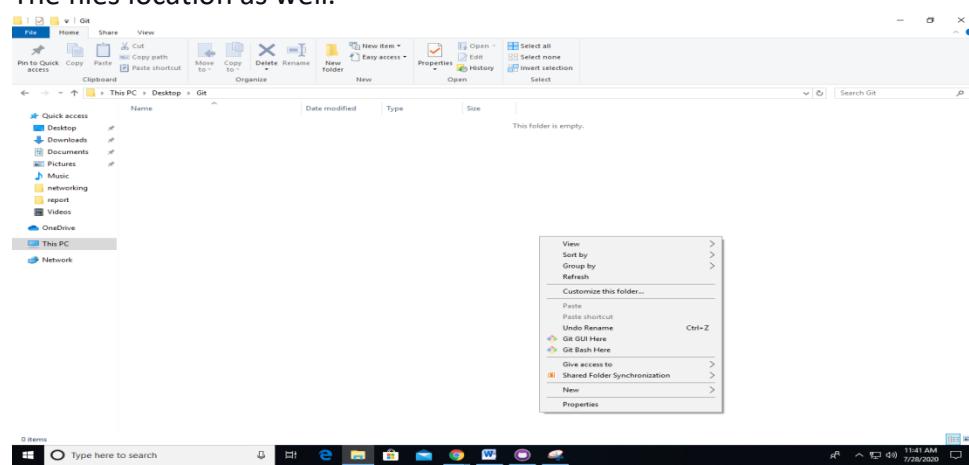


15) Create a folder on the desktop to make it remote repository. Which makes it will be only used
For programs or projects.

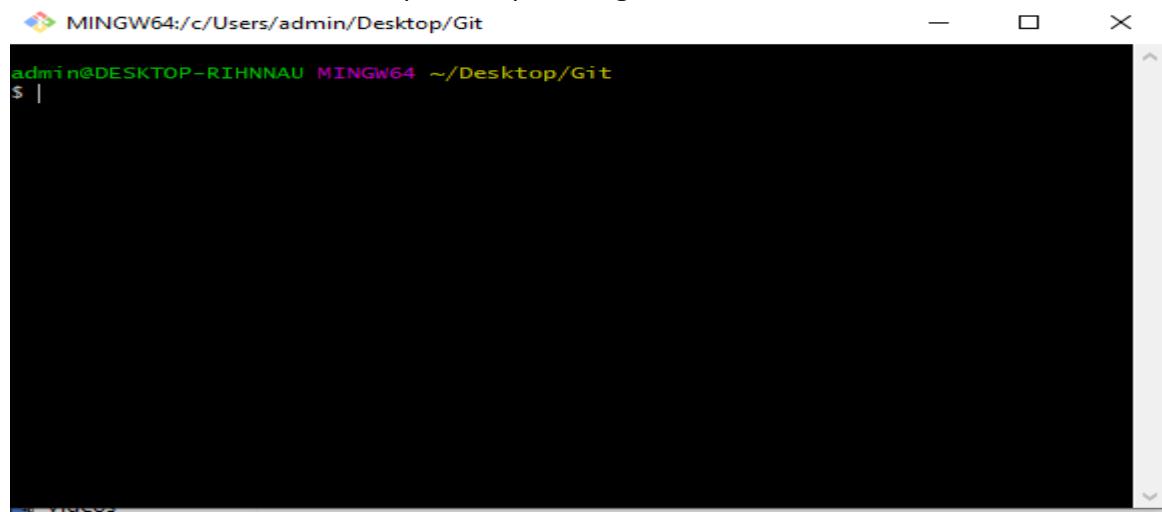


16) Get into the folder you made. Right click and you'd see options like given below. Select GitBash and it will open the GitBash terminal in there or you can use the cd command to go

The files location as well.



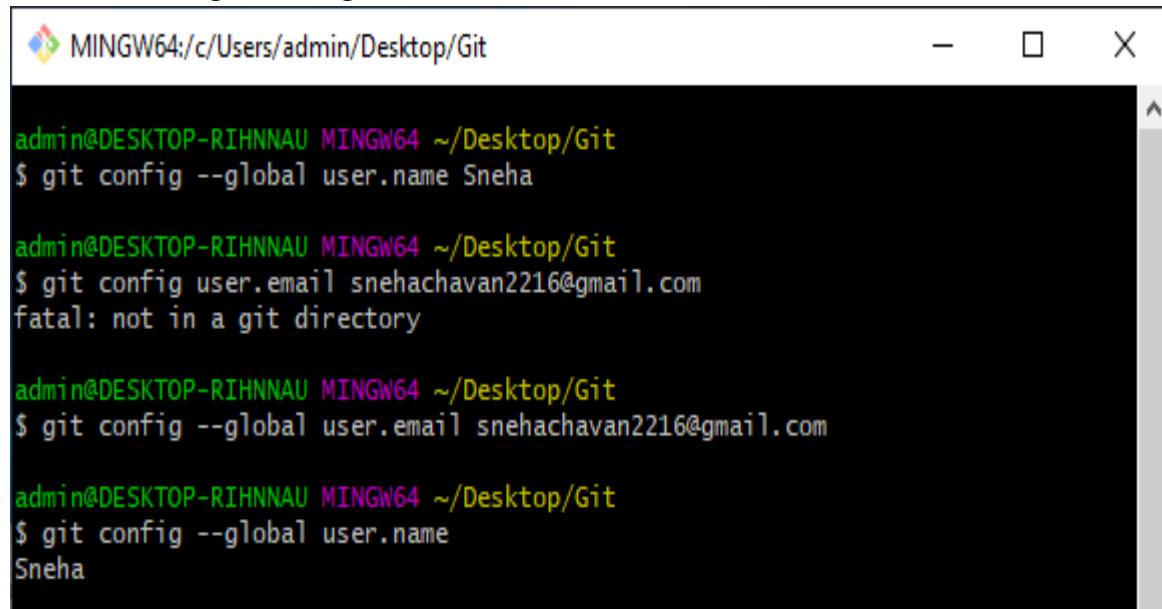
17) The Git Bash Terminal would open will path as git folder as shown below.



A screenshot of a Git Bash terminal window. The title bar says "MINGW64:/c/Users/admin/Desktop/Git". The command line shows the user's name "admin@DESKTOP-RIHNNAU" and the terminal path "MINGW64 ~/Desktop/Git". A single dollar sign (\$) is at the beginning of the line, indicating where the user can type a command. The rest of the window is a black space.

18) Later on, type these commands to set user name and email id which you will register on Github account. Given one wrong command the second one to show you how errors appear.

Even if one thing is missing in the command.



A screenshot of a Git Bash terminal window showing several commands being run:

```
admin@DESKTOP-RIHNNAU MINGW64 ~/Desktop/Git
$ git config --global user.name Sneha

admin@DESKTOP-RIHNNAU MINGW64 ~/Desktop/Git
$ git config user.email snehachavan2216@gmail.com
fatal: not in a git directory

admin@DESKTOP-RIHNNAU MINGW64 ~/Desktop/Git
$ git config --global user.email snehachavan2216@gmail.com

admin@DESKTOP-RIHNNAU MINGW64 ~/Desktop/Git
$ git config --global user.name
Sneha
```

Git commands and their usage:

SETUP configuring user information used across all local repositories

- 1) `gitconfig --global user.name "[firstname.lastname]"`
Set a name that is identifiable for credit when review version history.
- 2) `gitconfig --global user.email "[valid-email]"`
Set an email address that will be associated with each history marker.
- 3) `gitconfig --global color.ui`
Auto set automatic command line coloring for Git for easy reviewing.

STAGE & SNAPSHOT working with snapshots and the Git staging area

- 1) git status
Show modified files in working directory, staged for your next commit.
- 2) git add [file]
Add a file as it looks now to your next commit (stage).
- 3) git reset [file]
Unstage a file while retaining the changes in working directory.
- 4) git diff
diff of what is changed but not staged.
- 5) git diff --staged
diff of what is staged but not yet committed.
- 6) git commit -m “[descriptive message]”
Commit your staged content as a new commit snapshot.

SETUP & INIT configuring user information, initializing and cloning repositories

- 1) git init
Initialize an existing directory as a Git repository.
- 2) git clone [url]
Retrieve an entire repository from a hosted location via URL.

BRANCH & MERGE Isolating work in branches, changing context, and integrating changes

- 1) git branch
List your branches. a * will appear next to the currently active branch.
- 2) git branch [branch-name]
Create a new branch at the current commit.
- 3) git checkout
Switch to another branch and check it out into your working directory.
- 4) git merge [branch]
Merge the specified branch's history into the current one git log show all commits in the current branch's history.

PART-II

GitHub is platform which hosts repositories which people can see on public. Let's see how it is supposed to be installed.

1. Go to the download page by opening web browser: <https://desktop.github.com/> download it



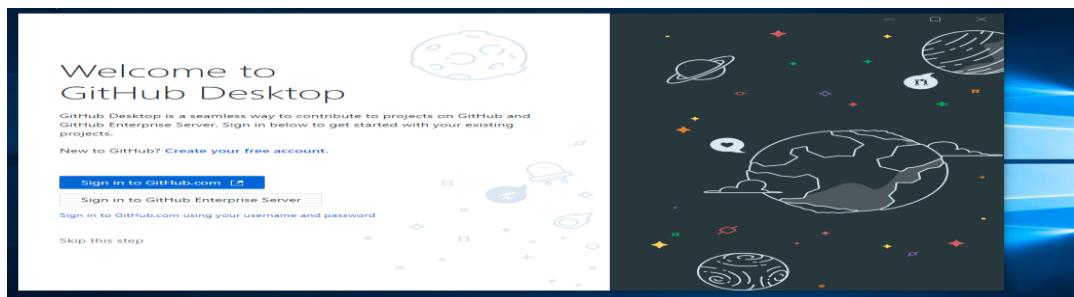
- a. Double click on the downloaded file.



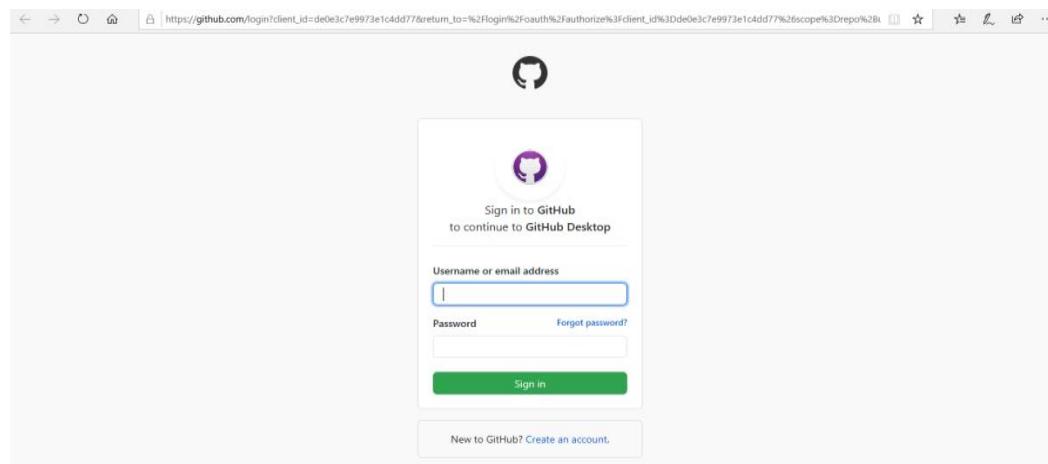
- b. Once you double click the image will appear on your screen.



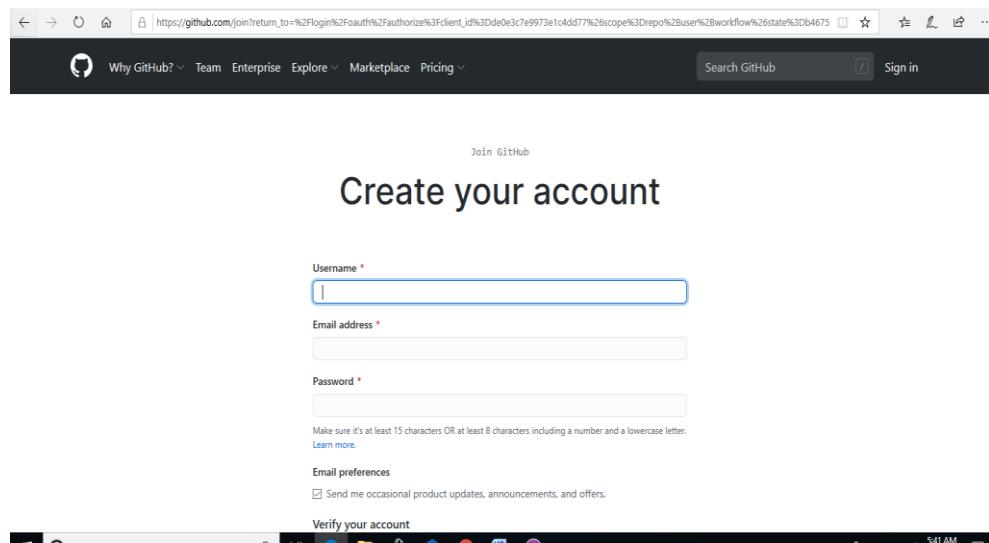
- c. A pop will appear on your screen like the image given below. Click on Sign in to GitHub.com



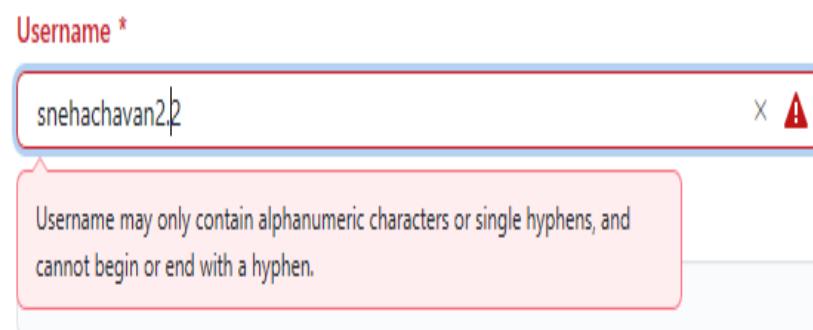
- d. Then registration/signup link will be showcased by the website. If you don't have an account. Click on Create an account.



- e. Start giving the required details which are mentioned.



- f. It will prompt if you have filled anything wrong just like the below image.



