आईएफटीएम विश्वविद्यालय, मुरादाबाद, उत्तर प्रदेश

IFTM University, Moradabad, Uttar Pradesh

NAAC ACCREDITED

E-Content

IFTM University, Moradabad

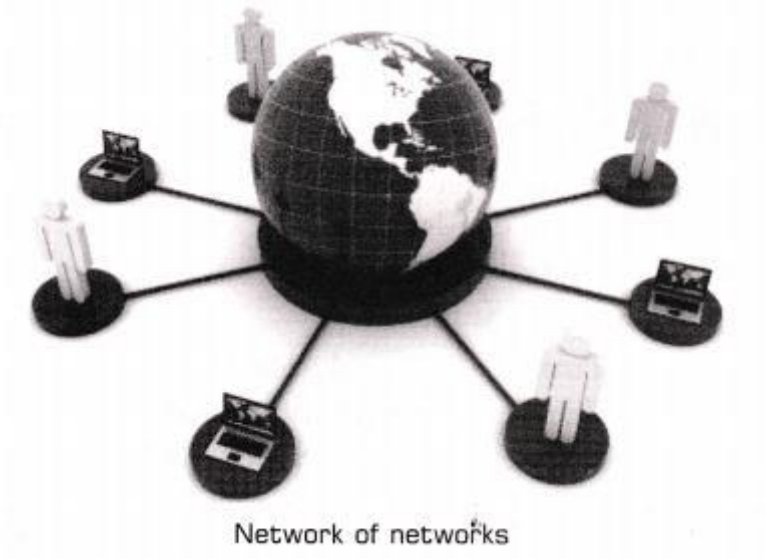
BCADS-116 Internet Basics & HTML

Unit-1

Internet

The Internet is an increasingly important part of everyday life for people around the world. The Internet is a global network of billions of computers and other electronic devices. With the Internet, it's possible to access almost any information, communicate with anyone else in the world, and do much more.

We can do all of this by connecting a computer to the Internet, which is also called **going online**. When someone says a computer is online, it's just another way of saying it's connected to the Internet.



Network of networks

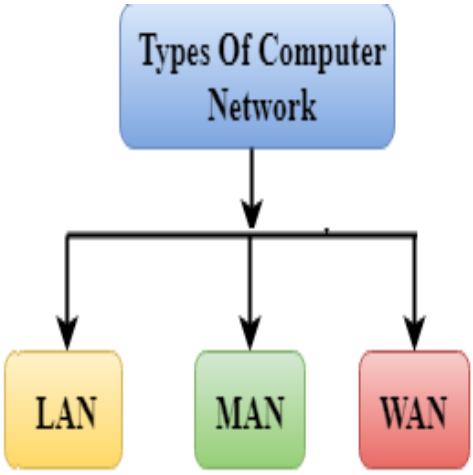
Sometimes referred to as a "network of networks," the Internet is **a vast network that connects computers all over the world**. Through the Internet, people can share information and communicate from anywhere with an Internet connection.

Network

A network is a group of computers linked to each other that enables the computer to communicate with another computer and share their resources, data, and applications. The devices can be connected by physical or wireless connections. The key is that there are at least two separate components, and they are connected.

A computer network can be categorized by their size. A computer network is mainly of three types:

- LAN (Local Area Network)
- MAN (Metropolitan Area Network)
- WAN (Wide Area Network)



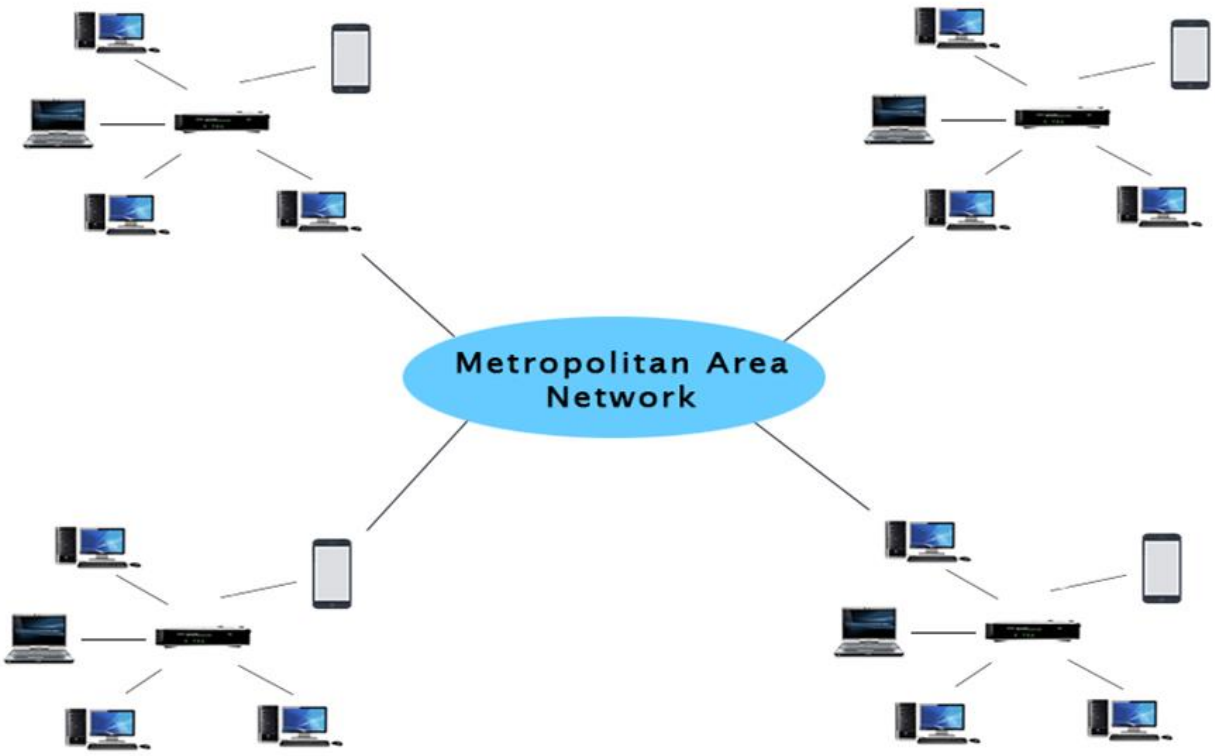
LAN (Local Area Network)

- Local Area Network is a group of computers connected to each other in a small area such as building, office.
- LAN is used for connecting two or more personal computers through a communication medium such as twisted pair, coaxial cable, etc.
- It is less costly as it is built with inexpensive hardware such as hubs, network adapters, and ethernet cables.
- The data is transferred at an extremely faster rate in Local Area Network.
- Local Area Network provides higher security.

MAN (Metropolitan Area Network)

A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network.

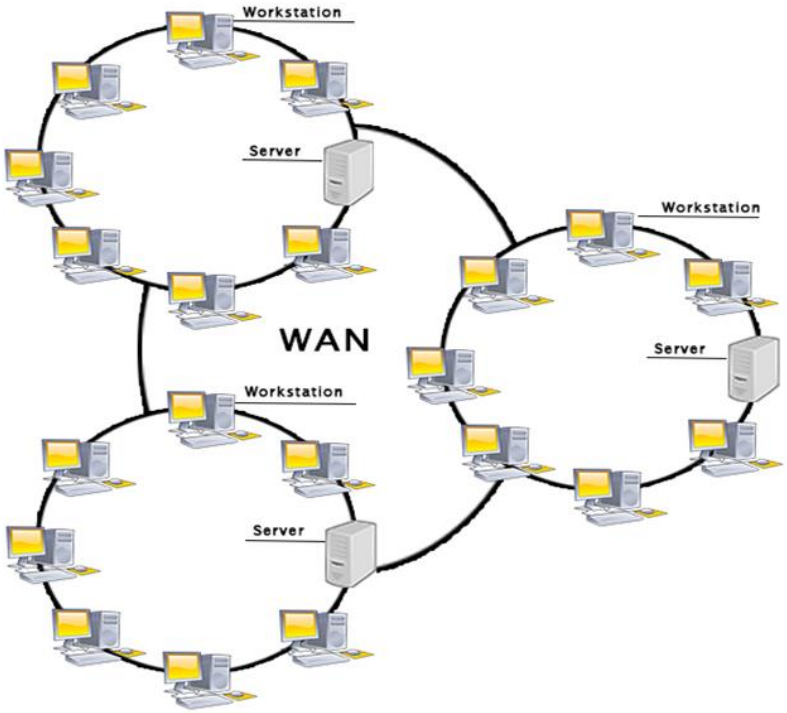
- Government agencies use MAN to connect to the citizens and private industries.
- In MAN, various LANs are connected to each other through a telephone exchange line.
- It has a higher range than Local Area Network (LAN).



WAN (Wide Area Network)

A Wide Area Network is a network that extends over a large geographical area such as states or countries.

- A Wide Area Network is quite bigger network than the LAN.
- A Wide Area Network is not limited to a single location, but it spans over a large geographical area through a telephone line, fibre optic cable or satellite links.
- The internet is one of the biggest WAN in the world.
- A Wide Area Network is widely used in the field of Business, government, and education.



History of Internet

Originally announced on **February 14, 1946**, the **Electronic Numerical Integrator and Computer** (**ENIAC**), was the first general-purpose electronic computer. After this many more computers were developed. Computers in the '60s were large and immobile. All were doing their respective tasks very well but they were not connected till date.

In order to make use of information stored in any one computer, one had to either travel to the site of the computer or have magnetic computer tapes sent through the conventional postal system.

Another catalyst in the formation of the Internet was the heating up of the Cold War. The Soviet Union's launch of the Sputnik satellite spurred the U.S. Defense Department to consider ways information could still be disseminated even after a nuclear attack.

The **Advanced Research Projects Agency (ARPA)**, an arm of the U.S. Defense Department, funded the development of the **Advanced Research Projects Agency Network (ARPANET)** in the late **1960s**. Its initial purpose was to link computers.

On **December 5, 1969**—the U.S. Department of Defense's **Advanced Research Projects Agency (ARPA)** connected four computer network nodes at the **University of California, Los Angeles, (U.C.L.A.),** the **Stanford Research Institute (S.R.I.)** in Menlo Park, Calif., **University of California, Santa Barbara (U.C.S.B.),** and the **University of Utah.**

ARPANET was a great success but membership was limited to certain academic and research organizations who had contracts with the Defense Department. In response to this, other networks were created to provide information sharing. Numerous new networks had emerged by the late **1970s**, including **CSNET** (Computer Science Research Network), **CDnet** (Canadian Network), **BITNET** (Because It's Time Network), and **NSFNET** (National Science Foundation Network).

What ARPANET could not do was talk to any of the other computing networks. So, in the spring of **1973**, **Vinton Cerf** and **Bob Kahn** began considering ways of connecting ARPANET with two other networks and named this project as **Internetworking-Project.**

One day, waiting in a hotel lobby, Cerf dreamed up a new computer communications protocol, a gateway between networks, which eventually became known as **Transmission-Control Protocol/Internet Protocol (TCP/IP)**. TCP/IP, which was first tested on ARPANET in 1977, was a way that one network could hand off data packets to another, then another, and another. Eventually, when the Internet consisted of a network of networks, Cerf's innovation would prove invaluable. It remains the basis of the modern Internet.

The term "**Internet**" was adopted in January 1, 1983, at about the same time that TCP/IP came into wide use. In 1983, ARPANET was divided into two parts, **MILNET**, to be used by military and defense agencies, and a civilian version of **ARPANET**. The word "Internet" was initially coined as an easy way to refer to the combination of these two networks, to their "internetworking."

Working of Internet

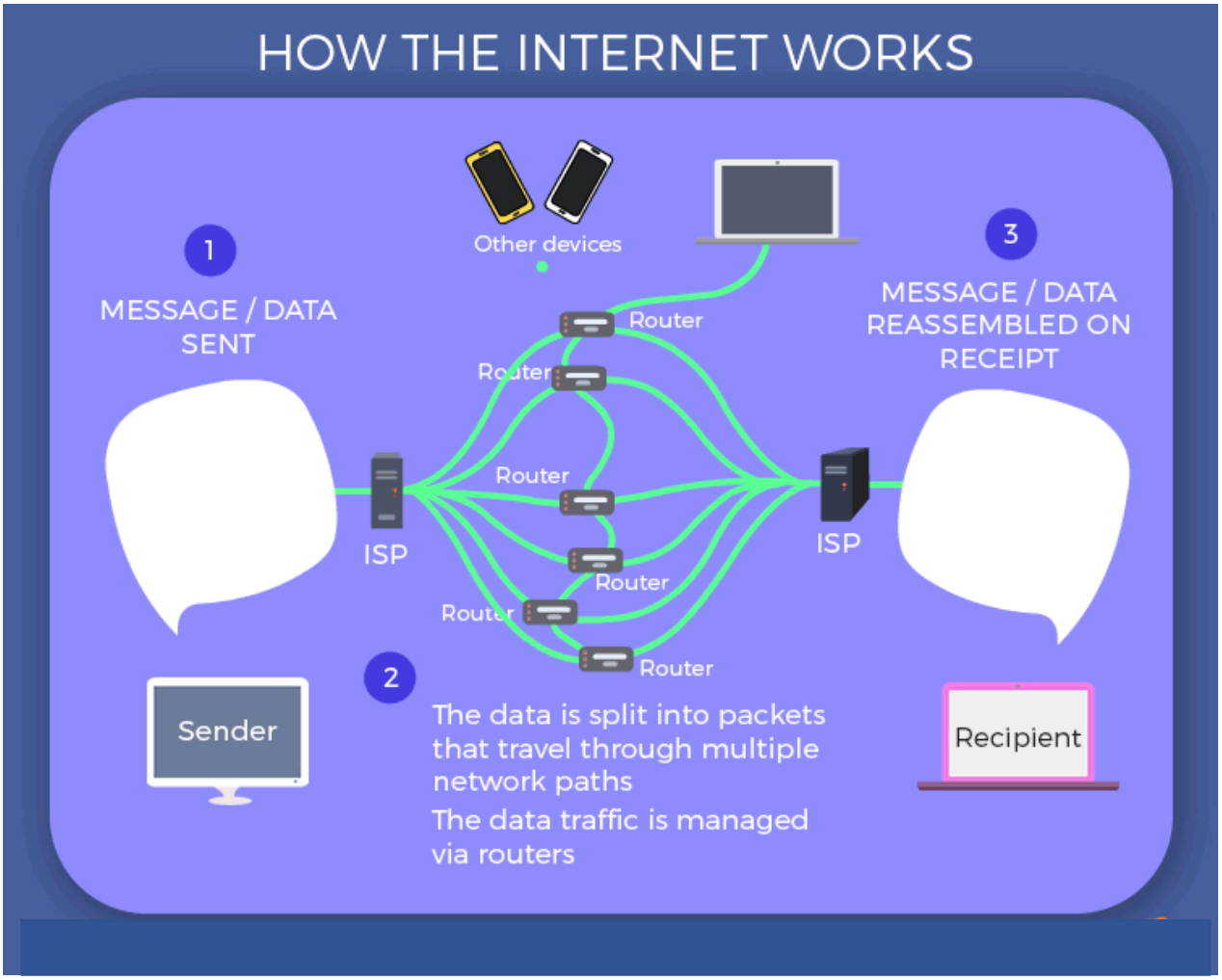
There are two main concepts that are fundamental to the way the Internet functions: packets and protocols.

Packets

In networking, a packet is a small segment of a larger message. Each packet contains both data and information about that data. The information about the packet's contents is known as the "header," and it goes at the front of the packet so that the receiving machine knows what to do with the packet. To understand the purpose of a packet header, think of how some consumer products come with assembly instructions.

When data gets sent over the Internet, it is first broken up into smaller packets, which are then translated into bits. The packets get routed to their destination by various networking devices such as **routers** and **switches**. When the packets arrive at their destination, the receiving device reassembles the packets in order and can then use or display the data.

Packets are sent across the Internet using a technique called **packet switching**. Intermediary routers and switches are able to process packets independently from each other, without accounting for their source or destination. This is by design so that no single connection dominates the network. If data was sent between computers all at once with no packet switching, a connection between two computers could occupy multiple cables, routers, and switches for minutes at a time. Essentially, only two people would be able to use the Internet at a time — instead of an almost unlimited number of people, as is the case in reality.



Protocols

Connecting two computers, both of which may use different hardware and run different software, is one of the main challenges that the creators of the Internet had to solve. It requires the use of communications techniques that are understandable by all connected computers, just as two people who grew up in different parts of the world may need to speak a common language to understand each other.

This problem is solved with standardized protocols. In networking, a protocol is a standardized way of doing certain actions and formatting data so that two or more devices are able to communicate with and understand each other.

There are protocols for sending packets between devices on the same network **(Ethernet)**, for sending packets from network to network **(IP)**, for ensuring those packets successfully arrive in order **(TCP)**, and for formatting data for websites and applications **(HTTP)**. In addition to these foundational protocols, there are also protocols for routing, testing, and encryption. And there are alternatives to the protocols listed above for different types of content — for instance, streaming video often uses UDP instead of TCP.

Because all Internet-connected computers and other devices can interpret and understand these protocols, the Internet works no matter who or what connects to it.

Each packet contains actual data and address part, i.e., addresses of destination and source upto 1500 characters. Functioning of TCP and IP are as follows:

- TCP-It breaks message into smaller packets that are transmitted over the Internet and also reassembles these smaller packets into the original message that are received from the Internet.
- IP-It handles the address part of each packet, so that the data is sent to the correct address. Each gateway on the network checks this address to see where to forward the message.

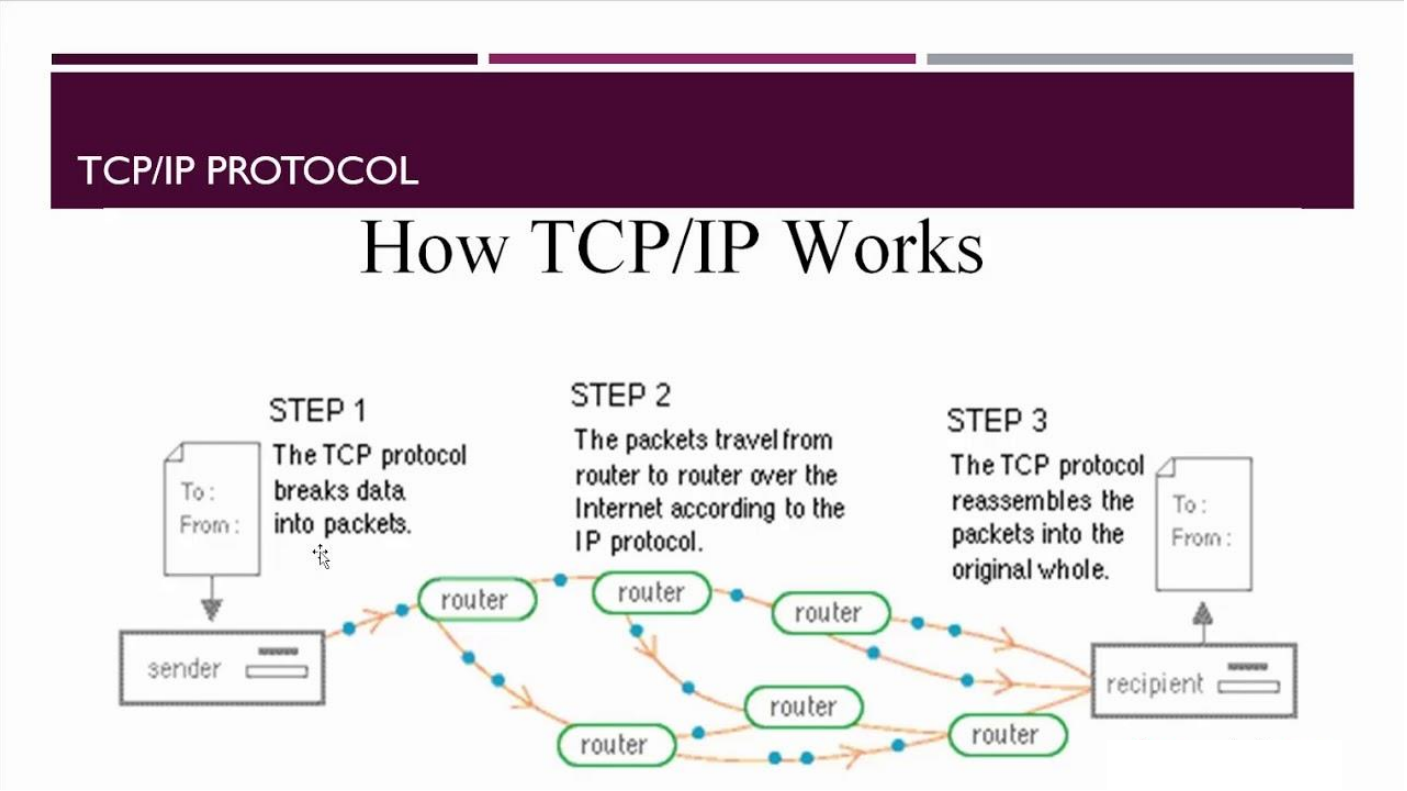
Internet Protocols

Networking makes the internet work, but neither can succeed without protocols. Common network protocols and functions are key for communication and connection across the internet. By protocols we mean some set of rules. Here are some important protocols for the functionating of Internet.

