

# Email

---

**Email:** Electronic mail (also known as email or e-mail) is one of the most commonly used services on the Internet, allowing people to send messages to one or more recipients. Email was invented by Ray Tomlinson in 1972.

It is a system used for creating, sending and storing textual data in digital form over a network. Earlier, the e-mail system was based on Simple Mail Transfer Protocol (SMTP) mechanism, a protocol used for sending e-mails from one server to another. Today's e-mail technology uses the store-and-forward model. In this model, the user sends and receives information on their own computer terminal. However, the computer is used only for connecting to the e-mail architecture. The creation, transmission and storage of e-mail takes place, only when the connection with this e-mail architecture is established.

E-mail is one of the many technological developments that have influenced our lives. It has changed the medium of communication. So, it becomes necessary for us to have a look at the benefits and harmful effects of this popular mailing tool.

General description of electronic mail : email is a computer based method of sending messages from one computer user to another . these messages usually consist of individual pieces of text which you can send to another computer user even if the other user is not logged in (i.e. using the computer) at the time you send your message. The message can then be read at a later time. This procedure is analogous to sending and receiving a letter.

Originally, email messages were restricted to simple text, but now many systems can handle more complicated formats, such as graphics and word processed documents. When mail is received on a computer system, it is usually stored in an electronic mailbox for the recipient to read later. Electronic mailboxes are usually special files on a computer which can be accessed using various commands. Each user normally has their individual mailbox

It is straightforward to send electronic mail between users of different computer

systems which are connected to major networks. Most major academic and research institutions and companies throughout the world can now be reached by electronic mail. In addition, a growing number of individuals can be contacted in this way. In the UK, most academic and research institutions are linked by a network called JANET (or SuperJANET). This is effectively part of the Internet, so email can be exchanged with most national and international networks.

### **Email Facilities**

All email systems have the ability to send, receive and discard mail. Most systems have facilities for storing mail which is to be kept rather than discarded. It is important to discard mail which does not need to be kept, as it uses storage space on disks. Mailboxes can soon accumulate a large number of mail messages making it difficult to read and process new mail, in addition to wasting disk space

There is almost always a connection between the email system and the computer's standard file system which allows mail to be read from files or written to files. This enables greater flexibility in how the mail system is used. For example, a mail message may be prepared in a normal file using a familiar text editor and then sent by the email system. Sections of other files may be included in the mail message as well.

Most systems have a reply facility, although some of these do not always work as expected. Care should be taken when using this facility in electronic mail, as replies do not always go back to the sender.

# Advantages and disadvantages of Email

**Advantages of Email:** The benefits of e-mail are huge in number.

- **Easy to use:** E-mail frees us from the tedious task of managing data for daily use. It helps us manage our contacts, send mails quickly, maintain our mail history, store the required information, etc.
- **Speed:** An e-mail is delivered instantly and anywhere across the globe. No other service matches the e-mail in terms of speed.
- **Easy to prioritize:** Because e-mails come with a subject line, it is easy to prioritize them and ignore the unwanted ones.
- **Reliable and secure:** Constant efforts are being taken to improve the security in electronic mails. It makes e-mail one of the secured ways of communication.
- **Informal and conversational:** The language used in e-mails is generally simple and thus, makes the process of communication informal. Sending and receiving e-mails takes less time, so it can be used as a tool for interaction
- **Automated e-mails:** It is possible to send automated e-mails using special programs like autoresponders. The autoresponders reply only to those messages with generalized, prewritten text messages.
- **Environment friendly:** Postal mails use paper as a medium to send letters. Electronic mail therefore, prevents a large number of trees from getting axed. It also saves the fuel needed for transportation.
- **Use of graphics:** Colorful greeting cards and interesting pictures can be sent through e-mails. This adds value to the e-mail service.
- **Advertising tool:** Nowadays, many individuals and companies are using the e-mail service to advertise their products, services, etc.
- **Info at your fingertips:** Storing data online means less large, space taking file cabinets, folders and shelves. You can access information far quicker if you learn how to use email this way. **Leverage:** Send the same message to any number of people. Adaptations are simple, too. If you have a product or service to sell, email is an effective medium to get

your message out. Send reminders to yourself. Do you use more than one account? Email yourself messages from work to home or vice versa.

**Disadvantages of Email:** The e-mail service, though beneficial in our day-to-day life, has got its own drawbacks that are off late coming to the fore.

- **Viruses:** There are computer programs which have the potential to harm a computer system.
- **Spam:** E-mails when used to send unsolicited messages and unwanted advertisements create nuisance and are termed as Spam. Checking and deleting these unwanted mails can unnecessarily consume a lot of time. It has therefore, become necessary to block or filter unwanted e-mails by means of spam filters. Spamming is practiced by sending hoax e-mails. E-mail spoofing is another common practice, used for spamming. The act of spoofing involves deceiving the recipient by altering e-mail headers or addresses from which the mail is sent.
- **Hacking:** The act of breaking into computer security is termed as hacking. In this form of security breach, e-mails are intercepted by hackers. An e-mail before being delivered to the recipient, "bounces" between servers located in different parts of the world; hence, it can be hacked by a professional hacker.
- **Misinterpretation:** One has to be careful while posting content through an e-mail. If typed in a hurry, the matter could be misinterpreted.
- **Lengthy mails:** If the mail is too long and not properly presented, the reader may lose interest in reading it.
- **Not suitable for business:** Since the content posted via e-mails is considered informal, there is a chance of business documents going unnoticed. Thus, urgent transactions and especially those requiring signatures are not managed through e-mails.
- **Lacks the personal touch:** Some things are best left untyped. Email will never beat a hand written card or letter when it comes to relationships.

Every new technology enters the social world with its share of benefits and drawbacks. Different people interpret and utilize it in different ways. In order to make the most of the available technology, users should try to understand both, the positive and negative sides of the tool they use. No technology can be deemed totally perfect!

# User id and Password

---

**User id and Password:** A computer system uses user ids and passwords together to grant access to the system. You need the correct combination of user id and password, to access your account.

**User id:** The user id identifies you to the computer.

User ids are typically some form of your name. (Your last name, for example). A user id must be unique throughout the computer system. This allows the computer to distinguish between you and some other person. One confusing issue is that different systems refer to a user id (the thing that identifies you to the computer) with different names. Some that you may run across are:

- Login ID
- Username
- User id

But, they all refer to that special name that identifies you to that particular computer system. Your user id is also used for communication with other people through electronic mail. Since the user id identifies you to the computer, it is natural for it to be used to identify you to other computer systems. For example: The person Joe, user could have this user id. "jbu3470". The user id is made up of the persons' initials and the digits of some personal identification number.

## Password

To prevent other people from using your account via your user id, you are required to have a password. A password allows you and only you to access the computer system through your user id. It basically proves to the computer system that you are who you say you are. So it should be obvious that you **DO NOT GIVE YOUR PASSWORD TO ANYONE!!!**

**User ID and Password Rules:** Please review the User ID and Password guidelines below before selecting a User ID or selecting or changing your Password,

User ID:

- User IDs must be 7-14 characters
- User IDs must contain at least one letter; numbers are allowed, but not required
- User IDs cannot contain spaces
- User IDs cannot contain your Social Security Number, Tax Identification Number, or customer Access Number
- No special characters are allowed, such as: ! @ # \$ % ^ &
- Use of an underscore is allowed but not required: \_
- Do not use your Password as your User ID

**Password:**

- Passwords must be 7-14 characters
- Passwords must include at least one letter and one number
- Passwords cannot contain spaces
- Semicolons cannot be part of a Password Passwords are case- sensitive
- Do not use your User ID as your Password
- If you forget your User ID or Password, you can retrieve them through the "User ID & Password Help" link.

## Email addresses

---

**Email addresses:** Electronic mail must be addressed properly, in much the same way as a normal letter needs to be addressed, in order that it can reach its intended recipient. However, with electronic mail the addressing has to be slightly different and must be more precise than a conventional mail address.

Most email uses is addressed using internet addresses. These are supported by what is known as the Domain Name Server (DNS). This is an internationally distributed naming scheme and the components of each address are structured in a hierarchical manner.

These start with the mailname of the person, followed by various address components, ending with the "Top level" domain (often a country code). In this respect, the addressing is in the same order as that of a conventional letter.

- Email addresses (both for senders and recipients) are two strings separated by the character "@" (the "at sign"): user@domain
- The right-hand part describes the domain name involved, and the left-hand part refers to the user who belongs to that domain. One or more mail servers correspond to each domain.
- Lowercase letters from a to z; Digits
- The characters ".", "\_", and "-" (full stop, underscore, and hyphen)

In practice, an email address often looks something like this:

**firstname.lastname@provider.domain For example**

E.V.Nebor@bham.ac.uk

The mailname comes first, indicating who the email is for. The "@" sign separates the mailname from the site. The next components indicate details of the system to which the email is sent.

bham = Birmingham ac = academic

uk = Great Britain

The final component ("top level" domain) is normally the ISO3166 standard two letter country code. There are, however, two main types of exception. One is Great Britain (gb) which uses "uk". The other exceptions are mainly some of American domains (eg edu, mil, gov).

These are at the same "level" in the naming scheme as countries. Addresses of this form are also known as rfc822 addresses, because that is the name of the Internet standard in which the addresses were originally described.

**Partially qualified addresses:** Some systems will allow the address to be specified in an incomplete form. For example,

**E.V.Nebor@bham.ac.uk**

could be expressed as

**E.V.Nebor@bham**

This is known as a partially qualified address. The use errors and much confusion. The full address should always be used.

## Message Components

---

**Header:** The header of an e-mail address is where all the important addressing information is kept, along with data about the message itself. The following table describes the elements that make up the header of an e-mail.

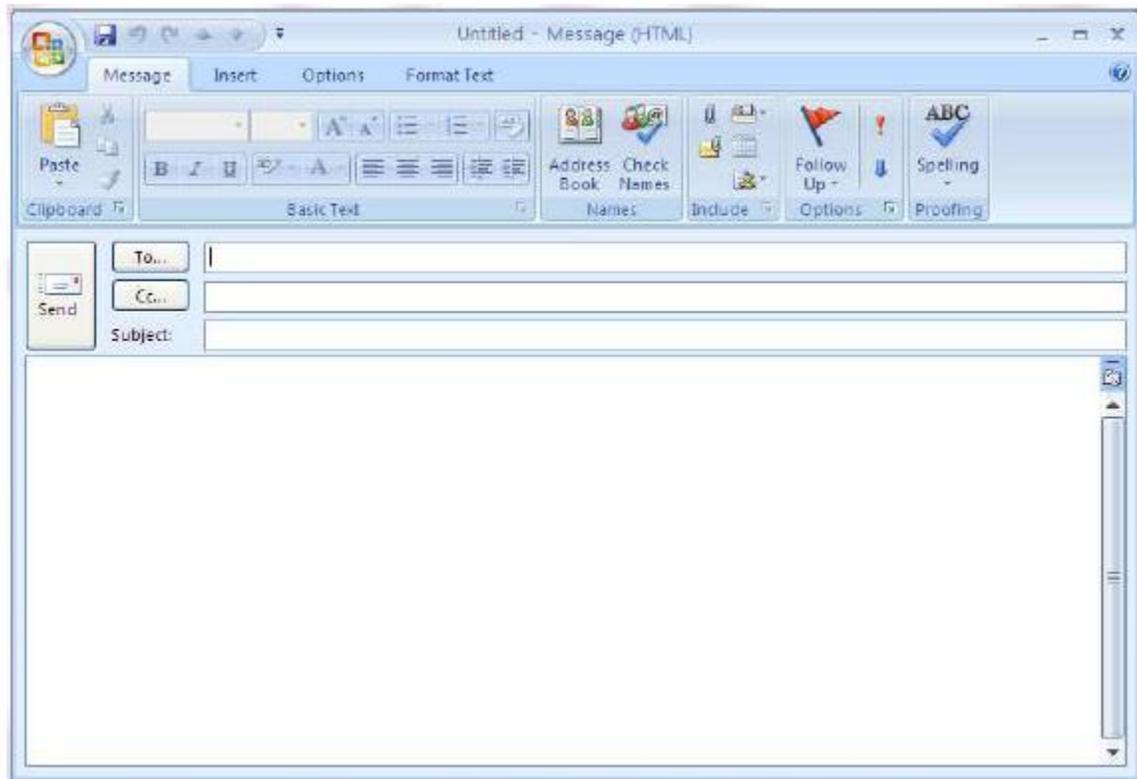
<b>To</b>	The <b>TO</b> Field is where the email address of the messages recipient is entered.
<b>Subject</b>	The <b>subject</b> is essentially a title, or a very short summary, of an e-mail. Ideally it should contain some sort of reference to the emails content
<b>CC</b>	To <b>Carbon Copy (CC)</b> a message to someone is to send them a copy of the e-mail, even though they are not the e-mail's intended recipient. This is a common practice in business communications, where many people in an organization often need to be made aware of certain communications, even if they are not actively involved in the conversation.
<b>BCC</b>	<b>BCC stands for Blind Carbon Copy.</b> Any people included in the BCC recipient list of an e-mail receive a copy of the message, but their name is not included in the message headers, and no one else who received the message knows that they were sent a copy.