- g. Once you fill all the details it will ask you to verify your account:

Join GitHub

Create your account

Username *
snehachavan22

Email address *
snehachavan2216@gmail.com

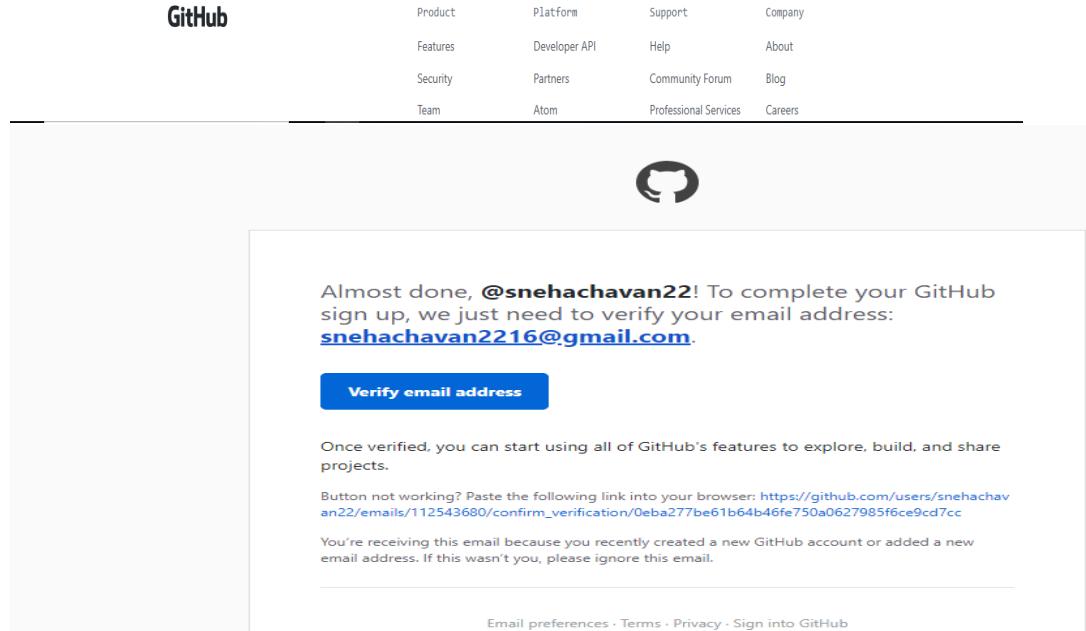
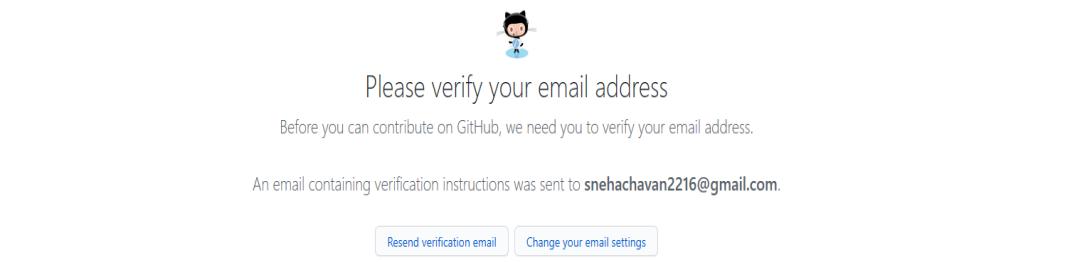
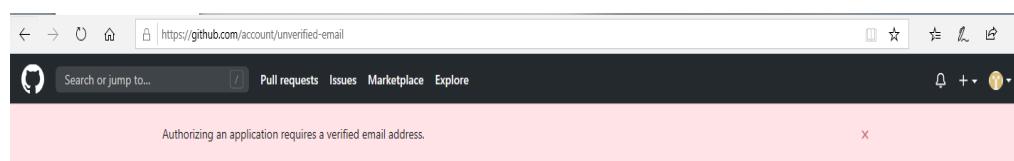
Password *
••••••••••••••••

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.
[Learn more.](#)

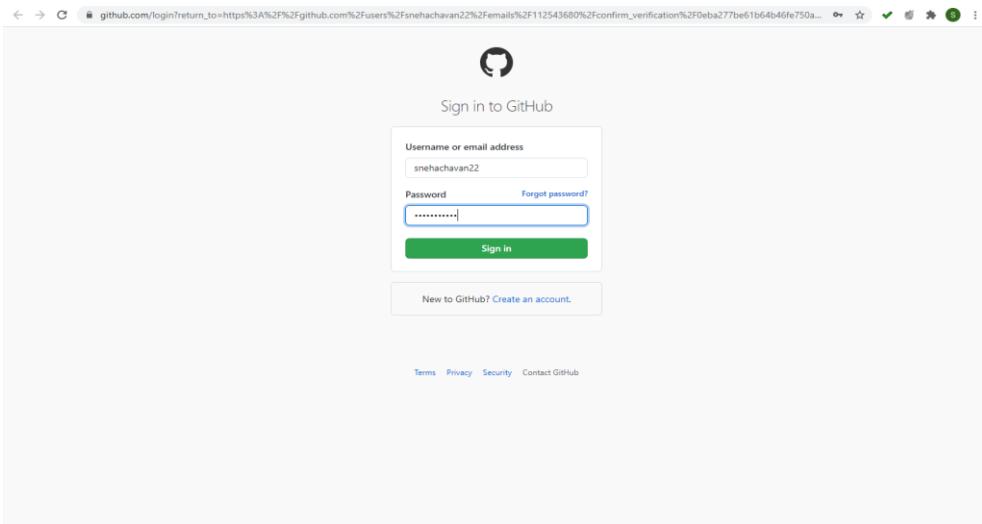
Email preferences
 Send me occasional product updates, announcements, and offers.

Verify your account

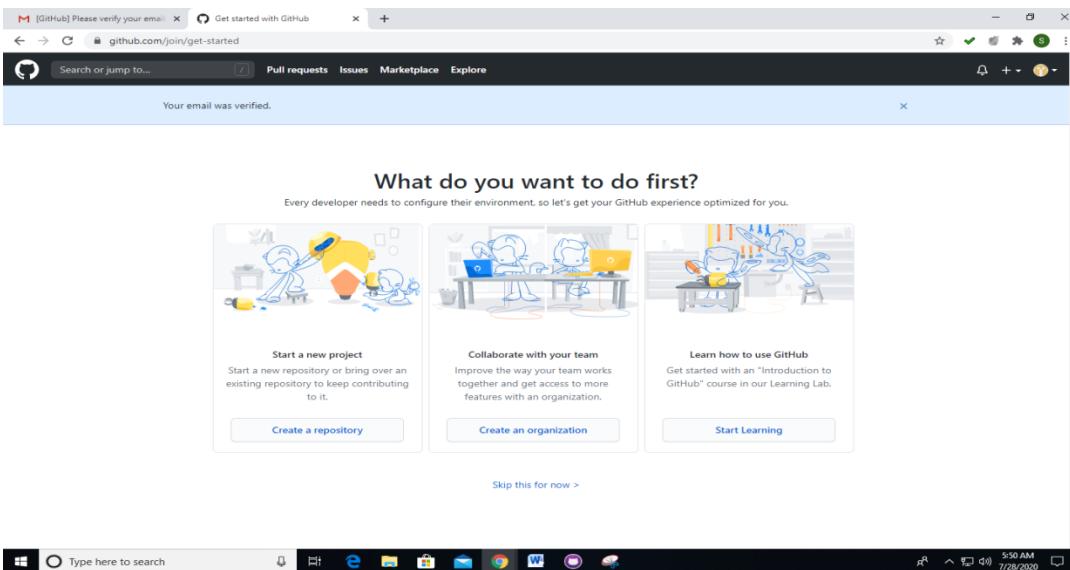
- h. Verification will be sent to your email id which you have given while registration. Go to your email and check once and verify it from there. Click on verify button.



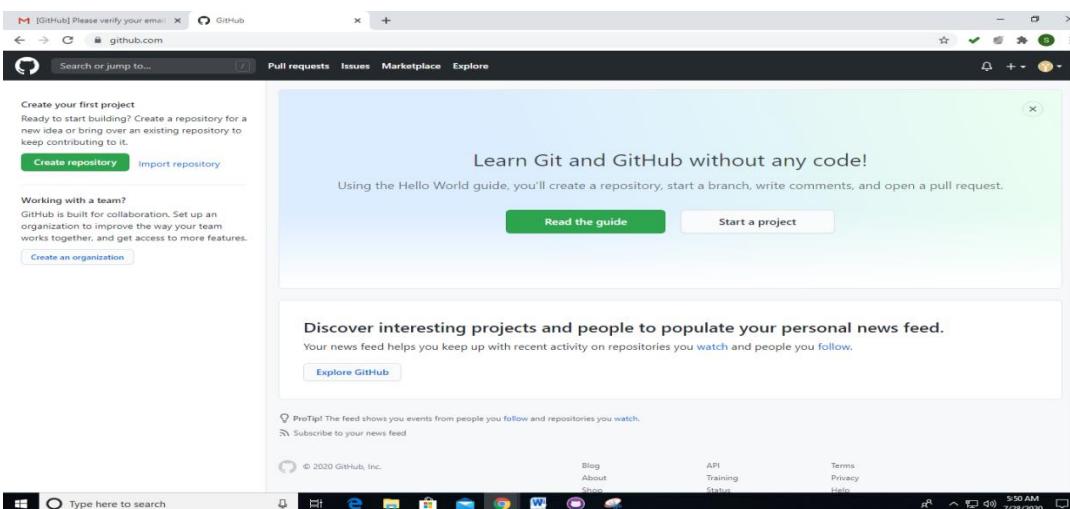
- i. Login later with the your credentials:



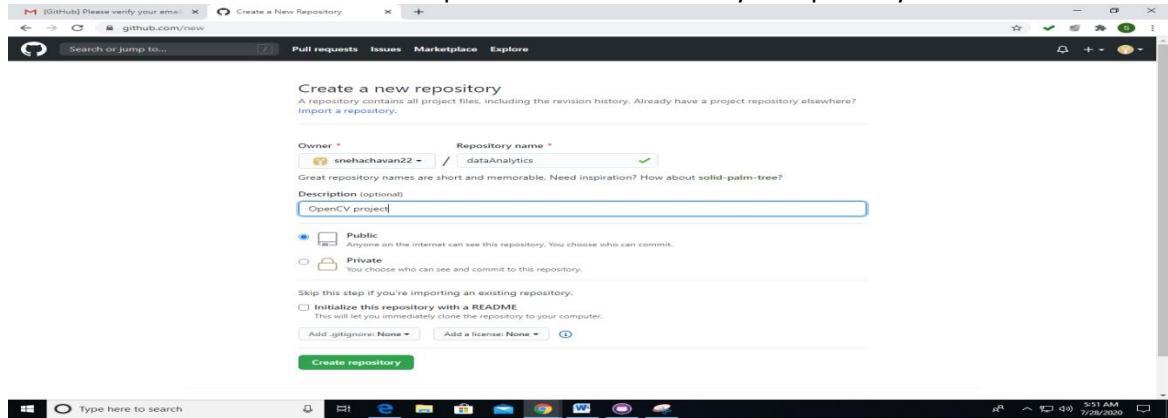
- j. You will land up to this page. There are options which you can explore. Click on Repository.



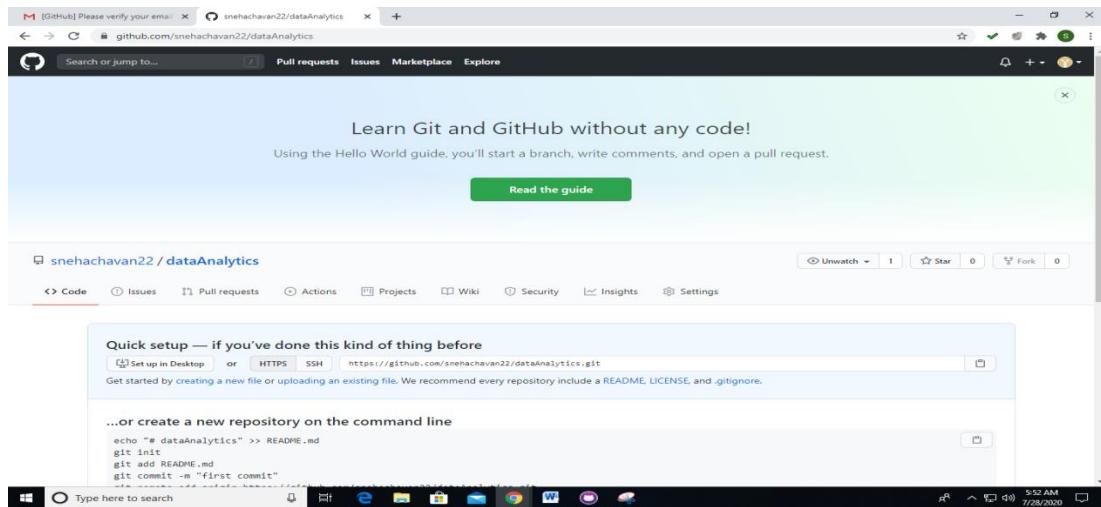
- k. When you click on Create a repository.



- I. Fill the fields and make sure it is public so others can see your repository.

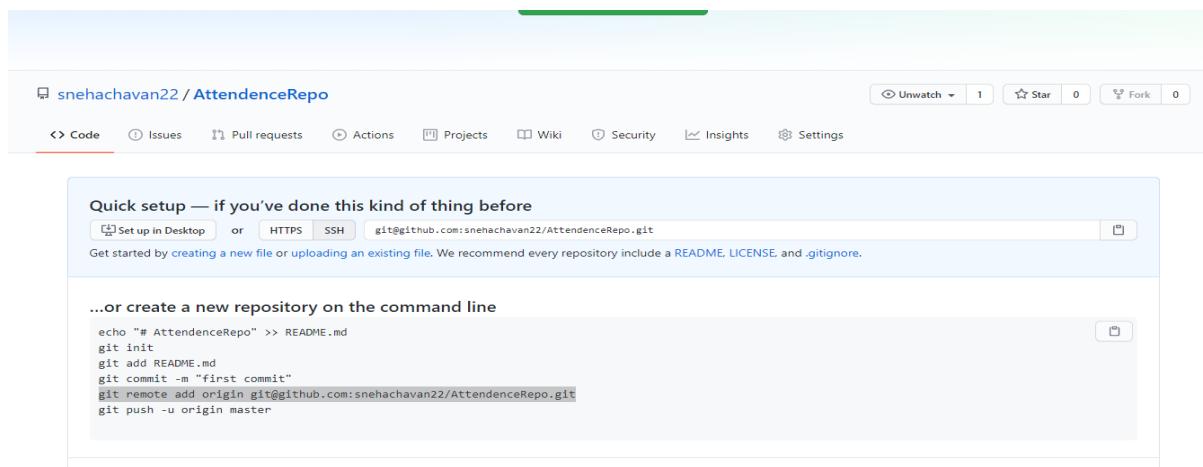


- m. Click on Create Repository. And you'd this.

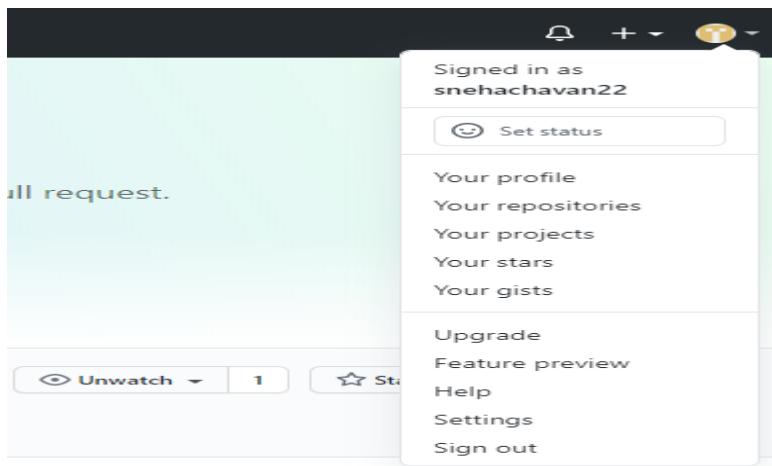


How to collaborate local repository with GitHub to host it:

- When I create a new repository it will take me to this particular page inside which there is a code section. You have to select that command and then paste it on the Git Bash from git folder which we made it earlier. Command: `git remote add origin "<your github url, not mine>"`.



- 2) In this you remote repository is added. Next, step would be to go settings at top right corner and click on settings.



- 3) Once you go on settings you have to go at and select SSH and GPG keys section on left middle section of web page.



- 4) Then in this you will have to select generate ssh keys which is highlighted for you. Click on it.

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

- 5) Then go on to the link which is highlighted below it would open in new tab.

[About SSH](#)

Using the SSH protocol, you can connect and authenticate to remote servers and services. With SSH keys, you can connect to GitHub without supplying your username or password at each visit.

[Checking for existing SSH keys](#)

Before you generate an SSH key, you can check to see if you have any existing SSH keys.

[Generating a new SSH key and adding it to the ssh-agent](#)

After you've checked for existing SSH keys, you can generate a new SSH key to use for authentication, then add it to the ssh-agent.

- 6) Then copy this link and then paste it on your GitBash Terminal.

[Generating a new SSH key](#)

- 1 Open Git Bash.
- 2 Paste the text below, substituting in your GitHub email address.

```
$ ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

This creates a new ssh key, using the provided email as a label.

```
> Generating public/private rsa key pair.
```

Copy the above command and instead of you email type your email id which you have registered it with github

And used it as user.email during git.

- 7) I have overrided it but in your case it wont have any option. When it asks for enter passphrase just keep on hitting

The enter button.

```
91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ ssh-keygen -t rsa -b 4096 -C "snehachavan2216@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/91897/.ssh/id_rsa):
/c/Users/91897/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/91897/.ssh/id_rsa
Your public key has been saved in /c/Users/91897/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vBoA4iiJtLCFUZ/CsBPLsZKSkfTkFP33zPVkKVBYxIc snehachavan2216@gmail.com
The key's randomart image is:
+--[RSA 4096]----+
| *.=+o      *+. |
| o%B. o      o E . |
| %+=oo .     . . . |
| O*.o      o . o + |
| *o .      S + . = |
| . . . + .   |
| .           o |
| .           . |
+----[SHA256]----+
91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ |
```

Key gets generated in here. Now, we want to deploy this ssh key.

- 8) Then run this command to get agent key.

Command: eval \$(ssh-agent -s)

Once you type this command hit enter then you'll see agent id as output.