- **TCP- Transmission-Control Protocol**

TCP stands for Transmission Control Protocol. It is a transport layer protocol that facilitates the transmission of packets from source to destination. It is a connection-oriented protocol that means it establishes the connection prior to the communication that occurs between the computing devices in a network. This protocol is used with an IP protocol, so together, they are referred to as a TCP/IP.

The main functionality of the TCP is to take the data from the application layer. Then it divides the data into a several packets, provides numbering to these packets, and finally transmits these packets to the destination. The TCP, on the other side, will reassemble the packets and transmits them to the application layer. As we know that TCP is a connection-oriented protocol, so the connection will remain established until the communication is not completed between the sender and the receiver.



- **IP- Internet Protocol**

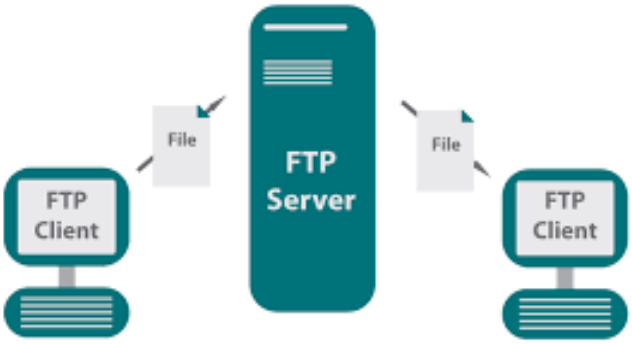
An IP address is a unique address that identifies a device on the internet or a local network. IP stands for "Internet Protocol," which is the set of rules governing the format of data sent via the internet or local network.

In essence, IP addresses are the identifier that allows information to be sent between devices on a network: they contain location information and make devices accessible for communication. The

internet needs a way to differentiate between different computers, routers, and websites. IP addresses provide a way of doing so and form an essential part of how the internet works. It handles the address part of each packet, so that the data is sent to the correct address. Each gateway on the network checks this address to see where to forward the message.

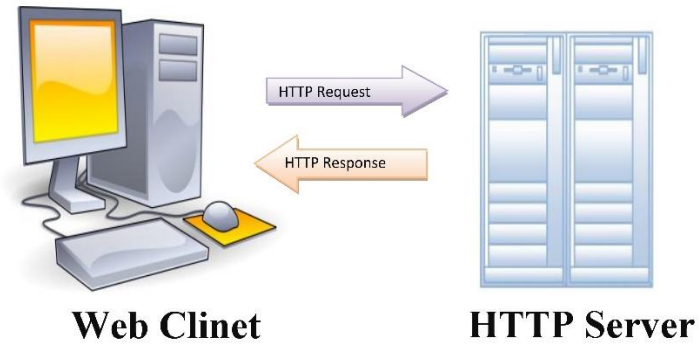
- **FTP- File Transfer Protocol**

  This was one of the first Internet services developed and it allows users to move files from one computer to another. Using the FTP program, a user can logon to a remote computer, browse through its files, and either download or upload files (if the remote computer allows).



- **HTTP- Hypertext Transfer Protocol**

  The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web, and is used to load web pages using hypertext links. It is a protocol used to access the data on the World Wide Web (www). The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on. This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.

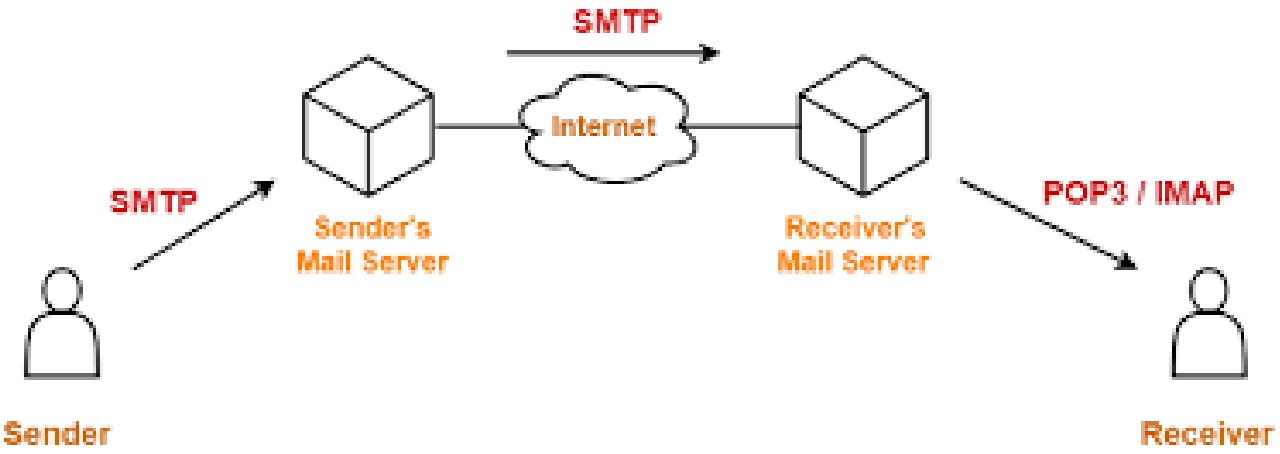


TCP/IP made the internet, and **HTTP** made the web.

There would be no web without HTTP and no internet without TCP and IP.

- **SMTP (Simple Mail Transfer Protocol)** is a TCP/IP protocol used in sending and receiving e-mail. However, since it is limited in its ability to queue messages at the receiving end, it is usually used with one of two other protocols, **POP3** (Post Office Protocol) or **IMAP (Internet Message Access Protocol)**, that let the user save messages in a server mailbox and download them periodically from the server. In other words, users typically use a program that uses SMTP for sending e-mail and either POP3 or IMAP for receiving e-mail.

SMTP works as a three-step process, using a client/server model. First, an e-mail server uses SMTP to send a message from an e-mail client, such as Outlook or Gmail, to an e-mail server. Second, the e-mail server uses SMTP as a relay service to send the e-mail to the receiving e-mail server. Third, the receiving server uses an e-mail client to download incoming mail via IMAP and place it in the inbox of the recipient.



Difference-

- ✓ POP3 will download all the emails to our system for us to view, and by doing so, all emails are removed from the mail server
- ✓ IMAP will send a copy of the emails to our system, but leaving the originals on our mail server.

- **MIME- Multipurpose Internet Mail Extensions**

Multipurpose Internet Mail Extension (MIME) is a standard that was proposed by Bell Communications in 1991 in order to expand the limited capabilities of email. MIME is a kind of add-on or a supplementary protocol that allows non-ASCII data to be sent through SMTP. It allows the users to exchange different kinds of data files on the Internet: audio, video, images, application programs as well.

MIME protocol is used to transfer e-mail in the computer network for the following reasons:

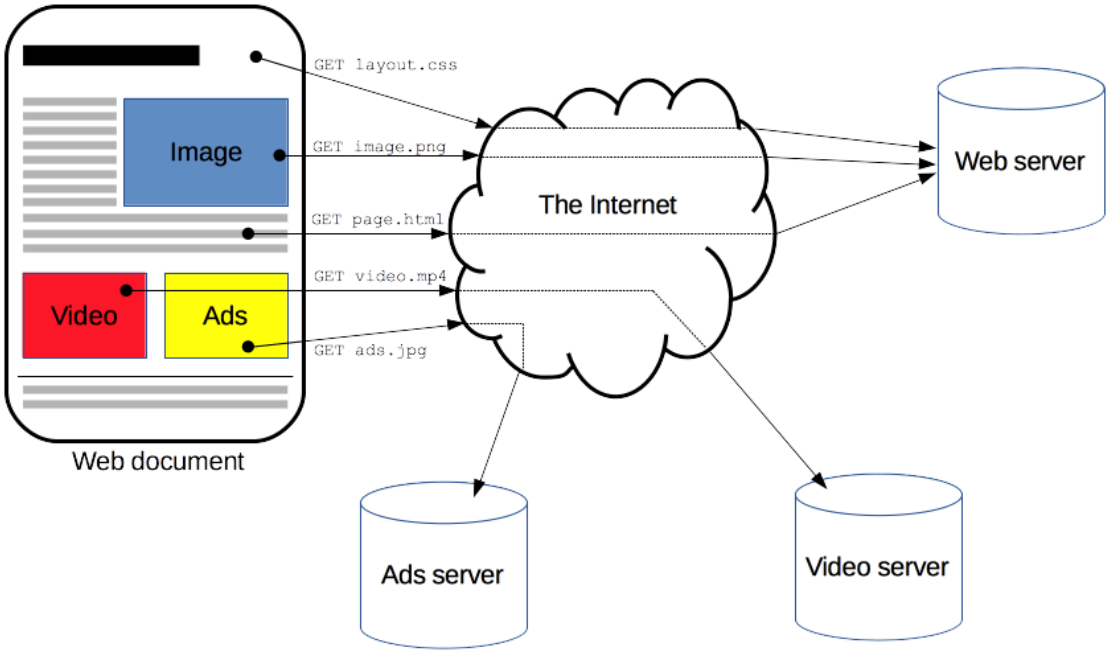
- ✓ The MIME protocol supports multiple languages in e-mail, such as Hindi, French, Japanese, Chinese, etc.
- ✓ Simple protocols can reject mail that exceeds a certain size, but there is no word limit in MIME.
- ✓ Images, audio, and video cannot be sent using simple e-mail protocols such as SMTP. These require MIME protocol.

World Wide Web- WWW

WWW stands for World Wide Web. A technical definition of the World Wide Web is: all the resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP). Basically, web is a large scale, **online store of information**. It is a system of **creating**, **organizing** and **linking** the documents.

Information is stored on WWW as a collection of documents that are **interconnected** with each other via **links**. The interconnected documents may be located on one or more than one computer, worldwide, thus, the name world wide web.

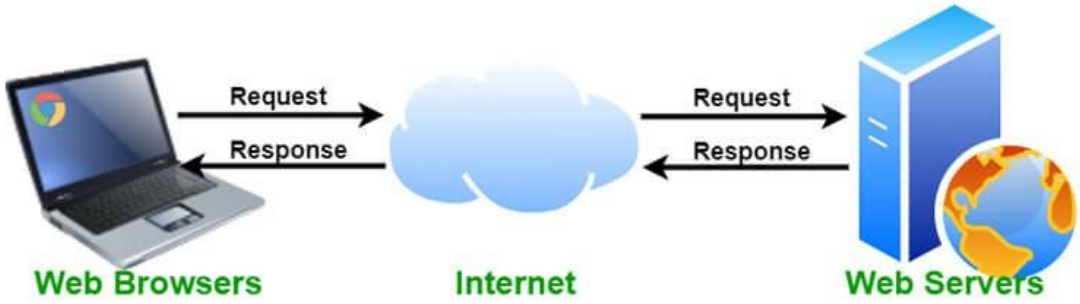
Internet and Web is not the same thing: Web uses internet to pass over the information.



- The documents on web are created in **hypertext format**. **Hypertext facilitates linking** of documents.
- The language used to create a hypertext format document is **HyperText Markup Language (HTML). HTML** allows the designers of the document to include text, pictures, videos, images, sound, graphics, movies etc, and also to link contents on the same document or different document using a **hyperlink**.
- The hypertext format document is transferred on the Web using **HyperText Transfer Protocol (HTTP).**
- A **single hypertext document** is called a **Web page.**
- A group of related web pages is called a **Web site.** A website displays related information on a specific topic.
- The first page or main page of a website is called **Homepage.**
- The web pages are stored on the Internet on the **Web Server. Web Servers are host computers that can store thousands of web pages.**
- The process of **storing a web page on a web server** is called **uploading.**
- The process of **retrieving a web page from a web server onto a user's computer** is **downloading.**
- The web pages stored on web server on the Internet, can be viewed from the user's computer using a software/tool called **Web Brower. Web Browser** is a software program that extracts information on user's request from the Internet and presents it as a web page to the user. It is also referred to as the user

interface of the web. **First web browser** was **Nexus**. Some of the popular web browsers are – **Internet Explorer** from Microsoft, **Mosaic** Brower, Google's **Chrome**, Mozilla's **Firefox**, **Netscape** Navigator from Netscape Inc etc.

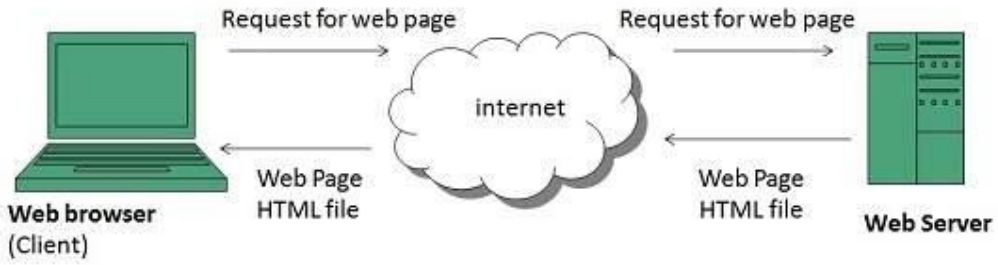
- The process of using browser to view information on the internet in known as **Browsing** or **Surfing**.
- Every web page is identified on Internet by its address, also called **Uniform Resource Locator (URL). URL** is the address on the Internet at which the web page resides. The user uses this address to get a web page from the Internet.



WWW Operation

WWW works on client- server approach. Following steps explains how the web works:

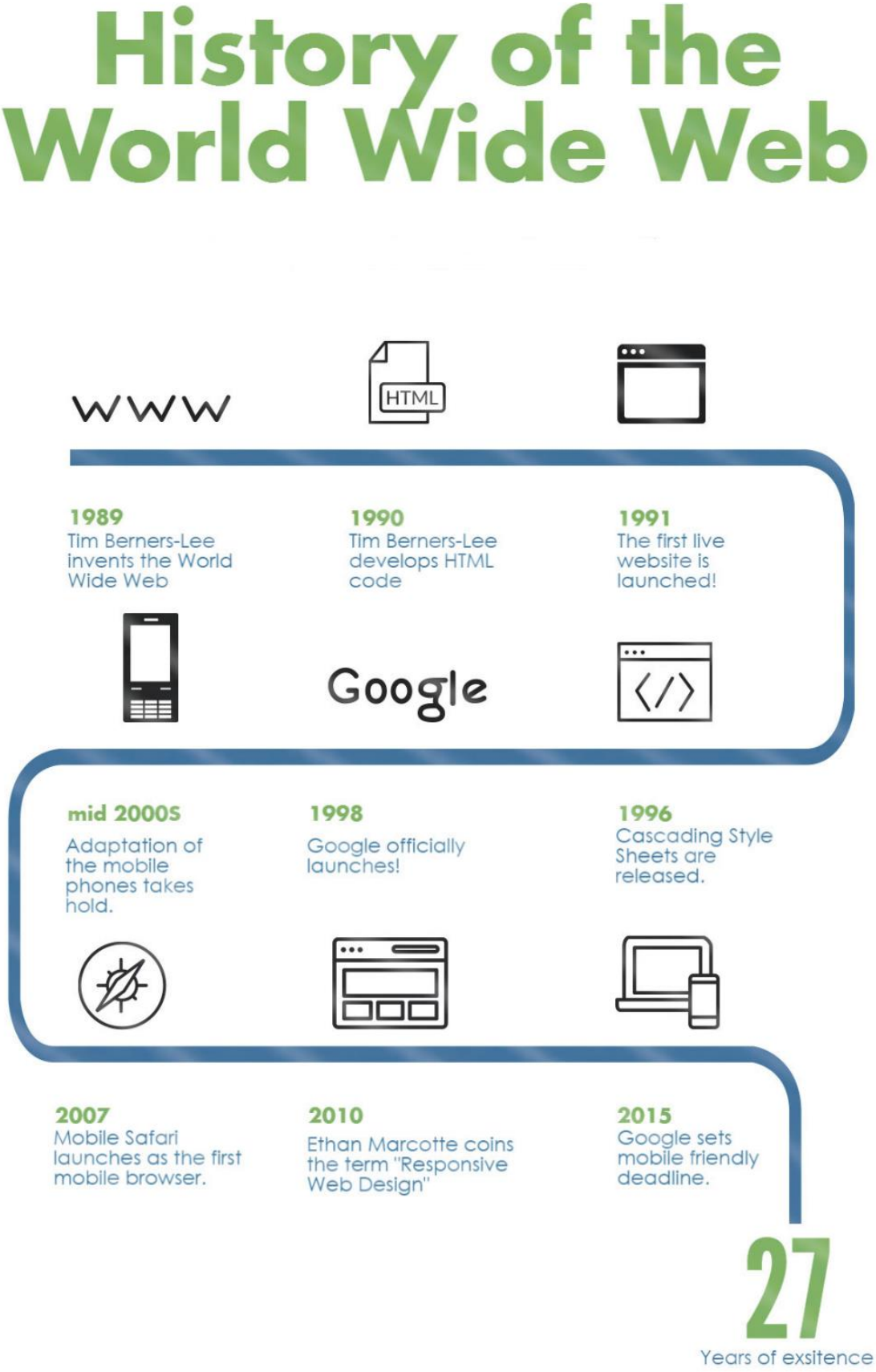
1. User enters the URL (say, **http://www.bigbrains.com**) of the web page in the address bar of web browser.
2. Then browser requests the Domain Name Server for the IP address corresponding to www.bigbrains.com.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, it interprets it and display the contents of web page in web browser's window.



Evolution of web

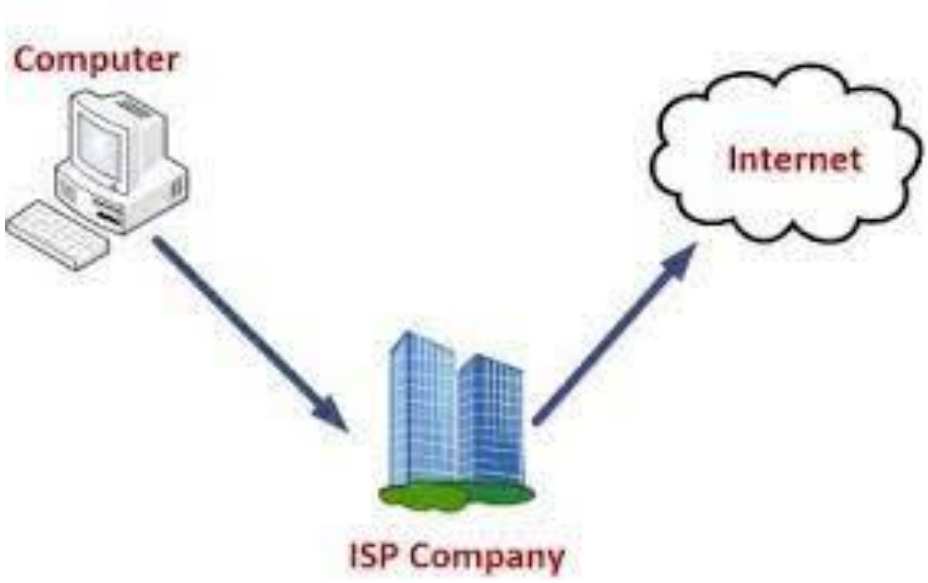
World Wide Web was created by **Tim Berners Lee** in **1989** at **CERN** in Geneva (Switzerland). World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at CERN. Eventually it became World Wide Web.