**Body:** The body of the message is, of course, the most important part.

Everything you need to say is entered here, and this is what the recipient will be reading.

Message Composition Steps are:

Launching your email program is usually as simple as clicking on the email icon on your desktop. Once the program is open, write a message by clicking on the Compose Mail, New Message or similar button.



- In the new message window, type the e-mail addresses of the recipients in the To field, or use the Address book to select one or more of the addresses.
- In the CC field, enter the address of anyone you want to receive a copy of the message.

**Type the e-mail address or choose it from the address book.**

- Type the subject of the message.
- Now type your message.
- Finally, click on the Send button.

# Composing and Sending Email

Before sending an email, we need to compose a message. When we are composing an email message, we specify the following things:

- Sender's address in To field
- Cc (if required)
- Bcc (if required)
- Subject of email message
- Text
- Signature

You should specify the correct email address; otherwise it will send an error back to the sender.

Once you have specified all the above parameters, It's time to send the email. The mailer program provides a Send button to send email, when you click Send, it is sent to the mail server and a message **mail sent successfully** is shown at the above.

## Reading Email

Every email program offers you an interface to access email messages. Like in Gmail, emails are stored under different tabs such as primary, social, and promotion. When you click one of tab, it displays a list of emails under that tab.

In order to read an email, you just have to click on that email. Once you click a particular email, it gets opened.

The opened email may have some file attached with it. The attachments are shown at the bottom of the opened email with an option called **download attachment**.

## Replying Email

After reading an email, you may have to reply that email. To reply an email, click **Reply** option shown at the bottom of the opened email.

Once you click on Reply, it will automatically copy the sender's address in to the To field. Below the To field, there is a text box where you can type the message.

Once you are done with entering message, click Send button. It's that easy. Your email is sent.

## Forwarding Email

It is also possible to send a copy of the message that you have received along with your own comments if you want. This can be done using **forward** button available in mail client software.

The difference between replying and forwarding an email is that when you reply a message to a person who has send the mail but while forwarding you can send it to anyone.

When you receive a forwarded message, the message is marked with a > character in front of each line and **Subject:** field is prefixed with **Fw.**

## Deleting Email

If you don't want to keep email into your inbox, you can delete it by simply selecting the message from the message list and clicking **delete** or pressing the appropriate command.

Some mail clients offers the deleted mails to be stored in a folder called deleted items or trash from where you can recover a deleted email.

Now a day, the mail client comes with enhanced features such as attachment, address book, and MIME support. Here in this chapter we will discuss all of these features which will give you a better understanding of added feature of a mail client program.

## Attachment

Ability to attach file(s) along with the message is one of the most useful features of email. The attachment may be a **word document**, **PowerPoint presentation**, **audio/video files**, or **images**.

- In order to attach file(s) to an email, click the attach button. As a result, a dialog box appears asking for specifying the name and location of the file you want to attach.
- Once you have selected the appropriate file, it is attached to the mail.
- Usually a paper clip icon appears in the email which indicates that it has an attachment.
- When adding an attachment it is better to compress the attached files so as to reduce the file size and save transmission time as sending and downloading large files consumes a lot of space and time.

## Address Book

Address book feature of a mail program allows the users to store information about the people whom they communicate regularly by sending emails. Here are some of the key features of an Address book:

- Address book includes the nick names, email addresses, phone number etc. of the people.
- Using address book allows us not to memorize email of address of a person, you just have to select recipient name from the list.
- When you select a particular name from the list, the corresponding email address link automatically get inserted in to the **To:** field.
- Address book also allows creating a group so that you can send a email to very member of the group at once instead of giving each person email address one by one.

## Mailer Features

---

**Mailer Features:** Most mailers provide common features which provide functionality for manipulating your mail box contents, composing messages and saving messages to the disk. The features are:

**Compose, File and Reply:** A Compose button provides following features :

New – Compose a message from scratch. Reply – Reply to the current message.

Forward – Pass the message on.

Vacation – You are going away and want automatic responses to be generated, and have email saved.

**A File button has the following functionality:**

- Save – Save the current message into a file on disk.
- Insert – Include a file in the body of the message being composed.

- Exit – Leave the mailer.
- Open – Open a file from the disk.
- Attach – Append a file to a message.

### **A Reply button consists of following items:**

- To sender.
- Include
- Include
- Include bracketed

## **E-mail Inner Working:**

If we split the mailing of a letter into three phases:

- In phase one, the steps needed to perform are compose, address, package, stamp and deposit the letter in a suitable place for pick up.
- In phase two, a mail person, on a fixed schedule, retrieves the letter from its place of deposit. Then the letter, if correctly addressed and with the proper postage, is routed to its final destination mailbox.
- In phase three, the recipient check for mail, retrieves the letter from the mailbox, opens the envelope, reads the mail, and perhaps files it away.
- Similar phases need to be carried out in the electronic setting:
- Mailer, Mail Server, and Mailbox – Three mail components are necessary for the email system to work.
- Mailers A mailer is also called a mail program, mail application, or mail client. A mailer is the software that allows you to manage, read, and compose email.
- Mail Servers The mail server is a computer whose function is to receive, store and deliver email.
- Mailboxes An electronic mailbox is a disk file specifically formatted to hold email messages and information about them,.

**There are several different ways in which users typically obtain their emails.**

- Store and Forward Features
- Central Mail Spool and IMAP

Email working follows the client server approach. In this client is the mailer i.e. the mail application or mail program and server is a device that manages emails.

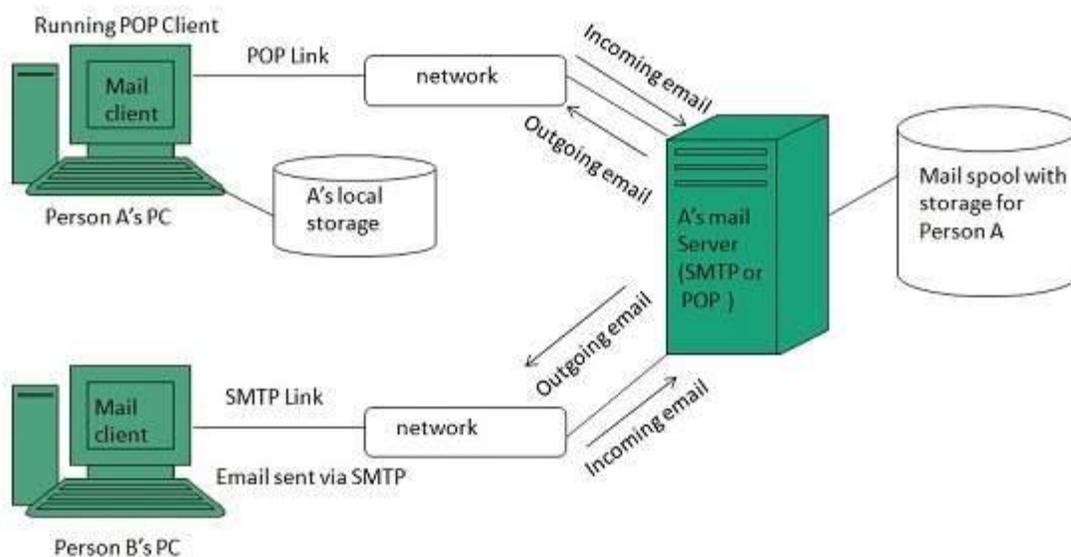
Following example will take you through the basic steps involved in sending and receiving emails and will give you a better understanding of working of email system:

- Suppose person A wants to send an email message to person B.
- Person A composes the messages using a mailer program i.e. mail client and then select Send option.
- The message is routed to **Simple Mail Transfer Protocol** to person B's mail server.
- The mail server stores the email message on disk in an area designated for person B.

The disk space area on mail server is called mail spool.

- Now, suppose person B is running a POP client and knows how to communicate with B's mail server.
- It will periodically poll the POP server to check if any new email has arrived for B. As in this case, person B has sent an email for person B, so email is forwarded over the network to B's PC. This message is now stored on person B's PC.

The following diagram gives pictorial representation of the steps discussed above:



Here we will discuss the operations that can be performed on an e-mail. But first of all

we will learn how to create an email account.

## Creating Email Account

There are various email service provider available such as **Gmail**, **hotmail**, **ymail**, **rediff mail** etc. Here we will learn how to create an account using Gmail.

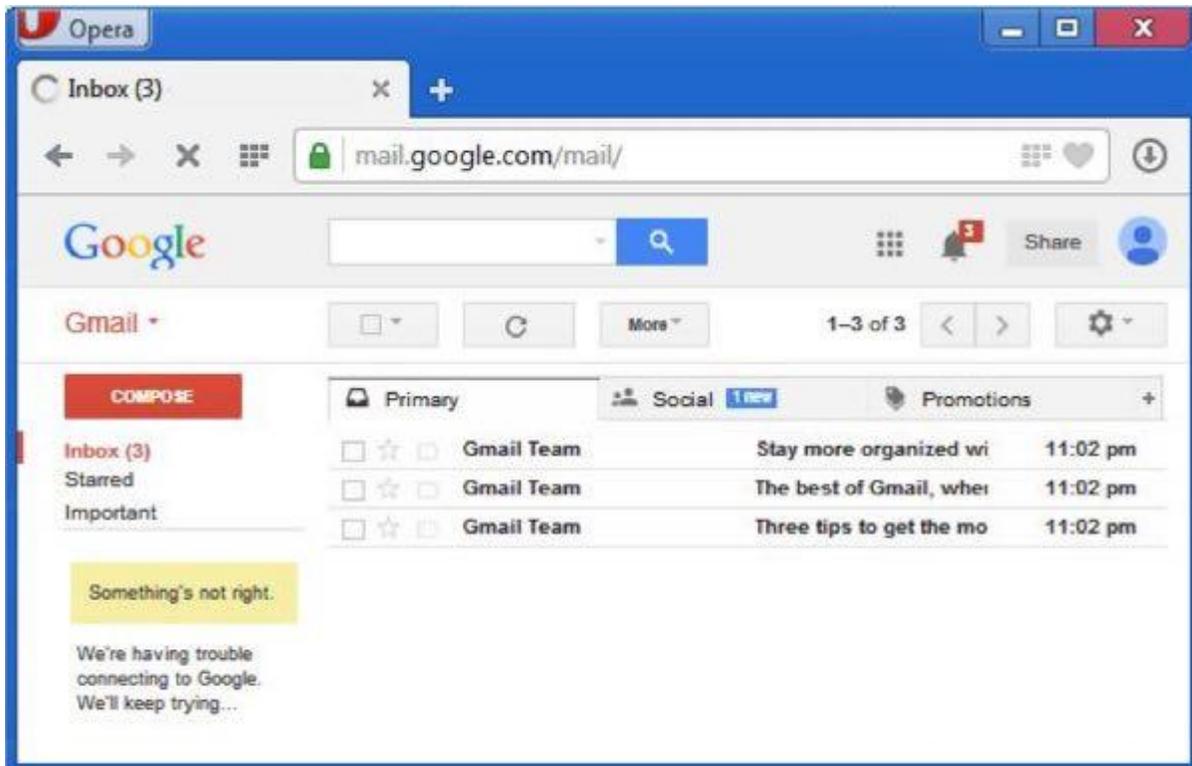
- Open gmail.com and click **create an account**.
- Now a form will appear. Fill your details here and click **Next Step**.