```
91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ eval $(ssh-agent -s)
Agent pid 1337
```

- 9) Then type this command which is given

```
91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/91897/.ssh/id_rsa (snehachavan2216@gmail.com)

91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ |
```

- 10) Then go back to the GitHub web page. Click on adding your ssh key to ssh-agent which is highlighted.

[Adding your SSH key to the ssh-agent](#)

Before adding a new SSH key to the ssh-agent to manage your keys, you should have checked for existing SSH keys and generated a new SSH key.

If you have [GitHub Desktop](#) installed, you can use it to clone repositories and not deal with SSH keys.

- 1 Ensure the ssh-agent is running. You can use the "Auto-launching the ssh-agent" instructions in "[Working with SSH key passphrases](#)", or start it manually:

```
# start the ssh-agent in the background
$ eval $(ssh-agent -s)
> Agent pid 59566
```

- 11) Which will take you to the section link which you'd have to link.

[Add the SSH key to your GitHub account](#)

- 12) The public key has to be copied.

After adding a new SSH key to your GitHub account, you can reconfigure any local repositories to use SSH. For more information, see "[Switching remote URLs from HTTPS to SSH](#)."

Note: DSA keys (SSH-DSS) are no longer supported. Existing keys will continue to function, but you cannot add new DSA keys to your GitHub account.

- 1 Copy the SSH key to your clipboard.

If your SSH key file has a different name than the example code, modify the filename to match your current setup. When copying your key, don't add any newlines or whitespace.

```
$ clip < ~/.ssh/id_rsa.pub
# Copies the contents of the id_rsa.pub file to your clipboard
```

Tip: If `clip` isn't working, you can locate the hidden `.ssh` folder, open the file in your favorite text editor, and copy it to your clipboard.

- 13) When you copy paste this command with cat. It will give you the following output.

```

91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDjUNddz5m82K7G+IrFF2fve0cy+05TQzZF/wM8UnCv
h+iaJNfES+qv+08iaRJI9ynjnXd9EYcYVbfCT8M2FidRQgxP5UXKWzyX8J7f+L2GECCuYR6XRWrSgj
0xj6JdIkxnWFDYDuQ3Z28B/BTiooXdD3wSZJNklnSe3ky53A8Hj4Y01cFJBAgO1P5rN2MVtPE1J1zds
PyWDX5exAEvPJbjWnddmFjIcZuazkpPC242gjj9IVF8eVZ9hMFJN0bAQzNp/SKNyfw1tsD4tAFLIJDC
8BNp5T+9QX7QchThjMP6Lz8IW/PbMtYpWABVRsvtawaRcCvnc1Je2/Fa28qA4jVG3rUZIVdIJP7FI1K
Sr652hyxDu6GPJ14aSzIWyDY9Q3Qbqueh1cYDbtD5PNHUYCKDbLVwDT04NXwD/uqVJUxgn/d4k4Sh3x1
CVWXEiGgPsHBC30uR8scvtQfwteXtdNjtttsXxuhkxeSSV8di0kG//zB4lpC6H470aiEhUvj+3PgKALR
WOO+1jqzhLwDrP9/g0A7eGvWjeB5044ZidtA8mJdEPxB0M5fYAKmpP6CwOJHlOTVua+gEhtDQBDqRbjM
Yukg/6Tpj9nGS851KosodMzKsB5af/pd27NIUJsrmmmy2707cq/TwCSC7Pj5EzREAE0u6/QPx8YsLFWe
qw== snehachavan2216@gmail.com

91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ |

```

- 14) Then you have to select the entire output copy it and then paste it at new ssh key. Which is marked in red

The screenshot shows the GitHub 'SSH keys' page for the user 'snehachavan22'. The page title is 'SSH keys'. On the right side, there is a green button labeled 'New SSH key' which is highlighted with a red box. The left sidebar has links for 'Personal settings', 'Profile', and 'SSH and GPG keys'.

- 15) Then add the copied ssh-key and give it appropriate name to it.

And then click on the green button named add ssh key.

The screenshot shows the GitHub 'SSH keys / Add new' page. The title is 'SSH keys / Add new'. In the 'Title' field, the text 'sneha's computer' is entered. In the 'Key' field, a long SSH public key is pasted. At the bottom, there is a green button labeled 'Add SSH key' which is highlighted with a red box. The left sidebar lists various GitHub settings sections like 'Profile', 'Account', 'Security log', etc., with 'SSH and GPG keys' selected.

16) Go to your repository and then you'd be able to see this. Then click on ssh which is highlighted.

The screenshot shows a GitHub repository page for 'snehachavan22 / AttendanceRepo'. At the top right, there is a green button labeled 'Read the guide'. Below the header, there are tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Under the 'Code' tab, there is a section titled 'Quick setup — if you've done this kind of thing before' with options for 'Set up in Desktop' or 'HTTPS' (which is selected) or 'SSH'. The SSH URL 'git@github.com:snehachavan22/AttendanceRepo.git' is highlighted with a yellow box. Below this, there is a note about creating a README, LICENSE, and .gitignore files. Another section below shows command-line instructions for creating a new repository: 'echo "# AttendanceRepo" >> README.md', 'git init', and 'git add README.md'. A copy icon is visible next to the command examples.

17) Then copy paste the part which is next to it. With simple command on git bash terminal.

The screenshot shows a GitHub repository page for 'snehachavan22 / AttendanceRepo'. At the top right, there is a green button labeled 'Read the guide'. Below the header, there are tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Under the 'Code' tab, there is a section titled 'Quick setup — if you've done this kind of thing before' with options for 'Set up in Desktop' or 'HTTPS' (which is selected) or 'SSH'. The SSH URL 'git@github.com:snehachavan22/AttendanceRepo.git' is highlighted with a yellow box. Below this, there is a note about creating a README, LICENSE, and .gitignore files. Another section below shows command-line instructions for creating a new repository: 'echo "# AttendanceRepo" >> README.md'. A copy icon is visible next to the command examples.

18) The commands with output is given in the below image.

```
91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ git remote set-url origin git@github.com:snehachavan22/AttendanceRepo.git

91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ git remote
origin

91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ |
```

19) If you type git remote -v you'll see url has been changed.

```
91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (opencv2)
$ git remote -v
origin  git@github.com:snehachavan22/AttendanceRepo.git (fetch)
origin  git@github.com:snehachavan22/AttendanceRepo.git (push)
```

20) Then type this command: to push all the files to your github using git

```
91897@DESKTOP-QQBU45R MINGW64 ~/Desktop/git (master)
$ git push -u origin master
The authenticity of host 'github.com (13.250.177.223)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,13.250.177.223' (RSA) to the list of known hosts.
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 2.95 KiB | 604.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:snehachavan22/dataAnalytics.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

If you check website you'll see all your files using git add, git commit command and in the end git push.

The screenshot shows a GitHub repository page for 'snehachavan22 / dataAnalytics'. The repository has 2 branches and 0 tags. The master branch is selected. There are 1 commit by 'snehabca07' titled "First module". The commit was made 4 hours ago. The commit history also lists 'sneha.html', 'varshitha.html', 'yashraj.html', 'DOD', and 'DOD.pub' files, all added 4 hours ago. The repository has 1 star and 0 forks. It includes sections for About (OpenCV project), Releases (No releases published, Create a new release), and Packages (No packages published, Publish your first package). A button to 'Add a README' is visible at the bottom.

AND WE'RE DONE! For more understanding of such concepts you can refer to the YouTube channel:
<https://www.youtube.com/c/DecodegitSimplified>

LAB Exercise

Set A:

- a) Create a GitHub account.
- b) Download and install git on computer or laptop.
- c) Create your own repository for each subject.
- d) Collaborate Remote Repository with GitHub using GitBash Terminal.
- e) Upload html programs on your repository and share the link with your professor to see.

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: WellDone []

Signature of the Instructor: ----- **Date:** -----

Assignment No 5: Creating List and Tables using HTML Tags

Author: Aparna Tushar Gohad

Allotted Slots: 2

Aim: To study different types of List, creating tables using HTML tags.

Pre-requisite: Basic HTML tags

The student should read following topics before starting exercise.

❖ List:

A List is a record of short piece of information usually written with a single thing on each line and ordered in the way that makes a particular thing easy to find.

Eg:

- To-do list
- Shopping list

HTML offers two types of List:

1. **Numbered List (Ordered List ie ol):** An ordered list is used when sequence of list items is important.
2. **Bulleted List (Unordered List ie ul):** An unordered list is a collection of related items that have no special order or sequence. Tags used to create lists are given in the following table.

Sr. No.	Tag	Description	Attribute	Example
	 	Used to specify list under ol or ul		
	 	The tag is used to specify the ordered list. By default it starts at 1 and always incremented by 1.	type = a/A/i/I Sets the numbering style to: <ol type = "1"> - Default Numerals. <ol type = "I"> - Upper Numerals. <ol type = "i"> - Lower Numerals. <ol type = "A"> - Upper Letters. <ol type = "a"> - Lower Letters. start = "4" Specifies the start value of the first list item in an ordered list. This value is always an integer when the numbering type is let romans. E.g., to start counting items from the letter "c" or the number "iii", use start="3".	 Pen Pencil Scale <ol type="a"> Pen Pencil Scale <ol type="a" start="4"> Pen Pencil Scale
	<	This tag defines uno list of items. Each it	type = disc/square/circle Sp the bullet type.	 Pune

		the list is marked with bullet.	<pre><ul type = "disc"> <ul type = "circle"> <ul type = "square"></pre>	<pre>Mumbai Nagpur <ul type="square"> Pune Mumbai Nagpur </pre>
--	--	---------------------------------	---	--

❖ **Table:**

A table is a two dimensional matrix, consisting of rows and columns. HTML tables are intended for displaying data in columns on a web page. Tables contains information such as text, images, forms, hyperlinks etc. Tags used to create table are given in the following table.

Sr. N	Tag	Description	Attribute
	<code><table> </table></code>	It is used to create a table.	border =number Draws an outline around the table rows and cells of width equal number. By default table have no borders number =0. width =number Defines width of the table. cellspacing =number Sets the amount of cell space between table cells. Default value cellpadding =number Sets the amount of cell space, in number of pixels between the cellborder and its contents. Default is 2 bgcolor="#rrggbb" Sets background color of the table bordercolor="#rrggbb" Sets border color of the table align=left right center Aligns the table. The default alignment is left. frame=void above below hsides lhs rhs vsides box border Tells the browser where to draw borders around the table
	<code><tr> </tr></code>	Creates a table row.	align=left right center Aligns the data in cell. The default alignment is left.
	<code><th> </th></code>	Creates a table heading.	
	<code><td> </td></code>	Data cells are inserted in a row table.	rowspan =number Specifies number of rows a cell should span. colspan =number Specifies number of columns a cell should span align=left right center Aligns the data in cell. The default alignment is left. bgcolor="#rrggbb" Sets background color of the table.

Sample program for HTML table.

```
<!DOCTYPE html>
<html>
<body>

<h2>Basic HTML Table</h2>
<table border=1 width=80% cellpadding=3 cellspacing=3 bgcolor="gray"
bordercolor=black>
<tr>
<th>Item No</th>
<th>Item Name</th>
<th>Price</th>
</tr>
<tr>
<td>1</td>
<td>Pen</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>Pencil</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Scale</td>
<td>30</td>
</tr>

<tr>
<td colspan=2>
</td>
<td>
<b>Total: 140</b>
</td>
</tr>
</table>
</body>
</html>
```

LAB Exercise

SET A

1. Write the HTML code to display the rainbow color names using Ordered List. Modify the code to display each color name with the same color of rainbow using font tag. Apply page background color as black.

[Hint: Red]

2. Write the HTML code which generates the following output.

SYBCA (Science) Course Objectives

- I. Data Structures
 - To understand algorithms and analysis of algorithms
 - To learn static and dynamic data structures
- II. Database Management Systems II
 - To understand advanced SQL features and procedural SQL
 - To study concurrency control and crash recovery techniques
- Computer Networks
 - To learn role of protocols at various layers in the protocol stacks
 - To study different techniques for framing, error control, flow control and routing

3. Write the HTML code to display list of any three car companies. For each company display list car models. Display car model as hyperlink. On click of car model it should display an image of that car in another window. Use ordered or unordered list of your choice.

[Hint: Create separate HTML page for each car model which will display an image.

Use Swift]

SET B

1. Write the HTML code to display day wise SYBCA time table in tabular format.
2. Write the HTML code which generates the following output. Add more colors in following table.

HTML Colors

Color	Name	hexadecimal	RGB value
	Salmon	FA8072	250-128-114
	Gold	FFD700	250-215-0

3. Write the HTML code which generates the following output.

Company wise Profit

Company Name	Year	Profit (In Crore)
Infosys	2018	6520
	2019	7250
	2020	7962
Wipro	2018	1803
	2019	1953
	2020	2529
Cognizant	2018	5420
	2019	5863
	2020	6293

SET C

1. Write the HTML code to display names of html text formatting tags and output in tabular format. Add more html text formatting tags in following table.

Tag name	Output
b	Bold
I	<i>Italic</i>
U	<u>Underline</u>

2. Write the HTML code to display Product and its subtypes in tabular format. Add more products of your choice in following table.

Sr. No	Product Name	Product subtypes
1	Pulses	<ul style="list-style-type: none"> • Toor daal • Moong daal • Udad daal
2	Everest Masala	<ul style="list-style-type: none"> • Sambhar masala • Pavbhaji masala • Kichen king masala

Assignment Evaluation

- | | | |
|--------------------------|-------------------|----------------------|
| 0: Not Done [] | 1: Incomplete [] | 2: Late Complete [] |
| 3: Needs Improvement [] | 4: Complete [] | 5: Well done [] |

Signature of the Instructor: ----- Date: -----

Assignment No 6: Creating Frames and Forms using HTML Tags

Author: Aparna Tushar Gohad

Allotted Slots: 2

Aim: To study frames for diving html page and designing input form using HTML tags.

Pre-requisite: Basic HTML tags

The student should read following topics before starting exercise.

❖ **Frames:** Using frames, one can divide the screen into multiple scrolling sections, each of which can display a different web page into it. It allows multiple HTML documents to be seen concurrently.

Tag	Description	Attributes	Example
<frameset> </frameset>	Splits browser screen frames.	rows =number Helps in dividing the browser screen into horizontal sections or frames. cols =number Divides the screen into vertical sections or frames. The number written in the rows and cols attribute can be given as absolute numbers percentage value or an asterisk can be used indicate the remaining space. border = number This attribute specifies the width of the border for each frame in pixels. For example, border = 10. A value of zero means no border. frameborder = number This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example frameborder = "0" specifies no border. framespacing = number This attribute specifies the amount of space between frames in a frameset. This can take integer value. For example framespacing = 10 means there should be 10 pixels spacing between each frames.	<frameset rows="20%, 30%">
<frame> </frame>	Used to define a single frame in a <frameset>	name =text Assigns a name to the frame src =url Specifies the location of the initial HTML document to be displayed by the frame. noresize By default, frame can be resized by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize = "noresize". scrolling This attribute controls the appearance of the scrollbars that appear on the frame. This tag	<html> <frameset rows = "50% 50%" border=10> <frameset cols = "50% 50%"> <frame src = "success.html" name = "frm1" noresize="noresize"> <frame src = "welcome.html"> </frameset> <frame src = "failure.html" noresize="noresize"> </frameset> </html>

		values either "yes", "no" or "auto". For example scrolling = "no" means it should not have scroll bars.	Note: Don't specify border=1 while using frameset tag
<iframe> </iframe>	This is used to define inline frame with HTML tag <iframe>. The <iframe> tag is not somehow related to <frameset> tag, instead can appear anywhere in your document. The <iframe> tag defines rectangular region within the document in which browser can display a separate document, including scrollbars and borders. An inline frame is used to embed another document within the current HTML document.	src=url This attribute is used to give the file name that should be loaded in the frame. name=text Assigns a name to the iframe height=number Specifies the height of iframe width=number Specifies the width of iframe	<html> <body> <iframe src="demo.htm" height=300 width=300></iframe> </body> </html>

Frame's Name and target attributes

The main use of frame is to place navigation bars in one frame and then load main pages into a separate frame.