The following diagram briefly defines evolution of World Wide Web:

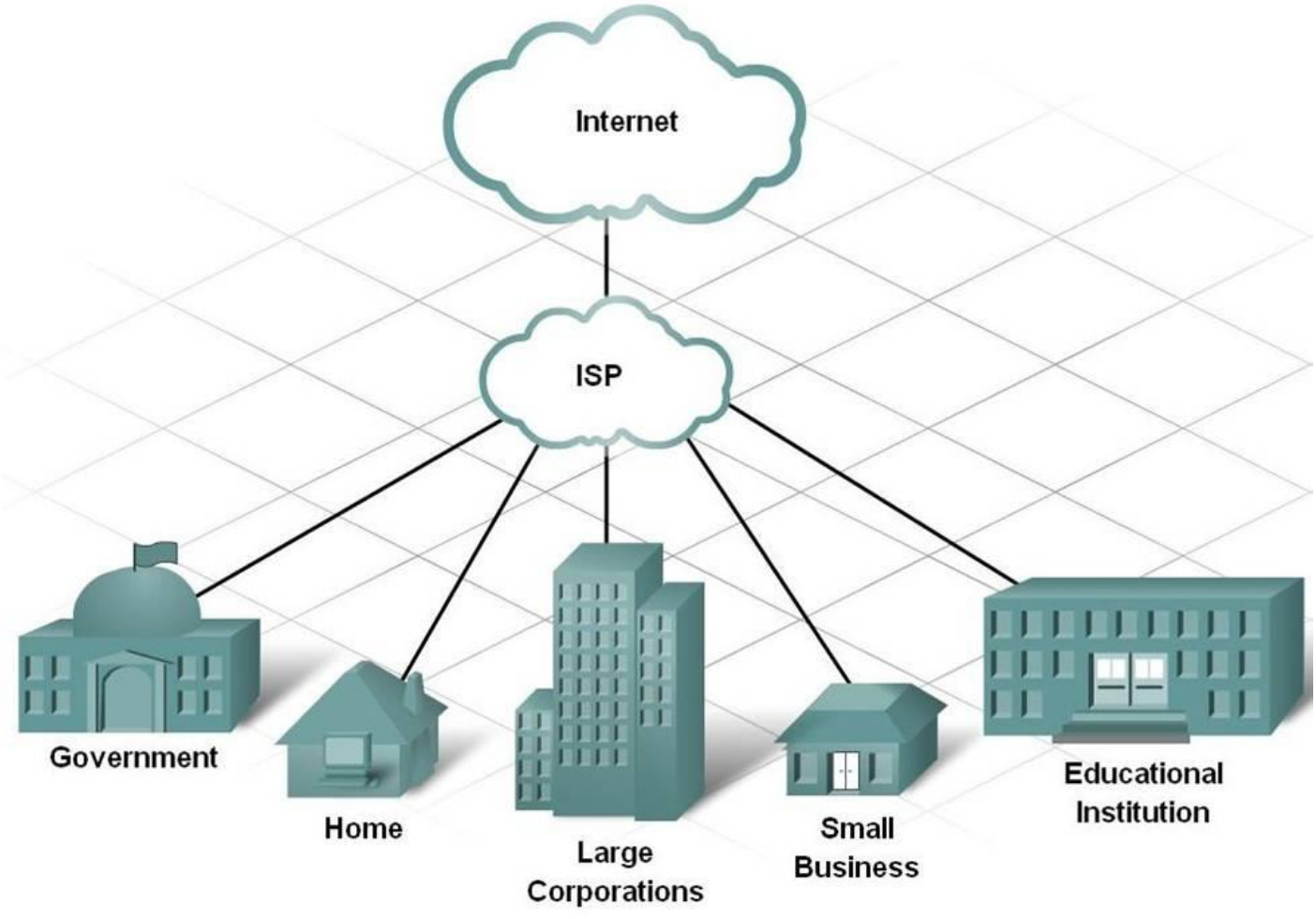


Internet Service Provider- ISP

Internet service provider (ISP) is an organisation that provides Internet connections and services to individuals and organizations. ISPs make it possible for their customers to surf the web, shop online, conduct business, and connect with family and friends—all for a fee.



An ISP has the equipment and the telecommunication line access required to have a point-of-presence on the Internet for the geographic area served.



Factors To Consider When Choosing an Internet Service Provider

For most businesses and even homes, internet access is just as important as their other utilities. Without internet access, some businesses simply can't operate. In addition to business needs, an increasing amount of home users are using the internet as their primary source of entertainment. Here are some important factors to consider when we choose an internet service provider (ISP) for our business or home.

- **Availability**

Before we figure out what kind of internet best suits our needs, we need to know what options are available in our location. If we live in a rural area, we will likely have fewer options than those who live in more urban areas. Typically, rural areas won't have as many providers offering cable or fiber-optic networks. The good news is that satellite internet (which is more widely available in rural areas) has dramatically improved in recent years.

- **Speed**

To some customers, speed is the most important factor when determining an ISP. They simply want the fastest internet they can get in their area. If you like to stream videos, you should be especially concerned about download and upload speed when choosing an ISP. That being said, you don't want to overdo it by purchasing extremely fast internet that you can't afford and that you probably won't need.

- **Cost**

Internet service providers can also charge you a wide range of prices for their services, depending on the type of service and where you are located. Only you can determine how much you're willing to spend on your Internet.

- **Type of Connection**

When it comes to the type of connection, you'll have a few different options. Each one has its pros and cons.

- **Dial-Up (Analog 56K).**

  Dial-up access is cheap but slow. A modem (internal or external) connects to the Internet after the computer dials a phone number.

- **DSL.** DSL stands for Digital Subscriber Line. It is an internet connection that is always "on". This uses 2 lines so your phone is not tied up when your computer is connected. There is also no need to dial a phone number to connect.
- **Cable.** Cable provides an internet connection through a cable modem and operates over cable TV lines. This is faster than DSL, but the speed will depend on how many people in your area are also using cable internet.
- **Wireless.** Wireless, or Wi-Fi, as the name suggests, does not use telephone lines or cables to connect to the internet. Instead, it uses radio frequency. Wireless is also an always on connection and it can be accessed from just about anywhere.
- **Satellite**—This is a widely available option, meaning if you're in an underserved market you'll probably still have this option. However, it's usually slower than other options.
- **Cellular.** Cellular technology provides wireless Internet access through cell phones. The speeds vary depending on the provider, but the most common are 3G and 4G speeds.

- **Reliability**

The best way to find out about the reliability of a certain internet service provider is to check out reviews online from customers in your area and see what their experience with that ISP has been.

It's important to remember that a provider alone cannot make their internet reliable. You have to also consider the type of internet you have and other outside factors such as the weather. For example, adverse weather conditions can mess with an internet connection. But if you have fiber-optic internet, the lines are buried underground, which usually means the connection will be strong regardless of the weather conditions.

E-mail (Electronic-Mail)

E-mail is defined as the transmission of messages on the Internet. It is one of the most commonly used features over communications networks that may contain text, files, images, or other attachments. It offers an efficient, inexpensive and real time mean of distributing information among people.

E-Mail Address

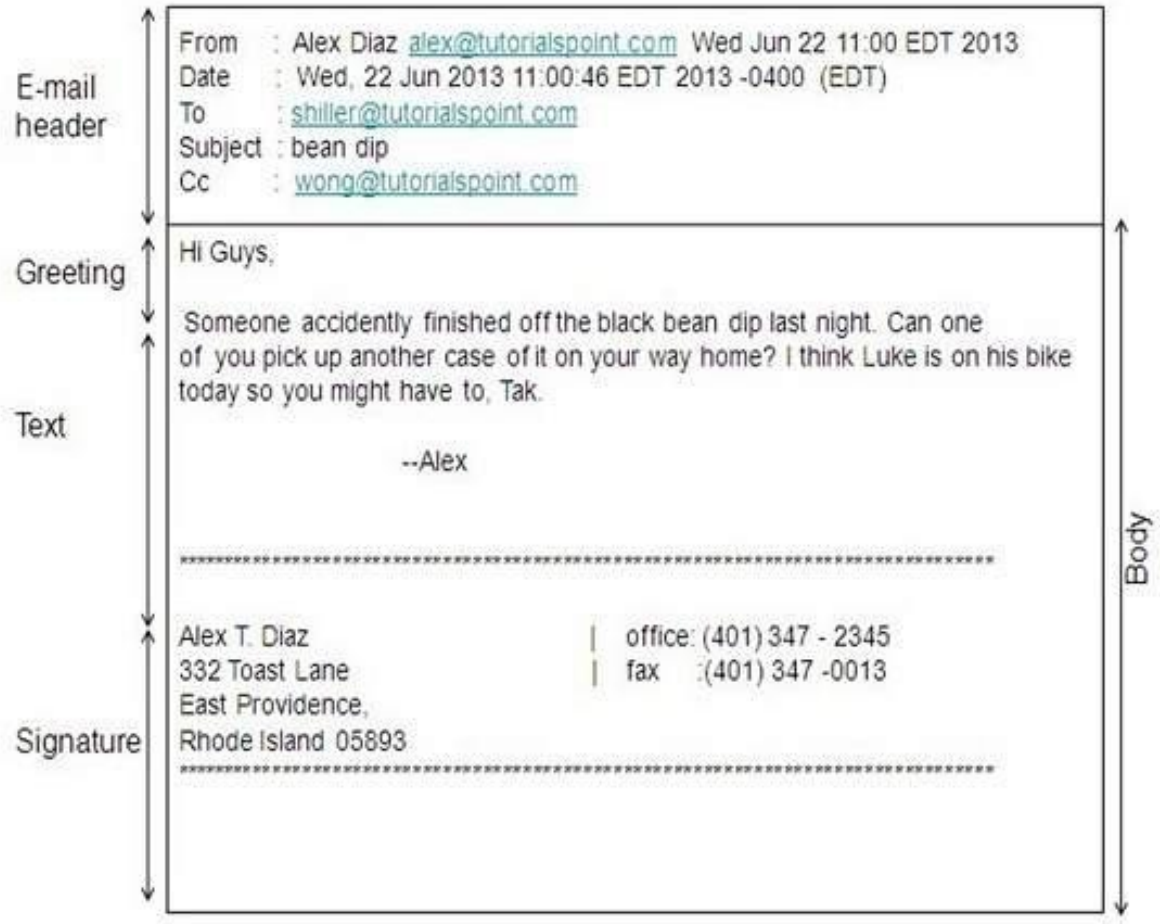
Each user of email is assigned a unique name for his email account. This name is known as E-mail address. Different users can send and receive messages according to the e-mail address.

E-mail is generally of the form username@domainname. For example, ritu@gmail.com is an e-mail address where ritu is username and gmail.com is domain name.

- The username and the domain name are separated by @ (at) symbol.
- E-mail addresses are not case sensitive.
- Spaces are not allowed in e-mail address.

E-mail Message Components

E-mail message comprises of different components: E-mail Header, Greeting, Text, and Signature. These components are described in the following diagram:



E-mail Header

The first five lines of an E-mail message is called E-mail header. The header part comprises of following fields:

- **From**
- **Date**
- **To**
- **Subject**
- **CC**
- **BCC**

From

The **From** field indicates the sender's address i.e., who sent the e-mail.

Date

The **Date** field indicates the date when the e-mail was sent.

To

The **To** field indicates the recipient's address i.e., to whom the e-mail is sent.

Subject

The **Subject** field indicates the purpose of e-mail. It should be precise and to the point.

CC

Cc stands for **carbon copy** which means that whose address appears after the Cc: header would receive a copy of the message. Also, the Cc header would also appear inside the header of the received message.

CC works in exactly the same way as adding more than one address to the 'To' field, each recipient will get the same copy of the email. Everyone in the To and CC fields will be able to see who else the email has been sent to.

BCC

Bcc stands for blind carbon copy which is similar to that of Cc except that the Email address of the recipients specified in this field do not appear in the received message header and the recipients in the To or Cc fields will not know that a copy sent to these addresses.

BCC recipient will receive a copy of the email, but other people in the To and CC fields won't know it's been sent to anyone else, whereas people in the BCC field will be able to see all the other recipients.

Greeting

Greeting is the opening of the actual message. e.g. Hi Sir or Hi Guys etc.

Text

It represents the actual content of the message.

Signature

This is the final part of an e-mail message. It includes Name of Sender, Address, and Contact Number.

Advantages

E-mail has proved to be powerful and reliable medium of communication. Here are the benefits of **E-mail:**

- **Reliable-**Many of the mail systems notify the sender if e-mail message was undeliverable.
- **Convenience-**There is no requirement of stationery and stamps. One does not have to go to post office. But all these things are not required for sending or receiving a mail.
- **Speed-**E-mail is very fast. However, the speed also depends upon the underlying network.
- **Inexpensive-**The cost of sending e-mail is very low.
- **Printable-**It is easy to obtain a hardcopy of an e-mail. Also, an electronic copy of an e-mail can also be saved for records.
- **Global-**E-mail can be sent and received by a person sitting across the globe.
- **Generality-**It is also possible to send graphics, programs and sounds with an e-mail.

URL (Uniform Resource Locator)

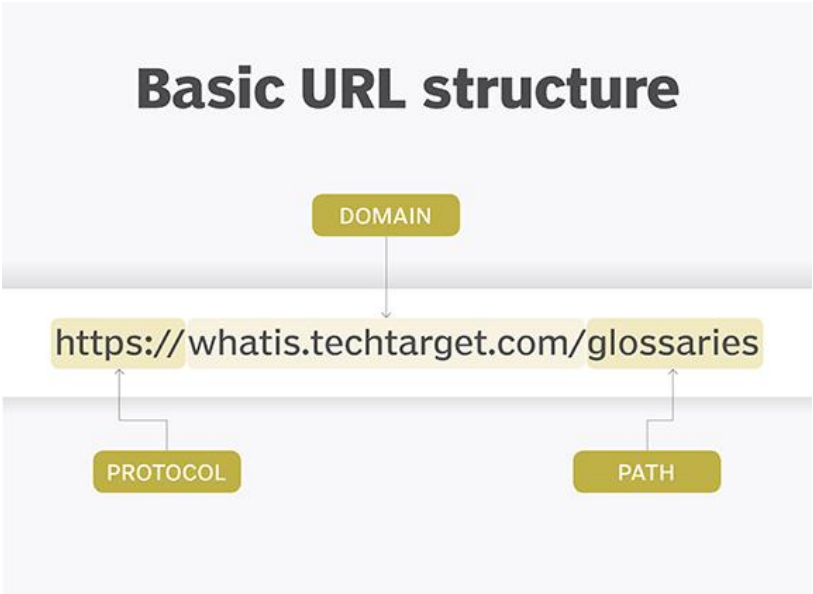
Uniform Resource Locator (URL) is another name for a web address. URLs are made of letters, numbers and other symbols in a standard form. People use them on computers, to make the computer fetch and show some specific resource (usually a web page) from another computer (web server) on the Internet.

URLs consist of multiple parts -- including a **protocol** and **domain name** -- that tell a web browser how and from where to retrieve a resource.

URL structure

The URL contains the name of the protocol needed to access a resource, as well as a resource name. The first part of a URL identifies what protocol to use as the primary access medium. The second part identifies the IP address or domain name -- and possibly subdomain -- where the resource is located.

URL protocols include HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure) for web resources, mail to for email addresses, FTP for files on a File Transfer Protocol (FTP) server, and telnet for a session to access remote computers. Most URL protocols are followed by a colon and two forward slashes; "mail to" is followed only by a colon.



Examples of URLs:

https://developer.mozilla.org

https://developer.mozilla.org/en-US/docs/Learn/

https://developer.mozilla.org/en-US/search?q=URL

Parts of a URL

Protocol

The protocol declares how your web browser should communicate with a web server when sending or fetching a web page or document. The most common protocol is http which stands for Hypertext Transfer Protocol. Another common protocol is https which stands for Hypertext Transfer Protocol Secure.

Domain name

A domain name is a unique reference that identifies a web site on the internet, for example bigbrains.com.

Path

The path typically refers to a file or location on the web server, e.g. /directory/file.php.

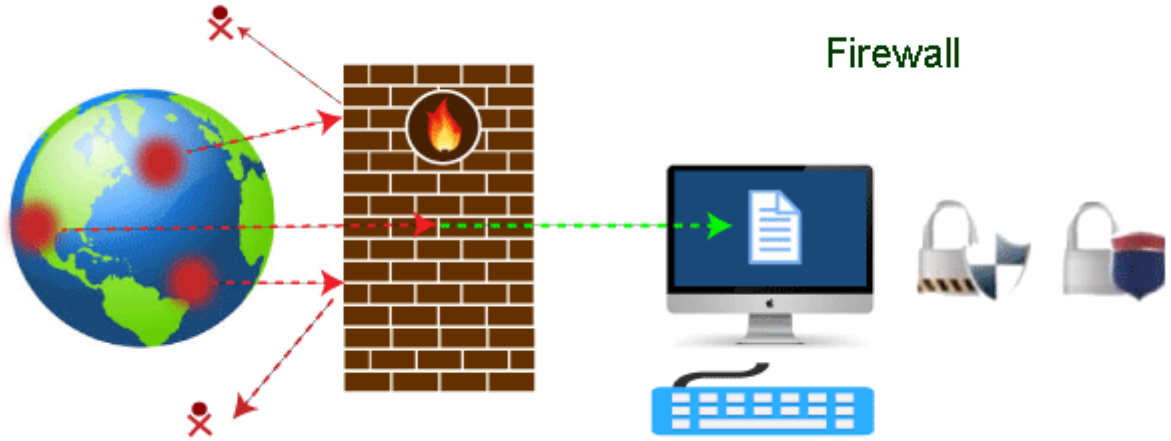
HTTP vs. HTTPs

Both HTTP and HTTPS are used to retrieve data from a web server to view content in a browser. The difference between them is that HTTPS uses a Secure Sockets Layer (SSL) certificate to encrypt the connection between the end user and the server.

Firewall

Nowadays, it is a big challenge to protect our sensitive data from unwanted and unauthorized sources. There are various tools and devices that can provide different security levels and help keep our private data secure. One such tool is a 'firewall' that prevents unauthorized access and keeps our computers and data safe and secure.

A firewall can be defined as a special type of network security device or a software program that monitors and filters incoming and outgoing network traffic based on a defined set of security rules. It acts as a barrier between internal private networks and external sources (such as the public Internet).



The primary purpose of a firewall is to allow non-threatening traffic and prevent malicious or unwanted data traffic for protecting the computer from viruses and attacks.

Firewall: Hardware or Software

This is one of the most problematic questions whether a firewall is a hardware or software. As stated above, a firewall can be a network security device or a software program on a computer. This means that the firewall comes at both levels, i.e., hardware and software, though it's best to have both.

Each format (a firewall implemented as hardware or software) has different functionality but the same purpose.

A hardware firewall is a physical device that attaches between a computer network and a gateway.

On the other hand, a software firewall is a simple program installed on a computer that works through port numbers and other installed software.

Why Firewall

Firewalls are primarily used to prevent malware and network-based attacks. Additionally, they can help in blocking application-layer attacks. These firewalls act as a gatekeeper or a barrier. They monitor every attempt between our computer and another network. They do not allow data packets to be transferred through them unless the data is coming or going from a user-specified trusted source.

Firewalls are designed in such a way that they can react quickly to detect and counter-attacks throughout the network. They can work with rules configured to protect the network and perform quick assessments to find any suspicious activity. In short, we can point to the firewall as a traffic controller.

How Firewall Works

Firewall match the network traffic against the rule set defined in its table. Once the rule is matched, associate action is applied to the network traffic. For example, Rules are defined as any employee from HR department cannot access the data from code server and at the same time another rule is defined like system administrator can access the data from both HR and technical department. Rules can be defined on the firewall based on the necessity and security policies of the organization.

Firewall Benefits

1. Monitors Network Traffic
2. Stops Virus Attacks
3. Prevents Hacking
4. Stops Spyware
5. Promotes Privacy

Domain Name System (DNS)

The Domain Name System (DNS) is the **Internet's system** for **mapping alphabetic names** to **numeric Internet Protocol (IP) addresses** like a phone book maps a person's name to a phone number.

A domain name is your website name. A domain name is the address where Internet users can access your website. A domain name is used for finding and identifying website on the Internet. Computers use IP addresses, which are a series of number. However, it is difficult for humans to remember strings of numbers. Because of this, domain names were developed and used to identify entities on the Internet rather than using IP addresses. The domain name must be registered before you can use it. Every domain name is unique. No two websites can have the same domain name.

A domain name takes the form of two main elements. For example, the domain name **Facebook.com** consists of the website's name (Facebook) and the domain name extension (.com).

Requirement

Every host is identified by the IP address but remembering numbers is very difficult for the people and also the IP addresses are not static therefore a mapping is required to change the domain name to IP address. So, DNS is used to convert the domain name of the websites to their numerical IP address.