The screenshot shows the Google Accounts sign-up page in an Opera browser. The browser window title is "Opera" and the tab is "Google Accounts". The address bar shows "accounts.google.com/SignUp". The page features the Google logo and a "Sign in" button. The main heading is "Create your Google Account". The form fields are:

- Name:** Two input boxes containing "Rahul" and "Kalotra".
- Choose your username:** A single input box containing "rahulkalotra20@gmail.com".
- Create a password:** A single input box with masked characters "\*\*\*\*\*".
- Confirm your password:** A single input box with masked characters "\*\*\*\*\*".
- Birthday:** A dropdown menu showing "August", a spin box showing "18", and a year input box showing "1989".
- Gender:** A label with a dropdown arrow, partially visible at the bottom.

- This step allows you to add your picture. If you don't want to upload now, you can do it later. Click **Next Step**.
- Now a welcome window appears. Click **Continue to Gmail**.
- Wow!! You are done with creating your email account with Gmail. It's that easy. Isn't it?
- Now you will see your Gmail account as shown in the following image:



### Key Points:

- Gmail manages the mail into three categories namely **Primary**, **Social** and **Promotions**.
- **Compose** option is given at the right to compose an email message.

**Inbox, Starred, Sent mail, Drafts** options are available on the left pane which allows you to keep track of your emails.

# E-mail Management

---

**E-mail Management:** Email is now the de facto standard for business communication across organizations at this time. Recent research shows the extensive use of emails in business and its projected growth in the future presents major information management challenges and risks for many organizations. Just as any type of business information and record, email must be included as part of, and adhered to the organizational standards addressing information and records:

- Capture
- Classification
- Storage
- Preservation
- Management
- Destruction

The vast quantities of emails held in inboxes, sent folders, and deleted item folders put the organization at risk and adversely impact the performance of email servers in the organizations. Email servers were never designed to act as repositories for such great quantities of emails and move control of this information away from the organization. Without the management of Emails, it is difficult for organization to meet their legal preservation requirements in the event of litigation and government investigations, increasing the effort and cost in responding to ediscovery and disclosure.

Email management systems centrally capture emails created and received by employees. Using a classification scheme to manage this content, retention periods and access controls can be applied to manage emails. Metadata associated with the emails can be captured to allow this information to be managed and retrieved. Email active archiving is one of the most common applications for email management. At their most basic, these solutions either copy or remove messages from the messaging application and store it at another location. Some approaches copy all messages coming into and out of the messaging application in real time, while others will physically remove the messages from the message store. In some

of these solutions, the messages are not archived, but attachments are and are replaced with either a stub or an outright link within the message.

These email management systems, while attractive and effective, are merely part of an overall solution. Organizations need to have records management programs, consisting of organization-wide policies and procedures, staff and activities, in which these computer applications can be successfully utilized.

Here are four simple email management rules to help you keep control of your inbox:

**Let your email program manage** your email as much as possible. Email management starts with setting up and using filters. If you're using an email program such as Outlook, you can configure email rules to send your spam directly to the trash

**Do not check your email on demand.** You don't need to see every piece of email the second it

arrives. If you're using an email program that announces the arrival of new email, turn off the program's announcement features, such as making a sound or having a pop-up screen announce the arrival of email. Checking email on demand can seriously interfere with whatever other tasks you're trying to accomplish because most people will read email when they check it.

**Don't read and answer your email all day long.** You may get anywhere from a handful to hundreds of emails each day that need to be answered, but they don't need to be answered immediately, interrupting whatever else you're doing. Instead, set aside a particular time each day to review and answer your email. Schedule the hour or whatever time it takes you to answer the volume of email you get, and stick to that schedule as regularly as possible.

**Don't answer your email at your most productive time of day.** For me, (and for many others, I suspect), my most productive work time is the morning. If I start my work day by answering my email, I lose the time that I'm at my most creative. If I'm writing a piece, for instance, it takes me twice as long to compose it in the afternoon or evening than it would in the morning, when I feel fresh and alert.

Answering email, on the other hand, isn't usually a task that calls for a great deal of

creativity. So by ignoring my email until the late afternoon, and answering it then, I get the dual benefit of saving my most productive time.

## **MIME (Multipurpose Internet Mail Extensions)**

**MIME (Multipurpose Internet Mail Extensions):** MIME is a standard which was proposed by Bell

Communications in 1991 in order to expand upon the limited capabilities of email, and in particular to allow documents (such as images, sound, and text) to be inserted in a message. It was originally defined by RFCs 1341 and 1342 in June 1992.

Using headers,

MIME describes the type of message content and the encoding used.

**MIME adds the following features to email service:**

- Be able to send multiple attachments with a single message;
- Unlimited message length;
- Use of character sets other than ASCII code; Use of rich text (layouts, fonts, colours, etc)
- Binary attachments (executables, images, audio or video files, etc.), which may be divided if needed.

Common MIME Types are listed in the table below:

Type	Subtype	Description of Content Type	File Extensions
Application	Postscript	Printable postscript document	.eps, .ps
	Tex	TEX document	.tex
	troff	Printable troff document	.t, .tr, .roff
Audio	Aiff	Apple sound	.aif, .aiff, .aifc
	Au	Sun Microsystems sound	.au, .snd
	Midi	Musical Instrument Digital Interface	.midi, .mid
	Realaudio	Progressive Networks sound	.ra, .ram
	wav	Microsoft sound	.wav
Image	Gif	Graphics Interchange Format	.gif
	Jpeg	Joint Photographic Experts Group	.jpeg, .jpg, .jpe
	Png	Portable Network Graphics	.png
	Tiff	Tagged Image File Format	.tiff, .tif
Model	Vrml	Virtual Reality Modeling Language	.wrl
Text	Html	Hypertext Markup Language	.html, .htm
	Plain	Unformatted text	.txt
	Sgml	Standard Generalized Markup Language	.sgml
Video	Avi	Microsoft Audio Video Interleaved	.avi
	Mpeg	Moving Picture Experts Group	.mpeg, .mpg
	Quicktime	Apple QuickTime movie	.qt, .mov
	Sgi-movie	Silicon Graphics movie	.movie

**News Groups:** A newsgroup is a continuous public discussion about a particular topic. You can join a newsgroup at any time to become part of a huge conversation between hundreds or even thousands of people

A newsgroup is a discussion about a particular subject consisting of notes written to a central Internet site and redistributed through Usenet, a worldwide network of news discussion groups. Usenet uses the Network News Transfer Protocol (NNTP).

Newsgroups are organized into subject hierarchies, with the first few letters of the newsgroup name indicating the major subject category and sub-categories represented by a subtopic name. Many subjects have kind of group and what kind of postings you will find there. Some major subject categories are:

- news,
- rec (recreation),
- soc (society),
- sci (science),
- comp (computers),
- and so forth (there are many more).
- Alt talk

**Use of Newsgroups:** We can use newsgroups for various purposes, for example:

- Newsgroups are an excellent way to find out web sites to visit in your particular area of interest or just pick up detailed information about area of interest.
- You can buy and sell stuff. People often advertise things for sale in some newsgroup.

Users can post to existing newsgroups, respond to previous posts, and create new newsgroups.

Newcomers to newsgroups are requested to learn basic Usenet netiquette and to get familiar with a newsgroup before posting to it. A frequently-asked questions is provided. The rules can be found when you start to enter the Usenet through your browser or an online service. You can subscribe to the postings on a particular newsgroup.

Some newsgroups are moderated by a designated person who decides which postings to allow or to remove. Most newsgroups are unmoderated.

# Usenet (USER NETWORK)

Like mailing lists Usenet is also a way of sharing information. It was started by Tom Truscott and Jim Ellis in 1979. Initially it was limited to two sites but today there are thousands of Usenet sites involving millions of people.

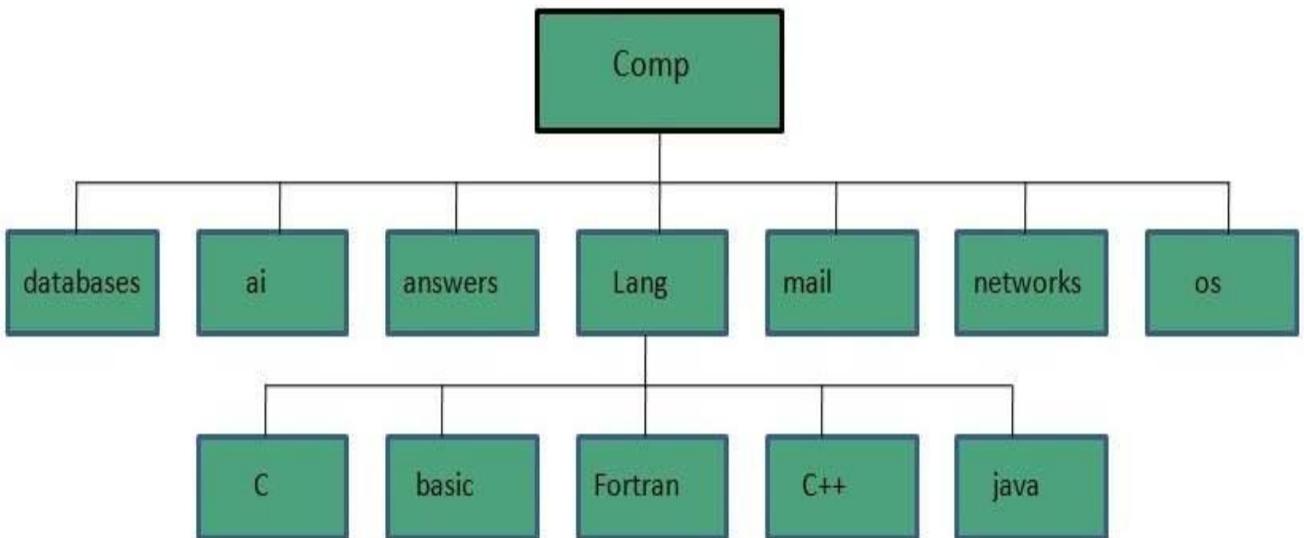
**Usenet** is a kind of discussion group where people can share views on topic of their interest. The article posted to a newsgroup becomes available to all readers of the newsgroup.

## Newsgroup Classification

There exist a number of newsgroups distributed all around the world. These are identified using a hierarchical naming system in which each newsgroup is assigned a unique name that consists of alphabetic strings separated by periods.