See the following code:

test.html

```

<!DOCTYPE html>
<html>
<head>
<title>HTML Target Frames</title>
</head>
<frameset cols = "20%, *">
<frame src = "menu.html" name = "menu_page" />
<frame src = "main.html" name = "main_page" />
</frameset>
</html>

```

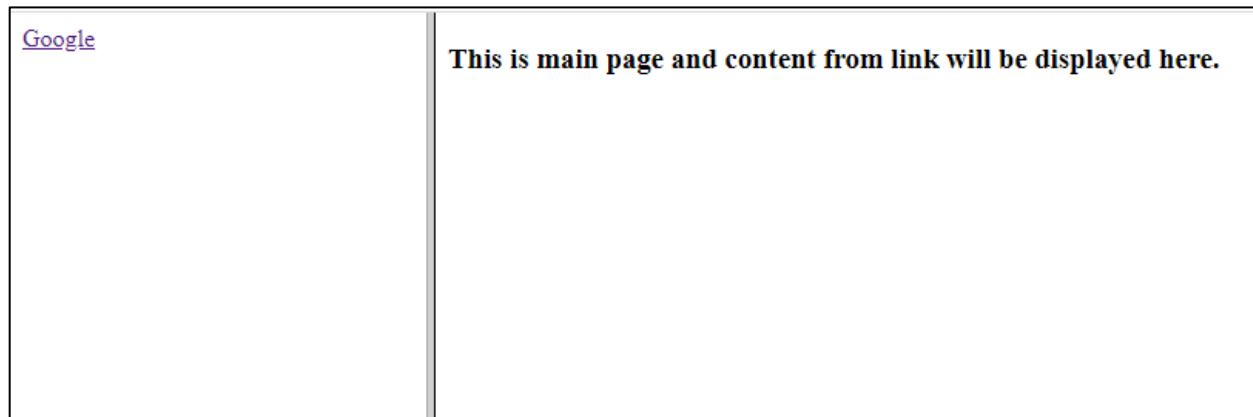
menu.html

```
<!DOCTYPE html>
<html>
<body>
<a href = "http://www.google.com" target = "main_page">Google</a>
</body>
</html>
```

main.html

```
<!DOCTYPE html>
<html>
<body>
<h3>This is main page and content from link will be displayed here.</h3>
</body>
</html>
```

To see the result open **test.html** in browser. It will display output as follows.



The target attribute can take one of following values:

Sr No	target value	Description
1	<u>_self</u>	Loads the page into the current frame.
2	<u>_blank</u>	Loads a page into a new browser window. Opening a new window.
3	<u>_parent</u>	Loads the page into the parent window, which in the case of a single frameset is the main browser window.
4	<u>_top</u>	Loads the page into the browser window, replacing any current frames.
5	<u>_targetframe</u>	Loads the page into a named targetframe.

❖ **Forms:** HTML provides better & more extensive support for collecting user inputs through forms.

A form can be placed anywhere inside the body of an HTML document. You can have more than one form in the document.

Sr	Tag	Description	Attributes	Example
1	<code><form> </form></code>	Creates a form	action=URL Gives the URL the application that is to & processes the forms data method=post/get Specifies which method form transmits data to server	<pre><!DOCTYPE html> <html> <body> <form method="post" action="next.html">

 <input type="submit" value="Submit"> </form> </body> </html></pre>
2	<code><input> </input></code>	This is used to input control of html form	name=text Used to name the maxlength=number Used to specify number of character allowed in field size=number The width of the input control in pixel. type="(checkbox/radio/reset/submit/image)" value to be submitted with the form (for a checkbox or radio button) or label for Reset or Submit buttons) src="source file for an image" checked It indicates that checkbox/radio button is checked. align="texttop/absmiddle/baseline/bottom"	<pre><input type="text" name="nm" width=20> <input type="radio" name="gender" value="male" checked>Male <input type="radio" name="gender" value="Female">Female <input type="checkbox" name="chess" value="chess"> Chess <input type="checkbox" name="Poker" value="Poker"> Poker</pre>
3	<code><select> </select></code>	Defines and displays a set optional list from which the user can select one or more items.	name=text Assigns a name to the control size=number Defines the number of items in the drop-down list multiple=multiple When set, it specifies that multiple items can be selected at a time	<pre><select name="item" size=1> </select></pre>
4	<code><option> </option></code>	Indicates a particular item within a select widget	selected=default selected value="data" Submitted the data, if this option is selected	<pre><select name="item" size=1> <option value="Chair" selected>Chair</option> <option value="Table">Table</option> <option value="Blackboard">Blackboard</option></pre>

				Blackboard</option></select>
5	<textarea> </textarea>	It defines multiline text It is used to userinputs comments reviews.	name=text assigns a name to the control rows=number Specifies the visible num lines in a text area cols=number Specifies the visible wid text area	<textarea name="comment" rows=10 cols=40></textarea>

➤ New input types in HTML5

HTML5 introduces 15 new input types. When viewed in a browser that doesn't support them, these input types fall back to text input.

Sr	Input types	Description	Attributes	Example
1	label	Define a Label for <input> element		<form> <label for> enter your name</label> <input type="text"> </form>
2	legend	Define a c for field element		<form> <legend>create new account </legend> </form>
3	number	Used representing numerical in	value: number The initial value. If o the field is initially bla the internal value is consistent across brows step: number How much to change the when you click on the down arrows of the c The default is 1 min, max: number The smallest and largest that can be selected the up/down arrows	<input type="number" min="0" max="20" step="2" value="10" name="weight"/>
4	range	For numerical input, but unlike number, the actual is not important.	value: number The initial value. The de halfway between the mi the max. step: number How much to change the when you click on the down arrows of the c The default is 1. min, max: The smalle	<input type="range" name="days"/>

			largest values that can be selected. The default for min is 0, and the default for max is 100	
5	date	For entering a date with no time zone	<p>value: date The initial value. The format is "yyyy-mm-dd".</p> <p>step: number The step size in days. The default is 1.</p> <p>min, max: number The smallest and largest dates that can be selected, formatted as date strings of the form "yyyy-mm-dd".</p>	<input type="date" name="bdate"/>
6	time	For entering a time value with hour, minute, seconds, and fractional seconds, but no time zone		<input type="time"/>
7	datetime-local	For entering a date and time with no time zone.		<input type="datetime-local"/>
8	color	For choosing color through a color well control	<p>value: number (in RGB, hex triplet) The initial value. The intention is that if a browser pops up a color chooser, the initial selection will match the current value.</p>	<input type="color"/>
9	email	Define a field for an e-mail address (validates automatically when submitted)	<p>value: emailid The initial value, as an absolute URL.</p> <p>Multiple: Allows multiple addresses</p>	Only one email address <input type="email" name="email"/> Allows multiple email addresses Separate each email address by comma <input type="email" id="email" name="emails" multiple>
10	URL	For entering a single URL.	value: url The initial value, as an absolute URL	<input type="url" name="weburl"/>

11	tel	For entering a telephone number.	value: The initial value as a number	<input type="tel" name="mobno">
12	placeholder	Gives the user a hint about what sort of data they should enter.	placeholder: A small hint. This differs from the “value” attribute in several ways. First, it will usually be rendered differently (e.g. gray). Second, it automatically disappears when you click in the textfield. Value: The initial value. If you specify both placeholder and value, the value is considered, and the placeholder is ignored.	<input type="text" placeholder="Firstname" name="fname"/>
13	autofocus	Focuses the input on the element when the page is loaded	value: The initial value TRUE/FALSE	<input id=" last name" type="text" autofocus="true">
14	autocomplete	For specifying that a field should not autocomplete or be pre-filled by the browser based on a user's past entries	value: The initial value ON/OFF	<input type="password" name="Password confirmation" autocomplete="off">
15	List/datalis	Represents a set of option elements that can be used in combination with the new list attribute for input to make dropdown menus.	list: The id of a separate “datalist” element that defines a list of choices for the user to choose among. The option element (inside “datalist”) label: Extra information that may be displayed in the autocomplete list. value: The value that should be inserted into the textfield when the entry is selected	<input type="text (or other)" list="some-id" name="some-name"/> <datalist id="email-choices"> <option label="Display Val 1"> value="Insert Val 1"> <option label="Display Val 2"> value="Insert Val 2"> <option label="Display Val 3"> value="Insert Val 3"> ...</datalist>

Type the sample Form code and view the content through a browser.

Sample Program:

```
<!DOCTYPE html>
<html>
<body>
<form action="" method="post">
<h3>Quick Contact</h3>
<h4>Contact us today, and get reply within 24 hours!</h4>
Name : <input type="text"></input><br><br>
Email id : <input type="text"></input><br><br>
Type your message : <textarea></textarea><br><br>
<input type="submit">Submit</input>
</form>
</body>
</html>
```

LAB Exercise:

SET A

1. Write the HTML code which generates the following output.

This is a header.	
Look in the box at the right for some information.	Here is some information.
This is a footer.	

2. Write the HTML code to divide the frame into different sections as shown below and add appropriate html files to each frame.

First Frame : Name and Address		
Second Frame	Third Frame	
Bulleted list of qualifications		Links to Favourite sites
Fourth Frame	Fifth Frame	Sixth Frame
Scrolling Message	Blinking reminders	image

3. Write the HTML code to divide page vertically in two sections. On left side display list of any three car companies. For each company display list car models. Use ordered or unordered list of your choice. Display car model as hyperlink. On click of car model it should display an image of that car in right sidewindow.

[Hint: Use target attribute of anchor tag]

SET B

1. Write the HTML code which generates the following output.

Feedback Form

Student Name

Student E-Mail

How do you know this Institute

How do you rate the faculty Poor Good Very Good Excellent

Suggestions for the betterment of faculty and institute

Nice to be here. If given more days for training, it is good.

2. Write the HTML code which generates the following output.

Enter Your Name	<input type="text"/>
Enter Your Password	<input type="password"/>
Which of the following Operating System have you used?	
<input checked="" type="checkbox"/> LINUX <input checked="" type="checkbox"/> Windows XP <input type="checkbox"/> Macintosh 8.0	
Which Operating System do you like the best?	
<input type="radio"/> LINUX <input checked="" type="radio"/> Windows XP <input type="radio"/> Macintosh 8.0	
You have Completed the Form .	<input type="button" value="Submit"/>

SET C

- Create the HTML page which gives details of your college, containing College Heading. Add nested list of courses offered by college for various streams. Add your college photograph and message. Save this page as “College.html”.
- Design the HTML form to take the information of a student registering for the course such as name, address, gender, date of birth, birthplace(to be selected from a list of city/country) ,telephone number, email ,course (to be selected from a list of courses) etc. One should provide button to submit as well as Reset the form contents.Save this page as “Register.html” and embed the above page “COLLEGE.HTML” in this page. (HINT: Use inline frame.)

Assignment Evaluation

0: Not Done [] 1: Incomplete [] 2: Late Complete []

3: Needs Improvement [] 4: Complete [] 5: Well done []

Signature of the Instructor: ----- **Date:** -----

Assignment No7 : Styling HTML pages using CSS

Author: Aparna Tushar Gohad

Allotted Slots: 3

Aim: To study designing of HTML in different ways using CSS.

Pre-requisite: Basic HTML tags

The student should read following topics before starting exercise.

CSS stands for Cascading Style Sheets. CSS is a language that describes style of an HTML document. It describes how the elements of HTML should display. CSS offers more options to provide layout to the HTML.

Advantages of CSS:

1. Control layout of many documents from one single style sheet.
2. More precise control of layout.
3. Apply different layout to different media-types (screen, print, etc.).
4. Numerous advanced and sophisticated techniques.

There are three ways one can apply CSS to an HTML document.

1. In-Line Method – By writing text next to it. (The attribute style).
2. Internal Method – By specifying tag at the top of the page. (The tag style).
3. External Method – By writing a separate CSS file. (Link to a style sheet).

Method 1:

In-Line Method – By writing text next to it. (The attribute style).

To apply CSS to HTML is by using the HTML attribute style. Here is the example of applying red background to HTML page and applying CSS to h1 tag:

Test.html

```
<html>
    <head>
        <title>Example</title>
    </head>
    <body style="background-color: #FF0000;">
        <h1 style="color:blue;margin-left:30px;">This is a heading</h1>
        <p>This is a red page</p>
    </body>
</html>
```

Method 2:

Internal Method – By specifying tag at the top of the page. (The tag style).

Another way is to include the CSS codes using the HTML tag <style>. For example:

Test.html

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            body
            {
                background-color: #FF0000;
            }
            h1
            {
                color:blue;
                margin-left:30px;
            }
        </style>
    </head>
    <body>
        <h1>Internal CSS example!</h1>
    </body>
</html>
```

Method 3:

External Method – By writing a separate CSS file. (Link to a style sheet).

Style sheets are separate files full of CSS instructions (with the file extension .css). When any web page includes an external style sheet, its look and feel will be controlled by this CSS file (unless you decide to override a style using one of these above two types). This is how you change a whole website at once. And that's perfect if you want to keep up with the latest fashion in web pages without rewriting every page!

In external CSS <link> tag defines the relationship between a document and an external resource.

For Example:

Test.html

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" type="text/css"
        href="mystyle.css">
    </head>
    <body>
        <h1>
            ExternalCSS example!
        </h1>
    </body>
</html>
```

mystyle.css

```
body
{
    background-color: #FF0000;
}
h1
{
    color: blue;
    margin-left: 30px;
}
```

Following table shows the list of properties used in CSS stylesheet:

Sr. No	Tag/Property	Property with Values
	Colors and Background	1) color: colorName 2) background-color: colorName 3) background-image: url(path/image) 4) background-repeat: repeat,repeat-x,repeat-y
	Font	1) font-family: fontName 2) font-style: italic,oblique 3) font-size: pixels/percentage 4) font-weight: bold,bolder,lighter(100-900)
	Text	1) text-decoration: underline,overline,line-through, blink 2) text-align: left,right,center,justify 3) text-transform: capitalize,uppercase,lowercase 4) text-indent: number 5) vertical-align: sub,super,top,middle,bottom
	Margin	1) margin-top: 100px 2) margin-bottom: 100px 3) margin-left: 100px 4) margin-right: 100px 5) margin: 100px 40px 10px 70px
	Border	1) border-style: solid,double,groove,inset,outset,ridge 2) border-color: colorName 3) border-width: number 4) border-top-width: number 5) border-bottom-width: number 6) border-left-width: number 7) border-right-width: number 8) border-top: width style color 9) border-bottom: width style color 10) border-left: width style color 11) border-right: width style color 12) border: width style color
	List	1) list-style: disc,circle,square,decimal,lower-roman, upper-roman,lower-alpha, upper-alpha

Applying CSS using id or class

Using id: The id attribute specifies a unique id for an HTML element. The value of the id attribute must be unique within the HTML document. The id attribute is used to point to a specific style declaration in a style sheet. To define CSS use # character followed by id name and then write CSS properties in {} curly braces.

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            #myHeader
            {
                background-color: lightblue;
                color: black;
                padding: 40px;
                text-align: center;
            }
        </style>
    </head>
    <body>
        <h1 id="myHeader">My Header</h1>
    </body>
</html>
```

Using class: A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page. To define CSS use dot (.) character followed by css class name and then write CSS properties in {} curly braces. The css class can be used multiple times.

```

<!DOCTYPE html>
<html>
    <head>
        <style>
            /* Style all elements with the class name "city" */
            .city
            {
                background-color: red;
                color: white;
                padding: 10px;
            }
        </style>
    </head>
    <body>

        <!-- Multiple elements with same class -->
        <h2 class="city">Maharashtra</h2>
        <h2 class="city">Nagpur</h2>
        <h2 class="city">Bangalore</h2>
    </body>
</html>

```

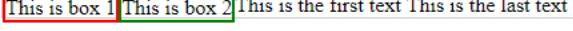
CSS float property

The CSS float property can make HTML elements float to the left or right inside their parent element. Content inside the same parent element will move up and wrap around the floating element. For example:

Code:	Output:
<pre> <!DOCTYPE html> <html> <body> <div style="border:1px solid #cccccc;"> This is the first text <div style="float: left; border: 2px red;"> This is box 1 </div> <div style="border: 2px solid green;"> This is box 2 </div> This is the last text </div> </body> </html> </pre>	

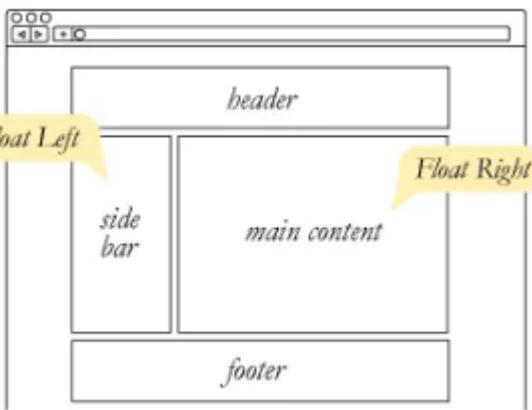
In above example the HTML code contains a div element which has a text, two div elements and another text inside its body.

Now, let us make the first nested and second nested div element float left using the float CSS property.

Code:	Output:
<pre data-bbox="213 318 804 937"><!DOCTYPE html> <html> <body> <div style="border:1px solid #cccccc;"> This is the first text <div style="float: left; border: 2px red;"> This is box 1 </div> <div style="float: left; border: 2px green;"> This is box 2 </div> This is the last text </div> </body> </html></pre>	

Here both the first and second nested div element is floating to the left inside their parent element. The text wraps nicely around the two floating elements.

Possible values of float

<p>none: the element does not float. This is the value.</p> <p>left: floats the element to the left of its container.</p> <p>right: floats the element to the right of its container.</p> <p>inherit: the element inherits the float direction from its parent.</p>	
---	--

LAB Exercise

SET A

1. Write the HTML code for generating the following layout as shown below.