Domain:

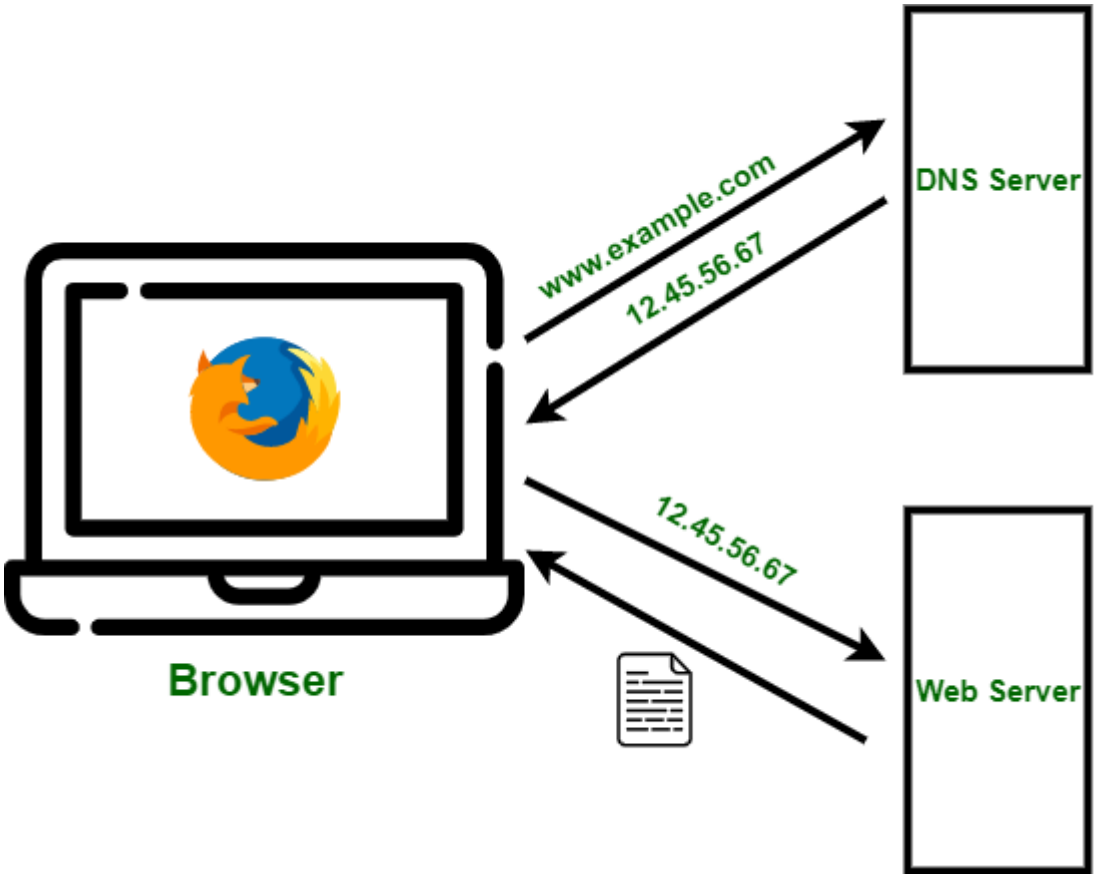
There are various kinds of DOMAIN:

- **Generic domain**: .com(commercial) .edu(educational) .mil(military) .org(non profit organization) .net(similar to commercial) all these are generic domain.
- **Country domain** .in (india) .us .uk
- **Inverse domain** if we want to know what is the domain name of the website. Ip to domain name mapping. So DNS can provide both the mapping for example to find the ip addresses of bigbrains.com then we have to type nslookup www.bigbrains.com

Working of Domain Name System (DNS) Server

DNS resolves names to numbers, to be more specific it resolves domain names to IP addresses. So if you type in a web address in your web browser, DNS will resolve the name to a number because the only thing computers know are numbers.

If you wanted to go to a certain website you would open up your web browser and type in domain name of that website. Let us use google.com. Now technically you really do not have to type in google.com to retrieve Google web page, you can just type in IP address instead if you already know what google's IP address is, but since we are not accustomed to memorizing and dealing with numbers, especially when there are millions of websites on Internet, we can just type in domain name instead and let DNS convert it to an IP address for us.



So back to our example, when you type google.com on your web browser DNS server will search through its cache to find a matching IP address for that domain name, and when it finds it it will resolve that domain name to IP address of Google web site, and once that is done then your computer is able to communicate with a Google web server and retrieve the webpage.

BCADS-116 Internet Basics & HTML

Unit-2

Basic principles involved in developing a web site

When designing a website there are many key factors that will contribute to how it is perceived. effective web design is judged by the users of the website and not the website owners. There are many factors that affect the usability of a website, and it is not just about form (how good it looks), but also function (how easy is it to use). Below we explore the top 9 web design principles that will make your website aesthetically pleasing, easy to use, engaging, and effective.

1. **Website Purpose**

   Good web design always caters to the needs of the user. Each page of your website needs to have a clear purpose, and to fulfill a specific need for your website users in the most effective way possible.

2. **Simplicity**

   Simplicity is the best way to go when considering the user experience and the usability of your website. The over-designed website may not work. Putting too many elements on the page may lead to distracting visitors from the main purpose of your website. Simplicity always works in an effective web page design. Keep your design as simple as possible so that the visitors can feel it easy-to-use and can find their ways easily.

3. **Colour**

   Colour has the power to communicate messages and evoke emotional responses. Finding a colour palette that fits your brand will allow you to influence your customer's behaviour towards your brand. Keep the colour selection limited to less than 5 colours. Complementary colours work very well. Pleasing colour combinations increase customer engagement and make the user feel good.

4. **Typefaces & Font**

   No matter how good your design is text still rules the website as it provides users the desired information. Typefaces generally means a particular design of type. In general, Sans Serif fonts such as Arial and Verdana are easier to read online (Sans Serif fonts are contemporary looking fonts without decorative finishes). The ideal font size for reading easily online is 16px and stick to a maximum of 3 typefaces in a maximum of 3-point sizes to keep your design streamlined.

5. **Imagery**

   Imagery is every visual aspect used within communications. This includes still photography, illustration, video and all forms of graphics. All imagery should be expressive and capture the spirit of the company. Most of the initial information we consume on websites is visual and as a first impression it is important that high quality images are used to form an impression.

6. **Readability**

   Consider using fonts that are easier to read. The modern sans-serif fonts as Arial, Helvetica, etc. can be used for the body texts. Make proper combinations of typefaces for each and every design element such as headlines, body texts, buttons, etc.

7. **Navigation**

   Navigation is about how easy it is for people to take action and move around your website. Website navigation is key to retaining visitors. If the websites navigation is confusing visitors will give up and find what they need elsewhere. Keeping navigation simple, intuitive and consistent on every page is key.

8. **Content**

   An effective web design has both great design and great content. Using compelling language great content can attract and influence visitors by converting them into customers.

9. **Load Time**

   Everybody hates a website that takes ages to load. Waiting for a website to load will lose visitors. Nearly half of web visitors expect a site to load in 2 seconds or less and they will potentially leave a site that isn't loaded within 3 seconds.

10. **Mobile Friendly (Responsive Web Design)**

    More people are using their phones or other devices to browse the web. It is important to consider building your website with a responsive layout where your website can adjust to different screens.

Planning Process in Website Development

Planning a website can take a lot of time and effort. It can also cost money to launch a website. The two most important factors that need to be taken into consideration are time and money. During the planning process, it is a good idea to make a timeline or have a goal of how long you might need to spend on each step. The more time that is taken during the first few steps of planning, the less you will have to do in the long run.

Before launching a website, the planning process is vital to ensuring everything goes smoothly down the road. In this day and age, there are so many sites out there on the internet. It is almost impossible for a business to do well without having a website that works well for both the company and its clients. Interestingly enough, there is a lot to be done before the coding of a website takes place. There are many steps which need to be followed for it to be properly developed. The purpose of development planning is to make sure nothing is forgotten.

There are seven steps that should be followed from start to finish when developing a website.

1. **Research and goal setting**
2. **Planning**
3. **Designing the layout**
4. **Writing the content**
5. **Coding**
6. **Testing and launching**
7. **Maintaining**

1. **Research and Goal Setting**

   It is important to do proper research and set goals before beginning. By setting goals, it will help the website to have a direction and will also help your business to achieve specific accomplishments. The planning and goal setting process could take about 1-2 weeks to complete. It is a very important first step to creating a website that sells. There are a few questions you should ask yourself during this phase:

   - What do I hope for my website to accomplish?
   - Who is the audience I would like to target?
   - What are the main goals of the website?

   By setting goals for your website, you will be helping the site to have a clear direction and purpose. This is important to the rest of the steps.

2. **Planning the Site**

   Planning the website involves creating a wireframe and sitemap. This is an important step because it is kind of like the skeleton of your site. This process can take about 2-6 weeks to complete. The sitemap allows the developer to get an outline of what the site will look like, what pages there will be and how they will interact with each other. This not only helps with planning but is also beneficial to the user experience.

   A user should be able to easily navigate a site, and this begins with the development of the sitemap. Before you begin to plan content, a sitemap lets you design what the structure will look like.

3. **Designing the Layout**

   The details of the layout are what will give your website character. This is the step where you get to be creative with pictures, videos and what kinds of things the customer will notice when they come to your site. This process can take about 4-12 weeks from start to finish. The timing depends on experience, time spent on the project, and how thorough the developer is. During this step, it is especially important to keep referring back to the target audience you wish to focus on.

   Consider colors, logos, and anything that will encourage your audience to interact with the site. By considering how you will create the layout of the website, you are attempting to bring the website to life.

4. **Writing the Content**

   This step may be going on simultaneously with the other development planning steps. The written content of a website is so important to its success. While this step may be happening during other steps, it is one that is crucial and deserves a lot of expertise. It could take from 5 to 15 weeks. When determining what words to use, it is important that they are not too hard to understand. A general rule is that you have to assume not everyone is going to want to read words that are higher vocabulary. A website should have a vocabulary that the average person can understand.

5. **Coding the Website**

   Once the website is laid out according to the sitemap, it should be tested before moving any further. If all works well, then the rest of the content should be added, and formatting should be completed. This phase involves having a deep understanding of the technology you are using. In fact, if you are looking to do most of the work yourself, you should at least consider getting a developer to code for you, so that you can make sure everything works as planned.

6. **Testing and Launching**

Before the website is launched, it is crucial that it is tested out by real users. All the links and content should be tested to see if it works. Not only is it important to test out all the buttons and everything on the site, but it is also important to test out what users think of it. There is user testing that can be completed to make sure the website is giving users what they need to be successful on the test.

Again, there are tools that can be used to determine if anything needs to be changed. Make sure to check all written content, including spelling and grammar. If your website has forms, ensure that they are working correctly as well. These might be important ways the users can get in touch with you or sign up for alerts and messages. Without these working properly, it can be very difficult for the user and also will be difficult for you to have a successful website.

Don't just check the website once, but check it over multiple times. When you are confident that everything is in working order, you can go ahead and launch your website live. When it comes to launching, you are finally ready and can do this by uploading it to the server. You will need FTP (File Transfer Protocol) for this process. It is also important to make sure everything is running smoothly immediately after launching. Testing and launching may take 2-4 weeks to complete.

7. **Maintenance**

You might think your job is done once the website is launched, but this is not the case. Since technology and products are changing more rapidly than ever before, it is important to stay up-to-date with what is happening on the internet. Maintaining a website is hard work, but the more effort put into its maintenance, the better. The last step involves updating and doing maintenance on the website often, even after it has been launched to ensure it is still working properly.

Design Concept

A design concept is the core idea driving the **design of a product**, explained via a collection of **sketches**, **images**, and a **written statement**. This helps the designers and, later, the developers stay on track throughout the creative process, ensuring they bring a product to market with value to target users.

An effective design concept makes the goal of the product explicit and serves as the foundation upon which the product is built. Developing a design concept demands a clear understanding of the type of problems to be solved, the ideal aesthetic style, the target audience, and the client's needs (if working on an external project).

Investing time into creating a solid design concept reduces the risk of running into dead ends once work is underway, and of investing money into a project without a clear purpose.

It is the idea behind a design. It's the underlying logic, thinking, and reasoning for how you'll design a website. Your concept will lead to your choices in color and type. It'll choose your aesthetic and determine your grid. Every design decision you make will fall back on your concept for direction.

Your design concept becomes the framework for all your design decisions.

We can think of design concepts in two ways.

- **Verbal** – the verbal parts of your concept might be words you use to describe the site. For example, your design concept might be one of sophisticated elegance. Verbal concepts tend toward the abstract. They're focused on the message your design is to communicate.
- **Visual** – the visual parts of your concept might be a specific image or color scheme. It might be an idea to use circles prominently. Visual concepts tend to be a little more concrete. They should come from the verbal part of your concept. Visual concepts are focused more on the how of conveying your message.

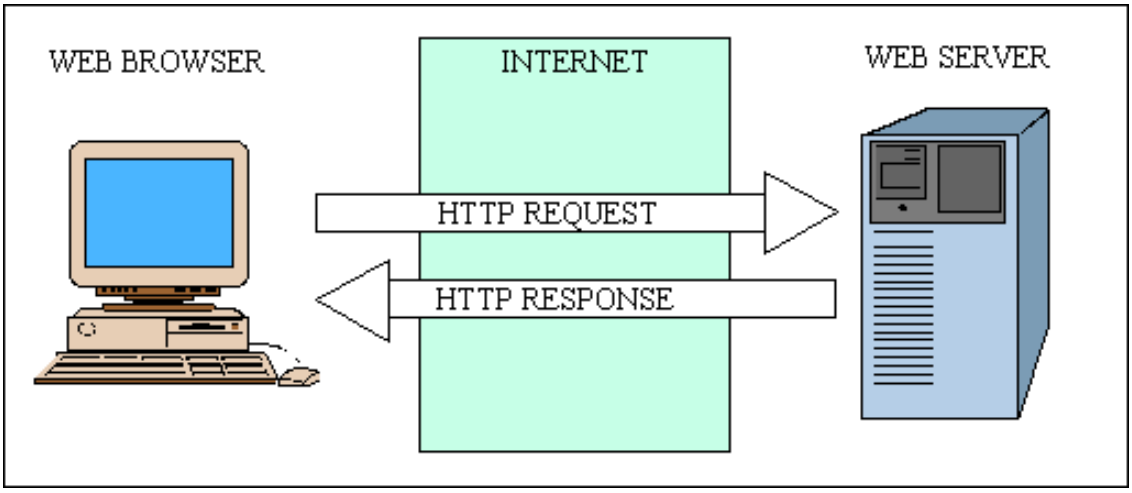
Generally verbal concepts come before visual concepts as the visual is really about how you'll communicate the verbal, though it likely depends on the individual and how you think best.

Web Browser

Web Browser is an application software that allows us to view and explore information on the web. User can request for any web page by just entering a URL into address bar.

Web browser can show text, audio, video, animation and more. It is the responsibility of a web browser to interpret text and commands contained in the web page.

Earlier the web browsers were text-based while now a days graphical-based or voice-based web browsers are also available. Following are the most common web browser available today:



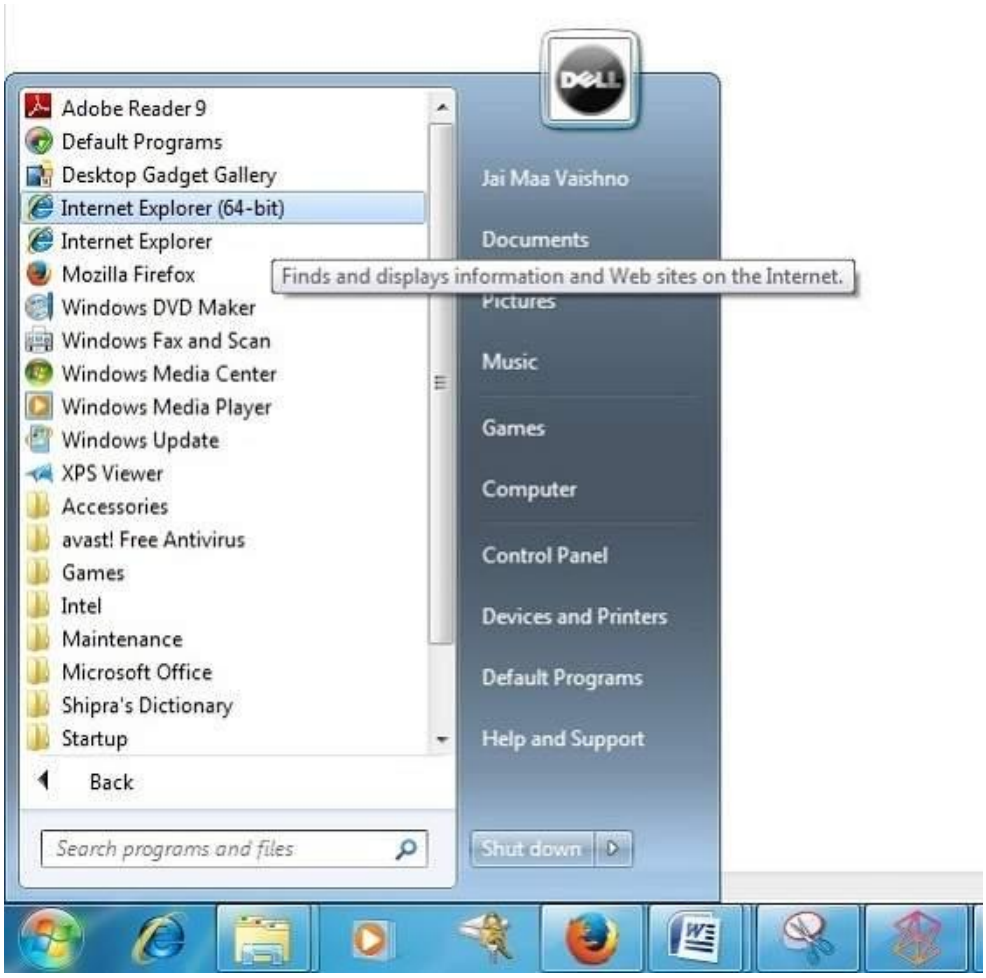
Browser	Vendor
Internet Explorer	Microsoft
Google Chrome	Google
Mozilla Firefox	Mozilla
Netscape Navigator	Netscape Communications Corp.
Opera	Opera Software
Safari	Apple
Sea Monkey	Mozilla Foundation

Starting Internet Explorer

Internet explorer is a web browser developed by Microsoft. It is installed by default with the windows operating system however, it can be downloaded and be upgraded.

To start internet explorer, follow the following steps:

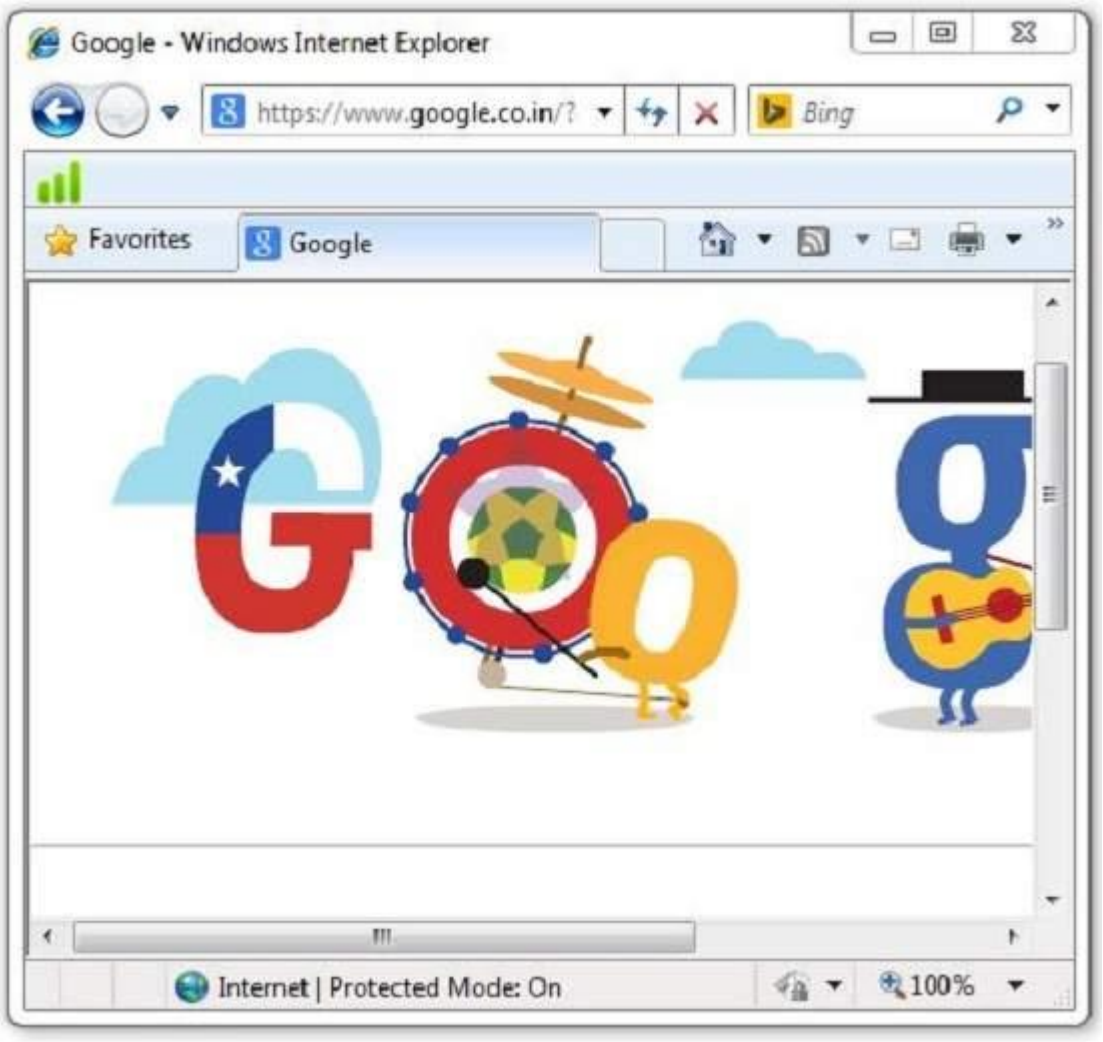
- Go to **Start** button and click **Internet Explorer.**



The **Internet Explorer** window will appear as shown in the following diagram:

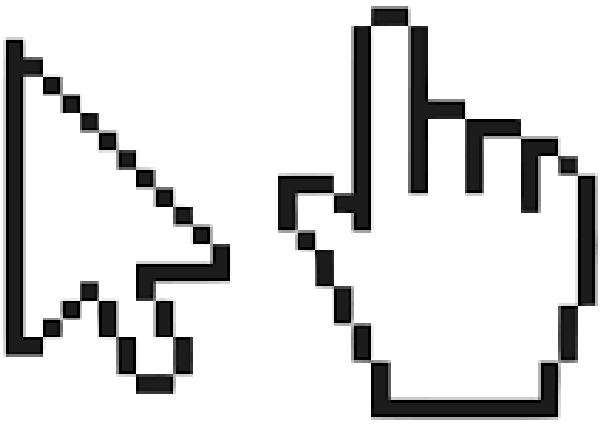
Accessing Web Page

Accessing web page is very simple. Just enter the **URL** in the address bar as shown the following diagram:



Navigation

A web page may contain **hyperlinks.** When we click on these links other web page is opened. These hyperlinks can be in form of text or image. When we take the mouse over a hyperlink, pointer change its shape to hand.