The leftmost portion of the name represents the top-level category of the newsgroup followed by subtopic. The subtopic can further be subdivided and subdivided even further (if needed).

For example, the newsgroup **comp.lang.C++** contains discussion on **C++** language. The leftmost part **comp** classifies the newsgroup as one that contains discussion of computer related topics. The second part identifies one of the subtopic **lang** that related to computer languages. The third part identifies one of the computer languages, in this case **C++**.



Newsgroup hierarchy of 'Comp'

The following table shows the top-level hierarchies of Usenet Newsgroup:

Comp.*	Computer related topics including computer hardware, software, languages etc.	Comp.lang.java.beans Comp.database.oracle
News.*	Newsgroup and Usenet topics	News.software.nntp
Rec.*	Artistic activities, hobbies, or recreational activities such as books, movies etc.	Rec.arts.animation
Sci.*	Scientific topics	Sci.bio.botany
Soc.*	Social issues and various culture	Soc.culture.india
Talk.*	Conventional subjects such as religion, politics etc.	Soc.politics.india
Humanities.*	Art, literature, philosophy and culture	Humanities.classics
Misc.*	Miscellaneous topics i.e. issues that may not fit into other categories	Misc.answers Misc.books.technical

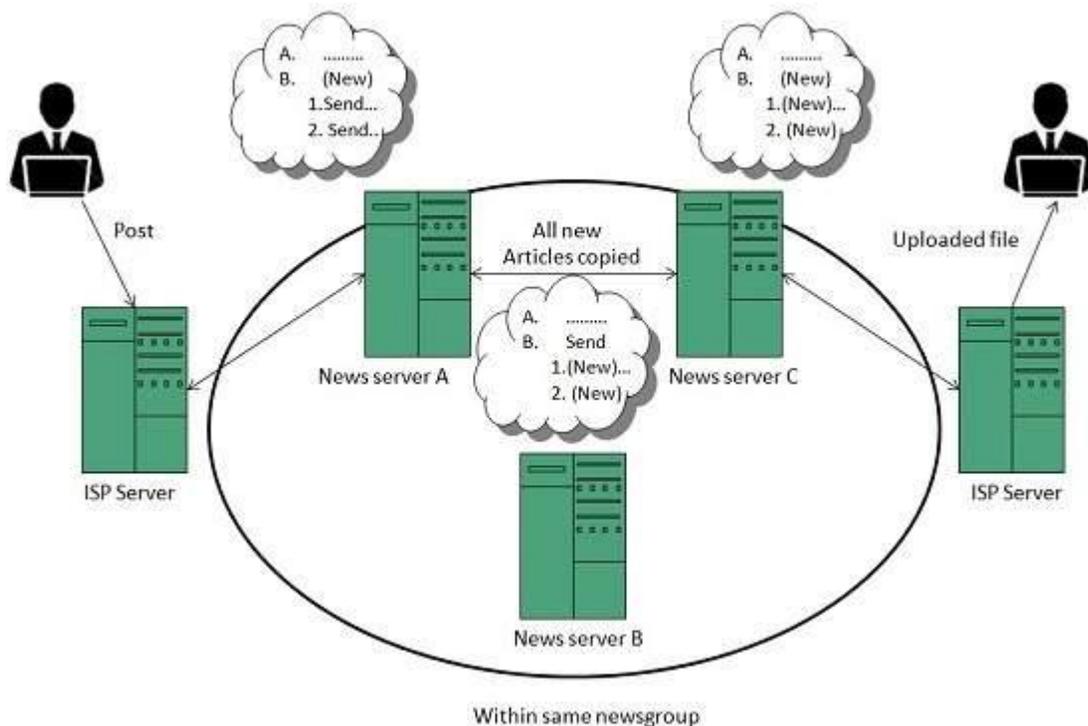
## Working of Usenet Newsgroup

When a newsreader such as outlook express connects to a news server, it downloads all the new messages posted in the subscribed newsgroup. We can either reply a message after reading or post a news article to the news server.

The article posted to a news server is appended to the file maintained for that newsgroup. Then the news server shares article with other news servers that are connected to it.

Then each news server compares if both carry the same newsgroup. If yes, then by comparing the files it checks that if there are any new articles in the file, if so they are appended to the file.

The updated file of the news servers is then sent to other news servers connected to it. This process is continued until all of the news servers have updated information.



## Reading Articles

If user wants to read article, user has to connect to the news server using the newsreader. The newsreader will then display a list of newsgroups available on the news server where user can subscribe to any of the news group. After subscription the newsreader will automatically download articles from the newsgroup.

After reading the article user can either post a reply to newsgroup or reply to sender by email. The newsreader saves information about the subscribed newsgroups and articles read by the user in each group.

## Posting an Article

In order to send new article to a newsgroup, user first need to compose an article and specify the names of the newsgroup to whom he/she wants to send. An article can be sent to one or more newsgroup at a time provided all the newsgroups are on same news server.

It is also possible to cancel the article that you have posted but if someone has downloaded an article before cancellation then that person will be able to read the article.

## Replying an Article

After reading the article user can either post a reply to newsgroup or reply to sender by email. There are two options available Reply and Reply group. Using Reply, the reply

mail will be sent to the autor of the article while Reply group will send a reply to whole of the newsgroup.

## Cancelling an Article

To cacle an article after it is sent, select the message and click Message > Cancel message. It will cacle the message from the news server. But if someone has downloaded an article before cancellation then that person will be able to read the article.

## Usenet netiquette

While posting an article on a newsgroup, one should follow some rules of netiquette as listed below:

- Spend some time in understanding a newsgroup when you join it for first time.
- Article posted by you should be easy to read, concise and grammatically correct.
- Information should be relevant to the article title.
- Don't post same article to multiple newsgroups.
- Avoid providing your business email address while subscribing to a newsgroup as may be used by spammers.
- Avoid using capital letters as someone may interpret as shouting.
- Prefer to use plain text wherever possible in your article.

## Mailing list vs. Newsgroup

S.N.	Mailing List	Newsgroup
1.	Messages are delivered to individual mailboxes of subscribed member of group.	Messages are not posted to individual mailboxes but can be viewed by anyone who has subscribed to that newsgroup.
2.	Working with mailing list is easier than newsgroup. It is easy to compose and receive emails.	Working with a particular newsgroup requires proper knowledge of that newsgroup.
3.	In order to send or receive mails, you required an email program.	It requires a newsgroup reader.

4.	Messages are delivered to certain group of people.	Messages are available to public.
5.	Mailing list does not support threaded discussion.	Newsgroup supports threaded discussion.
6.	Messages delivered to listed subscribers can not be cancelled.	Article posted on a newsgroup can be cancelled.

## Mailing Lists

---

**Mailing Lists:** A mailing list is a collection of names and addresses used by an individual or an organization to send material to multiple recipients. The term is often extended to include the people subscribed to such a list, so the group of subscribers is referred to as "the mailing list", or simply "the list".

**Types of mailing lists:** At least two types of mailing lists can be defined:

An announcement list is closer to the original sense, where a "mailing list" of people was used as a recipient for newsletters, periodicals or advertising. Traditionally this was done through the postal system.

With the rise of email, the electronic mailing list became popular. The second type allows members to post their own items which are broadcast to all of the other mailing list members. This second category is usually known as a discussion list.

## More definitions

- When similar or identical material is sent out to all subscribers on a mailing list, it is often referred to as a mailshot or a blast. A list for such use can also be referred to as a distribution list. In legitimate mailing lists, individuals can subscribe or unsubscribe themselves.
- Mailing lists are often rented or sold. If rented, the renter agrees to use the mailing list for only contractually agreed-upon times. The mailing list owner typically enforces this by "salting" (known as "seeding" in direct mail) the mailing list with fake addresses and creating new salts for each time the list is rented. Unscrupulous renters may attempt to bypass salts by renting several lists and merging them to find the common, valid addresses.
- Mailing list brokers exist to help organizations rent their lists. For some organizations, such as specialized niche publications or charitable groups, their lists may be some of their most valuable assets, and mailing list brokers help them maximize the value of their lists.
- A mailing list is simply a list of e-mail addresses of people that are interested in the same subject, are members of the same work group, or who are taking class together. When a member of the list sends a note to the group's special address, the e-mail is broadcast to all of the members of the list. The key advantage of a mailing list over things such as web-based discussion is that as new message becomes available they are immediately delivered to the participants' mailboxes.

## Chat Rooms

---

**Chat Rooms:** A chat room is a Web site, part of a Web site, or part of an online service such as America Online, that provides a venue for communities of users with a common interest to communicate in real time. Forums and discussion

groups, in comparison, allow users to post messages but don't have the capacity for interactive messaging. Most chat rooms don't require users to have any special software; those that do, such as Internet Relay Chat (IRC) allow users to download it from the Internet.

Chat room users register for the chat room of their choice, choose a user name and password, and log into a particular room (most sites have multiple chat rooms). Inside the chat room, generally there is a list of the people currently online, who also are alerted that another person has entered the chat room. To chat, users type a message into a text box. The message is almost immediately visible in the larger communal message area and other users respond. Users can enter chat rooms and read messages without sending any, a practice known as lurking.

Because chatroom messages are spontaneous and instantly visible, there is a potential for abuse, which may or may not be intentional. Site hosts typically post a frequently asked question list to appropriate chat room behavior, such as introducing yourself when you enter a room, making it clear when you are directing a question or response to a specific user, and reporting disruptive users, for example. Disruptive users may verbally abuse other chatters, monopolize the conversation, or even just disable it by repeatedly typing the same word or phrase into the conversation, a practice (much frowned upon) known as scrolling.

Chat rooms can be found that focus on virtually any aspect of human endeavor or interest: there are current communities based on classic movies, Irish ancestry, baton twirling, and psychic readings, for example. Various sites, such as Yahoo, provide a directory of chat sites. Others, such as MSN Web Communities, guide users through the steps required to create their own chat room.

# E-mail Hacking

Email hacking can be done in any of the following ways:

- Spam
- Virus
- Phishing

## Spam

E-mail spamming is an act of sending **Unsolicited Bulk E-mails (UBI)** which one has not asked for. Email spams are the junk mails sent by commercial companies as an advertisement of their products and services.

## Virus

Some emails may incorporate with files containing malicious script which when run on your computer may lead to destroy your important data.

## Phishing

Email phishing is an activity of sending emails to a user claiming to be a legitimate enterprise. Its main purpose is to steal sensitive information such as usernames, passwords, and credit card details.

Such emails contains link to websites that are infected with malware and direct the user to enter details at a fake website whose look and feels are same to legitimate one.

## E-mail Spamming and Junk Mails

Email spamming is an act of sending Unsolicited Bulk E-mails (UBI) which one has not asked for. Email spams are the junk mails sent by commercial companies as an advertisement of their products and services.

Spams may cause the following problems:

- It floods your e-mail account with unwanted e-mails, which may result in loss of important e-mails if inbox is full.
- Time and energy is wasted in reviewing and deleting junk emails or spams.
- It consumes the bandwidth that slows the speed with which mails are delivered.
- Some unsolicited email may contain virus that can cause harm to your computer.

## Blocking Spams

Following ways will help you to reduce spams:

- While posting letters to newsgroups or mailing list, use a separate e-mail address than the one you used for your personal e-mails.
- Don't give your email address on the websites as it can easily be spammed.
- Avoid replying to emails which you have received from unknown persons.
- Never buy anything in response to a spam that advertises a product.

## E-mail Cleanup and Archiving

In order to have light weighted Inbox, it's good to archive your inbox from time to time. Here I will discuss the steps to clean up and archive your Outlook inbox.