```
<!DOCTYPE html>
<html>
<body>
<div id="container" style="width:500px;">
<div id="header" style="background-color:#FFA500;">
<h1 style="margin-bottom:0;">Main Title of Web Page</h1></div>
<div id="menu" style="background-color:#FFD700;height:200px;width:90px;float:left; border-right: 1px solid black ">
<b>Menu</b><br>
HTML<br>
CSS<br>
JavaScript</div>
<div id="content" style="background-color:#EEEEEE;height:200px;width:390px;float:left; margin-left:10px">
Content goes here</div>
<div id="footer" style="background-color:#FFA500;clear:both;text-align:center;">
This is footer part </div>
</div>
</body>
</html>
```

2. Write the HTML code to generate same output as above by using External CSS.

SET B

1. Write the HTML code which generates the following output.



2. Write the HTML code which generates the following output.

Pay Now

Credit or Debit Card

Pay with Net Banking

PayPal

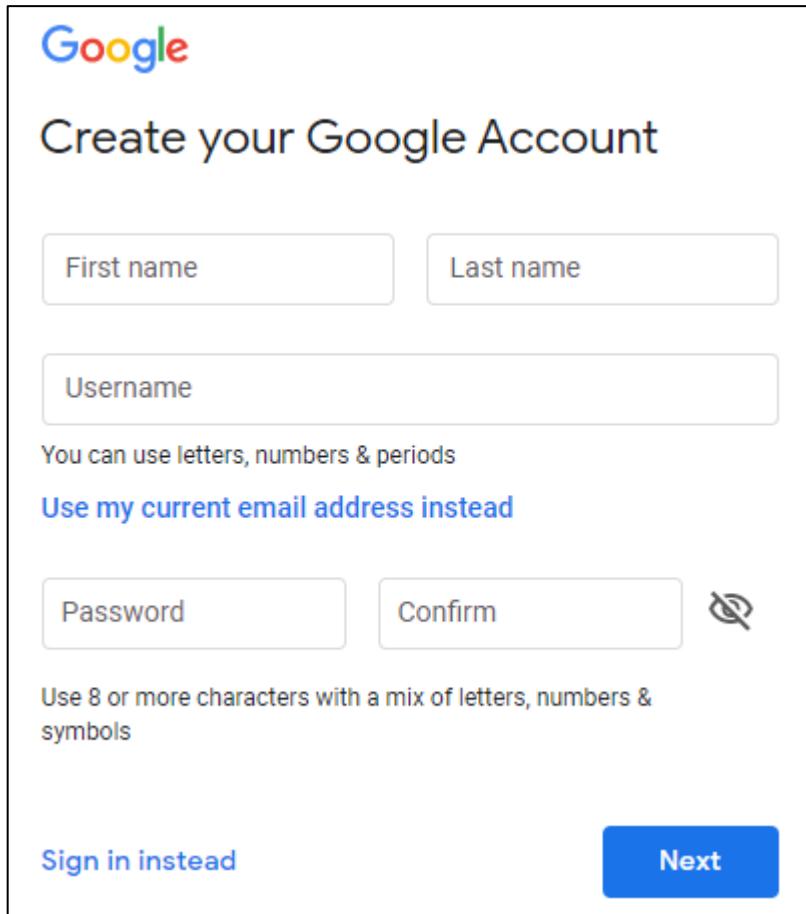
Pay by Mobile Using Local Methods

Name on Card		
Card Number		
MM MM	YYYY YYYY	Security Code

By completing your purchase you agree to these [Terms of Service](#).

SET C

1. Write the HTML code which generates the following output



The image shows a screenshot of the Google Account creation form. At the top left is the Google logo. Below it is the heading "Create your Google Account". There are two input fields: "First name" and "Last name". Below these is a larger input field for "Username", with a note below it stating "You can use letters, numbers & periods". A link "Use my current email address instead" is present. Below the username field are two more input fields: "Password" and "Confirm". To the right of the "Confirm" field is a small icon of a lock with a slash through it. A note below the password fields says "Use 8 or more characters with a mix of letters, numbers & symbols". At the bottom left is a link "Sign in instead", and at the bottom right is a large blue "Next" button.

Assignment Evaluation

0: Not Done [] 1: Incomplete [] 2: Late Complete []

3: Needs Improvement [] 4: Complete [] 5: Well done []

Signature of the Instructor: ----- Date: -----

Assignment No. 8: Introduction to JavaScript

Author: Sneha Ganesh Chavan

Allotted Slots:2

Prerequisite:

- 1) You must know HTML well with CSS.
- 2) Need to know how web pages work.

What is Scripting Language?

Scripting language doesn't need compiler but uses interpreter which means each line of the code is interpreted one by one at run time. It is cross platform but no privacy. Slow speed during execution. Mostly they are used to automate the Task.

JavaScript is scripting language and not any programming language. Before we deep dive into learning JavaScript we need to understand why we are learning JavaScript and its role in website making. JavaScript makes the

Web page response which means if you created a web page and you want your client to fill whatever is in the web page.

Once the Client is done with filling the form on web page he/she would click on submit button and they would expect some action occurs and it redirects them to different page or give them a message. JavaScript is used for making the Web pages interactive with end user, moving images and for validation purpose as well.

Other Scripting languages can be: VBScript, Python, JavaScript, PHP, Ruby, Lua and many more.

- **Structure of a JavaScript program:**

- 1) The <script> ending with </script> tag is used inside the html code to include JavaScript code in it.
- 2) It is inside the <head> ending with </head> tag and can be used inside the <body> ending with </body> tag as well.
- 3) The file containing JavaScript must be saved with html or.htm extension.

Syntax: -

```
<html>
  <head>
    <script language="||JavaScript||">
      //JavaScript code
    </script>
```

OR

```
<html>
  <body>
    <script language="||JavaScript||">
      //JavaScript code
    </script>
```

- **Points to keep in mind:**

- 1) JavaScript is a Case sensitive language.
- 2) The JavaScript program code ends with semicolon.
- 3) Comments are defined using // or /**/.

Variables: A variable is container which stores the value and can be changed over period of time.

Declaration: var<nameOfTheVariable> = <value>;

Example: varaccountBalance = 1500;

Pro-tip: Give your variable name a meaningful name related to the program. It is good practice.

Identifier: As the name suggest it is used to identify. Identifiers are names like variable names, keywords, Functions and labels. There are some rules for naming as well like:

First character must be a letter or underscore (_) or a dollar sign (\$), which means numbers aren't allowed as first character while naming.

Operators: An operator is a symbol that tells to perform a certain mathematical, logical manipulation.

Operators in JavaScript are:

- 1) Assignment Operators: which are used to assign (=), (+=), (-=), (*=), (/=), (%=).
Eg: var balance = 300; (this shows balance variable is assigned 300 value).
- 2) Arithmetic Operators: In this you'd see Addition (+), Subtraction (-), Multiplication (*) and many more.
- 3) Comparison Operators: Used for comparing two values.
- 4) Logical Operators: Boolean conditions like &&, || and !
- 5) Conditional/ Ternary Operators: Condition? expression 1 : expression 2
It takes three operands and returns the value of expression1 if the condition is true or else it will return Value of expression 2.

- **Special Operators used in JavaScript:**

1. The equal(==) and not equal(!=) operators perform type conversions before testing for equality. For.e.g. -5!= 5 evaluate to true.
2. The strictly equal(==) and strictly not equal(!==) operators do not perform type conversions before testing for equality. For.e.g. -5== 5 evaluate to false.
3. String Operator:
In JavaScript string concatenation(+) operator is used to joint two strings.
For.e.g. -Hello+ -World produces output as HelloWorld.

- **Control Structures:** They are just the loops which we have learned in earlier classes.
Let us see the loops one by one.

- 1) if ... else
- 2) switch case
- 3) do while loop
- 4) while loop
- 5) for loop

1) If ... else

The if statement is the fundamental control statement that allows JavaScript to make decisions and execute statements conditionally.

Syntax:

```
if(expression){
```

Statement(s) to be executed if expression is true

```
}
```

Example:

```
<script type="text/javascript">  
<!--<br/>    var age =20;  
    if( age >18)  
    {  
        document.write("<b>Qualifies for driving</b>");  
    }  
//-->  
</script>
```

2) Switch case

The basic syntax of the switch statement is to give an expression to evaluate and several different statements to execute based on the value of the expression. The interpreter checks each case against the value of the expression until a match is found. If nothing matches, a default condition will be used.

Syntax:

```
switch(expression){  
    case condition 1: statement(s)
```

```
    break;
```

```
    case condition 2: statement(s)
```

```
    break;
```

```
    .....
```

```
    case condition n: statement(s)
```

```
    break;
```

```
    default: statement(s)
```

```
}
```

Example:

```
<script type="text/javascript">  
<!--<br/>var grade='A';  
document.write("Entering switch block<br/>");  
switch(grade){  
    case'A':document.write("Good job<br/>");  
    break;
```

```

        case'B':document.write("Pretty good<br/>");
        break;
        case'C':document.write("Passed<br/>");
        break;
        case'D':document.write("Not so good<br/>");
        break;
        case'F':document.write("Failed<br/>");
        break;
        default:document.write("Unknown grade<br/>")

    }

document.write("Exiting switch block");

//-->

</script>

```

3) Do while Loop

The do...while loop is similar to the while loop except that the condition check happens at the end of the loop. This means that the loop will always be executed at least once, even if the condition is false.

Syntax:

```

do{
    Statement(s) to be executed;
}while(expression);

```

Example:

```

<script type="text/javascript">

<!--

var count =0;
document.write("Starting Loop"+ "<br/>");
do{
    document.write("Current Count : "+ count + "<br/>");
    count++;
}while(count <0)
document.write("Loop stopped!");

//-->

</script>

```

This will produce following result:

Starting Loop

Current Count: 0

Loop stopped!

4) While Loop

The purpose of a while loop is to execute a statement or code block repeatedly as long as expression is true. Once expression becomes false, the loop will be exited.

Syntax:

```
while(expression){  
    Statement(s) to be executed if expression is true  
}
```

Example:

```
<script type="text/javascript">  
    <!--  
        var count = 0;  
        document.write("Starting Loop" + "<br/>");  
        while(count < 5){  
            document.write("Current Count : " + count + "<br/>");  
            count++;  
        }  
        document.write("Loop stopped!");  
    //-->  
</script>
```

This will produce following result:

Starting Loop

Current Count : 0

Current Count : 1

Current Count : 2

Current Count : 3

Current Count : 4

Current Count : 5

Loop stopped!

5) For Loop

The for loop is the most compact form of looping and includes the following three important parts –

The loop initialization where we initialize our counter to a starting value. The initialization statement is executed before the loop begins. The test statement which will test if the given condition is true or not. If condition is true then code given inside the loop will be executed otherwise loop will come out. The iteration statement where you can increase or decrease your counter.

Syntax:

```
for(initialization; test condition; iteration statement){  
    Statement(s) to be executed if test condition is true  
}
```

Example:

```
<script type="text/javascript">
```

```
    <!--  
        var count;  
        document.write("Starting Loop"+<br/>);  
        for(count =0; count <4; count++){  
            document.write("Current Count : "+ count );  
            document.write("<br/>");  
        }  
        document.write("Loop stopped!");  
    //-->  
</script>
```

This will produce following result which is similar to while loop –

Starting Loop

Current Count : 0

Current Count : 1

Current Count : 2

Current Count : 3

Current Count : 4

Loop stopped!

Dialog Boxes:-

There are three types of dialog boxes.

Sr.No.	Types and Syntax	Description
1	alert() dialogbox Syntax:- alert(-message);	The alert() dialog box displays the string passed to it as well as OK button. It can be used to display message.
2	confirm() dialog box Syntax :- confirm(-message);	The confirm dialog box used to confirm user action. It displays a predefined message and OK and Cancel button. a) Clicking on OK button returns True to the program. b) Clicking on Cancel button returns False to the program.
3	prompt dialog box Syntax:- prompt(-message ,-defaultValue);	The prompt dialog box is used to accept input from the user. It displays predefined message, a text box with default valueand OK and Cancelbutton.

1. Sample Program for Confirm Dialogbox:-

```
<html>
<head>
<script>
    if(confirm("do u want to continue?"))
        document.write("Welcome");
    else
        document.write("Good bye");

```

2. Sample Program for Prompt Dialogbox:-

```
<html>
<head>
<script>
    var fn=prompt("Enter your First Name");
    var ln=prompt("Enter your Surname");
    document.write("My name is" + fn + " " + ln);
</script>
```

LAB Exercise

Set A:

1. Write a JavaScript program to print factorial of a given number.
2. Write a JavaScript program to check whether given number is perfect or not.
3. Write a JavaScript program to check whether given number is Armstrong number or not.
4. Write a JavaScript program to accept a number from user and display that number in word (e.g. 226 → Two Two Six)

Set B:

1. Write a JavaScript program to print prime numbers between 1 to 89.
2. Write a JavaScript program to print number of even and odd numbers occur between 50 to 100.
3. Write a JavaScript program to print the reverse of a number.
4. Write a JavaScript program to store a keyword in a variable. Accept a new keyword from user till it matches with the value of variable. Allow three chances only.

Set C:

1. Write a JavaScript program to print sum of a digit of a number.
2. Write a JavaScript program to print the sum of first and last digit of a number.
3. Write a JavaScript program to print Fibonacci series.
4. Write a JavaScript to check given year value is leap year or not.

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Signature of the Instructor: ----- **Date:** -----

Assignment No. 9: Functions in JavaScript

Author: Sneha Ganesh Chavan

Allotted Slots: 2

A Function is a block of code that performs a specific task and often returns a value. A JavaScript function takes zero or more parameters. For example: Imagine you have to design a calculator in which you need functionalities like add, subtract, multiply and divide. All of them are different operations. So, according to these you'll have to design 4 functions in the class named calculator. 4 functions would be add, subtract, multiply and divide.

Structure of Function:

```
function add(number1, number2){  
    return number1+number2  
}
```

Small Program:

```
varaddingTwoNum = add(3,4); // add() invokes the function with arguments  
given.  
function add(number1, number2){ // here number1 and number2 are 3, 4 which are parameters.  
    return number1+number2  
}
```

There are two types of functions namely:

- A. Predefined Functions** :- JavaScript provides several predefined functions which means JavaScript has functions made already which you just need to use it. Few functions are listed below:

Sr. No.	Name	Types	Description
1	Conversion Function	eval()	eval() function can be used to convert a string expression to a numeric value.
		parseInt()	This function used to convert a string value to an integer. parseInt() returns the first integer contained in a string or NaN(Not a number) if the string does not begin with an integer.
		parseFloat()	This function returns the first floating number contained in a string or NaN if string does not begin with a valid floating point number.
2	String Function	bold()	Returns string in bold face.
		italics()	Returns string in italic.
		Length	Returns number of characters in string.
		toLowerCase()	Returns string with its entire uppercase letters converted to lowercase letters.
		toUpperCase()	Returns string with its entire lowercase letters converted to uppercase letters.
		charAt(index)	Returns the character at the specified index.
		charCodeAt(index)	Returns the Unicode of the character at the specified index.

		concat(string)	Joins two or more strings, and returns a copy of the joined strings
		indexOf(string, startIndex)	Returns the position of the first found occurrence of a specified value in a string.
		substr(startIndex, length)	Extracts the characters from a string, beginning at a specified start position, and through the specified number of character.
		replace(find, replaceWith)	Searches for a match between a substring (or regular expression) and a string, and replaces the matched substring with a new Substring
		search(stringToFind)	Searches for a match between a regular expression and a string, and returns the position of the match
3	Date Functions .	getDate()	Returns the day of the month (from 1-31)
		getDay()	Returns the day of the week (from 0-6)
		getMonth()	Returns the month (from 0-11)
		getFullYear()	Returns the year.
		getHours()	Returns the hour
		getMinutes()	Returns the minutes
		getSeconds()	Returns the seconds.
		getMilliseconds()	Returns the milliseconds
4	Math Function	Math.pow(x,y)	returns the value of x to the power of y
		Math.round(x)	returns the value of x rounded to its nearest integer
		Math.sqrt(x)	returns the square root of x
		Math.abs(x)	returns the absolute (positive) value of x
		Math.ceil(x)	returns the value of x rounded up to its nearest integer
		Math.floor(x)	returns the value of x rounded down to its nearest integer

1) Sample JavaScript function Program for String to Integer Conversion:-