Key Points

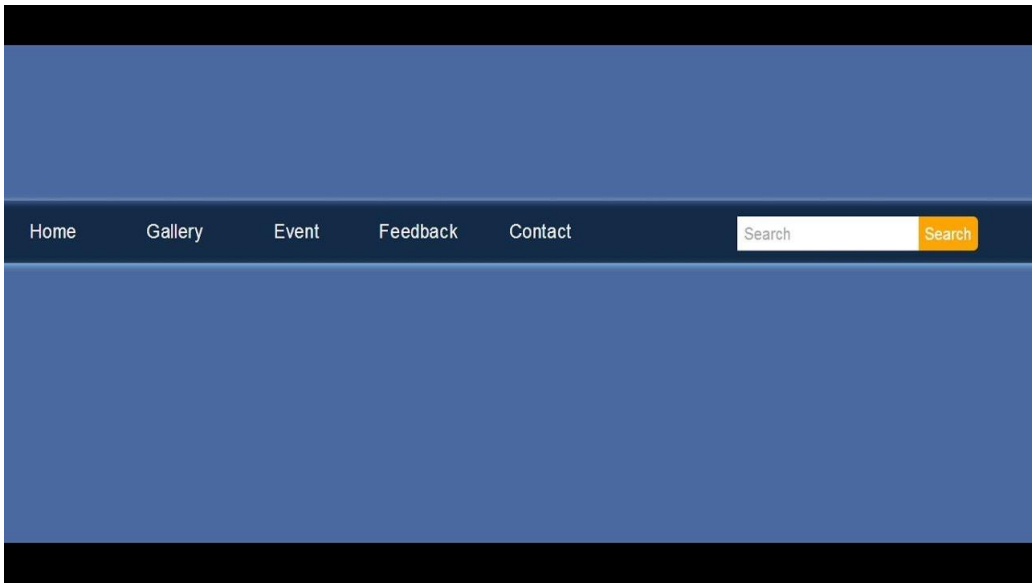
- In case, you have accessed many web pages and willing to see the previous webpage then just click back button.
- You can open a new web page in the same tab, or different tab or in a new window.

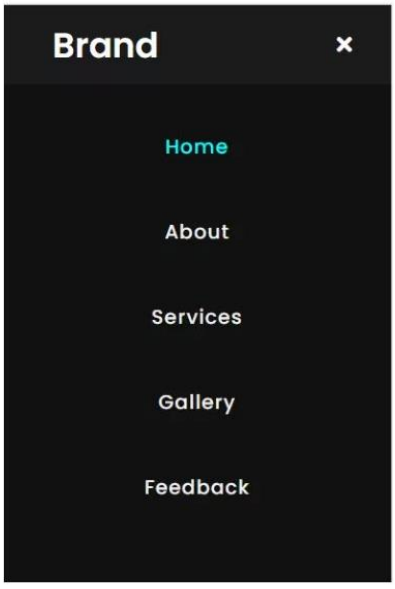
Navigation Bar

A navigation bar is a user interface element within a webpage that contains links to other sections of the website. In most cases, the navigation bar is part of the main website template, which means it is displayed on most, if not all, pages within the website. This means that no matter what page you are viewing, you can use the navigation bar to visit other sections of the website.

A website navigation bar is most commonly displayed as horizontal list of links at the top of each page. It may be below the header or logo, but it is always placed before the main content of the page. In some cases, it may make sense to place the navigation bar vertically on the left side of each page. This type of navigation bar is also called a sidebar, since it appears to the side of the primary content. Some websites have both a horizontal navigation bar at the top and a vertical navigation bar on the left side of each page.

The navigation bar is an important element of a website's design since it allows users to quickly visit any section within the site. If you've ever visited a website without a navigation bar, you may have found it is difficult to locate the page you need. You may have also had to click "Back" several times in order to find a link to the next section you wanted to visit.





BCADS-116 Internet Basics & HTML

Unit-3

HTML- (Hyper Text Markup Language)

HTML stands for **HyperText Markup Language**. It is used to design web pages using a markup language. HTML is the combination of **Hypertext** and **Markup language**. **Hypertext** defines the **link between the web pages**. A **markup language** is used to **define the text document within tag** which **defines the structure of web pages**. This language is used to annotate text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

HTML consists of a series of elements, which we use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on. For example, take the following line of content:

```
I am a student of Computer Application Department.
```

If we wanted the line to stand by itself, we could specify that it is a paragraph by enclosing it in paragraph tags:

```
<p> I am a student of Computer Application Department. </p>
```

The major points of HTML are given below:

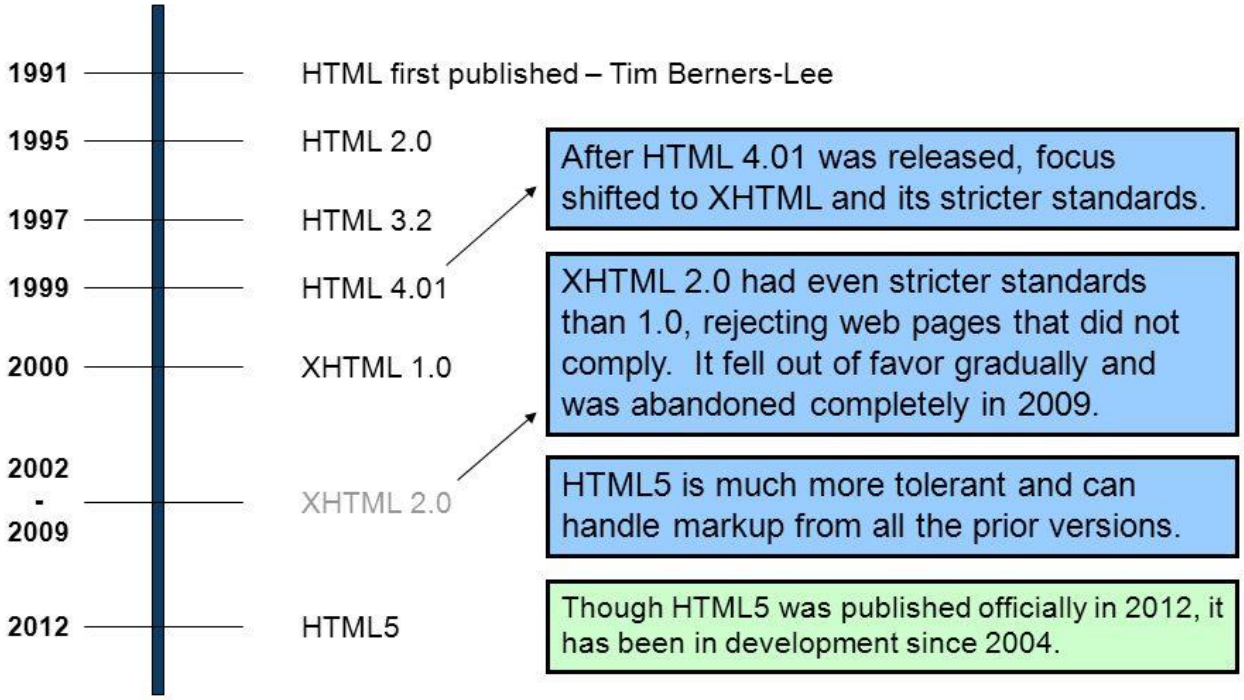
- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

History of HTML

HTML was created by Sir Tim Berners-Lee in late 1991 but was not released officially, published in 1995 as HTML 2.0. HTML 4.01 was published in late 1999 and was a major version of HTML.

HTML is a very evolving markup language and has evolved with various versions updating. Long before its revised standards and specifications are carried in, each version has allowed its user to create web pages in a much easier and prettier way and make sites very efficient.

History of HTML



- HTML 1.0 was released in 1993 with the intention of sharing information that can be readable and accessible via web browsers. But not many of the developers were involved in creating websites. So the language was also not growing.
- Then comes the HTML 2.0, published in 1995, which contains all the features of HTML 1.0 along with that few additional features, which remained as the standard markup language for designing and creating websites until January 1997 and refined various core features of HTML.
- Then comes the HTML 3.0, where Dave Raggett who introduced a fresh paper or draft on HTML. It included improved new features of HTML, giving more powerful characteristics for webmasters in designing web pages. But these powerful features of new HTML slowed down the browser in applying further improvements.
- Then comes HTML 4.01, which is widely used and was a successful version of HTML before HTML 5.0, which is currently released and used worldwide. HTML 5 can be said for an extended version of HTML 4.01, which was published in the year 2012.

A Simple HTML Document

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading </h1>
<p>My first paragraph. </p>

</body>
</html>
```

Example Explained

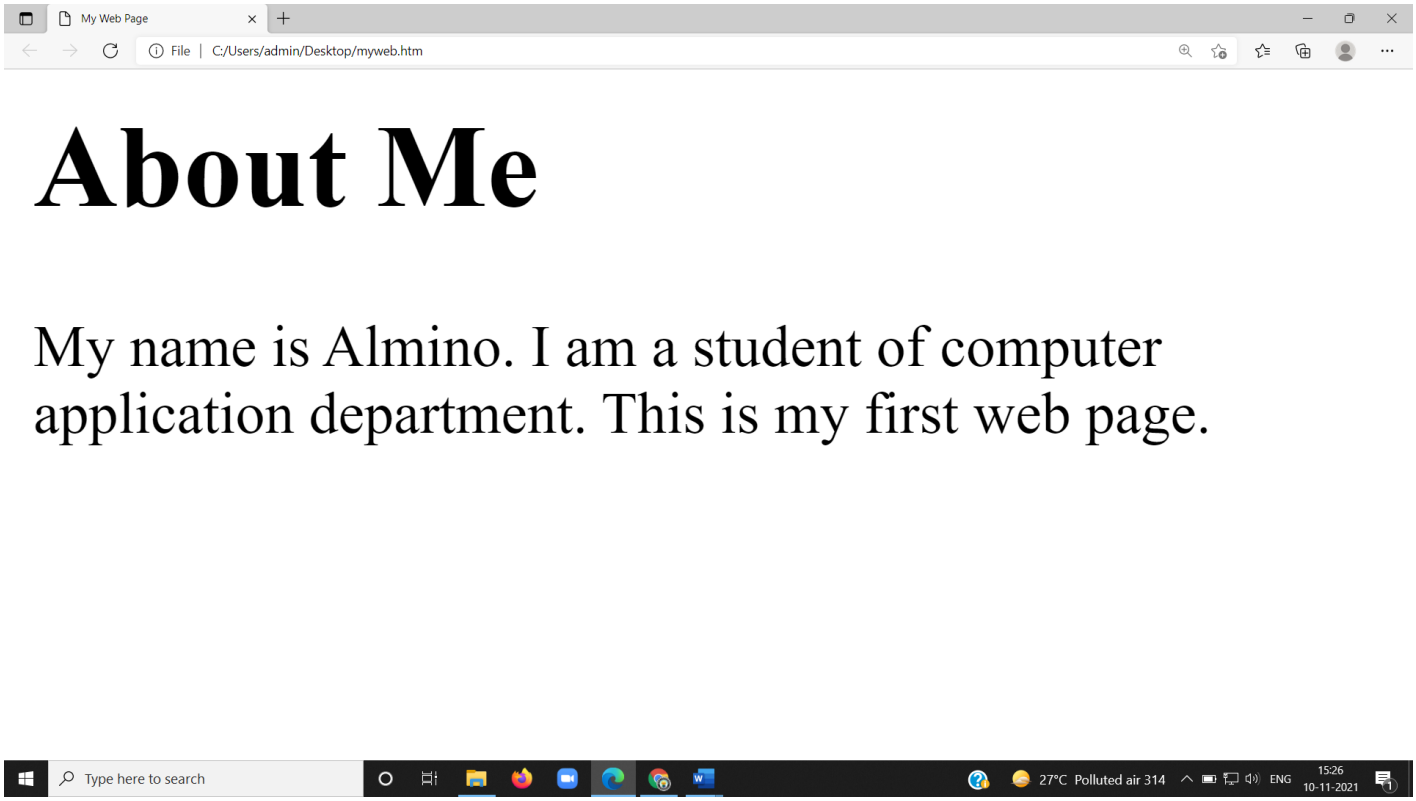
- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

Example

```
<!DOCTYPE html>
<html>
<head>
<title>My Web Page</title>
</head>
<body>

<h1>About Me</h1>
<p>My name is Almino. I am a student of computer application department. This is my first web page. </p>
```

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

Output



HTML Editors

To write HTML code, we need an HTML editor, and we can do this with the help of a **notepad** on our computer. Some computers with different Operating-system have different text editors.

A web page is a text file in which a hypertext language is written according to HTML grammar. This HTML code is displayed by the browser converting it to a web page. **Programs** in which **HTML code is written or modified** are called **HTML editors**.

Follow the steps below to create your first web page with Notepad.

Step 1: Open Notepad (PC)

Windows 8 or later:

Open the **Start Screen** (the window symbol at the bottom left on your screen). Type **Notepad**.

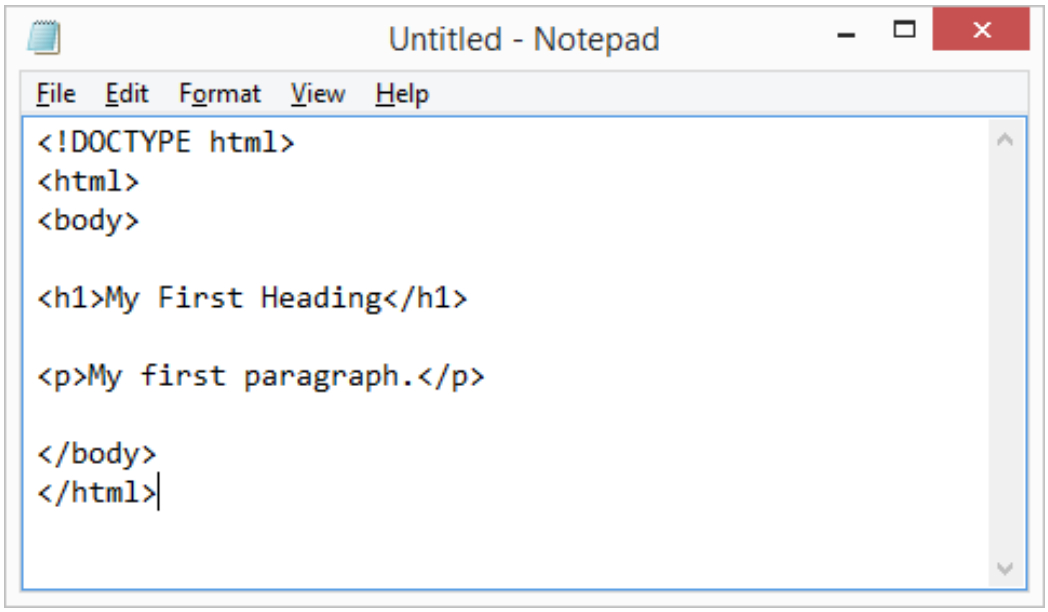
Windows 7 or earlier:

Open **Start** > **Programs** > **Accessories** > **Notepad**

Step 2: Write Some HTML

Write or copy the following HTML code into Notepad:

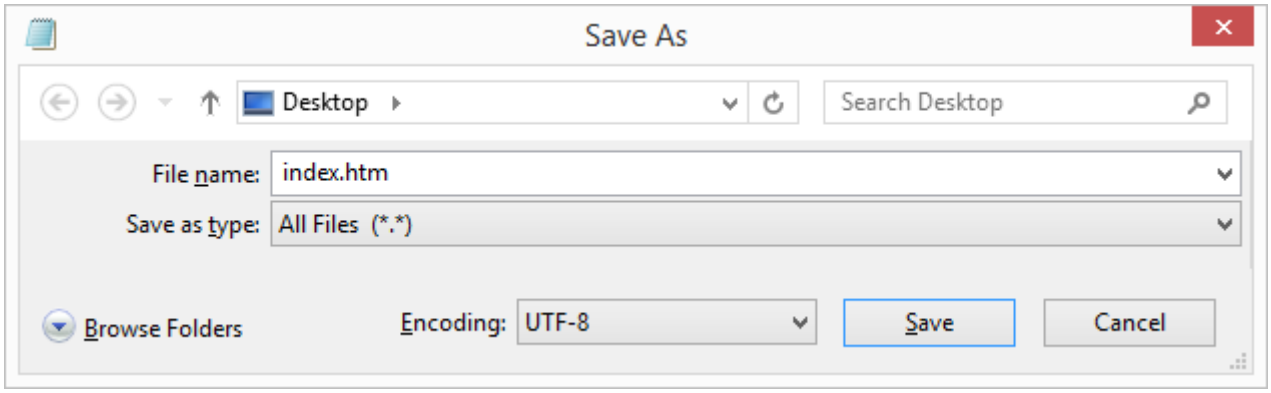
```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph. </p>