- Select **File** tab on the mail pane.
- Select **Cleanup Tools** button on account information screen.
- Select **Archive** from cleanup tools drop down menu.
- Select **Archive this folder and all subfolders** option and then click on the folder that you want to archive. Select the date from the **Archive items older than:** list. Click **Browse** to create new **.pst** file name and location. Click **OK**.

## SMTP

**SMTP** stands for **Simple Mail Transfer Protocol**. It was first proposed in 1982. It is a standard protocol used for sending e-mail efficiently and reliably over the internet.

### Key Points:

- SMTP is application level protocol.
- SMTP is connection oriented protocol.
- SMTP is text based protocol.
- It handles exchange of messages between e-mail servers over TCP/IP network.
- Apart from transferring e-mail, SMTP also provides notification regarding incoming mail.
- When you send e-mail, your e-mail client sends it to your e-mail server which further contacts the recipient mail server using SMTP client.

- These SMTP commands specify the sender's and receiver's e-mail address, along with the message to be send.
- The exchange of commands between servers is carried out without intervention of any user.
- In case, message cannot be delivered, an error report is sent to the sender which makes SMTP a reliable protocol.

## SMTP Commands

The following table describes some of the SMTP commands:

S.N.	Command Description
1	<b>HELLO</b> This command initiates the SMTP conversation.
2	<b>EHELLO</b> This is an alternative command to initiate the conversation. ESMTP indicates that the sender server wants to use extended SMTP protocol.
3	<b>MAIL FROM</b> This indicates the sender's address.
4	<b>RCPT TO</b> It identifies the recipient of the mail. In order to deliver similar message to multiple users this command can be repeated multiple times.
5	<b>SIZE</b> This command let the server know the size of attached message in bytes.
6	<b>DATA</b> The <b>DATA</b> command signifies that a stream of data will follow. Here stream of data refers to the body of the message.
7	<b>QUIT</b>

	This commands is used to terminate the SMTP connection.
8	<b>VERFY</b> This command is used by the receiving server in order to verify whether the given username is valid or not.
9	<b>EXPN</b> It is same as VRFY, except it will list all the users name when it used with a distribution list.

# Pine & Pico

## Introduction

Pine is an easy-to-learn electronic mail system. It is menu driven, which means that you will be presented with command choices at the bottom of each screen, as well as on-line Help. This document also describes Pico, Pine's default text editor, which you automatically use when you compose a message.

## *Conventions*

In this document, prompts and other on-screen information will be presented in typewriter style type. Information you must enter will be presented in **bold typewriter style type**. The character **^**, which accompanies many commands, represents the **control** key.

## *Entering Pine*

When you log on to the fog server, **pine** at the Unix prompt ('>' or '%'), and press **<return>**. You will be presented with Pine's Main Menu.

```
PINE 3.96    MAIN MENU    Folder:INBOX 7 Messages
?   HELP      - Get help using Pine
C   COMPOSE MESSAGE - Compose and send a message
I   FOLDER INDEX - View messages in current folder
L   FOLDER LIST  - Select a folder to view
```

- A ADDRESS BOOK - Update address book
- S SETUP - Configure or update Pine
- Q QUIT - Exit the Pine program

Copyright 1989-1997. PINE is a trademark of the Univ. of Washington.  
FOLDER "INBOX" opened with 7 Messages

? Help P PrevCmd R RelNotes

O OTHER CMDS L [ListFldrs] N NextCmd K KBLock

### ***Communications Software***

## **Reading Your Mail**

To view new messages, from the Main Menu simply press **I** (Folder Index). You will be shown an index of the messages in your "INBOX," the folder that contains new messages and messages you have not deleted or moved to another folder. The index screen provides the following information, from left to right:

- \* "+" means the message was sent directly to your account (i.e. - not as part of a cc: or mailing list)
- the message status (N - new, D - deleted, A - answered)
- the number of the message
- the date the message was received
- the email address or name of the person who sent it
- the length of the message (in bytes)
- the subject of the message (if the sender has given one)

To read a message, select it using your arrow keys and then press **<return>**. If the message is more than one screen long, press the **space bar** (NextPage) or the down arrow until you reach the end of the message. To view another message, press **N** (NextMsg) for next message or **I** (Folder Index) to return to the index. You can also delete the message when you are finished reading it (see "Deleting a Message" below). Again, as with all Pine screens, your options are shown on the menu bar at the bottom of the screen. The options within Folder Index are extensive and continue on a second and third menu, shown by pressing **O** (Other Cmds). These options can be accessed directly by pressing the letter without going to the additional menus.

## **Sending a Message**

**REMINDER:** The ^ means hold down the **Control** key while pressing the accompanying letter.

Sending a message from Pine is a simple matter; the text editor, Pico, is easy to learn and use. To write a message, enter **C** (Compose). Your cursor will appear at the **To:** prompt in the Compose Menu. Enter the email address of the person to whom you are sending the message, then press **<return>**, or the **down arrow**. If you want to send carbon copies, enter the email addresses after the **Cc:** prompt and press **<return>**. (If you want to attach a file, see the document *Pine/Pico Advanced*). Press **<return>** again and give a brief description of what your message is about after the **Subject:** prompt. The next **<return>** will transfer you to the Message Text portion of the screen. (The menu commands shown below will appear when you move into the message text portion of the screen.)

PINE 3.96 COMPOSE MESSAGE Folder:"folder name" 0 Messages

To : **user@e-mail.address**

Cc :

Attchmnt:

Subject : **test**

----- Message Text -----

**This is a test message. To enter your message, simply type in what you want to say.**

**^G** Get Help **^X** Send **^R** Rich Hdr **^Y**PrvPg/Top**^K** Cut Line **^O** Postpone  
**^C** Cancel **^D** Del Chr **^J** Attach **^V** NxtPg/End**^U** UnDel Line**^T** To AddrBk

Press **^G** at any time for help with any section.

### ***Typing your Message with Pico***

Now that you are in the Message Text area, you can type in the text of your message using the Pico text editor. Move the cursor by using the **arrow keys**. Pico automatically word-wraps the text of your message. You can insert blank lines for neater formatting by pressing **<return>**. If you delete words or lines, you can press **^J** to re justify the text so that unusual line breaks will not occur in your letter. If you want to delete a line, press **^K**. Pine has a word find command (**^W**) and also a spell-checker (**^T**). When you run the spell-checker, you can use the **arrow keys** to move to the part of the word that needs correction (that is, you do not need to re-spell the entire word). Unfortunately, the spell checker does not provide you with the correct spellings for the words it flags. When you have finished writing your message, press **^X** to send it.

### Moving Blocks of Text

The simplest method of moving a large block of text is with the Mark Set command. To set the mark, simultaneously depress the **control key**, the **shift key** and the **^**. You should see the Mark Set prompt appear. Next, use the arrow key and scroll to the end of the block of text you wish to move. The entire block of text should be highlighted. Press **^K** to cut this section to the clipboard. To paste the selection into your letter, relocate your cursor to the place where you wish the block of text to appear and press **^U**. (**Note:** some terminal emulations do not allow you to use this command.)

Instead, use repeated **^K** commands to delete the block of text. Then move your cursor and press **^U** to paste the entire set of deletions into your letter).

## Replying to a Message

If you would like to reply to the message you have just read, press **R** (Reply). Using the Reply function has the advantage that you do not need to type in the person's address. You will be asked if you want the original message included in your reply, and if you do, press **<return>** or type **y**. (Note: Including the entire text of a large message clogs up user space and may prove expensive to those who download their mail at a cost. However, you may want to include the original message to remind the person what you're talking about. You can also edit the original message in your Reply so that it only contains the relevant parts.) You will also be asked whether you want to reply to all the recipients, such as people who were CC:'d or on the mailing list. The default for both questions is no. After this, you will be presented with a screen that already has the original sender's address filled in; simply type in your message and press **^X** (Send) to send the message.

## Forwarding a Message

If you would like to forward the message you have just read in the Folder Index to someone else, press **F** (Forward). You will be asked to enter the email address of the person to whom you want to forward the message. After you have entered the address, enter an introductory message above the forwarded message if you feel it needs some explanation. Press **^X** (Send) to send it.

### ***Bounce***

The Bounce command is similar to the Forward command, but it does not allow you to edit or add to the original message. An exact copy of the message is sent to the address you enter.

## Deleting a Message

When you have finished reading a message, you can delete it or save it. **All messages remain in the INBOX folder until you either eXpunge them or save them to another folder.** To delete a message, simply highlight it from the index using your arrow keys, then press **D** (Delete). The letter **D** will appear next to the message, which means that when you exit Pine or press **X** (eXpunge), the message will be deleted. If you make a mistake, press **U** to undelete the message. You can also delete messages while you are reading them by simply pressing **D** (Delete). You will then move to the next message in the current folder.

## **Saving a Message and Using Folders**

If you want to save some of your messages, you should think about a system of folder organization. You may, for example, want to create one folder to contain all messages from John Smith, another to contain messages concerning SAS, a third to contain messages about meetings and conferences, and a fourth to contain miscellaneous messages. To save a message to a folder, highlight the message, then choose **S** (Save). You will be asked to give the name of the folder in which you wish to save it. If you have never used this folder before, Pine will ask you if you want the folder created; if it already exists, the message will be saved in the existing folder. Example:

```
SAVE to folder in <mail/[]> [saved-messages]: sas
```

If the folder does not exist, you will be given a message that looks like this:

```
Folder "sas" in <mail/[]> doesn't exist. Create?
```

When the message is saved, you will be notified like this:

```
[Message 2 copied to folder "sas" in <mail/[]> and marked deleted]
```

The message is now in the designated folder and will be marked for deletion from the INBOX. To look at all of your folders, press **L** (Folder List).

### ***Moving Around Folders***

When you enter Pine, the INBOX folder is automatically opened. If you want to view a message you have saved in another folder, from the Main Menu choose **L** for Folder List. Use the **arrow keys** to highlight the folder you wish to visit and press **return**. If you are already in the Folder Index, you can press **G** (GotoFldr) and enter the name of the folder you wish to move to. (Note: if you can't remember the folder name, press **^T** for a list of all your folders.)

### **The Address Book**

If you send messages frequently to a particular person or group, you can create a shortcut so you do not need to enter complete email addresses each time.

#### ***Nicknames***

A nickname is a shortcut for a particular person. To create an nickname, select **A** (Address Book) from Pine's main menu. Then press **A** (Add). You will be asked to enter the person's full name. After this, you will be asked to enter a one-word nickname, such as Harry; this is the name you will actually enter when sending messages to the person. You will then enter the full email address of the person. The next time you are composing a message to that person, you need only enter the

nickname (e.g. Harry) at the To: prompt; Pine will supply the person's full name and his or her email address.

### ***Take Address Command***

A simpler method of adding nicknames is by using the "Take Address" command. From either the message screen or with a message highlighted in the Folder Index, enter **T** (Take Address). This function will search the message for any email addresses, list those available, and allow you to create a nickname for them. When prompted, enter a nickname, the person's full name, and any other information you require. In newer versions of Pine, you have to enter **Control-X** in order to save this information to the addressbook. Next time you wish to write to this person, enter the nickname on the To: field and Pine will expand it to the appropriate address.

### ***Address Lists***

You may also have a certain group of people to whom you regularly send the same message. To avoid having to enter the email address for every person on that list, you can create an address list. Once you do this, you simply send the message to the group name you designate, and Pine will mail the message to each member's email address. To create an address list, select **A** (Addresses) from Pine's main menu.