```
<!DOCTYPE html>

<html>
<body>

<script>

    functionconvertStoI() {

        varr = parseInt("1011", 2);
        vark = parseInt("234", 8);

        document.write('Integer value is '+ r);
        document.write("<br>");
        document.write("integer value is "+ k);
        document.write("<br>");
        document.write(parseInt("226 Successful"));

    }

    convertStoI();

</script>
</body>
</html>
```

2) Sample Program for Math Function:-

```
<!DOCTYPE html>

<html>
<body>
<script>

document.write("Power="+Math.pow(2,3));
document.write("<br>Round="+Math.round(4.7));
document.write("<br>Round="+Math.round(4.4));
document.write("<br>Root="+Math.sqrt(16));
document.write("<br>Absolute="+Math.abs(-2.7));
document.write("<br>Ceil="+Math.ceil(4.4));
document.write("<br>Ceil="+Math.ceil(4.7));
document.write("<br>Floor="+Math.floor(4.7));

</script>
</body>
</html>
```

3) Sample Program for String and Date Function:-

```
<!DOCTYPE html>

<html>
<body>
<head>
<title>Show good morning good night wish as per time Javascript</title>
</head>
<script type="text/javascript">

    document.write("<center><font size=+3 style='color: green;'>");

    var day = new Date();

    var hr = day.getHours();

    if (hr >= 0 && hr < 12) {

        document.write("Good Morning!");

    } else if (hr == 12) {

        document.write("Good Noon!");

    } else if (hr >= 12 && hr <= 17) {

        document.write("Good Afternoon!");

    } else {

        document.write("Good Evening!");

    }

    document.write("</font></center>");

</script>
</html>
</body>
</html>
```

B. User defined Functions:-

Functions are declared and created using the function keyword and we can customize it according to our needs.

A function has,

- 1 A name for thefunction.
- 2 A list of parameters / arguments that will be accepting values passed to the function whencalled.
- 3 A block of JavaScript code that defines what the functiondoes.

Syntax :-

```
function functionName(parameter1,parameter2,----)  
{  
    //JavaScript Code  
}
```

Place of Declaration:

Functions can be declared anywhere within an HTML file, but if the function is called before it is declared and parsed, it will lead to an error condition. Therefore declaring functions within the `<head>---</head>` tags of the HTML file, ensures that all functions will be parsed before they are invoked or called.

Function Call:- To call a function, use function's name and its parameters as a statement.

Returning a value:- User define

functions can return values using return statement. The return statement can be used to return any valid expression that evaluates to a single value.

1. Sample program for user defined function

```
<html>  
<body>  
    <script>  
        function cubeOfNumber(num)  
        {  
            return (num*num*num);  
        }  
        var cube=cubeOfNumber(4);  
        document.write("Cube of a Number=" + cube);  
    </script>  
</body>  
</html>
```

Array:- An array is a collection of similar data type variables.

- **Declaration Syntax:-**

```
vararrName=new Array(arr_length);
```

Example:-

```
varplayerName=new Array();  
player_Name[50]=Sachin Tendulkar;  
player_Name[100]=Rahul Dravid;
```

TYPES OF ARRAYS:

- **Dense Array:-**

A dense array is an array that has been created with each of its elements being assigned a specific value. They are declared and initialized at the same time. Listing the values of the array elements in the array declaration creates dense arrays.

Syntax :-

```
vararrName=new Array(val0,val1,val2,-----,valn);
```

Example :-

1. var n=new Array(10,20,30,40,50);
- 2 vararr=new Array(abc,10,20,pqr);

Since array is a **JavaScript object**, arrays have several methods associated with them via which the array and its elements content can be manipulated. These methods are,

Sr. No.	Method Name and Syntax	Description
1	concat() Syntax :- array1.concat(array2,arry3, -----);	The concat() method is used to join two or more arrays. This method does not change the existing arrays, it only returns a copy of the joined arrays.
2	slice() Syntax :- array.slice(start, end)	The slice() method selects a part of an array, and returns the new array
3	join() Syntax :- array.join(separator);	returns all elements of the array joined together as a single string. This takes one argument, a string to be used as a separator between each element of the array
4	sort() :- Syntax :- array.sort();	The sort() method sorts the elements of an array.
5	reverse() :- Syntax :- array.reverse();	reverses the order of the element in the array.
6	push() Syntax :- array.push(newEle);	adds a new element to an array (at the end)
7	pop() Syntax :- array.pop();	removes the last element from an array
8	shift() Syntax :- array.shift();	removes the first array element and "shifts" all other elements to a lower index
9	unshift() Syntax :- array.unshift(newEle);	adds a new element to an array (at the beginning), and "unshifts" older elements

1. Sample program for an Array

```
<html>
<body>
<script>
    vararr=new Array(10,20,30,40);
    document.write("Join="+arr.join()+"<br>");
    document.write("Join=x"+arr.join("x")+"<br>");
    document.write("Reverse"+arr.reverse()+"<br>")
    ; document.write("Length=" + arr.length3 + "<br>");
</script>
</body>
</html>
```

LAB Exercise

Set A

1. Write a JavaScript code to greet the user according to the current timing.
2. Write a JavaScript Program to read a number from user, store its factors into the array and display that array.
3. Write a menu driven program using JavaScript to perform the following operations on an array
 - a. Display an array
 - b. Sort elements of an array
 - c. Reverse elements of an array
 - d. Search a given element from an array

Set B

1. Write a menu driven program using JavaScript to find square-root, power and absolute value of a given number.
2. Write a JavaScript code to accept date from the user. If date entered by the user is
1st January
then print - Happy New Year!, if 25th December then print - Merry Christmas,
if 14th January then - happy Makarsankranti, otherwise print - Have a Good Day message ;
3. Write a menu driven program using JavaScript to perform the following operations.
 1. Insert an element in stack
 2. Delete an element from stack
 3. Display the contents of stack
 4. Insert an element in queue
 5. Delete an element from queue
 6. Display the contents of queue

Set C

1. Write a JavaScript program to display a Multiplication table in tabular format using function.
2. Write a JavaScript code to calculate maximum, minimum, sum and average of numbers in an array.
3. Write a JavaScript code to accept birth-date and print the age of a user.

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Signature of the Instructor: ----- **Date:** -----

Assignment No. 10(a) : Validation using JavaScript

Author: Sneha Ganesh Chavan

Allotted Slots:2

Validation is the process of checking whether the specification captures the customer's needs. It includes testing and validating the actual product.

Verification is the process of checking that the software meets the specification. It includes checking documents, design, codes and programs. A website can before getting live it goes under validation stage.

JavaScript provides a way to validate form's data on the client's computer before sending it to the web server. Form validation generally performs two functions.

Basic Validation – First of all, the form must be checked to make sure all the mandatory fields are filled in. It would require just a loop through each field in the form and check for data.

Data Format Validation – Secondly, the data that is entered must be checked for correct form and value. Your code must include appropriate logic to test correctness of data.

For example: If a form field (firstName) is empty, this function alerts a message, and returns false, to prevent the form from being submitted.

Sample example:

```
<!DOCTYPE html>
<html>
<head>
<script>
    function validateForm() {
        var x = document.forms["myForm"]["firstName"].value;
        if (x == "") {
            alert("Name must be filled out");
            return false;
        }
    }
</script>
</head>
<body>

<form name="myForm" action="/action_page.php" onsubmit="return validateForm()"
method="post">
    Name: <input type="text" name="firstName">
    <input type="submit" value="Submit">
</form>

</body>
</html>
```

For example this will be your registration form:

Registration Form

User id: Required and must be of length 5 to 12.

Password: Required and must be of length 7 to 12.

Name: Required and alphabates only.

Address: Optional.

Country: (Please select a country) Required. Must select a country.

ZIP Code: Required. Must be numeric only.

Email: Required. Must be a valid email.

Sex: Male Female Required.

Language: English Non English Required.

About: Optional.

So, there are:

- Textbox
- Checkbox
- Drop down
- Radio button
- Textarea (it is optional to fill)
- Button

You have to make sure the textbox is not empty as well as you have to make user input correct values in the textbox's. Drop down must be selected same goes with radio button. The checkbox must be checked otherwise it would accept Register Now button getting clicked. If everything is filled perfectly only then it should go on to next page.

Let's see how it is supposed to be coded.
From HTML, CSS to JavaScript.

CODE:

- (HTML using BootStrap)

```
<!DOCTYPE html>

<html lang="en"><head>

<meta charset="utf-8">

<title>JavaScript Form Validation using a sample registration form</title>

<meta name="keywords" content="example, JavaScript Form Validation, Sample registration form" />

<meta name="description" content="This document is an example of JavaScript Form Validation using a sample registration form. " />

<link rel='stylesheet' href='js-form-validation.css' type='text/css' />

<script src="sample-registration-form-validation.js"></script>

</head>

<body onload="document.registration.userid.focus();">

<h1>Registration Form</h1>

Use tab keys to move from one input field to the next.

<form name='registration' onSubmit="return formValidation();">

<ul>

<li><label for="userid">User id:</label></li>

<li><input type="text" name="userid" size="12" /></li>

<li><label for="passid">Password:</label></li>

<li><input type="password" name="passid" size="12" /></li>

<li><label for="username">Name:</label></li>

<li><input type="text" name="username" size="50" /></li>

<li><label for="address">Address:</label></li>

<li><input type="text" name="address" size="50" /></li>

<li><label for="country">Country:</label></li>

<li><select name="country">

<option selected="" value="Default">(Please select a country)</option>
```

```
<option value="AF">Australia</option>
<option value="AL">Canada</option>
<option value="DZ">India</option>
<option value="AS">Russia</option>
<option value="AD">USA</option>
</select></li>
<li><label for="zip">ZIP Code:</label></li>
<li><input type="text" name="zip" /></li>
<li><label for="email">Email:</label></li>
<li><input type="text" name="email" size="50" /></li>
<li><label id="gender">Sex:</label></li>
<li><input type="radio" name="msex" value="Male" /><span>Male</span></li>

<li><input type="radio" name="fsex" value="Female" /><span>Female</span></li>
<li><label>Language:</label></li>
<li><input type="checkbox" name="en" value="en" checked /><span>English</span></li>
<li><input type="checkbox" name="nonen" value="noen" /><span>Non English</span></li>
<li><label for="desc">About:</label></li>
<li><textarea name="desc" id="desc"></textarea></li>
<li><input type="submit" name="submit" value="Submit" /></li>
</ul>
</form>
</body>
</html>
```

- Now let's see the Cascading Style Sheet, make sure you save it as .css file or you can embed it as well in html but it is mostly advised to keep css apart from html code.

```
h1 {  
margin-left: 70px;  
}  
form li {  
list-style: none;  
margin-bottom: 5px;  
}  
  
form ul li label{  
float: left;  
clear: left;  
width: 100px;  
text-align: right;  
margin-right: 10px;  
font-family: Verdana, Arial, Helvetica, sans-serif;  
font-size: 14px;  
}  
  
form ul li input, select, span {  
float: left;  
margin-bottom: 10px;  
}  
  
form textarea {  
float: left;  
width: 350px;  
height: 150px;  
}  
  
[type="submit"] {  
clear: left;  
margin: 20px 0 0 230px;  
font-size: 18px  
}
```

```
p {  
margin-left: 70px;  
font-weight: bold;  
}
```

Now it is time to see JavaScript validation: using the function formValidation.

```
function formValidation()
{
var uid = document.registration.userid;
var passid = document.registration.passid;
var uname = document.registration.username;
var uadd = document.registration.address;
var ucountry = document.registration.country;
var uzip = document.registration.zip;
var uemail = document.registration.email;
var umsex = document.registration.msex;
var ufsex = document.registration.fsex; if(userid_validation(uid,5,12))
{
if(passid_validation(passid,7,12))
{
if(allLetter(uname))
{
if(alphanumeric(uadd))
{
if(countryselect(ucountry))
{
if(allnumeric(uzip))
{
if(ValidateEmail(uemail))
{
if(validsex(umsex,ufsex))
{
}
}
}
}
}
}
}
}
}
}
}
}
}
}
return false;
}
```

LAB Exercise

Set A

1. Write a JavaScript code to design registration form.
2. Draw a flowchart for one entire module and show its working.
3. Write JavaScript code for formValidation.

Set B

1. Write a menu driven program using JavaScript to find square root, power and absolute value of a given number and validate them.
2. Write a JavaScript code using Bootstrap for sign-up form.
3. Write a JavaScript function for Calculator and verify them.

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: WellDone []

Signature of the Instructor: ----- **Date:** -----

Assignment No. 10(b): Functions for Validation using JavaScript & Event Handling.

Author: Sneha Ganesh Chavan

Allotted Slots: 2

In the last assignment we have seen that how registration form is created and what is the importance of the validation with difference between validation and verification. Continuing from the last assignment we will check how each function for each field which is your textbox can be validated.

□ Function for validating uid:

```
function userid_validation(uid,mx,my)
{
    var uid_len = uid.value.length;
    if (uid_len == 0 || uid_len >= my || uid_len < mx)
    {
        alert("User Id should not be empty / length be between "+mx+" to "+my);
        uid.focus();
        return false;
    }
    return true;
}
```

□ Function for password validation:

```
function passid_validation(passid,mx,my)
{
    var passid_len = passid.value.length;
    if (passid_len == 0 || passid_len >= my || passid_len < mx)
    {
        alert("Password should not be empty / length be between "+mx+" to "+my);
        passid.focus();
        return false;
    }
    return true;
}
```

Function for validating username:

```
function allLetter(uname)
{
    var letters = /^[A-Za-z]+$/;
    if(uname.value.match(letters))
    {
        return true;
    }
    else
    {
        alert('Username must have alphabet characters only');
        uname.focus();
        return false;
    }
}
```

Function for emailValidation:

```
function ValidateEmail(uemail)
{
    var mailformat = /^[\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$];
    if(uemail.value.match(mailformat))
    {
        return true;
    }
    else
    {
        alert("You have entered an invalid email address!");
        uemail.focus();
        return false;
    }
}
```

```
}
```

```
}
```

Here is the entire JavaScript used for validation of the form.