</body>
</html>
```



Step 3: Save the HTML Page

Save the file on your computer. Select **File > Save as** in the Notepad menu.

Name the file **"index.htm"** and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).

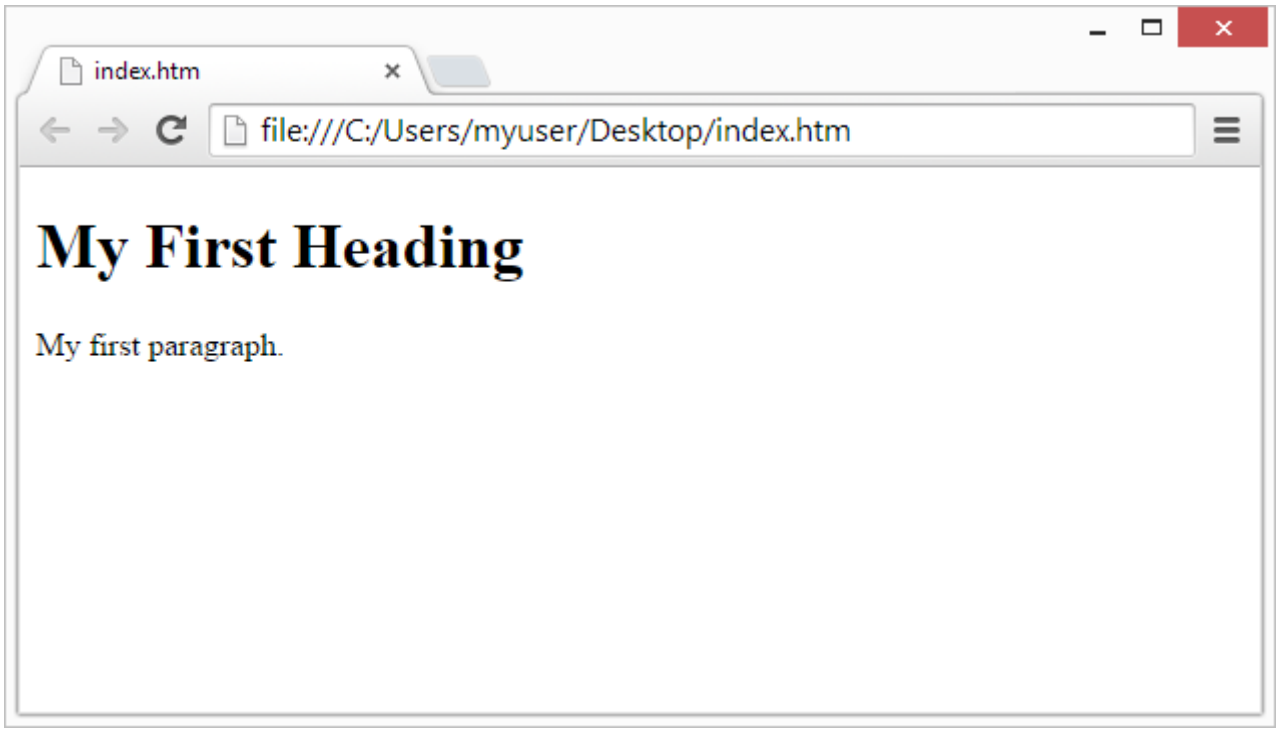


Tip: You can use either .htm or .html as file extension. There is no difference, it is up to you.

Step 4: View the HTML Page in Your Browser

Open the saved HTML file in your favourite browser (double click on the file, or right-click - and choose "Open with").

The result will look much like this:



HTML element

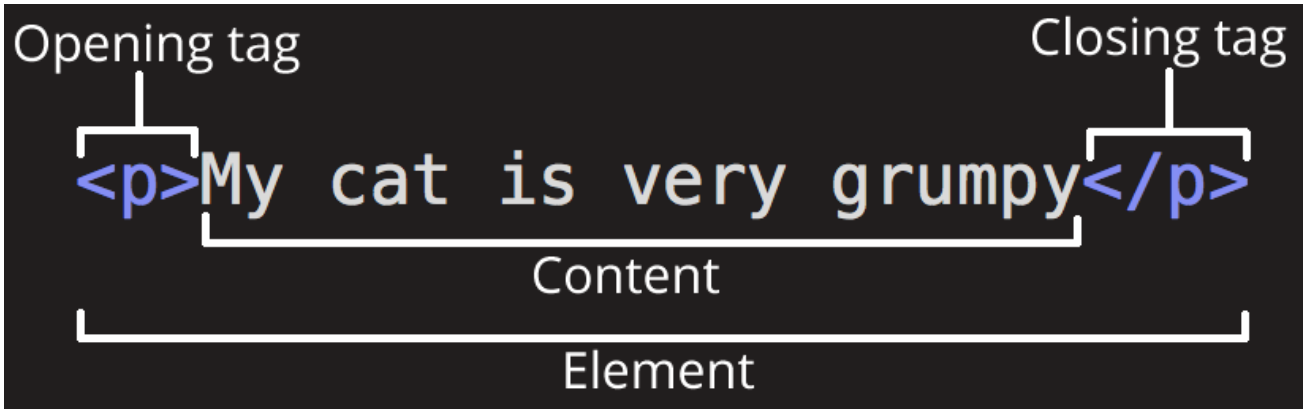
The HTML **element** is everything from the **start tag** to the **end tag**:

<tagname>Content goes here...</tagname>

Examples of some HTML elements:

<h1>My First Heading </h1>

My first paragraph.



The main parts of our element are as follows:

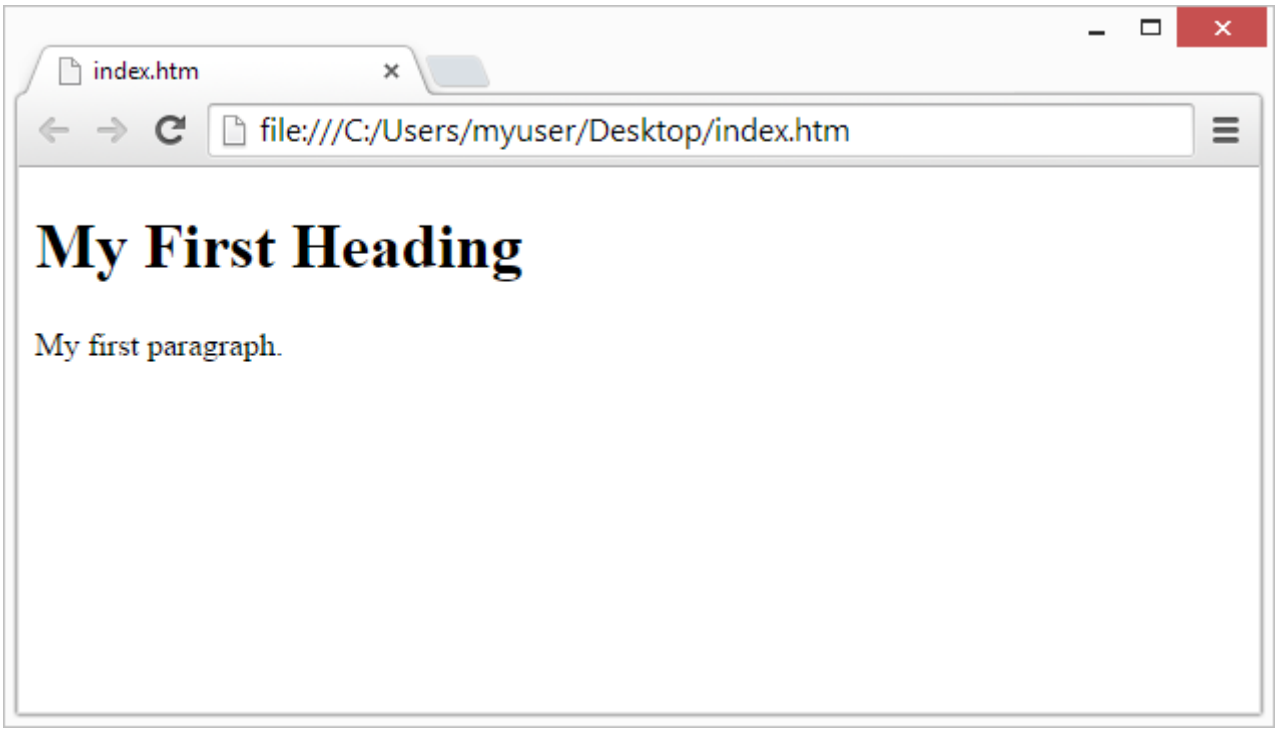
1. **The opening tag:** This consists of the name of the element (in this case, p), wrapped in opening and closing **angle brackets**. This states where the element begins or starts to take effect — in this case where the paragraph begins.
2. **The closing tag:** This is the same as the opening tag, except that it includes a *forward slash* before the element name. This states where the element ends — in this case where the paragraph ends. Failing to add a closing tag is one of the standard beginner errors and can lead to strange results.
3. **The content:** This is the content of the element, which in this case, is just text.
4. **The element:** The opening tag, the closing tag, and the content together comprise the element.

Note: Some HTML elements have no content (like the <br>, <hr> element). These elements are called empty elements. Empty elements do not have an end tag!

Web Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:



HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

HTML Documents

All HTML documents must start with a document type declaration: <!DOCTYPE html>.

The HTML document itself begins with <html> and ends with </html>.

The visible part of the HTML document is between <body> and </body>.

Example

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The <!DOCTYPE> declaration is not case sensitive.

HTML Headings

HTML headings are defined with the <h1> to <h6> tags.

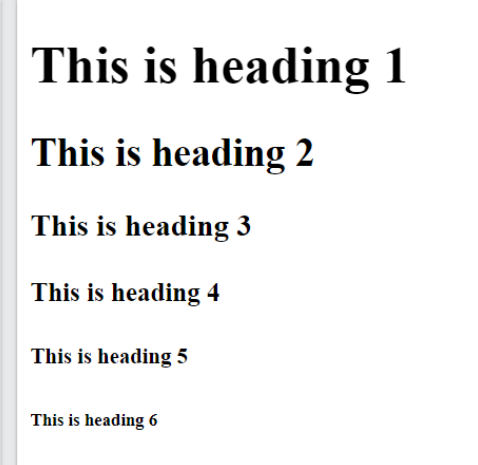
<h1> defines the most important heading. <h6> defines the least important heading:

Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Example with output

```
<!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>
```



HTML Paragraphs

HTML paragraphs are defined with the tag:

Example

This is a paragraph.

This is another paragraph.

Example with output

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

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

This is a paragraph.

This is another paragraph.

HTML Links

HTML links are defined with the <a> tag. The **<a> tag** (**anchor tag**) in HTML is used to create a hyperlink on the webpage. This **hyperlink** is used to **link the webpage to other webpages**.

Syntax: Link Name

Example

This is a link

The link's destination is specified in the href attribute.

Attributes are used to provide additional information about HTML elements.

Example with output

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

<h2>HTML Links</h2>
<p>HTML links are defined with the a tag:</p>

<a href="https://www.google.com">This is a link</a>

</body>
</html>
```

HTML Links

HTML links are defined with the a tag:

This is a link

HTML Images

HTML images are defined with the <img> tag.

The source file (src), width, and height are provided as attributes:

Example

<img src="google.jpg" width="104" height="142">

Example with output

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

<h2>HTML Images</h2>
<p>HTML images are defined with the img tag:</p>

<img src="google.jpg" width="104" height="142">

</body>
</html>
```

HTML Images

HTML images are defined with the img tag:

The <br> tag defines a line break, and is an empty element without a closing tag:

Example

```
<p>This is a <br> paragraph with a line break. </p>
```

Example with output

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

<p>This is from IFTM University. <br> Many courses are running here.
</p>

</body>
</html>
```

This is from IFTM University.
Many courses are running here.

HTML | Attributes

- All HTML elements can have attributes
- Attributes provide additional information about elements
- Attributes are always specified in the start tag

The href Attribute

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

The src Attribute

The <img> tag is used to embed an image in an HTML page.

The width and height Attributes

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels).

The alt Attribute

The required alt attribute for the <img> 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.

HTML Lists

HTML lists allow web developers to group a set of related items in lists. <ul type = "square">

<ul type = "disc">

<ul type = "circle">

Example

An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

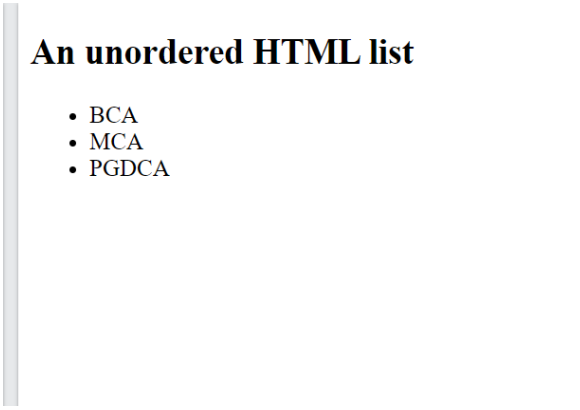
Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

Example

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

<h2>An unordered HTML list</h2>

<ul>
  <li>BCA</li>
  <li>MCA</li>
  <li>PGDCA</li>
</ul>

</body>
</html>
```



Ordered HTML List

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag. If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using <ol> tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>. Following are the possible options −

<ol type = "1"> - Default-Case Numerals.

<ol type = "I"> - Upper-Case Numerals.

<ol type = "i"> - Lower-Case Numerals.

<ol type = "A"> - Upper-Case Letters.

<ol type = "a"> - Lower-Case Letters.

The list items will be marked with numbers by default:

Example

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

<h2>An ordered HTML list</h2>

<ol>
  <li>BCA</li>
  <li>MCA</li>
  <li>PGDCA</li>
</ol>

</body>
</html>
```

An ordered HTML list

1. BCA
2. MCA
3. PGDCA

The start Attribute

You can use **start** attribute for <ol> tag to specify the starting point of numbering you need. Following are the possible options −

<ol type = "1" start = "4"> - Numerals starts with 4.

<ol type = "I" start = "4"> - Numerals starts with IV.

<ol type = "i" start = "4"> - Numerals starts with iv.

<ol type = "a" start = "4"> - Letters starts with d.

<ol type = "A" start = "4"> - Letters starts with D.

HTML Tables

HTML table tag is used to display data in tabular form (row * column). There can be many columns in a row.

We can create a table to display data in tabular form, using <table> element, with the help of <tr> , <td>, and <th> elements.

In Each table, table row is defined by <tr> tag, table header is defined by <th>, and table data is defined by <td> tags.

Define an HTML Table

A table in HTML consists of table cells inside rows and columns

Example:

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

<h2 align="center">HTML table</h2>

<table border=1 align="center">

   <tr>
                <td> S.No </td>
                <td> Name </td>
                        <td> Course </td>
```

```
  </tr>

 <tr>
        <td>1</td>
        <td>Ram</td>
        <td>BCA</td>
 </tr>

 <tr>
        <td>2</td>
        <td>Shyam</td>
        <td>MCA</td>
 </tr>
  <tr>
        <td>3</td>
        <td>Geeta</td>
        <td>MCA</td>
 </tr>
  <tr>
        <td>4</td>
        <td>Harshit</td>
        <td>B.Tech</td>
 </tr>

</table>

</body>
</html>
```

Output

HTML table

S.No	Name	Course
1	Ram	BCA
2	Shyam	MCA
3	Geeta	MCA
4	Harshit	B.Tech

Table Heading

Table heading can be defined using **<th>** tag. This tag will be put to replace <td> tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row. Headings, which are defined in <th> tag are cantered and bold by default.

Example

```
<!DOCTYPE html>
<html>
<body>
<h2 align="center">HTML table</h2>
<table border=1 align="center">
    <th> S.No  </th>
    <th> Name </th>
    <th> Course </th>

 <tr> <td>1</td>  <td>Sonu  </td>  <td>BCA </td>   </tr>
 <tr> <td>2</td>  <td>Hari  </td>  <td>MCA </td>   </tr>
 <tr> <td>3</td>  <td>Neeta </td>  <td>MCA  </td>  </tr>
 <tr> <td>4</td>  <td>Heera </td>  <td>B.Tech</td> </tr>
</table>
</body>
</html>
```

HTML table

S.No	Name	Course
1	Sonu	BCA
2	Hari	MCA
3	Neeta	MCA
4	Heera	B.Tech

Cellpadding and Cellspacing Attributes

There are two attributes called cellpadding and cellspacing which you will use to adjust the white space in your table cells. The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell.

Example

```
<!DOCTYPE html>
<html>
<body>
<h2 align="center">HTML table</h2>
<table border=1 cellspacing=5 cellpadding=5 align="center">
    <th> S.No  </th>
    <th> Name </th>
    <th> Course </th>

 <tr> <td>1</td>  <td>Sonu  </td>  <td>BCA </td>   </tr>
 <tr> <td>2</td>  <td>Hari  </td>  <td>MCA </td>   </tr>
 <tr> <td>3</td>  <td>Neeta </td>  <td>MCA  </td>  </tr>
 <tr> <td>4</td>  <td>Heera </td>  <td>B.Tech</td> </tr>
</table>
</body>
</html>
```

HTML table

S.No	Name	Course
1	Sonu	BCA
2	Hari	MCA
3	Neeta	MCA
4	Heera	B.Tech

Colspan and Rowspan Attributes

You will use colspan attribute if you want to merge two or more columns into a single column. Similar way you will use rowspan if you want to merge two or more rows.

```
<!DOCTYPE html>
<html>
   <body>
      <table border = "1">
         <tr>
            <th>S.No</th>
            <th>Name</th>
            <th>Course</th>
         </tr>
         <tr>
            <td rowspan = "2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
         </tr>
         <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
         </tr>
         <tr>
            <td colspan = "3">Row 3 Cell 1</td>
         </tr>
      </table>
   </body>
</html>
```

S.No	Name	Course
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

Tables Backgrounds

You can set table background using one of the following two ways −

bgcolor attribute − You can set background color for whole table or just for one cell.

background attribute − You can set background image for whole table or just for one cell.

You can also set border color also using bordercolor attribute.

Table Height and Width

You can set a table width and height using **width** and **height** attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

Example

```
<!DOCTYPE html>
<html>
<body>
<h2 align="center">HTML table</h2>
<table border=1 cellspacing=5 cellpadding=5 align="center" bgcolor="yellow" width=500 height=200>
    <th> S.No  </th>
    <th> Name </th>
    <th> Course </th>

 <tr> <td>1</td>  <td>Sonu  </td>  <td>BCA </td>   </tr>
 <tr> <td>2</td>  <td>Hari  </td>  <td>MCA </td>   </tr>
 <tr> <td>3</td>  <td>Neeta </td>  <td>MCA  </td>  </tr>
 <tr> <td>4</td>  <td>Heera </td>  <td>B.Tech</td> </tr>
</table>
</body>
</html>
```

Output

HTML table

S.No	Name	Course
1	Sonu	BCA
2	Hari	MCA
3	Neeta	MCA
4	Heera	B.Tech

HTM Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

Example

First name:

John

Last name:

Doe

Submit

The <form> Element

The HTML <form> element is used to create an HTML form for user input:

```
<form>
.
form elements
.
</form>
```

The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

The <input> Element

The HTML <input> element is the most used form element.

An <input> element can be displayed in many ways, depending on the type attribute.

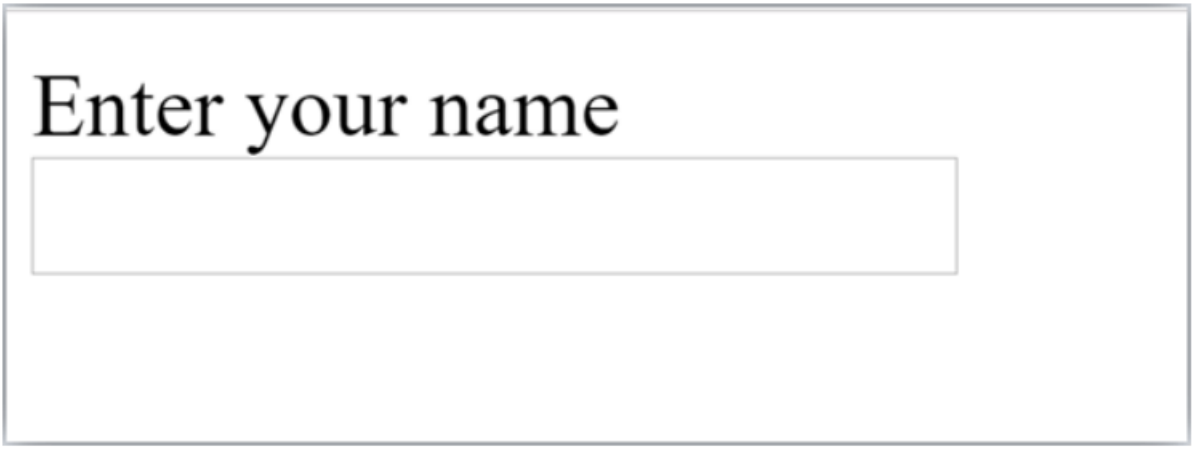
Here are some examples:

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="password">	Defines a password field.
<input type="email">	Defines an input field that should contain an e-mail address.
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

Example:

```
<body>
 <form>
   Enter your name  <br>
   <input type="text" name="username">
 </form>
</body>
```

Output:



Label Tag in Form

It is considered better to have label in form. As it makes the code browser/user friendly.

If you click on the label tag, it will focus on the text control. To do so, you need to have for attribute in label tag that must be same as id attribute of input tag.

- The <label> tag defines a label for many form elements.
- The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.
- The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.
- The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

NOTE: It is good to use <label> tag with form, although it is optional but if you will use it, then it will provide a focus when you tap or click on label tag. It is more worthy with touchscreens.

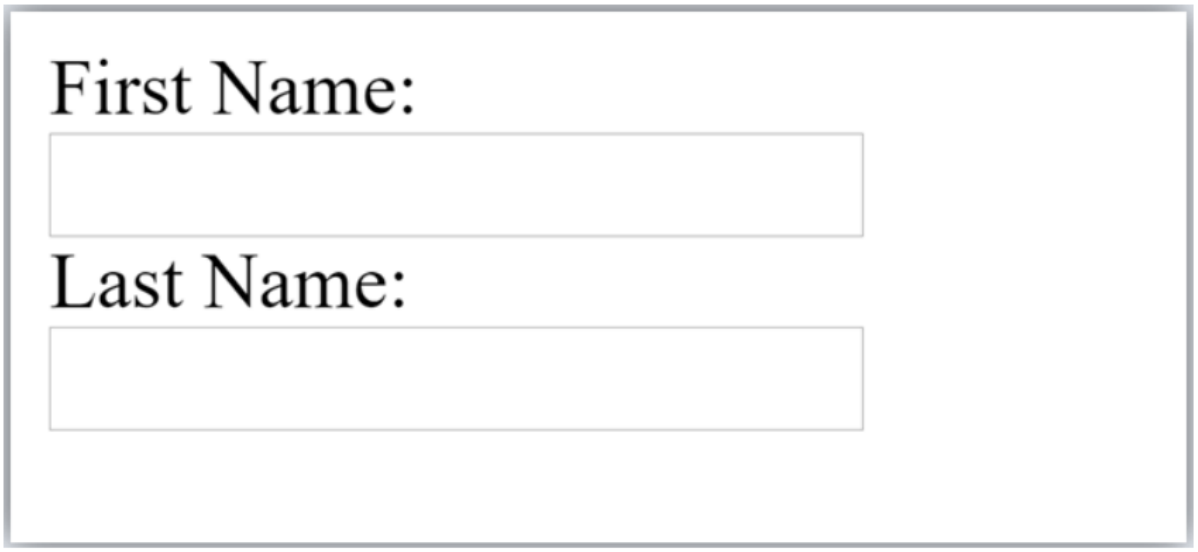
Example 1

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

<h2>HTML Form</h2>

<form>
  <label>First name:</label>
  <input type="text"><br>
  <label>Last name: </label>
   <input type="text">
</form>

</body>
</html>
```