Press **A** (AddNew). Enter a nickname for the list, and if you want to give it a more descriptive title, enter one in the **Fullname** field. You can skip the FCC and Comment fields. In the **Addresses** field, enter the email addresses that you want in the list separated by a comma. You can also enter nicknames that are currently in your Address Book. When you are done, press **Ctrl-X** and choose **Y** to save the changes. When you are composing a message, you need only enter the list nickname at the To: prompt. Pine will supply the email address for each member of the group. You can edit and add to the list as needed.

## **Sending Files**

### ***Inserting Files into Messages***

Pine users may have files in their Unix directories that they want to send by email. To place a file from your directory into the body of your email message, simply enter **^R** (Read File) at the point in your text where you want the file to be. You will be asked to supply the file name. After you have done so, Pine will insert the file wherever your cursor is located.

### ***Copy / Paste Options for Microcomputers***

Another useful tool that is present on many microcomputer systems is the copy and paste option. If you have a document written in a word processing program on your microcomputer (i.e., it has not been uploaded to the mainframe), you can often use the copy command to copy the text and then the paste command to place the text into the

body of your email message. (Note: this does not work with all word processing programs or microcomputer operating systems.)

## **Saving and Reading Attached Files Sent to You**

When you receive a message with an attached file, a message will appear under the Subject line that says Parts/Attachments. Use the View Attachments command (**v**) to save the attachment as a file in your Unix directory. After typing **v**, enter the attachment number that you would like to save (usually 2 since in most cases 1 will be an attachment comment). Choose Save (**s**) or press **<return>**. You can change the name of the attachment at this point. A message will appear saying the file has been written to your home directory with the name you selected. Enter **E** to exit the View Attachments menu. If you wish to look at this file and it is in text format, quit Pine. (**q**) At the Unix prompt ('%' or '>'), type the command **more** and the filename.

For example: **%more textfile**

Use the **space bar** to scroll downward, one screen at a time.

### ***Downloading Files to Your Microcomputer***

If the attachment is not in text format (e.g., an Excel spreadsheet or an MS Word document) you will need to download it to a microcomputer that has the necessary software to read it. If you want to move the file onto your desktop computer, please consult [Transferring Files](#).

## **Printing Email Messages**

Printing email messages and files from Pine differs according to what kind of microcomputer, printer, and communications software you are using. When reading an email message within Folder Index, you can select **Y** (Print) to print it on your local printer. If this does not work, make sure Pine is set to printer option "attached-to-ansi" (the default option) by selecting **S** from the main menu, then **P** (Printer). It should be stressed, however, that the print option in Pine is very buggy; there's simply no way for the program running on Unix to know what kind of terminal emulation and/or printer you are using. In many cases, you will have to export the particular letter, download it to your local computer, and print it via a local word processing program. To export a letter, highlight it in the INBOX, then enter **E** to store a copy of the letter in your home directory. From there, you can follow the downloading instructions detailed in [Transferring Files](#).

## **Miscellaneous Troubleshooting**

***^C throws you out of Pine and back in to Unix or prints out [Interrupt Process]:***

This is a problem common with NCSA Telnet for the Mac. Once you have connected from a Mac to Pine via Telnet, you should select **Setup Keys** from the **Session** menu.

Delete **^C** from the "Interrupt Process" box. **^C** is both a Pine and a Telnet command, but since Telnet is supporting the Pine session, it interprets the command first. You can save this change by selecting Save under the File menu. (Note: Unix also uses the **^C** command, to escape from hung or runaway processes.)

***The "backspace" doesn't work correctly:***

For NCSA Telnet on the Mac, you have the option of using either the **Backspace** key or the **Delete** key, but not both. To pick one or the other, select Backspace from the Session menu.

***READ ONLY shows up when you start Pine:***

This means that you actually have two sessions of Pine running at the same time. To correct this problem so that you can delete mail and manage your files again, do the following:

1. Quit Pine.
2. At the Unix prompt (%), type **ps**.
3. Note the PID# associated with pine.
4. Type **kill -9** and the PID# of that Pine session. An example is shown below.

```
% ps
PID  TT  STAT      TIME COMMAND
898  q3  S        0:00 -csh[schneide]
913  q3  T        0:00 pine
930  q3  R        0:00 ps
% kill -9 913
%
[1] Terminated      pine
```

***Exceeding your disk quota:***

Free email users have approximately 5 megabytes of disk space. If you receive any message/error that seems to indicate that you have reached or exceeded this limit, we suggest you use the following commands to identify the names of your files and their size and to remove any unwanted files.

```
%du | sort -rn | more
```

This command lists your files from largest to smallest, one page at a time. (Note: use the **space bar** to scroll to the next page, type **q** to quit the listing).

```
%ls -l
```

This command will list your files and the file sizes. In addition to the files listed, the Unix system makes use of a number of "invisible" files, some of which can

become quite large. To see these files, add the **a** argument to the ls command: `ls -la`. **WARNING:** Several of these files are extremely important, and removing or tinkering with them can destroy your ability to login and/or read your mail. Never remove your `.login`, `.cshrc`, `.addressbook`, or `.pinerc` files. However, the `.pine-debug#` files are only necessary when you are having problems.

`%rm filename`

Where filename is the file to be deleted. **Note:** You may want to consider downloading files prior to deleting/editing them.

## Summary of Pico Commands

The **^** indicates that the <Ctrl> key should be held down when typing the following letter.

COMMAND	FUNCTION
<b>^G</b> (F1)	Display help text.
<b>^F</b>	Move forward a character.
<b>^B</b>	Move backward a character.
<b>^P</b>	Move to the previous line.
<b>^N</b>	Move to the next line.
<b>^A</b>	Move to the beginning of the current line.
<b>^E</b>	Move to the end of the current line.
<b>^V</b> (F8)	Move forward a page of text.
<b>^Y</b> (F7)	Move backward a page of text.
<b>^W</b> (F6)	Search for text, neglecting case.
<b>^L</b>	Refresh the display.
<b>^D</b>	Delete the character at the cursor position.
<b>^^</b>	Mark cursor position as beginning of selected text. Note: Setting mark when already set unselects text.
<b>^K</b> (F9)	Cut selected text (displayed in inverse characters) or a whole line. Note: The selected text's boundary on the cursor side ends at the left edge of the cursor. So, with selected text to the left of the cursor, the character under the cursor is not selected.
<b>^U</b> (F10)	Uncut (paste) last cut text, inserting it at the current cursor position.
<b>^I</b>	Insert a tab at the current cursor position.
<b>^J</b> (F4)	Format (justify) the current paragraph. Note: Paragraphs are delimited by blank lines or indentation.
<b>^T</b> (F12)	Invoke the spell checker.
<b>^C</b> (F11)	Report current cursor position.
<b>^R</b> (F5)	Insert an external file at the current cursor position.

- ^O (F3)      Output the current buffer to a file, saving  
                 it. (Postpones mail message.)
- ^X (F2)      Exit pico, saving buffer. (Sends mail  
                 message.)

# LANGUAGES

## HTML

**HTML** stands for **Hyper Text Markup Language**. It is a formatting language used to define the appearance and contents of a web page. It allows us to organize text, graphics, audio, and video on a web page.

### **Key Points:**

- The word Hypertext refers to the text which acts as a link.

- The word markup refers to the symbols that are used to define structure of the text. The markup symbols tells the browser how to display the text and are often called tags.
- The word Language refers to the syntax that is similar to any other language.

HTML was created by **Tim Berners-Lee** at **CERN**.

## HTML Versions

The following table shows the various versions of HTML:

Version	Year
HTML 1.0	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.0	1999
XHTML	2000
HTML5	2012

## HTML Tags

**Tag** is a command that tells the web browser how to display the text, audio, graphics or video on a web page.

### Key Points:

- Tags are indicated with pair of angle brackets.
- They start with a less than (<) character and end with a greater than (>) character.
- The tag name is specified between the angle brackets.
- Most of the tags usually occur in pair: the start tag and the closing tag.

- The start tag is simply the tag name is enclosed in angle bracket whereas the closing tag is specified including a forward slash (/).
- Some tags are the empty i.e. they don't have the closing tag.
- Tags are not case sensitive.
- The starting and closing tag name must be the same. For example <b> hello </i> is invalid as both are different.
- If you don't specify the angle brackets (<>) for a tag, the browser will treat the tag name as a simple text.
- The tag can also have attributes to provide additional information about the tag to the browser.

## Basic tags

The following table shows the Basic HTML tags that define the basic web page:

Tag	Description
<html> </html>	Specifies the document as a web page.
<head> </head>	Specifies the descriptive information about the web documents.
<title> </title>	Specifies the title of the web page.
<body> </body>	Specifies the body of a web document.

The following code shows how to use basic tags.

```
<html>
  <head> Heading goes here...</head>
  <title> Title goes here...</title>
  <body> Body goes here...</body>
</html>
```

## Formatting Tags

The following table shows the HTML tags used for formatting the text:

Tag	Description
<b> </b>	Specifies the text as bold. Eg. <b>this is bold text</b>
<em> </em>	It is a phrase text. It specifies the emphasized text. Eg. <i>Emphasized text</i>
<strong> </strong>	It is a phrase tag. It specifies an important text. Eg. <b>this is strong text</b>
<i> </i>	The content of italic tag is displayed in italic. Eg. <i>Italic text</i>
<sub> </sub>	Specifies the subscripted text. Eg. X <sub>1</sub>
<sup> </sup>	Defines the superscripted text. Eg. X <sup>2</sup>
<ins> </ins>	Specifies the inserted text. Eg. The price of pen is now 15.
<del> </del>	Specifies the deleted text. Eg. The price of pen is now 15.
<mark> </mark>	Specifies the marked text. Eg. It is raining

## Table Tags

Following table describe the commonly used table tags:

Tag	Description
<table> </table>	Specifies a table.
<tr> </tr>	Specifies a row in the table.

<code>&lt;th&gt; &lt;/th&gt;</code>	Specifies header cell in the table.
<code>&lt;td&gt; &lt;/td&gt;</code>	Specifies the data in an cell of the table.
<code>&lt;caption&gt; &lt;/caption&gt;</code>	Specifies the table caption.
<code>&lt;colgroup&gt; &lt;/colgroup&gt;</code>	Specifies a group of columns in a table for formatting.

## List tags

Following table describe the commonaly used list tags:

Tag	Description
<code>&lt;ul&gt; &lt;/ul&gt;</code>	Specifies an unordered list.
<code>&lt;ol&gt; &lt;/ol&gt;</code>	Specifies an ordered list.
<code>&lt;li&gt; &lt;/li&gt;</code>	Specifies a list item.
<code>&lt;dl&gt; &lt;/dl&gt;</code>	Specifies a description list.
<code>&lt;dt&gt; &lt;/dt&gt;</code>	Specifies the term in a description list.
<code>&lt;dd&gt; &lt;/dd&gt;</code>	Specifies description of term in a description list.

## Frames

**Frames** help us to divide the browser's window into multiple rectangular regions. Each region contains separate html web page and each of them work independently.

A set of frames in the entire browser is known as frameset. It tells the browser how to divide browser window into frames and the web pages that each has to load.

The following table describes the various tags used for creating frames:

Tag	Description
<frameset> </frameset>	It is replacement of the <body> tag. It doesn't contain the tags that are normally used in <body> element; instead it contains the <frame> element used to add each frame.
<frame> </frame>	Specifies the content of different frames in a web page.
<base> </base>	It is used to set the default target frame in any page that contains links whose contents are displayed in another frame.

## Forms

**Forms** are used to input the values. These values are sent to the server for processing. Forms uses input elements such as text fields, check boxes, radio buttons, lists, submit buttons etc. to enter the data into it.

The following table describes the commonly used tags while creating a form:

Tag	Description
<form> </form>	It is used to create HTML form.
<input> </input>	Specifies the input field.
<textarea> </textarea>	Specifies a text area control that allows to enter multi-line text.
<label> </label>	Specifies the label for an input element.

**CSS** is acronym of **Cascading Style Sheets**. It helps to define the presentation of HTML elements as a separate file known as CSS file having **.css** extension.

CSS helps to change formatting of any HTML element by just making changes at one place. All changes made would be reflected automatically to all of the web pages of the website in which that element appeared.

## CSS Rules

CSS Rules are the styles that we have to create in order to create style sheets. These rules define appearance of associated HTML element. The general form of CSS syntax is as follows:

```
Selector {property: value;}
```

### Key Points

- Selector is HTML element to which CSS rule is applied.
- Property specifies the attribute that you want to change corresponding to the selector.
- Property can take specified value.
- Property and Value are separated by a colon (:).
- Each declaration is separated by semi colon (;).

Following are examples of CSS rules:

```
P { color : red;}  
  
h1 (color : green; font-style : italic )  
  
body { color : cyan; font-family : Arial; font- style : 16pt}
```

## Embedding CSS into HTML

Following are the four methods to add CSS to HTML documents.

1. Inline Style Sheets
2. Embedded Style Sheets
3. External Style Sheets
4. Imported Style Sheets

### Inline Style Sheets

**Inline Style Sheets** are included with HTML element i.e. they are placed inline with the element. To add inline CSS, we have to declare style attribute which can contain any CSS property.

## Syntax:

```
<Tagname STYLE = " Declaration1 ; Declaration2 " > .... </Tagname>
```

Let's consider the following example using Inline Style Sheets:

```
<p style="color: blue; text-align: left; font-size: 15pt">
```

Inline Style Sheets are included with HTML element i.e. they are placed inline with the element.

To add inline CSS, we have to declare style attribute which can contain any CSS property.

```
</p>
```

## Output -



## Embedded Style Sheets

**Embedded Style** Sheets are used to apply same appearance to all occurrence of a specific element. These are defined in <head> element by using the <style> element.

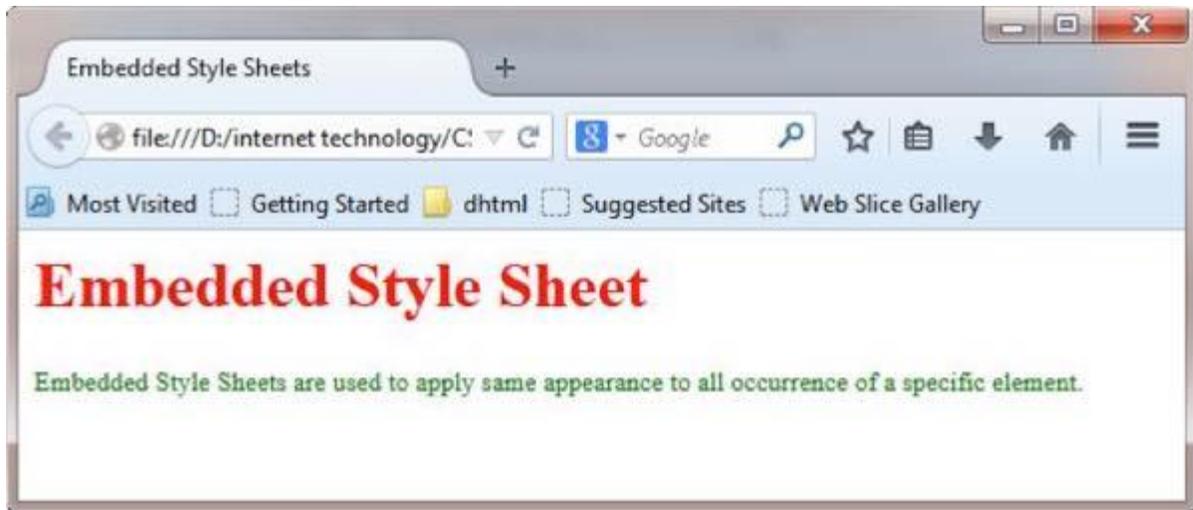
The <style> element must include **type** attribute. The value of **type** attribute specifies what type of syntax it includes when rendered by the browser.

## Syntax

```
<head> <title> .... </title>  
  <style type = "text/css">  
    .....CSS Rules/Styles....  
  </style>  
</head>
```

Let's consider the following example using Embedded Style Sheets:

```
<style type="text/css">  
  p {color:green; text-align: left; font-size: 10pt}  
  h1 { color: red; font-weight: bold}  
</style>
```



## External Style Sheets

**External Style Sheets** are the separate **.css** files that contain the CSS rules. These files can be linked to any HTML documents using `<link>` tag with `rel` attribute.

### Syntax:

```
<head> <link rel= "stylesheet" type="text/css" href= "url of css file">
</head>
```

In order to create external css and link it to HTML document, follow the following steps:

- First of all create a CSS file and define all CSS rules for several HTML elements. Let's name this file as `external.css`.

```
p{
  Color: orange;   text-align: left;   font-size: 10pt;
}
h1{
  Color: orange;   font-weight: bold;
}
```

- Now create HTML document and name it as **externaldemo.html**.

```
<html>
<head>
  <title> External Style Sheets Demo </title>
  <link rel="stylesheet" type="text/css" href="external.css">
</head>
<body>
  <h1> External Style Sheets</h1>
  <p>External Style Sheets are the separate .css files that contain the CSS rules.</p>
</body>
</html>
```



## Imported Style Sheets

**Imported Style Sheets** allow us to import style rules from other style sheets. To import CSS rules we have to use `@import` before all the rules in a style sheet.

### Syntax:

```
<head><title> Title Information </title>
  <style type="text/css">
    @import URL (cssfilepath)
    ... CSS rules...
  </style>
</head>
```

Let's consider the following example using Inline Style Sheets:

```
<html>
  <head>
    <title> External Style Sheets Demo </title>
    <style>
      @import url(external.css);
    </style>
  </head>
  <body>
    <h1> External Style Sheets</h1>
    <p>External Style Sheets are the separate .css files that contain the CSS rules.</p>
  </body>
</html>
```



# XML Basics

---

## XML Basics XML

- XML stands for EXtensible Markup Language XML is a markup language much like HTML
- XML was designed to carry data, not to display data
- XML tags are not predefined. You must define your own tags
- XML is designed to be self-descriptive XML is a W3C Recommendation

## The Difference between XML and HTML

XML is not a replacement for HTML.

XML and HTML were designed with different goals:

- XML was designed to transport and store data, with focus on what data is
- HTML was designed to display data, with focus on how data looks

HTML is about displaying information, while XML is about carrying information. With XML You Invent Your Own Tags documents can only use tags defined in the HTML standard (like <p>, <h1>, etc.). XML allows the author to define his/her own tags and his/her own document structure.

## **XML is Not a Replacement for HTML**

**XML is a complement to HTML.**

It is important to understand that XML is not a replacement for HTML. In most web applications, XML is used to transport data, while HTML is used to format and display the data.

## **XML Separates Data from HTML**

If you need to display dynamic data in your HTML document, it will take a lot of work to edit the HTML each time the data changes. With XML, data can be stored in separate XML files. This way you can concentrate on using HTML for layout and display, and be sure that changes in the underlying data will not require any changes to the HTML. With a few lines of JavaScript code, you can read an external XML file and update the data content of your web page.

## **XML Simplifies Data Sharing**

In the real world, computer systems and databases contain data in incompatible formats. XML data is stored in plain text format. This provides a software- and hardware-independent way of storing data. This makes it much easier to create data that can be shared by different applications.

## **XML Simplifies Data Transport**

One of the most time-consuming challenges for developers is to exchange data between incompatible systems over the Internet. Exchanging data as XML greatly reduces this complexity, since the data can be read by different

incompatible applications.

### **XML Simplifies Data Transport**

Upgrading to new systems (hardware or software platforms), is always time consuming. Large amounts of data must be converted and incompatible data is often lost. XML data is stored in text format. This makes it easier to expand or upgrade to new operating systems, new applications, or new browsers, without losing data.

### **XML Makes Your Data More Available**

Different applications can access your data, not only in HTML pages, but also from XML data sources. With XML, your data can be available to all kinds of "reading machines" (Handheld computers, voice machines, news feeds, etc), and make it more available for blind people, or people with other disabilities. XML is Used to Create New Internet Languages

A lot of new Internet languages are created with XML.

### **Here are some examples:**

- XHTML
- WSDL for describing available web services WAP and WML as markup
- RSS languages for news feeds
- RDF and OWL for describing resources and ontology
- SMIL for describing multimedia for the web

# DHTML | Introduction

DHTML stands for Dynamic HTML, it is totally different from HTML. The browsers which support the dynamic HTML are some of the versions of Netscape Navigator and Internet Explorer of version higher than 4.0. The DHTML is based on the properties of the HTML, javascript, CSS, and DOM (Document Object Model which is used to access individual elements of a document) which helps in making dynamic content. It is the combination of HTML, CSS, JS, and DOM. The DHTML make use of Dynamic object model to make changes in settings and also in properties and methods. It also makes uses of Scripting and it is also part of earlier computing trends.

DHTML allows different scripting languages in a web page to change their variables, which enhance the effects, looks and many others functions after the whole page have been fully loaded or under a view process, or otherwise static HTML pages on the same. But in true ways, there is noting that as dynamic in DHTML, there is only the enclosing of different technologies like CSS, HTML, JS, DOM, and different sets of static languages which make it as dynamic.

DHTML is used to create interactive and animated web pages that are generated in real-time, also known as dynamic web pages so that when such a page is accessed, the code within the page is analyzed on the web server and the resulting HTML is sent to the client's web browser.

**Key Features:** Following are the some major key features of DHTML:

- Tags and their properties can be changed using DHTML.
- It is used for real-time positioning.
- Dynamic fonts can be generated using DHTML.
- It is also used for data binding.
- It makes a webpage dynamic and be used to create animations, games, applications along with providing new ways of navigating through websites.
- The functionality of a webpage is enhanced due to the usage of low-bandwidth effect by DHTML.
- DHTML also facilitates the use of methods, events, properties, and codes.

**Why**

**Use**

**DHTML?**

DHTML makes a webpage dynamic but Javascript also does, the question arises that what different does DHTML do? So the answer is that DHTML has the ability to change a webpages look, content and style once the document has loaded on our demand without changing or deleting everything already existing on the browser's webpage. DHTML can change the content of a webpage on demand without the browser having to erase everything else, i.e. being able to alter changes on a webpage even after the document has completely loaded.

**Advantages:**

- Size of the files are compact in compared to other interational media like Flash or Shockwave, and it downloads faster.
- It is supported by big browser manufacturers like Microsoft and Netscape.
- Highly flexible and easy to make changes.

- Viewer requires no extra plug-ins for browsing through the webpage that uses DHTML, they do not need any extra requirements or special software to view it.
- User time is saved by sending less number of requests to the server. As it is possible to modify and replace elements even after a page is loaded, it is not required to create separate pages for changing styles which in turn saves time in building pages and also reduces the number of requests that are sent to the server.
- It has more advanced functionality than a static HTML. it is capable of holding more content on the web page at the same time.

#### **Disadvantages:**

- It is not supported by all the browsers. It is supported only by recent browsers such as Netscape 6, IE 5.5, and Opera 5 like browsers.
- Learning of DHTML requires a lot of pre-requisites languages such as HTML, CSS, JS, etc should be known to the designer before starting with DHTML which is a long and time-consuming in itself.
- Implementation of different browsers are different. So if it worked in one browser, it might not necessarily work the same way in another browser.
- Even after being great with functionality, DHTML requires a few tools and utilities that are some expensive. For example, the DHTML text editor, Dreamweaver. Along with it the improvement cost of transferring from HTML to DHTML makes cost rise much higher.

#### **Difference between HTML and DHTML:**

- HTML is a markup language while DHTML is a collection of technologies.
- HTML is used to create static webpages while DHTML is capable of creating dynamic webpages.
- DHTML is used to create animations and dynamic menus but HTML not used.
- HTML sites are slow upon client-side technologies whereas DHTML sites are comparatively faster.
- Web pages created using HTML are rather simple and have no styling as it uses only one language whereas DHTML uses HTML, CSS, and Javascript which results in a much better and way more presentable webpage.
- HTML cannot be used as server side code but DHTML used as server side code.
- DHTML needs database connectivity but not in case of HTML.
- Files in HTML are stored using .htm or .html extension while DHTML uses .dhtm extension.
- HTML requires no processing from the browser but DHTML does.

## **JAVASCRIPT**

**JavaScript** is a lightweight, interpreted programming language with object-oriented capabilities that allows you to build interactivity into otherwise static HTML pages.

JavaScript code is not compiled but translated by the translator. This translator is embedded into the browser and is responsible for translating javascript code.

## Key Points

- It is Lightweight, interpreted programming language.
- It is designed for creating network-centric applications.
- It is complementary to and integrated with Java.
- It is complementary to and integrated with HTML
- It is an open and cross-platform

## JavaScript Statements

JavaScript statements are the commands to tell the browser to what action to perform. Statements are separated by semicolon (;).

JavaScript statement constitutes the JavaScript code which is translated by the browser line by line.

Example of JavaScript statement:

```
document.getElementById("demo").innerHTML = "Welcome";
```

Following table shows the various JavaScript Statements –

Sr.No.	Statement	Description
1.	switch case	A block of statements in which execution of code depends upon different cases. The interpreter checks each case against the value of the expression until a match is found. If nothing matches, a <b>default</b> condition will be used.
2.	If else	The <b>if</b> statement is the fundamental control statement that allows JavaScript to make decisions and execute statements conditionally.
3.	While	The purpose of a while loop is to execute a statement or code block repeatedly as long as expression is true. Once expression becomes false, the loop will be exited.
4.	do while	Block of statements that are executed at least once and continues to be executed while condition is true.

5.	for	Same as while but initialization, condition and increment/decrement is done in the same line.
6.	for in	This loop is used to loop through an object's properties.
7.	continue	The continue statement tells the interpreter to immediately start the next iteration of the loop and skip remaining code block.
8.	break	The break statement is used to exit a loop early, breaking out of the enclosing curly braces.
9.	function	A function is a group of reusable code which can be called anywhere in your programme. The keyword function is used to declare a function.
10.	return	Return statement is used to return a value from a function.
11.	var	Used to declare a variable.
12.	try	A block of statements on which error handling is implemented.
13.	catch	A block of statements that are executed when an error occur.
14.	throw	Used to throw an error.

## JavaScript Comments

JavaScript supports both C-style and C++-style comments, thus:

- Any text between a // and the end of a line is treated as a comment and is ignored by JavaScript.
- Any text between the characters /\* and \*/ is treated as a comment. This may span multiple lines.

- JavaScript also recognizes the HTML comment opening sequence <!--. JavaScript treats this as a single-line comment, just as it does the // comment.-->
- The HTML comment closing sequence --> is not recognized by JavaScript so it should be written as //-->.

### Example

```
<script language="javascript" type="text/javascript">
  <!--
    // this is a comment. It is similar to comments in C++

    /*
     * This is a multiline comment in JavaScript
     * It is very similar to comments in C Programming
    */
  //-->
</script>
```

### JavaScript variable

Variables are referred as named containers for storing information. We can place data into these containers and then refer to the data simply by naming the container.

### Rules to declare variable in JavaScript

Here are the important rules that must be followed while declaring a variable in JavaScript.

- In JavaScript variable names are case sensitive i.e. a is different from A.
- Variable name can only be started with a underscore ( \_ ) or a letter (from a to z or A to Z), or dollar ( \$ ) sign.
- Numbers (0 to 9) can only be used after a letter.
- No other special character is allowed in variable name.

Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the var keyword as follows –

```
<script type="text/javascript">
  <!--
    var money;
    var name, age;
  //-->
</script>
```

Variables can be initialized at time of declaration or after declaration as follows –

```
<script type="text/javascript">
  <!--
    var name = "Ali";
    var money;
    money = 2000.50;
  //-->
</script>
```

## Javascript Data Type

There are two kinds of data types as mentioned below –

- Primitive Data Type
- Non Primitive Data Type

The following table describes **Primitive Data Types** available in JavaScript

Sr.No.	Datatype Description
1.	<b>String</b> Can contain groups of character as single value. It is represented in double quotes. E.g. var x= "tutorial".
2.	<b>Numbers</b> Contains the numbers with or without decimal. E.g. var x=44, y=44.56;
3.	<b>Booleans</b> Contain only two values either true or false. E.g. var x=true, y= false.
4.	<b>Undefined</b> Variable with no value is called Undefined. E.g. var x;
5.	<b>Null</b> If we assign null to a variable, it becomes empty. E.g. var x=null;

The following table describes **Non-Primitive Data Types** in JavaScript

Sr.No.	Datatype Description
--------	----------------------

1.	<b>Array</b> Can contain groups of values of same type. E.g. var x={1,2,3,55};
2.	<b>Objects</b> Objects are stored in property and value pair. E.g. var rectangle = { length: 5, breadth: 3};

## JavaScript Functions

Function is a group of reusable statements (Code) that can be called any where in a program. In javascript function keyword is used to declare or define a function.

### Key Points

- To define a function use function keyword followed by functionname, followed by parentheses ().
- In parenthesis, we define parameters or attributes.
- The group of reusable statements (code) is enclosed in curly braces {}. This code is executed whenever function is called.

### Syntax

```
function functionname (p1, p2) {  
    function coding...  
}
```

## JavaScript Operators

Operators are used to perform operation on one, two or more operands. Operator is represented by a symbol such as +, =, \*, % etc. Following are the operators supported by javascript –

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators
- Conditional (or ternary) Operators
- Arithmetic Operators

### Arithmetic Operators

Following table shows all the arithmetic operators supported by javascript –

Operator	Description	Example
+	Add two operands.	10 + 10 will give 20
-	Subtract second operand from the first.	10 – 10 will give 0
*	Multiply two operands.	10 * 30 will give 300
/	Divide numerator by denominator	10/10 will give 1
%	It is called modulus operator and gives remainder of the division.	10 % 10 will give 0
++	Increment operator, increases integer value by one	10 ++ will give 11
--	Decrement operator, decreases integer value by one	10 – will give 9

### Comparison Operators

Following table shows all the comparison operators supported by javascript –

Operator	Description	Example
==	Checks if values of two operands are equal or not, If yes then condition becomes true.	10 == 10 will give true
!=	<b>Not Equal to operator</b> Checks if the value of two operands is equal or not, if values are not equal then condition becomes true.	10 !=10 will give false

>	<b>Greater Than operator</b> Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	20 > 10 will give true
<	<b>Less than operator</b> Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	10 < 20 will give true
>=	<b>Greater than or equal to operator</b> Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	10 >=20 will give false
<=	<b>Less than or equal to operator</b> Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	10 <=20 will give true.

## Logical Operators

Following table shows all the logical operators supported by javascript –

Operator	Description	Example
&&	Logical <b>AND</b> operator returns true if both operands are non zero.	10 && 10 will give true.
	Logical <b>OR</b> operator returns true If any of the operand is non zero	10    0 will give true.
!	Logical <b>NOT</b> operator complements the logical state of its operand.	!(10 && 10) will give false.

## Assignment Operators

Following table shows all the assignment operators supported by javascript –

Operator	Description	Example
=	<b>Simple Assignment operator</b> Assigns values from right side operands to left side operand.	C = A + B will assign value of A + B into C
+=	<b>Add AND assignment operator</b> It adds right operand to the left operand and assign the result to left operand	C += A is equivalent to C = C + A
-=	<b>Subtract AND assignment operator</b> It subtracts right operand from the left operand and assign the result to left operand	C -= A is equivalent to C = C - A
*=	<b>Multiply AND assignment operator</b> It multiplies right operand with the left operand and assign the result to left operand	C *= A is equivalent to C = C * A
/=	<b>Divide AND assignment operator</b> It divides left operand with the right operand and assign the result to left operand	C /= A is equivalent to C = C / A
%=	<b>Modulus AND assignment operator</b> Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	C %= A is equivalent to C = C % A

### Conditional Operator

It is also called ternary operator, since it has three operands.

Operator	Description	Example
?:	Conditional	If Condition is true? Then value X : Otherwise

	Expression	value Y
--	------------	---------

## Control Structure

Control structure actually controls the flow of execution of a program. Following are the several control structure supported by javascript.

- if ... else
- switch case
- do while loop
- while loop
- for loop

### If ... else

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

#### Syntax

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

#### Example

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

### Switch case

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

#### Syntax

```
switch (expression) {
  case condition 1: statement(s)
    break;
  case condition 2: statement(s)
    break;
  ...
  case condition n: statement(s)
    break;
  default: statement(s)
}
```

### Example

```
<script type="text/javascript">
  <!--
    var grade='A';
    document.write("Entering switch block<br/>");
    switch (grade) {
      case 'A': document.write("Good job<br/>");
        break;
      case 'B': document.write("Pretty good<br/>");
        break;
      case 'C': document.write("Passed<br/>");
        break;
      case 'D': document.write("Not so good<br/>");
        break;
      case 'F': document.write("Failed<br/>");
        break;
      default: document.write("Unknown grade<br/>")
    }
    document.write("Exiting switch block");
  //-->
</script>
```

### Do while Loop

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

### Syntax

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

### Example

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

This will produce following result –

```
Starting Loop
Current Count : 0
Loop stopped!
```

While Loop

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

### Syntax

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

### Example

```
<script type="text/javascript">
  <!--
    var count = 0;
    document.write("Starting Loop" + "<br/>");
    while (count < 10){
      document.write("Current Count : " + count + "<br/>");
      count++;
    }
    document.write("Loop stopped!");
  //-->
</script>
```

This will produce following result –

```
Starting Loop
Current Count : 0
Current Count : 1
Current Count : 2
```

Current Count : 3  
Current Count : 4  
Current Count : 5  
Current Count : 6  
Current Count : 7  
Current Count : 8  
Current Count : 9  
Loop stopped!

## For Loop

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

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

## Syntax

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

## Example

```
<script type="text/javascript">  
    <!--  
        var count;  
        document.write("Starting Loop" + "<br/>");  
        for(count = 0; count < 10; count++){  
            document.write("Current Count : " + count );  
            document.write("<br/>");  
        }  
        document.write("Loop stopped!");  
    //-->  
</script>
```

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

Starting Loop  
Current Count : 0  
Current Count : 1  
Current Count : 2

Current Count : 3  
Current Count : 4  
Current Count : 5  
Current Count : 6  
Current Count : 7  
Current Count : 8  
Current Count : 9  
Loop stopped!

## Creating Sample Program

Following is the sample program that shows time, when we click in button.

```
<html>
  <body>
    <button onclick="this.innerHTML=Date()">The time is?</button>
    <p>Click to display the date.</p>
    <button onclick="displayDate()">The time is?</button>
    <script>
      function displayDate() {
        document.getElementById("demo").innerHTML = Date();
      }</script>

      <p id="demo"></p>
    </script>
  </body>
</html>
```

## Output