```
function formValidation()
{
    var uid = document.registration.userid;
    var passid = document.registration.passid;
    var uname = document.registration.username;
    var uadd = document.registration.address;
    var ucountry = document.registration.country;
    var uzip = document.registration.zip;
    var uemail = document.registration.email;
    var umsex = document.registration.msex;
    var ufsex = document.registration.fsex; if(userid_validation(uid,5,12))
    {
        if(passid_validation(passid,7,12))
        {
            if(allLetter(uname))
            {
                if(alphanumeric(uadd))
                {
                    if(countryselect(ucountry))
                    {
                        if(allnumeric(uzip))
                        {
                            if(ValidateEmail(uemail))
                            {
                                if(validsex(umsex,ufsex))
                                {
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
        }
    }
}
}
}
}

return false;

} function userid_validation(uid,mx,my)
{
var uid_len = uid.value.length;
if (uid_len == 0 || uid_len >= my || uid_len < mx)
{
alert("User Id should not be empty / length be between "+mx+" to "+my);
uid.focus();
return false;
}
return true;
}

function passid_validation(passid,mx,my)
{
var passid_len = passid.value.length;
if (passid_len == 0 || passid_len >= my || passid_len < mx)
{
alert("Password should not be empty / length be between "+mx+" to "+my);
passid.focus();
return false;
}
return true;
}

function allLetter(uname)
```

```
{  
var letters = /^[A-Za-z]+$/;  
if(uname.value.match(letters))  
{  
    return true;  
}  
else  
{  
    alert('Username must have alphabet characters only');  
    uname.focus();  
    return false;  
}  
}  
  
function alphanumeric(uadd)  
{  
var letters = /^[0-9a-zA-Z]+$/;  
if(uadd.value.match(letters))  
{  
    return true;  
}  
else  
{  
    alert('User address must have alphanumeric characters only');  
    uadd.focus();  
    return false;  
}  
}  
  
function countryselect(ucountry)  
{  
if(ucountry.value == "Default")
```

```
{  
alert('Select your country from the list');  
ucountry.focus();  
return false;  
}  
  
else  
{  
return true;  
}  
  
}  
  
function allnumeric(uzip)  
{  
var numbers = /^[0-9]+$/;  
if(uzip.value.match(numbers))  
{  
return true;  
}  
else  
{  
alert('ZIP code must have numeric characters only');  
uzip.focus();  
return false;  
}  
}  
  
function ValidateEmail(uemail)  
{  
var mailformat = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*(\\.\\w{2,3})+$/;  
if(uemail.value.match(mailformat))  
{  
return true;  
}
```

```
}

else
{
alert("You have entered an invalid email address!");
uemail.focus();
return false;
}

} function validsex(umsex,ufsex)
{
x=0;

if(umsex.checked)
{
x++;
} if(ufsex.checked)
{x++;
}
if(x==0)
{
alert('Select Male/Female');
umsex.focus();
return false;
}
else
{
alert('Form Succesfully Submitted');
window.location.reload()
return true;
}
}
```

A. EventHandling

An event occurs when something happens in a browser window.

The kinds of events that might occur are due to:

- a) A documentloading
- b) The user clicking a mousebutton
- c) The browser screen changingsize

When a function is assigned to an event handler, that function will run when that event occurs.

Syntax:-

```
eventHandler="clickHandler()"
```

This handler will cause the function clickHandler() to be executed whenever the event is triggered.

Standard event handlers

EventHandler	Description
onabort	Occurs when loading of image was interrupted
onblur	Occurs when element loses focus
onchange	Occurs when element gets modified
onclick	Occurs when element gets clicked
ondblclick	Occurs when element gets double clicked
onfocus	Occurs when an element received focus
onkeydown	Occurs when a key was pressed when an element has focus
onkeypress	Occurs when a keystroke was received by the element
onkeyup	Occurs when a key was released when the element has focus
onload	Occurs when an element was loaded
onmousedown	Occurs when the mouse button was pressed on the element
onmousemove	Occurs when the mouse pointer moves while inside the element
onmouseout	Occurs when the mouse pointer was moved outside the element
onmouseover	Occurs when the mouse pointer was moved onto the element
onmouseup	Occurs when the mouse button was released on the element.
onreset	Occurs when the form's reset button was clicked

Onresize	Occurs when the containing window or frame was resized
Onselect	Occurs when text within the element was selected
Onsubmit	Occurs when a form is being submitted
Onunload	Occurs when the content is being unloaded (e.g. window being closed)
Onscroll	Occurs when the user scrolls (in any direction and with any means).

1 Sample program to display message using double click event of bodytag.

```
<html>
    <body ondblclick="alert('Hello world!');">
        </body>
```

2 Sample program to display message using double click event of body tag(usingFunction)

```
<html>
<head>
    <script language=JavaScript>function
        printMsg()
    {
        alert(Hello world!);
    }
</script>
```

LAB Exercise

Set A

- a. Write a JavaScript function to validate email-id using regular expression.
- b. Write a JavaScript program for accepting name and mobile number from user and perform following validation:
 - i. Check all fields should not contain a null value
 - ii. Check name field contains only alphabets
 - iii. Mobile No. field should be of 10 digits long.
- c. Write a JavaScript program to compare the values of password and confirmed password field and display message accordingly. Also perform the validation to check any of the field should not be empty.

User Login

User Name:

Password :

Confirmed Password :

Submit

Set B

1. Write a JavaScript program to read employee details and generate payslip which will calculate net salary of an employee.

Payslip

Name of Employee:

Department

Designation:

Basic Salary

HRA

DA

Submit

2. Write a Java Script program to design Customer Account Details Form and perform validation on pan number field. (pan number is of only 10 characters long, out of which first 5 characters are alphabets, next 4 characters are digits and last character is alphabet)
3. Design a form to accept product name, quantity, rate and discount for the product purchased by the user. Write a JavaScript code to calculate total bill. If quantity is less than 5 then there is no discount, If quantity is greater than 5 and less than 25 then discount is 5%, If quantity is greater than 25 and less than 50 then discount is 15%, If quantity is greater than 50 discount is 20%.

Set C

1. Write a JavaScript code to change the background color of the webpage.
2. Write a menu driven program using JavaScript code to perform following operations,
 - a. Jump to the previous webpage
 - b. Jump to the next webpage
 - c. Jump to the two page before from current page
 - d. Jump to the two page after from current page.

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: WellDone []

Signature of the Instructor: ----- **Date:** -----

Assignment No 11: To study designing of HTML in different way using CSS

Author: Priyamvada U. Patil.

Allotted Slots: 3

Aim: Pre-requisite: Basic HTML tags and CSS

SET A: Write the appropriate code to display the output below :

1.



2.

Subscribe to our mailing list

* indicates required

Email Address

 *

First Name

Last Name

Subscribe

3. Write the HTML code to display Top 5 resorts in Goa. Make use of HTML tags and CSS that you have studied till now.

The 5 Best Goa Resorts

Check out our pick of great resorts in Goa

Filter by:

Review score

- Superb: 9+
- Very good: 8+
- Good: 7+
- Pleasant: 6+



Agonda Serenity Resort

📍 Agonda

Situated in Agonda, 50 metres from Agonda Beach, Agonda Serenity **Resort** features accommodation with a restaurant, free private parking, a bar and a garden. The accommodation offe...

[Show more](#)

Superb
220 reviews
9.1



Andores Resort And Spa

📍 Calangute Beach, Calangute

Boasting an outdoor **swimming pool**, Andores **Resort And Spa** offers rooms in Calangute, a 12-minute walk from Infantaria and 1.2 km from St. Alex Church. 2.6 km from Tito's...

[Show more](#)

Superb
129 reviews
9.1

5
at

SET B:

1. Write the appropriate code to display the output below :

SUBSCRIBE

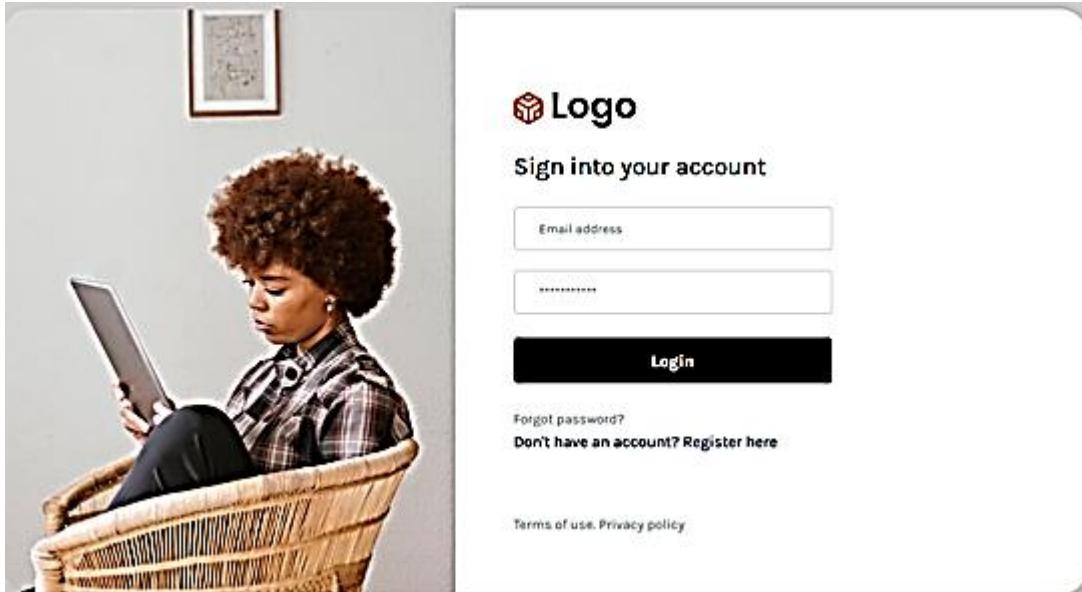


Subscribe to get the latest news & updates.

Your email address

Subscribe

- 2.



SET C:

1. Write the appropriate code to display available Vegetable Baskets.

 <p>Mini Vegetable Basket - 3.5 to 4.5 Kg (7M-MINI4)</p> <p>97 Reviews</p> <p>₹270.00</p> <p>Add to Cart</p>	 <p>Medium Vegetable Basket - 6 to 6.5 Kg (7M-MED6)</p> <p>31 Reviews</p> <p>₹379.00</p> <p>Add to Cart</p>	 <p>Mega Vegetable Basket - 8 to 9.5 Kg (7M-MEGA9)</p> <p>51 Reviews</p> <p>₹499.00</p> <p>Add to Cart</p>
---	--	---

- 2.

Personal Information

First Name:

Last Name:

Date of Birth: Month Day Year ?

Gender: Choose a gender ?

Account Information

Email:
(Your email address will be your username)

Re-type Email:

Password:
(Min. 8 characters, 1 number, case-sensitive)

Re-type Password:

Security Question: Choose a security question ?

Security Answer:
(Not case-sensitive)

Contact Information

Address:

City:

State: Choose a state

Zip Code: Optional

Phone: ? Mobile ?
No spaces or dashes

Assignment Evaluation

0: Not Done [] 1: Incomplete [] 2: Late Complete []

3: Needs Improvement [] 4: Complete [] 5: Well done []

Signature of the Instructor: ----- **Date:** -----

Assignment No 12: File uploads , Field & Form Validation

Authors: Mrs. Veena K. Gandhi

& Preeti Gawade

Allotted Slots: 2

A PHP script can be used with a HTML form to allow users to upload files to the server. Initially files are uploaded into a temporary directory and then relocated to a target destination by a PHP script.

In your "php.ini" file, search for the file_uploads directive, and set it to On:

file_uploads = On

There is one global PHP variable called **\$_FILES**. This variable is an associate double dimension array and keeps all the information related to uploaded file.

The keys are:

name

The name of the file, as supplied by the browser. It's difficult to make meaningful use of this, as the client machine may have different filename conventions than the web server (e.g., if the client is a Windows machine that tells you the file is *D:\PHOTOS\ME.JPG*, while the web server runs Unix, to which that path is meaningless).

type

The MIME type of the uploaded file, as guessed at by the client.

size

The size of the uploaded file (in bytes). If the user attempted to upload a file that was too large, the size is reported as 0.

tmp_name

The name of the temporary file on the server that holds the uploaded file. If the user attempted to

upload a file that was too large, the name is reported as "none".

For example :

- `$_FILES['filename']['tmp_name']` – the uploaded file in the temporary directory on the webserver.
- `$_FILES['filename']['name']` – the actual name of the uploaded file.
- `$_FILES['filename']['size']` – the size in bytes of the uploaded file.
- `$_FILES['filename']['type']` – the MIME type of the uploaded file.
- `$_FILES['filename']['error']` – the error code associated with this file upload.

The correct way to test whether a file was successfully uploaded is to use the function `is_uploaded_file()`, as follows:

```
if (is_uploaded_file($_FILES['toProcess']['tmp_name']))  
  
{  
  
    // successfully uploaded  
  
}
```

To move a file, use the `move_uploaded_file()`

function: `move_uploaded_file($_FILES['toProcess']['tmp_name'], "path/to/put/file/$file);`

The call to `move_uploaded_file()` automatically checks whether it was an uploaded file. When a script finishes, any files uploaded to that script are deleted from the temporary directory.

Consider sample program to upload file

```
<html>  
  
<body>  
  
<form action="upload.php" method="post" enctype="multipart/form-data">  
  
Select File to upload:  
  
<input type="file" name="uploadedfile" id="fileToUpload">  
  
<input type="submit" value="Upload File/Image" name="submit">  
  
</form>  
  
</body>  
</html>
```

```

<?php

$target_path = "D:/uploads/";

$target_path = $target_path . basename( $_FILES['uploadedfile']['name']);

if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target_path))
{
    echo "The file ". basename( $_FILES['uploadedfile']['name']). " has been uploaded";
}
else{
    echo "There was an error uploading the file, please try again!";
}

?>

```

Form Validation:

When you allow users to input data, you typically need to validate that data before using it or storing it for later use. There are several strategies available for validating data like Form Fields should not be empty. Check type of data entered by user or length of data entered by user. (e.g check age of person is integer or cannot be negative etc.) Check specific conditions for form fields(e.g. email validation).

PHP provides empty() function to check a variable is empty. We can use this function to check if all the text fields are empty or not. iset() function can be used to check the gender radio button is checked or not. We can validate email format by using PHP filter_var() function.

- **empty(var_name)** : The empty() function is used to check whether a variable is empty or not.
- **isset(var_name)** : this function determines if a variable is set and is not NULL
- **filter_var(var, filtername, options)** : The filter_var() function filters a variable with the specified filter.

Consider the following script to validate email.

```
<?php
$email = "john.doe@example.com";
if (!filter_var($email, FILTER_VALIDATE_EMAIL) === false) {
    echo("$email is a valid email address");
} else {
    echo("$email is not a valid email address");
}
?>
```

Example : Design form which accepts name and gender . If fields are blank display error message “All fields are required” otherwise display information

```
<html>
<head>
<style>
span.error {color: red;}
</style>
</head>
<body>
<?php
$nameErr = $genderErr = "";
$name = $gender = "";

if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    if (empty($_POST["name"]))
        $nameErr = "Name is required";
    else
        $name = $_POST["name"];
}
```

```
if (!isset($_POST["gender"]))
$genderErr = "Gender is required";

else

$gender =$_POST["gender"];
}

?>

<h2>PHP Form Validation Example</h2>
<p><span class="error">* Required field.</span></p>

<form method="POST" action=<?php echo $_SERVER['PHP_SELF']?>">

Name: <input type="text" name="name" value=<?php if ($name) echo "$name";?>"><span class="error">* <?php echo $nameErr;?></span>

<br><br>

<input type="radio" name="gender" value="female" <?php if($gender=="female") { echo "checked"; }?>>Female

<input type="radio" name="gender" value="male" <?php if($gender=="male") { echo "checked"; }?>>Male
<span class="error">* <?php echo $genderErr;?></span>

<br><br>

<input type="submit" name="submit"
value="Submit"></form>

<?php

if($name && $gender)

{
echo "<h2>Your Input:</h2>";

echo $name;

echo "<br>";

echo $gender;

}

?>
</body>

</html>
```

Lab Assignments:**SET A:**

1. Write a PHP Program to Upload the file and display its information like name, size, type , etc.
2. Write a PHP program to accept student information like name, address, class and Upload student photo and display same on form.

SET B:

1. **Write** a PHP program to accept Name , address , Pincode ,Gender information. If any field isblank , it display error message “all fields are required” . If pincode is more than 6 digits , it should give error.
2. Write a PHP program to accept empno, name, pan card information, email . If any field is blank , form should display error message “all fields are required”. Pan card number should be 10 digits and First 5 characters should be letter , next 4 characters should be digit and last character should be letter.
3. Write a PHP script to create a form that accept theusers full name and their email addresses. Use case conversion function to capitalize the first letter of each name, user submits and print result back to browser. Check that the user’s email address contains the @ symbol.

SET C:

- 1) Write a PHP script for creating a self-processing page for a form. The form should allow the user to enter the following attributes: Username, user city preference(pune/Mumbai/Chennai/kolkata),user birth date, occupation, sex. If any of the values is not entered by the user, the page is presented again with a message specifying the attributes that are empty. Any form attributes that the user already entered, are set to the values the user entered. The text of submit button changes from “create” to “continue”, when the user is correcting the form. Display the details entered by the user on the next form

Assignment Evaluation

0: Not Done []

1: Incomplete []

2: Late Complete []

3: Needs Improvement []

4: Complete []

5: Well Done []

Signature of the Instructor:----- Date:-----

Assignment No 13: Cookies and Sessions

Authors: Mrs. Veena K. Gandhi
& Preeti Gawade

Allotted Slots: 2

Cookies :

- A cookie is basically a string that contains several fields. A server can send one or more cookies to a browser in the headers of a response. Some of the cookie's fields indicate the pages for which the browser should send the cookie as part of the request.
- The setcookie() function is used to send a cookie to the browser:
- `setcookie(name[,value[,expire[,path[,domain[,secure]]]]]);` where parameters are as
- **name** : A unique name for a particular cookie. You can have multiple cookies with different names and attributes. The name must not contain whitespace or semicolons.
- **value** : The arbitrary string value attached to this cookie. The original Netscape specification limited the total size of a cookie (including name, expiration date, and other information) to 4 KB, so while there's no specific limit on the size of a cookie value, it probably can't be much larger than 3.5 KB.
- **expire** : The expiration date for this cookie. If no expiration date is specified, the browser saves the cookie in memory and not on disk. When the browser exits, the cookie disappears. The expiration date is specified as the number of seconds.
- **path** : The browser will return the cookie only for URLs below this path. The default is the directory in which the current page resides.
- **domain** : The browser will return the cookie only for URLs within this domain. The default is the server hostname.
- **secure** : The browser will transmit the cookie only over https connections. The default is false, meaning that it's okay to send the cookie over insecure connections

This function creates the cookie string from the given arguments and creates a Cookie header with that string as its value. Because cookies are sent as headers in the response, setcookie() must be called before any of the body of the document is sent.

When a browser sends a cookie back to the server, you can access that cookie through the `$_COOKIE` array. The key is the cookie name, and the value is the cookie's value field.

Consider a sample program to keep track of number of times the web page has been accessed

```
<?php  
    if(isset($_COOKIE['accesses']))  
  
        $cnt = $_COOKIE['accesses'];  
    else  
  
        $cnt = 0 ;  
  
  
    setcookie('accesses', ++$cnt);  
  
    echo "You have visited this page $cnt times ";  
?>
```

Deleting Cookies

You can delete a cookie by calling the same setcookie() function with the cookie name and any value (such as an empty string) however this time you need to set the expiration date in the past.

Consider the sample example for deleting cookie

```
<?php  
// Deleting a cookie  
setcookie("username", "", time()-3600);  
?>
```

Sessions :

Session allow us to easily create multi page forms, save user authentication information from page to page, and store persistent user preferences on a site. A session can be defined as a series of related interactions between a single client and the Web server. The session may consist of multiple requests to the same script or a variety of different resources on the same web site.

To enable sessions for a page, call session_start() before any of the document has been generated:

```
<?php session_start() ?>
```

```
<html>
```

```
...
```

```
</html>
```

This assigns a new session ID if it has to, possibly creating a cookie to be sent to the browser, and loads any persistent variables from the store. into the associative array \$HTTP_SESSION_VARS. The keys are the variables' names (e.g., \$HTTP_SESSION_VARS['hits'])

Functions :

- bool **session_start** ([array \$options = []]) :to enable session for a page. This function assigns a new session ID to the new session.
- bool **session_register** (mixed \$name [, mixed \$...]) : to register a variable with the session by passing the name of the variable. When a session is started, you can store any number of variables in the \$_SESSION superglobal array and then access them on any session enabled page.
- string **session_id** ([string \$id]) : **session_id()** is used to get or set the session id for the current session.
- bool **session_destroy** (void) : **session_destroy()** destroys all of the data associated with the current session. It does not unset any of the global variables associated with the session, or unset the session cookie. To use the session variables again, session_start() has to be called.
- bool **session_unregister**(string \$name) : **session_unregister()** unregisters the global variablenamed name from the current session.

Consider example which prints visitor count of web page

```
<?php  
session_start(); //start the PHP_session function  
if(isset($_SESSION['page_count']))  
$_SESSION['page_count'] += 1;  
else
```

```
$_SESSION['page_count'] = 1;  
  
echo 'You are visitor number ' . $_SESSION['page_count']; ?>
```

Lab Exercise

SET A

1. A web application that lists all cookies stored in the browser on clicking “list cookies” button, add cookies if necessary.
2. Write a PHP program to store current date-time in a COOKIE and display the ‘Last visited on’ date-time on the web page upon reopening of the same page.
3. Write a script to keep track of number of times the web page has been accessed using session.
4. Create a login form with a username and password. Once the user logs in, the second form should be displayed to accept user details (name, city, phoneno). If the user doesn’t enter information within a specified time limit, expire his session and give a warning otherwise Display Details using sessions.
5. Write PHP program to store Customer information like customer no, name, address, mobile no. On second page, accept product code, product name, Qty, Rate. Display Bill on third page including customer and product details.

SET B

1. Write a PHP script to accept username and password. If in the first three chances, username and password entered is correct, then display second form, otherwise display error message.
2. Create a form to accept student information (name, class, address). Once the student information is accepted, accept marks in next form (Java, PHP, SE, OS, Pract1, and Pract2). Display the mark sheet for the student in the next form containing name, class, marks of the subject, total and percentage using cookies.
3. Change the preferences of your web page like font style, font size, font color, background color using cookie. Display selected settings on next web page and actual implementation (with new settings) on third web page.
4. Create a form to accept employee details like name, address and mobile number. Once the employee information is accepted, then accept LIC information like policy_no, name, premium. Display employee details and LIC details on next form.
5. Write a PHP script to accept Employee details (Eno, Ename, Add.) on first page. On second page accept earning (Basic, DA, HRA). On third page Print Employee Information (Eno, Ename, Add, Basic, DA, HRA, Gross)

SET C

1. Write a program to create a shopping mall. User must be allowed to do purchase from three pages. Each page should have a page total. The fourth page should display a bill, which consists of a page total of whatever the purchase has been done and print the total. (Use `$_SESSION`).

Assignment Evaluation

0: Not Done[] 1: Incomplete[] 2: Late Complete []

3: Needs Improvement[] 4: Complete [] 5: Well Done []

Signature of the Instructor:-----**Date:**-----

Assignment No 14: XML

**Authors: Mohsin Tamboli
&Preeti Gawade**

Allotted Slots: 2

Introduction to XML :

XML stands for EXtensible Markup Language. It is a text-based markup language derived from Standard Generalized Markup Language (SGML). XML was designed to store and transport data. XML was designed to be both human- and machine-readable. XML is a markup language much like HTML. XML was designed to describe data. XML tags are not predefined . You must define your own tags.XML is self describing.

XML document are well – formed and valid. A well - formed XML document follows the basic XML syntax rules.A valid document also follows the rules imposed by a DTD or an XSD.

A simple document is shown in the following example –

```
<?xml version = "1.0"?>
<contact-info>
    <name>TanmayPatil</name>
    <company>TutorialsPoint</company>
    <phone>(011) 123-4567</phone>
</contact-info>
```

The following image depicts the parts of XML document.



Document Prolog Section :

Document Prolog comes at the top of the document, before the root element. This section contains –

1. XMLdeclaration
2. Document typedeclaration

Document Elements Section:

Document Elements are the building blocks of XML. These divide the document into a hierarchy of sections, each serving a specific purpose.

XML declaration :

It contains details that prepare an XML processor to parse the XML document. It is optional, but when used, it must appear in the first line of the XMLdocument.

```
<?xml version="version_number" encoding="encoding_declaration" standalone="standalone_status" ?>
```

An XML declaration should abide with the following rules:

- The XML declaration is case sensitive and must begin with "<?xml>" where "xml" is written in lower-case. If the XML declaration is included, it must contain version numberattribute.
- The Parameter names and values are case-sensitive. The names are always in lowercase.
- The order of placing the parameters is important. The correct order is:*version*, *encoding* and *standalone*. Either single or double quotes may be used.
- The XML declaration has no closing tag i.e.</?xml>

Example of XML declaration:

- <?xml>
- <?xmlversion="1.0">
- <?xml version="1.0" encoding="UTF-8" standalone="no"?>
- <?xml version='1.0' encoding='iso-8859-1' standalone='no'?>

DTD :Document Type Declaration :

- The XML Document Type Declaration, commonly known as DTD, is a way to describe XML languageprecisely.
- DTDs check vocabulary and validity of the structure of XML documents against grammatical rules of appropriate XMLlanguage.
- An XML DTD can be either specified inside the document, or it can be kept in a separate document and then likedseparately.
- Basic syntax of a DTD is asfollows:
- <!DOCTYPE element DTD identifier [
- declaration1 declaration2

.....

]>

XML Tags :

XML tags are case sensitive. The tag <Letter> is different from the tag <letter>. Opening and closing tags must be written with the same case.

For example,

```
<Message>This is incorrect</message>
<message>This is correct</message>
```

XML Elements :

- An XML file is structured by several XML-elements, also called XML-nodes or XML-tags. XML-elements' names are enclosed by triangular brackets <>.
- Each XML-element needstobeclosed eitherwithstartorwithendelementsasshownbelow:
- <element> </element>.
- An XML document can have only one rootelement
- An XML-element can contain multiple XML-elements as its children, but the children elements must notoverlap.
- In XML, all elements must be properly nested within eachother.

XML attributes:

- An XML-element can have one or moreattributes.
- Attribute names in XML (unlike HTML) are case sensitive. That is, *HREF* and *href*are considered two different XMLattributes.
- Same attribute **cannot have two values in asyntax**

So XML follows tree structure

```
<root>
<child>
<subchild> .....</subchild>
</child>
</root>
<?xml version = “1.0” ?>

<BookStore>
<Books>
<PHP>
</PHP>
<PHP>
</PHP>
</Books>
</BookStore>
<title>Programming PHP</title>
<publication>O’RELLY</publication>
<title>Beginners PHP</title>
<publication>WROX</publication>
```

SimpleXML :

- SimpleXML is an extension that allows us to easily manipulate and get XML data.
- The SimpleXML extension is the tool of choice for parsing an XML document.
- SimpleXML turns an XML document into a data structure you can iterate through like a collection of arrays and objects.
- The SimpleXML extension includes interoperability with the DOM for writing XML files and built-in XPath support.
- SimpleXML is easier to code than the DOM, as its name implies.

SimpleXMLElement class represents an element in an XML document.

- To create root element of xml document, first create object of SimpleXMLElement class and initialize with rootelement.
- For example:
- \$bk=new SimpleXMLElement("<bookstore/>");

Methods or functions of simpleXMLElement class

Function name	Description	syntax	Example
addChild()	The addChild() function adds a child element to a SimpleXMLElement object.	addChild(name,value);	\$book = \$bk->addChild("book");
addAttribute()	adds an attribute to a SimpleXMLElement object.	addAttribute(name,value);	\$book->addAttribute("Category", "Technical");
getName()	Returns the name of the XML tag referenced by the SimpleXMLElement object.	getName();	\$bk->getName();
asXML()	Returns a well-formed XML string (XML 1.0) from a SimpleXMLElement object	asXML([filename]);	echo \$bk->asXML();
children()	Returns the children of the specified node as an array.	children();	foreach (\$book->children() as \$child) { echo "Child node: " . \$child . " "; }
attributes();	Returns the attributes/values of element	attributes();	foreach (\$book->attributes () as \$k=>\$v) { echo \$k : \$v . " "; }
count();	The count() function counts the children of the specified node.	count();	\$cnt=\$book->count();
simplexml_load_file()	Converts an XML file into a SimpleXMLElement object	simplexml_load_file(filename);	\$xml=simplexml_load_file("note.xml");
simplexml_load_string()	The simplexml_load_string() function converts a well-formed XML string into a SimpleXMLElement object.		<?php \$note=<<<XML <note> <to>Tove</to> </note> XML; \$xml=simplexml_load_string(\$note);

Reading XML document

```
<?php  
$bk = simplexml_load_file("book.xml"); echo htmlspecialchars($bk->asXML());  
?>
```

- With SimpleXML, all the elements in XML document are represented as tree of SimpleXMLElement objects. Any given element's children are available as properties of elements SimpleXMLElement object.
- For example , We can access element name as properties \$book->title , \$book->publisher etc.

Consider an application that reads “Book.xml” file into simple XML object. Display attributes and elements.

```
//book .xml  
<?xml version='1.0' encoding='UTF-8'?>  
<bookstore>  
  <book category="Technical">  
    <title> LET US C </title>  
    <author> YASHWANT KANETKAR </author>  
    <year> 1980 </year>  
  </book>  
  <book category="Cooking">  
    <title> COOKING EVERYDAY </title>  
    <author> TARALA DALAL</author>  
    <year> 2000 </year>  
  </book>  
  <book category="YOGA">  
    <title> LIGHT ON YOGA </title>  
    <author> B.K.IYENGAR </author>  
    <year> 1990 </year>  
  </book>  
</bookstore>  
  
// book.php  
<?php  
$xml = simplexml_load_file("book.xml");  
  
echo $xml->getName() . "<br />"; foreach($xml->children() as $child)  
{  
  echo $child->getName() . "<br />"; foreach($child->attributes() as $k=>$v)  
  {  
    echo $k . "=" . $v . "<br />"; foreach($child->children() as $i=>$j)  
    {  
      echo $i . ":" . $j . "<br />";  
  
    }  
  }  
}  
?>
```

XSLT Introduction

- XSL (eXtensibleStylesheet Language) is a styling language for XML.
- XSLT stands for XSL Transformations.
- This tutorial will teach you how to use XSLT to transform XML documents into other formats (like transforming XML into HTML).
- XSLT is the most important part of XSL.
- XSLT is used to transform an XML document into another XML document, or another type of document that is recognized by a browser, like HTML and XHTML. Normally XSLT does this by transforming each XML element into an (X)HTML element.
- With XSLT you can add/remove elements and attributes to or from the output file. You can also rearrange and sort elements, perform tests and make decisions about which elements to hide and display, and a lot more.
- A common way to describe the transformation process is to say that **XSLT transforms an XML source-tree into an XML result-tree**.

```
?xml version="1.0"?>

<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
    <body>
      <h2>My CD Collection</h2>
      <table border="1">
        <tr bgcolor="#9acd32">
          <th>Title</th>
          <th>Artist</th>
        </tr>
        <xsl:for-each select="catalog/cd">
          <tr>
            <td><xsl:value-of select="title"/></td>
            <td><xsl:value-of select="artist"/></td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

LAB Exercise:

SETA:

- 1) Create a XML file which gives details of movies available in “Mayanagari CD Store” from following categories a) Classical b) Action c) Horror

Elements in each category are in the following format <Category>

<Movie Name>----</Movie Name>

<Release Year>----</Release Year>

</Category>

Save the file with name “movies.xml”.

- 2) Create a XML file which gives details of books available in “ABC Bookstore” from following categories.

- 1) Technical
- 2) Cooking
- 3) Yoga

and elements in each category are in the following format
<Book>

<Book_PubYear> -----</Book_PubYear>

<Book_Title> -----</Book_Title>

<Book_Author>-----</Book_Author>

</Book>

Save the file as “Book.xml”

- 3) Write a PHP script to create XML file named “Course.xml”

<Course><Computer
Science>

<Student name>.....</Student
name><Class name>.....</Class
name><percentage>.....</percentage></
Computer Science>

</Course>

Store the details of 5 students who are in TYBSc.

SET B:

- 1) Write PHP script to generate an XML code in the following format <?xml version="1.0" encoding="ISO-8859-1" ?>

```
<CD
Store><Movie>

<Title>Mr. India</Title>

<Release Year>1987</ Release Year
></Movie>

<Movie>

<Title>Holiday</Title>

<Release Year>2014</ Release Year
></Movie>

<Movie>

<Title>LOC</Title>

<Release Year>2003</ Release Year
></Movie>

</CD Store>
```

- 2) Write a script to create “cricket.xml” file with multiple elements as shown below:

```
<CricketTeam>
<Team country="India">

<player>____</player>

<runs>____</runs>
<wicket>____</wicket>

</Team>

</CricketTeam>
```

Write a script to add multiple elements in “cricket.xml” file of category, country=”Australia”.

- 3) Write a PHP script to accept an XML file which should comprise the following:

```
<cricket>

    <player>abc</player>
    <runs>1000</runs>

    <wickets>50</wickets>
    <noofnotout>10</noofnotout>

</cricket>
```

For at least 5 players. Display the details of players who have scored more than 1000 runs and at least 50 wickets.

SET C:

- 1) Write a PHP script to accept an XML file which should comprise the following :

```
<languages><lang
name="C"><appeared>1972</appeared
><creator>Dennis
Ritchie</creator></lang>

</languages>
```

For at least 5 records. Display the details of C language.

Assignment Evaluation

0: Not Done[]

1: Incomplete[]

2: Late Complete []

3: Needs Improvement[]

4: Complete []

5: Well Done []

Signature of the Instructor:-----Date:-----