Example 2

```
<form>
  <label for="firstname">First Name: </label> <br/>
        <input type="text" id="firstname" name="firstname"/> <br/>
  <label for="lastname">Last Name: </label>
        <input type="text" id="lastname" name="lastname"/> <br/>
</form>
```

Output:

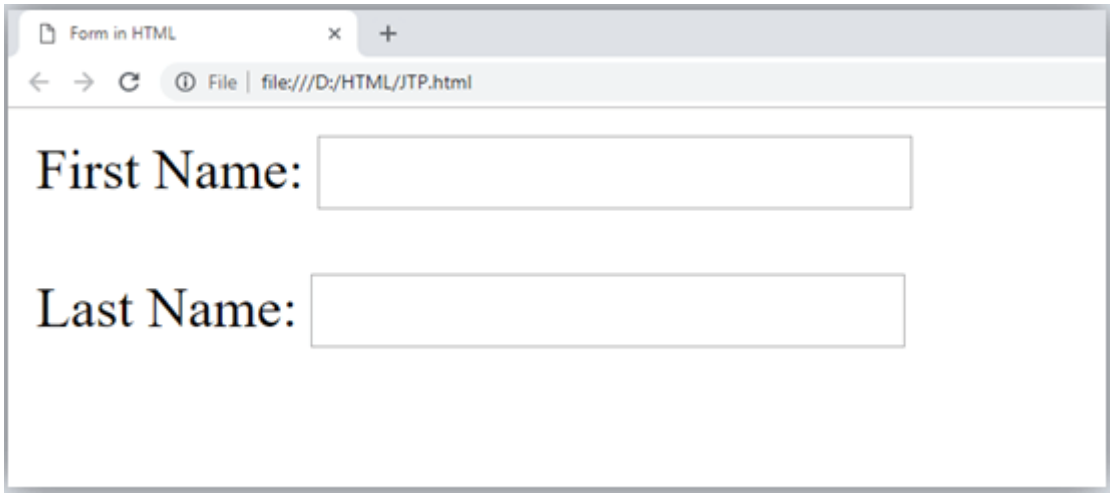


Text Field Control

The type="text" attribute of input tag creates textfield control also known as single line textfield control. The name attribute is optional, but it is required for the server-side component such as JSP, ASP, PHP etc.

```
<form>
  First Name: <input type="text" name="firstname"/> <br/>
  Last Name:  <input type="text" name="lastname"/> <br/>
</form>
```

Output:



Radio Button Control

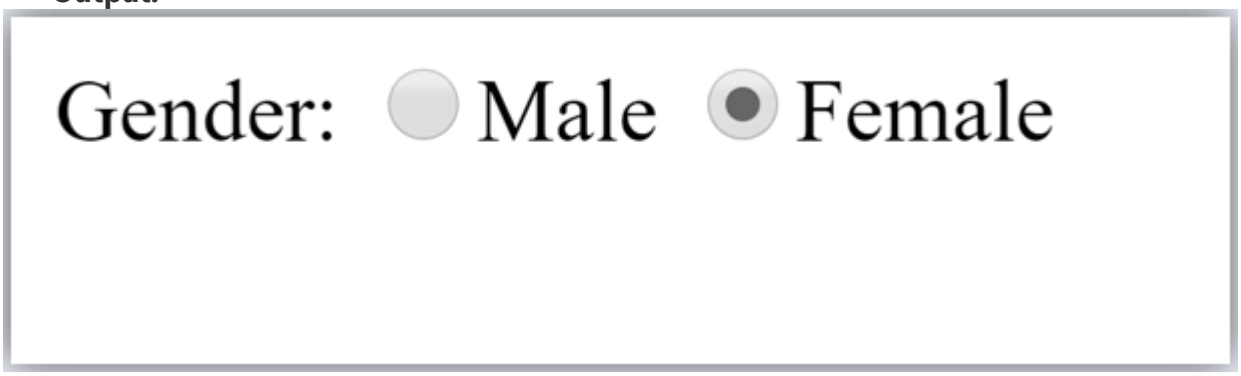
The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc.

If you use **one name for all the radio buttons**, only one radio button can be selected at a time.

Using radio buttons for multiple options, you can only choose a single option at a time.

```
<form>
  <label for="gender">Gender: </label>
        <input type="radio" id="gender" name="gender" >Male
        <input type="radio" id="gender" name="gender" >Female <br/>
</form>
```

Output:

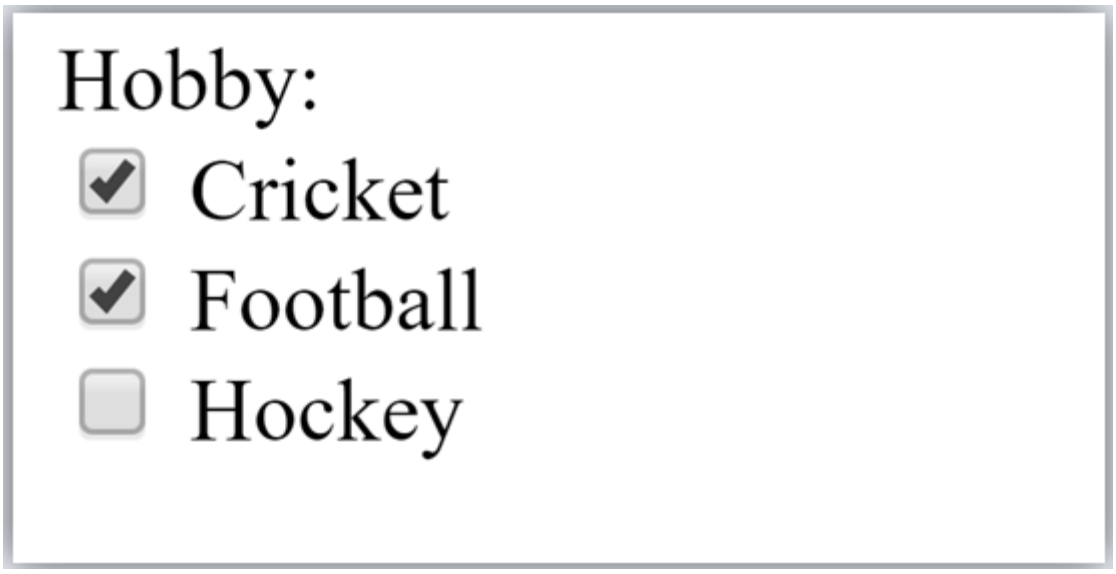


Checkbox Control

- The checkbox control is used to check multiple options from given checkboxes.
- The checkbox is shown as a square box that is ticked (checked) when activated.
- Checkboxes are used to let a user select one or more options of a limited number of choices.

```
<form>
Hobby:<br>
        <input type="checkbox" id="cricket" name="cricket">
          <label for="cricket">Cricket</label> <br>
        <input type="checkbox" id="football" name="football" >
          <label for="football">Football</label> <br>
        <input type="checkbox" id="hockey" name="hockey" >
          <label for="hockey">Hockey</label>
</form>
```

Output:



If we want to select any checkbox by default, then we have to set the checked attribute with the "yes" value as described in the following syntax:

```
<input type="checkbox"   checked="yes">
```

Example with output:

```
<html>
<head>
</head>
<body>
<form>
Programming Languages: <br>
              <input type="checkbox" name="C" checked="yes">
                 <label>C</label> <br>
              <input type="checkbox" name="Java">
                 <label>Java</label> <br>
              <input type="checkbox" name="Python">
                 <label>Python</label> <br>
         <input type="checkbox" name="PHP">
                 <label>PHP</label>
</form>
</body>
</html>
```

Programming Languages:
- [x] C
- [ ] Java
- [ ] Python
- [ ] PHP

Email Field Control

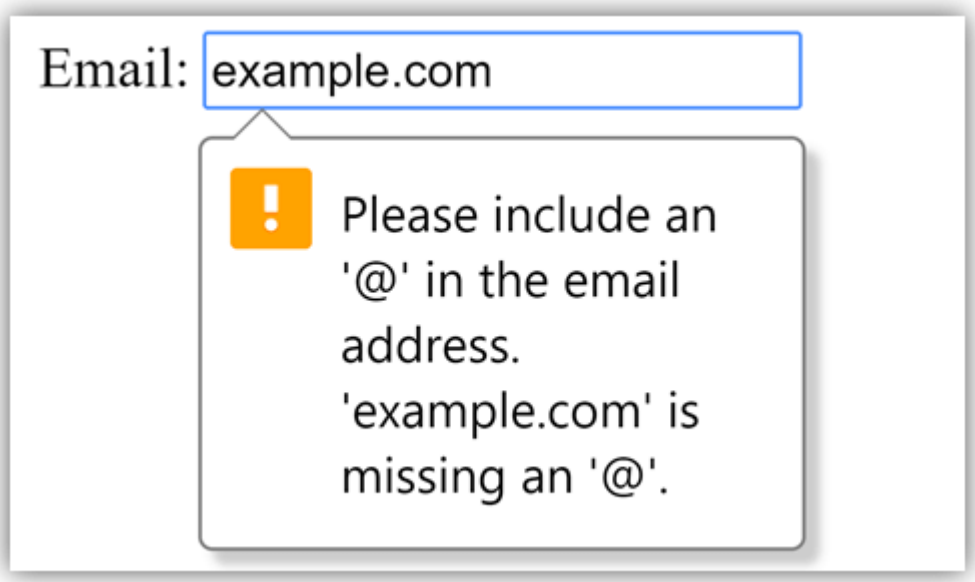
The email field in new in HTML 5. It validates the text for correct email address. You must use @ and . in this field.

```
<form>
  <label for="email">Email: </label>
       <input type="email" id="email" name="email"/> <br/>
</form>
```

Output:

Email:

Note: If we will not enter the correct email, it will display error like:



Password Field Control

The password is not visible to the user in password field control.

```
<form>
  <label for="password">Password: </label>
      <input type="password" id="password" name="password"/> <br/>
</form>
```

Output:

Password:

Submit button control

HTML **<input type="submit">** are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.

Syntax:

```
<input type="submit" value="submit">
```

The type = submit , specifying that it is a submit button

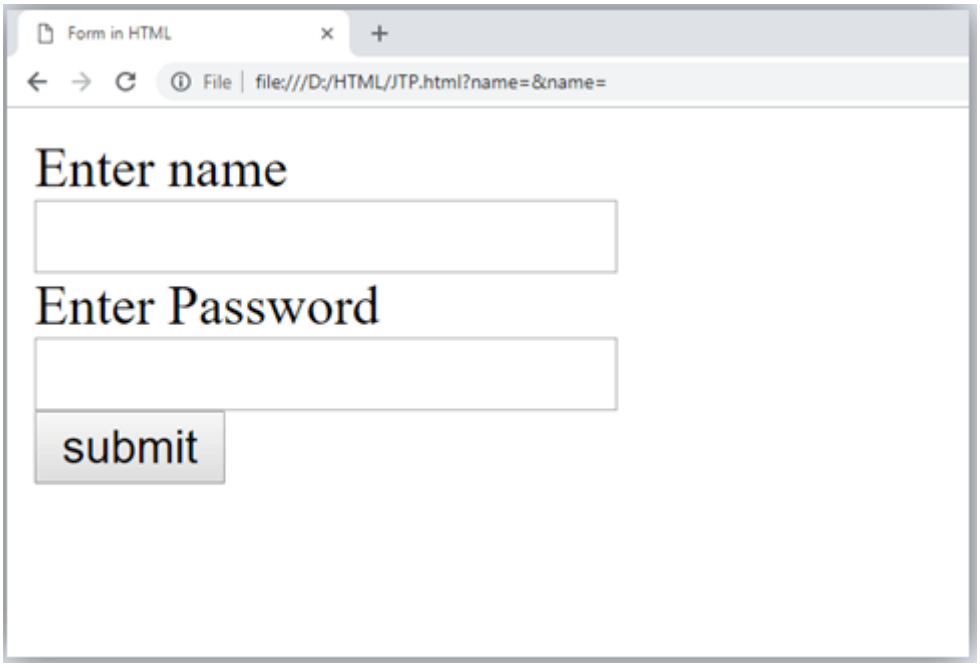
The value attribute can be anything which we write on button on web page.

The name attribute can be omit here.

Example:

```
<form>
  <label for="name">Enter name</label><br>
  <input type="text" id="name" name="name"><br>
  <label for="pass">Enter Password</label><br>
  <input type="Password" id="pass" name="pass"><br>
  <input type="submit" value="submit">
</form>
```

Output:



Input Type Button

<input type="button"> defines a **button**:

Example

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

<h2>Input Button</h2>

<input type="button" onclick="alert('Hello World!')" value="Click
Me!">

</body>
</html>
```

HTML - Frames

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

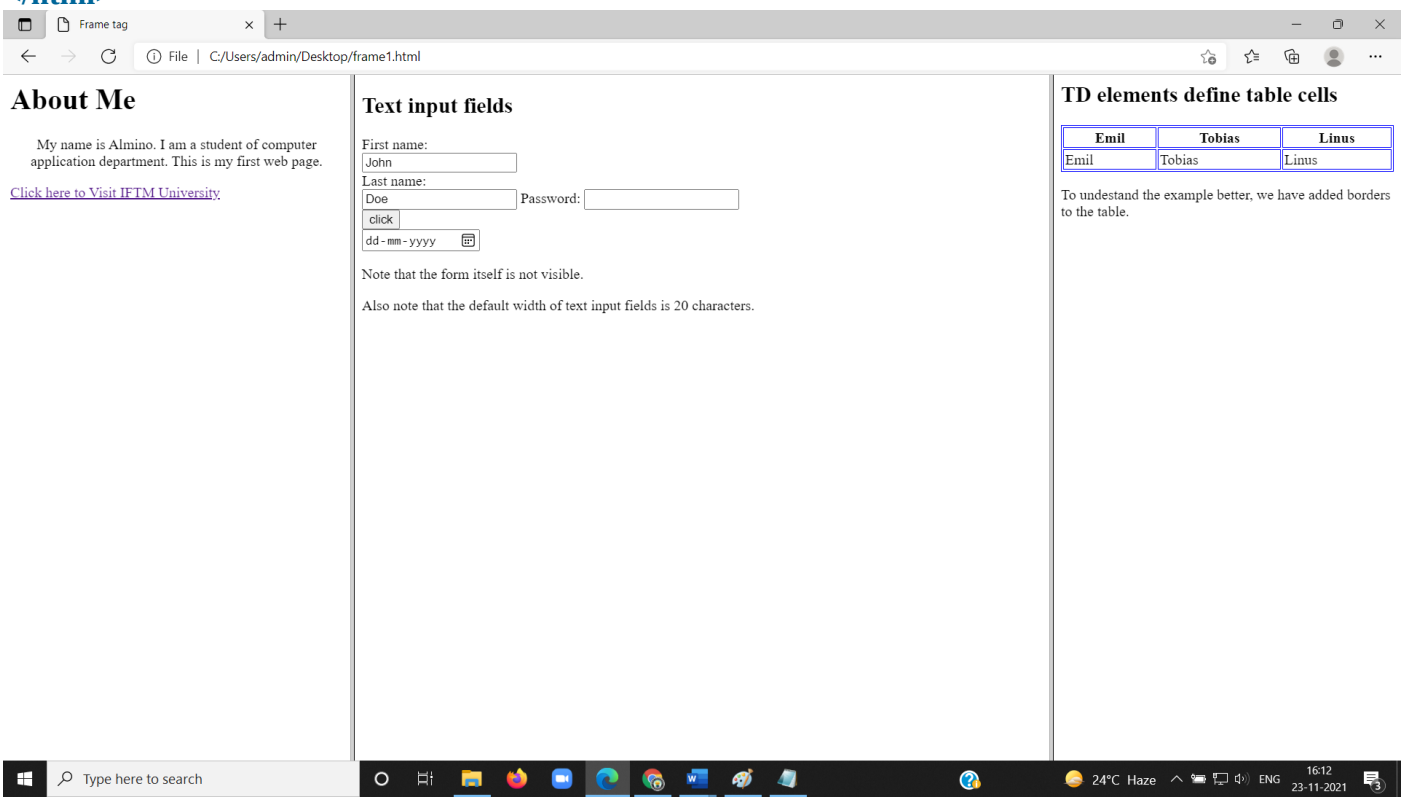
Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages −

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back* button might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Frame tag</title>
</head>
 <frameset cols="25%, 50%, 25%">
    <frame src="myweb.html" >
    <frame src="form.html">
    <frame src="table.html">
 </frameset>
</html>
```



CSS: Cascading Style Sheets

CSS stands for **Cascading Style Sheet**. The CSS was developed by the **World Wide Web Consortium (W3C)** in the year of **1996**. It is a style sheet language used to shape the HTML elements that will be displayed in the browsers as a web-page. Without using CSS, the website which has been created by using HTML, will look dull. Basically, CSS gives the outer cover on any HTML elements. If you consider HTML as a skeleton of the web-page then the CSS will be the skin of the skeleton.

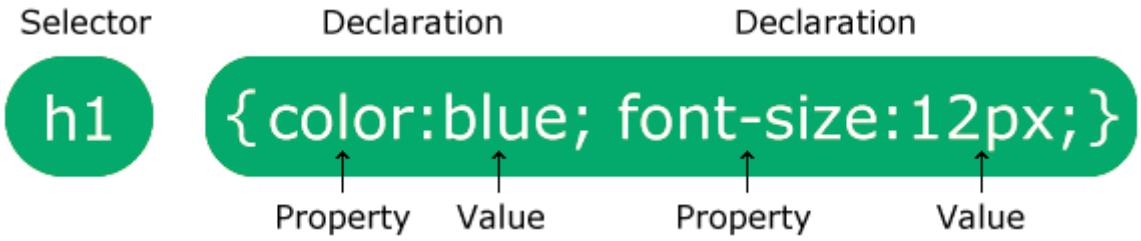
- CSS is the language we use to style an HTML document.
- CSS describes how HTML elements should be displayed.
- The CSS can be applied to HTML documents in different ways.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once

Characteristics of CSS:

- **Maintenance:** It is easy to maintain, changing in a single place will affect globally in your web site. No need to change every specific place.
- **Time-saving:** You can easily use any single CSS script at multiple places.
- **Support:** CSS is supported by all the browsers and search engines.
- **Native front-end:** CSS contains a huge list of attributes and function that is helpful to design the HTML page.
- **Selectors:** In CSS, there are lots of selectors (ID selectors, Class Selectors, etc.) that will be helpful to perform specific tasks.

CSS Syntax:

A CSS rule consists of a **selector** and a **declaration** block.



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Examples 1:

In this example all <p> elements will be center-aligned, with a red text color:

```
p {
color: red;
text-align:
center;}
```

Examples 2:

```
body {background-color: lightblue;}
h1 {color: white; text-align: center;}
p {font-family: times-new-roman; font-size: 20px;}
```

Advantages of CSS:

- CSS is compatible with all the devices.
- With the help of CSS, website maintenance is easy and faster.
- CSS support consistent and spontaneous changes.
- CSS make the website faster and enhances search engine capabilities to crawl the web pages
- It holds a special feature that is the ability to re-position.

CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1 {
  text-align: center;
  color: red;
}

h2 {
  text-align: center;
  color: red;
}

p {
  text-align: center;
  color: red;
}
```

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

Example

In this example we have grouped the selectors from the code above:

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can select selector based on **name**, **id** and **class**.

CSS element Selector

The element selector selects HTML elements based on the element name.

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

```
p {
  text-align: center;
  color: red;
}
```

The CSS id Selector

The **id selector** uses the **id attribute** of an **HTML element** to select a specific element.

The id of an element is **unique within a page**, so the id selector is used to select one unique element! i.e;

We cannot have more than one element with the same id in an HTML document.

To select an element with a specific id, **write a hash (#)** character, followed by the id of the element.

Note: An id name cannot start with a number!

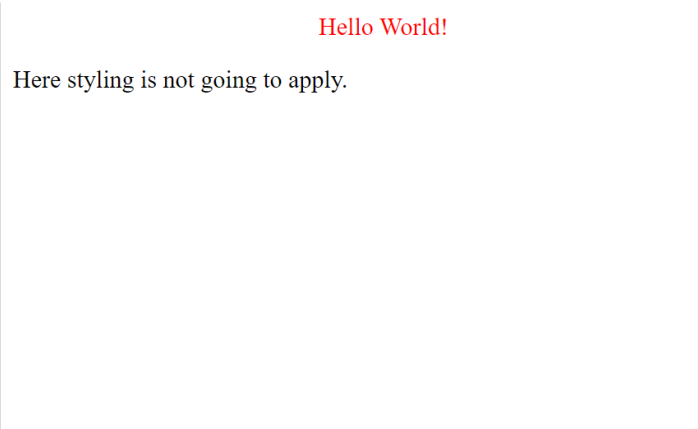
Example

The CSS rule below will be applied to the HTML element with id="para1":

```
#para1
{
  text-align: center;
  color: red;
}
```

Example with output:

```
<html>
<head>
<style>
#para1 {
  text-align: center;color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>Here styling is not going to apply.</p>

</body>
</html>
```



The CSS class Selector

The **class selector** selects HTML elements with a specific class attribute.

Multiple HTML elements can share the same class.

To select elements with a specific class, **write a dot (.) character**, followed by the class name.

Note: An class name cannot start with a number!

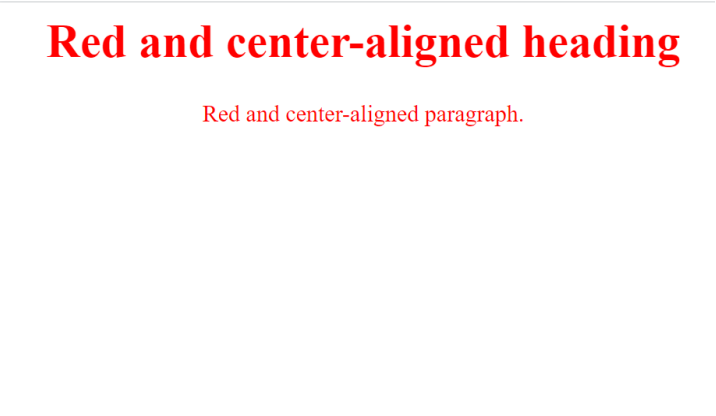
Example

In this example all HTML elements with class="center" will be red and center-aligned:

```
.center {
  text-align: center;
  color: red;
}
```

Example with output:

```
<html>
<head>
<style>
.center {
  text-align: center;  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```



CSS Basic Properties:

Text Properties

- **Font Family**

The **font-family** property allows us to change the particular font we are using. You may select any font which is installed on the clients computer with this property. Some of the more commonly used fonts include:

- Times New Roman
- Arial
- Verdana

It is possible to specify and include non-standard fonts but this is a little more complex.

Example:

```
h1 {
font-family: Times New Roman;
}

p {
font-family: Verdana;
}
```

- **Font Size**

Font-size may be specified using a few different types but the easiest to work with is pixels (px).

Example:

```
h1 {
font-size: 30px;
}
p {
font-size: 16px;
}
```

- **Font Weight**

The **font-weight** css property allows us to specify how thick the lines of the characters are. It may be one of the following values:

- lighter
- normal
- bold
- bolder

Example:

```
h1 {
font-weight: normal;
}

p {
font-weight: bold;
}
```

Colour Properties

Colour is a reasonably complex thing to define. There are several ways in which we can do this. Here we will just work with it in its simplest form as:

- **Color**

The **color** property allows us to specify the colour of the text.

```
h1 {
color: red;
}

p {
color: blue;
}
```

- **Background Colour**

The **background-color** property allows us to specify the background colour for the element.

```
h1 {
background-color: lightblue;
}
p {
background-color: yellow;
}
```

The CSS can be applied to HTML documents in different ways.

1. **Inline CSS**
2. **Internal CSS**
3. **External CSS**

Inline CSS

An inline style may be used to apply a unique style for a single element.

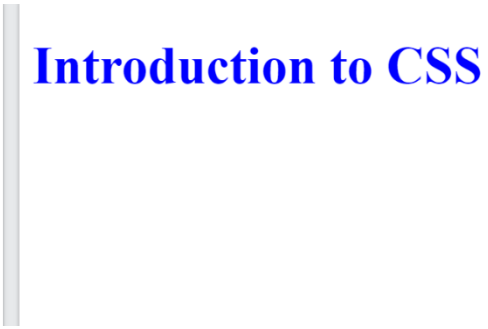
To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property. Inline styles are defined within the "**style**" attribute of the relevant element:

Example

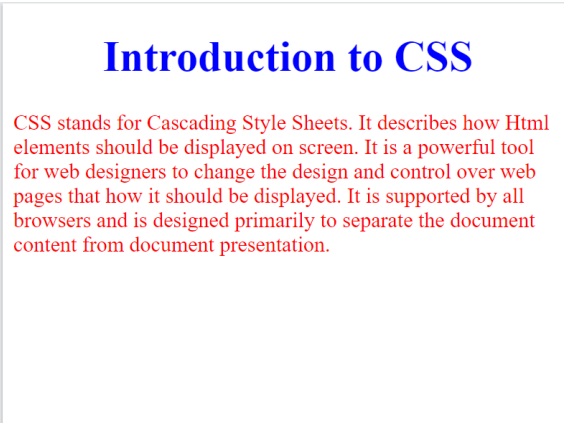
<h1 style="color: blue;">Introduction to CSS </h1>

Some Examples of Inline CSS are as follows

```
<!DOCTYPE html>
<html>
<head> </head>
<body>
<h1 style ="color:blue"> Introduction to CSS
</h1>

</body>
</html>
```




```
<!DOCTYPE html>
<html>
<body>
<h1 style ="color:blue" align="Center"> Introduction to CSS </h1>

<p style ="color:red">
CSS stands for Cascading Style Sheets. It describes how Html
elements should be displayed on screen. It is a powerful tool for
web designers to change the design and control over web pages
that how it should be displayed. It is supported by all browsers
and is designed primarily to separate the document content from
document presentation.
</p>

</body>
</html>
```



Internal CSS

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

The internal style is defined inside the **<style>** element, inside the <**head**> section.

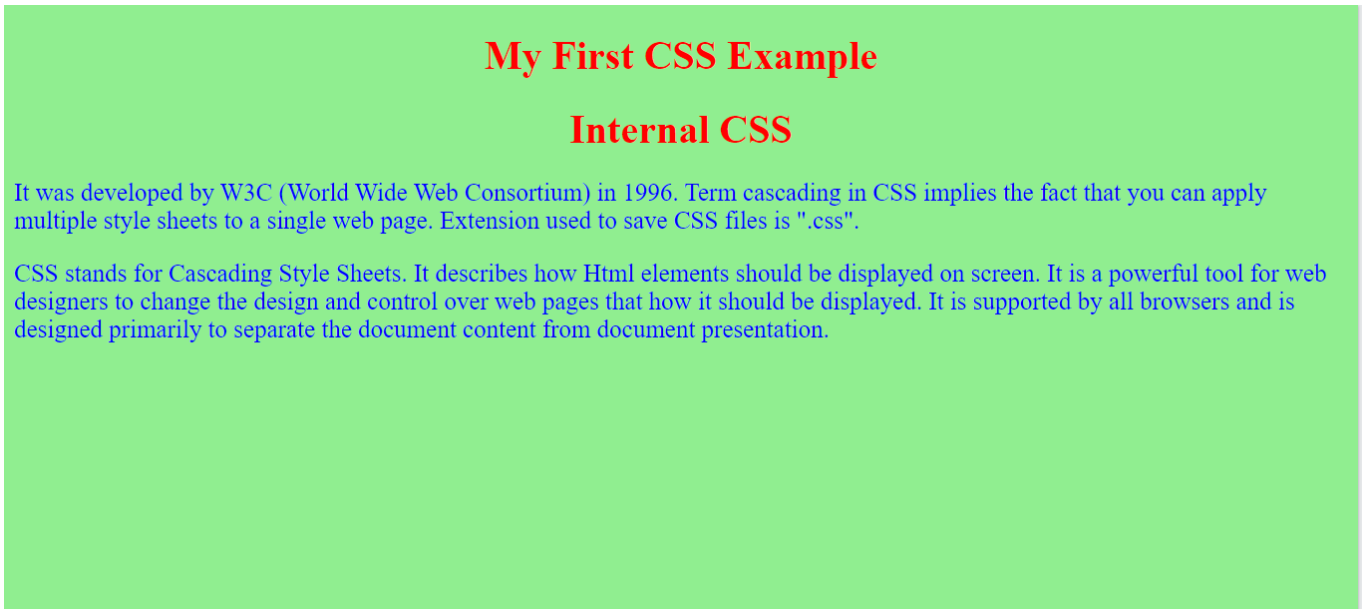
The internal CSS style that will look like below code:

```
<!DOCTYPE html>
<html>
<head>
<style>
 body {
      background-color: linen;
      }

 h1 {
  color: maroon;
  margin-left: 40px;
```

```
 }
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightgreen;
}

h1 {
  color: red;
  text-align: center;
}

p {
color: blue;
  font-family: times new roman;
  font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
<h1>Internal CSS</h1>
<p>It was developed by W3C (World Wide Web Consortium) in 1996. Term cascading in CSS implies
used to save CSS files is ".css".</p>

<p>CSS stands for Cascading Style Sheets. It describes how Html elements should be displayed o
over web pages that how it should be displayed. It is supported by all browsers and is designe
</body>
</html>
```



External CSS

With an external style sheet, we can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the **<link>** element, inside the **<head>** section.

Example

External styles are defined within the **<link>** element, inside the **<head>** section of an HTML page:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

An external style sheet can be written in any text editor, and must be saved with a **.css** extension.

The external .css file should not contain any HTML tags.

Here is how the "mystyle.css" file looks:

"mystyle.css"

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

CSS Background Color, Colors, Fonts and Sizes

Here, we will demonstrate some commonly used CSS properties.

The CSS color property defines the text color to be used.

The CSS font-family property defines the font to be used.

The CSS font-size property defines the text size to be used.

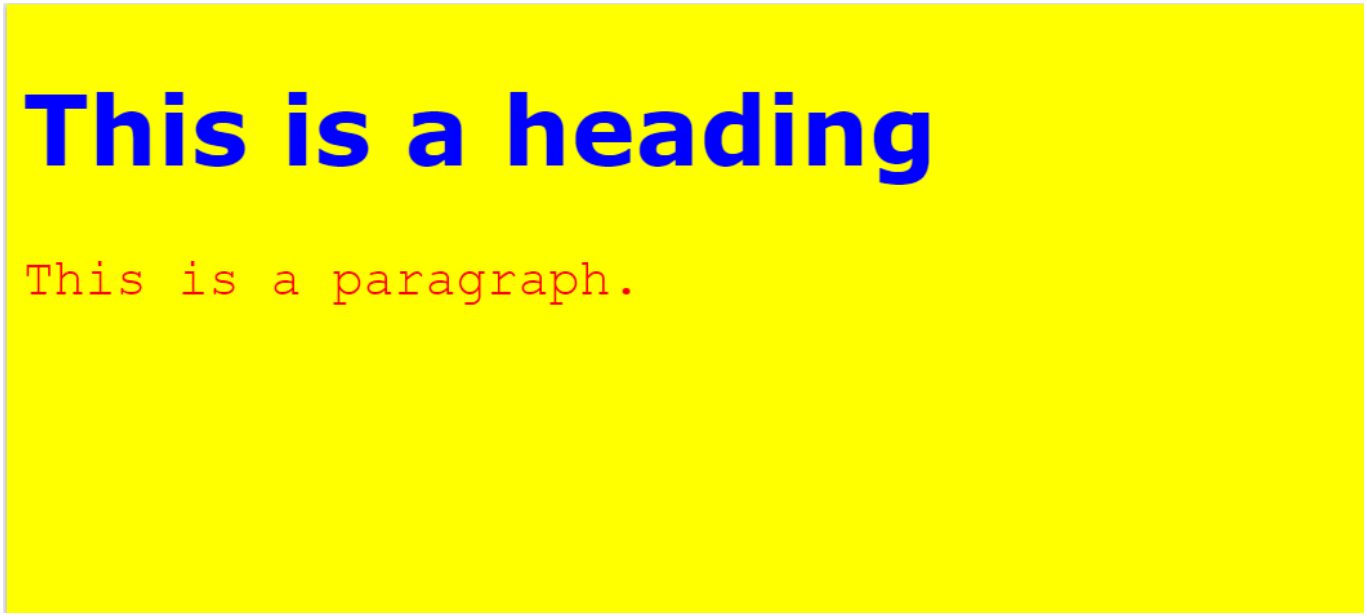
Example

Use of CSS color, font-family and font-size properties:

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

```
<head>
<style>
body {
background-color: yellow;
}
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Output:



CSS Border

The CSS border property defines a border around an HTML element.

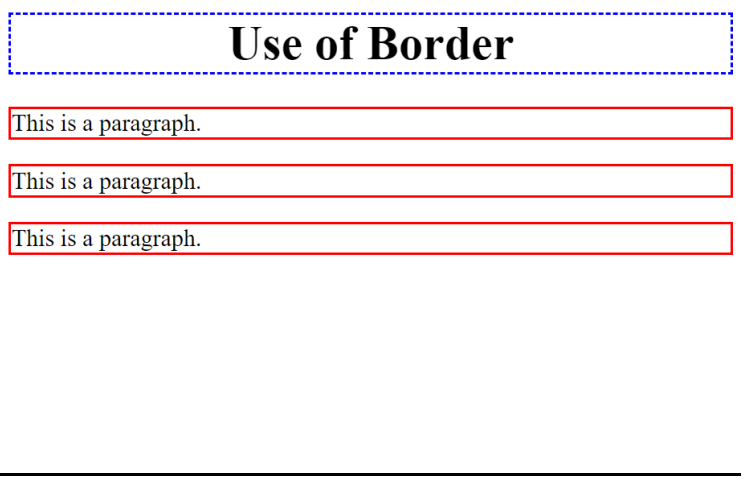
Example

Use of CSS border property:

```
p {
  border: 2px solid powderblue;
}
h1 {
   border: 2px dashed blue;
}
```

Example with output:

```
<html>
<head>
<style>
p {border: 2px solid red;}
h1{border: 2px dashed blue; }
</style>
</head>
<body>

<h1 style="text-align:center;">Use of Border</h1>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

</body>
</html>
```



CSS Padding

The CSS padding property defines a padding (space) around the text.

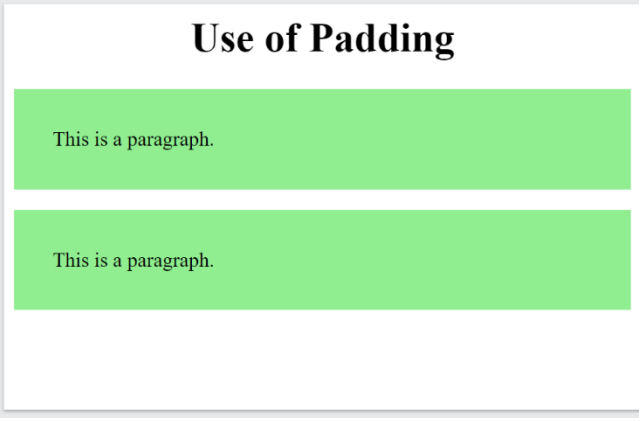
Example

Use of CSS padding properties:

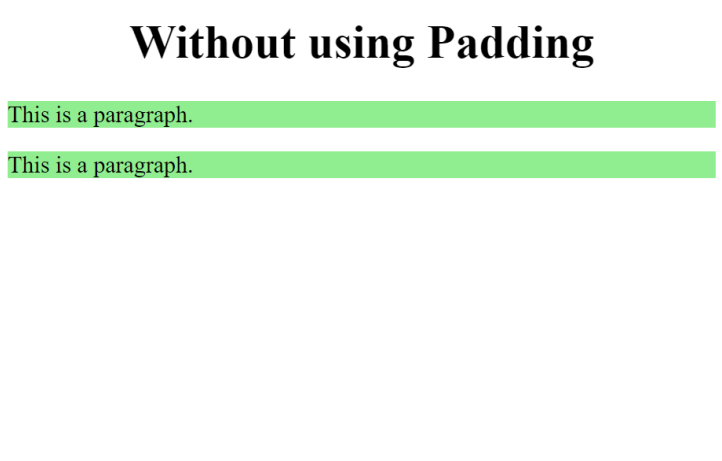
```
p {
   padding: 30px;
}
```

Example with output:

```
<html>
<head>
<style>
    p {
  padding: 30px;background-color: lightgreen;
    }
</style>
</head>
<body>

<h1 style="text-align:center">Use of Padding</h1>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

</body>
</html>
```



```
<html>
<head>
<style>
    p {
  background-color: lightgreen;
    }
</style>
</head>
<body>

<h1 style="text-align:center">Without using Padding</h1>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

</body>
</html>
```



CSS background-image Property

The background-image property sets background images for an element.

By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

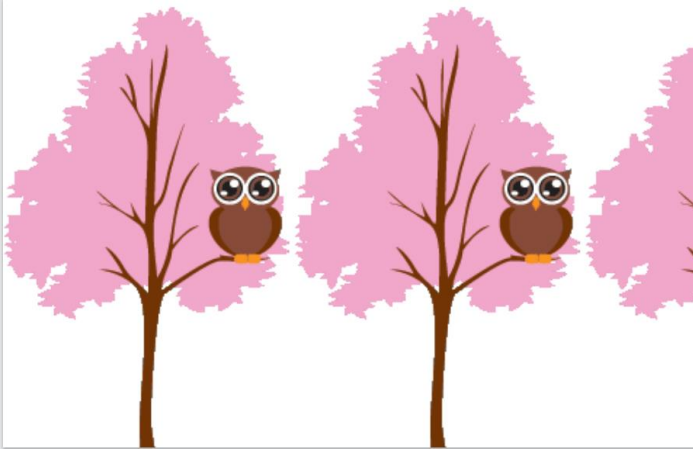
Example

Set background images for the <body> element:

```
body {
  background-image: url("img_tree.gif");
  background-color: #cccccc;
}
```

Example with output:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("img_tree.gif");
}
</style>
</head>
<body>

</body>
</html>
```



Designing a Navigation bar

A navigation bar is a user interface element within a webpage that contains links to other sections of the website. This means that no matter what page you are viewing, you can use the navigation bar to visit other sections of the website. A website navigation bar is most commonly displayed as horizontal list of links at the top of each page. It may be below the header or logo, but it is always placed before the main content of the page.

create a top navigation bar in HTML with CSS.

Top Navigation Bar

If we want to make a navigation bar in Html, then we have to follow the steps which are given below. Using these steps, we can easily create the Navigation bar.

Step 1: Firstly, we have to type the Html code in any text editor or open the existing Html file in the text editor in which we want to make a Navigation Bar.

```
<!Doctype Html>
<Html>
<Head>
<Title>
Make a Navigation Bar
</Title>
</Head>
<Body>
</Body>
</Html>
```

Step 2: Now, we have to define the <nav> tag in the <body> tag where we want to make the bar. The <nav> tag defines a set of navigation links.

```
<Body>
<nav>


</nav>
</Body>
</Html>
```

Step 3: After then, we have to define the <ul> tag, which is used to show the unordered list. And, then we have to define the list items in the <li> tag. We have to define those items which we want to show in the navigation bar.

```
<Body>
<nav>
<ul>
<li>
```

```
<a href="#"> Home </a>
</li>
<li>
<a href="#"> About </a>
</li>
<li>
<a href="#"> Contact </a>
</li>
<li> <a href="#"> Terms of use </a>
</li>
<li>
<a href="#"> Join Us </a>
</li>
</ul>
</nav>
</Body>
</Html>
```

Step 4: After then, we have to place the cursor in the <head> just after the closing of the title tag. And then, we have to define the <style> tag

```
<Head>
<Title>
Make a Navigation Bar
</Title>
<style >



</style>
</Head>
```

Step 5: Now, we have to specify different id attributes which are used to set the position, color of the navigation bar. So, we have to use the following code in the head tag. We can also change the value of properties according to our requirements.

```
<style >
body
{
height: 125vh;
margin-top: 80px;
padding: 30px;
```

```
background-size: cover;
font-family: sans-serif;
}
header
{
background-color: orange;
position: fixed;
left: 0;
right: 0;
top: 5px;
height: 30px;
display: flex;
align-items: center;
box-shadow: 0 0 25px 0 black;
}
header *
{
display: inline;
}
header li
{
margin: 20px;
}
header li a
{
color: blue;
text-decoration: none;
}
</style>
```

Step 6: After that, we have to type the <header> tag just before the opening <nav> tag. And we have to also close this tag. And, at last, we have to save the Html file and then run the file in the browser.

```
<!Doctype Html>
<Html>
<Head>
<Title>
Make a Navigation Bar
</Title>
<style type=text/css>
```

```
body
{
height: 125vh;
margin-top: 80px;
padding: 30px;
background-size: cover;
font-family: sans-serif;
}
header {
background-color: orange;
position: fixed;
left: 0;
right: 0;
top: 5px;
height: 30px;
display: flex;
align-items: center;
box-shadow: 0 0 25px 0 black;
}
header * {
display: inline;
}
header li {
margin: 20px;
}
header li a {
color: blue;
text-decoration: none;
}
</style>
</Head>
<Body>
<header>
<nav>
<ul>
<li>
<a href="#"> Home </a>
</li>
```

```
<li>
<a href="#"> About </a>
</li>
<li>
<a href="#"> Contact </a>
</li>
<li> <a href="#"> Terms of use </a>
</li>
<li>
<a href="#"> Join Us </a>
</li>
</ul>
</nav>
</header>
</Body>
</Html>
```

Output: