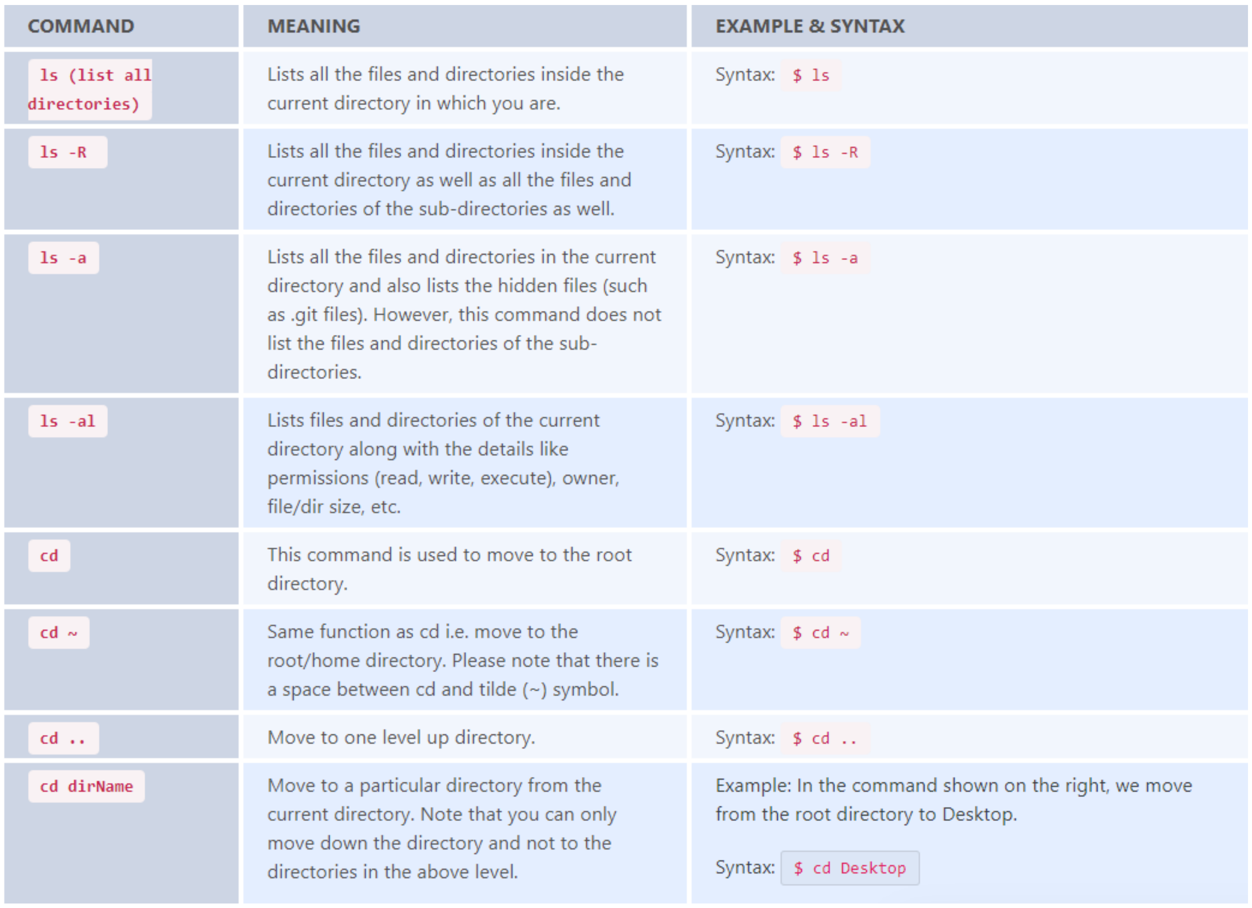
1. ls - list the contents of a directory
2. cd - change directory
3. pwd - print working directory
4. cp - copy files and directories
5. mv - move or rename files and directories
6. rm - remove files and directories
7. mkdir - make a new directory
8. rmdir - remove a directory
9. touch - create a new empty file
10. cat - concatenate and display files
11. less - display a file one page at a time
12. head - display the first few lines of a file
13. tail - display the last few lines of a file
14. grep - search for a pattern in a file
15. find - search for files and directories
16. ps - display information about running processes
17. top - display system resource usage and process information
18. kill - send a signal to a process to terminate it
19. su - switch user
20. sudo - execute a command with superuser privileges
21. chmod - change file permissions
22. chown - change file ownership
23. tar - create or extract tar archives
24. gzip - compress or decompress files with gzip compression
25. unzip - extract files from zip archives
26. ping - test network connectivity to a host
27. netstat - display network connections and network statistics
28. ifconfig - display network interface configuration
29. ssh - securely connect to a remote server
30. scp - securely copy files between hosts
31. curl - transfer data from or to a server using various protocols
32. wget - download files from the internet
33. du - display disk usage information
34. df - display disk space information
35. mount - mount a filesystem
36. umount - unmount a filesystem
37. passwd - change user password
38. history - display command history

39. which - display the location of a command
40. tar - create or extract tar archives
41. gzip - compress or decompress files with gzip compression
42. unzip - extract files from zip archives
43. ps - display information about running processes
44. df - display disk space information
45. uname - display system information
46. whoami - display the current username
47. id - display user and group information
48. last - display a log of last logins
49. date - display or set the system date and time
50. uptime - display system uptime and load averages



COMMAND	MEANING	EXAMPLE & SYNTAX
`ls (list all directories)`	Lists all the files and directories inside the current directory in which you are.	Syntax: `$ ls`
`ls -R`	Lists all the files and directories inside the current directory as well as all the files and directories of the sub-directories as well.	Syntax: `$ ls -R`
`ls -a`	Lists all the files and directories in the current directory and also lists the hidden files (such as .git files). However, this command does not list the files and directories of the sub-directories.	Syntax: `$ ls -a`
`ls -al`	Lists files and directories of the current directory along with the details like permissions (read, write, execute), owner, file/dir size, etc.	Syntax: `$ ls -al`
`cd`	This command is used to move to the root directory.	Syntax: `$ cd`
`cd ~`	Same function as cd i.e. move to the root/home directory. Please note that there is a space between cd and tilde (~) symbol.	Syntax: `$ cd ~`
`cd ..`	Move to one level up directory.	Syntax: `$ cd ..`
`cd dirName`	Move to a particular directory from the current directory. Note that you can only move down the directory and not to the directories in the above level.	Example: In the command shown on the right, we move from the root directory to Desktop. Syntax: `$ cd Desktop`

`mkdir`	This command creates a directory.	Example: The command shown in the right will create a directory named "exampleDir" in the current directory in which we are. Syntax: `$ mkdir exampleDir`
`cat > fileName`	This command creates a file in the current directory.	Example: The command shown in the right creates a new file in the current directory and the name of the file will be file1 with an extension of '.txt'. Syntax: `$ cat > file1.txt`
`cat fileName`	This command displays the content in a file. If a file is not present in the current directory, it gives a message showing no such file exists.	Example: The command shown on the right displays the content of the file file1.txt. "Hello there!" is the content inside it. Syntax: `$ cat file1.txt Hello there!`

`cat f1 f2 > f3`	This command joins the content of two files and stores it in the third file. If the third file does not exist, it is first created and then the joined content is stored.	Example: The command in the right stores the joined content of file1 and file2 in file3. File1 has "Hello there!" and file2 has "What's up?" in their content. We have displayed the content of file3.
		
		Syntax:
		`$ cat file1.txt  file2.txt > file3.txt`
		`$ cat file3.txt Hello there! What's up?`

`rmdir dirName`	This command is the remove directory command. It deletes a directory.	Example: The remove directory command for deleting a directory named "exampleDir" is shown on the right. Syntax: `$ rmdir exampleDir`

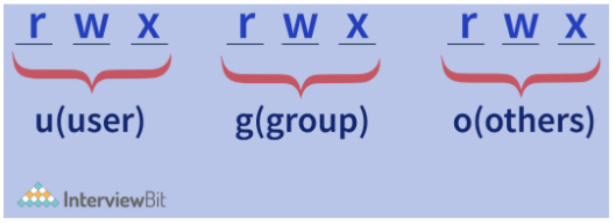
`mv fileName “new file path”`	This command is the move file command. It moves the file to the new path specified.	Example: The mv command moves the file file1.txt to “Docs” directory. Syntax: `$ mv file1.txt “Docs/”`
`mv fileName newName`	This command changes the name of the file from the old name i.e. the fileName to the newName.	Example: The command in the right changes the name of the file file1 to file2. Syntax: `$ mv file1.txt file2.txt`
`find <starting position to search> <expression determining what to find> <options> <what to find>`	This command is used for walking a file hierarchy. It is used to find files/directories and perform operations on them. We can search by file, folder, name, creation date, modification date, etc. There are a number of options available. For instance, exec searches the file that meets the criteria and returns 0 as exit status for successful command execution.	Example: The command in the right is for searching a file with the name file1.txt in the Docs directory. Syntax: `$ find ./Docs -name file1.txt`
`grep <options> pattern fileName`	The full form of this command is a global search for regular expression and printout. This command searches a file for a particular pattern of characters and displays all the lines that contain that pattern. The pattern being searched is called a regular expression (regex). There are a lot of <options> available. For instance, c is an option that is used to only count the number of lines in the file that matches the pattern.	Example: The command to count the number of lines that have “abc” in them in the file file1.txt is shown on the right. Syntax: `$ grep  -c “abc” file1.txt`

`history`	This command displays the list of all the typed commands in the current terminal session.	Syntax: `$ history`
`clear`	Clears the terminal i.e. no previous command will be visible on the screen now.	Syntax: `$ clear`
`hostname`	Shows the name of the system host.	Syntax: `$ hostname`
`hostid`	Displays the id of the host of the system.	Syntax: `$ hostid`
`sudo`	Allows a regular user to run the programs with the security privileges of a superuser or root.	Syntax: `$ sudo`
`apt-get`	This command is used to install and add new packages.	Syntax: `$ apt-get`
`date`	This command is used to show the current date and time.	Example: The command and its output are shown on the right. Syntax: `$ date` `Fri Feb 25 14:58:08 IST 2022`
`cal`	Shows the calendar of the current month.	Example: The command cal and its output is shown on the right. Syntax: `$ cal`
`whoami`	This command displays the name with which you are logged in.	Example: The command is typed in and it shows the username with which the user has logged in. Syntax: `$ whoami Guneet Malhotra`

`whereis [options] fileName`	This command is used to find the location of source/binary file of a command and manuals sections for a specified file in Linux System. This command is similar to the find command but this command is faster as it produces more accurate results by taking less time compared to the find command. There are again a number of options available.	Example: The command to locate apropos command in Linux System is given on the right. Syntax: `$ whereis apropos`

3. File Permission Commands

There are 3 types of people who can use a file and each type has 3 types of access to the file. This is shown in the diagram given below:



The diagram shows that there are 3 types of people accessing a file and they are:

1. User (u)
2. Group (g)
3. Others (o)

Also, the access that we want to give to each of them is of three types:

1. Read (r)
2. Write (w)
3. Execute (x)

So, each of them can have 0 or more out of these 3 permissions. Now let us understand the Linux commands that help us give these permissions to the files.

COMMAND	MEANING	EXAMPLE & SYNTAX
`ls -l fileName`	This command is used to show the file permissions along with the owner and other details of the specified file.	Example: The file permissions along with the owner and other details is shown for the file file1.txt on the right. Syntax: `$ ls -l file1.txt -rw-r--r-- 1 Guneet Malhotra 197121 0 Feb 25 10:51 file1.txt`
`r`	This command represents the read permission.	Example: The command shown in the right adds the read permission to the o (other) class for the file file1.txt. Syntax: `$ chmod o+r file1.txt`
`w`	This command represents the write permission.	Example: This commands adds the write permission for a(all) i.e. user, group and others. Syntax: `$ chmod a+w file1.txt`
`x`	This command represents the execute permission.	Example: This command adds the execution permission for the user. Syntax: `$ chmod u+x file1.txt`

- **Numerical Method for granting file permissions**

There are numeric codes for each permission. They are as follows:

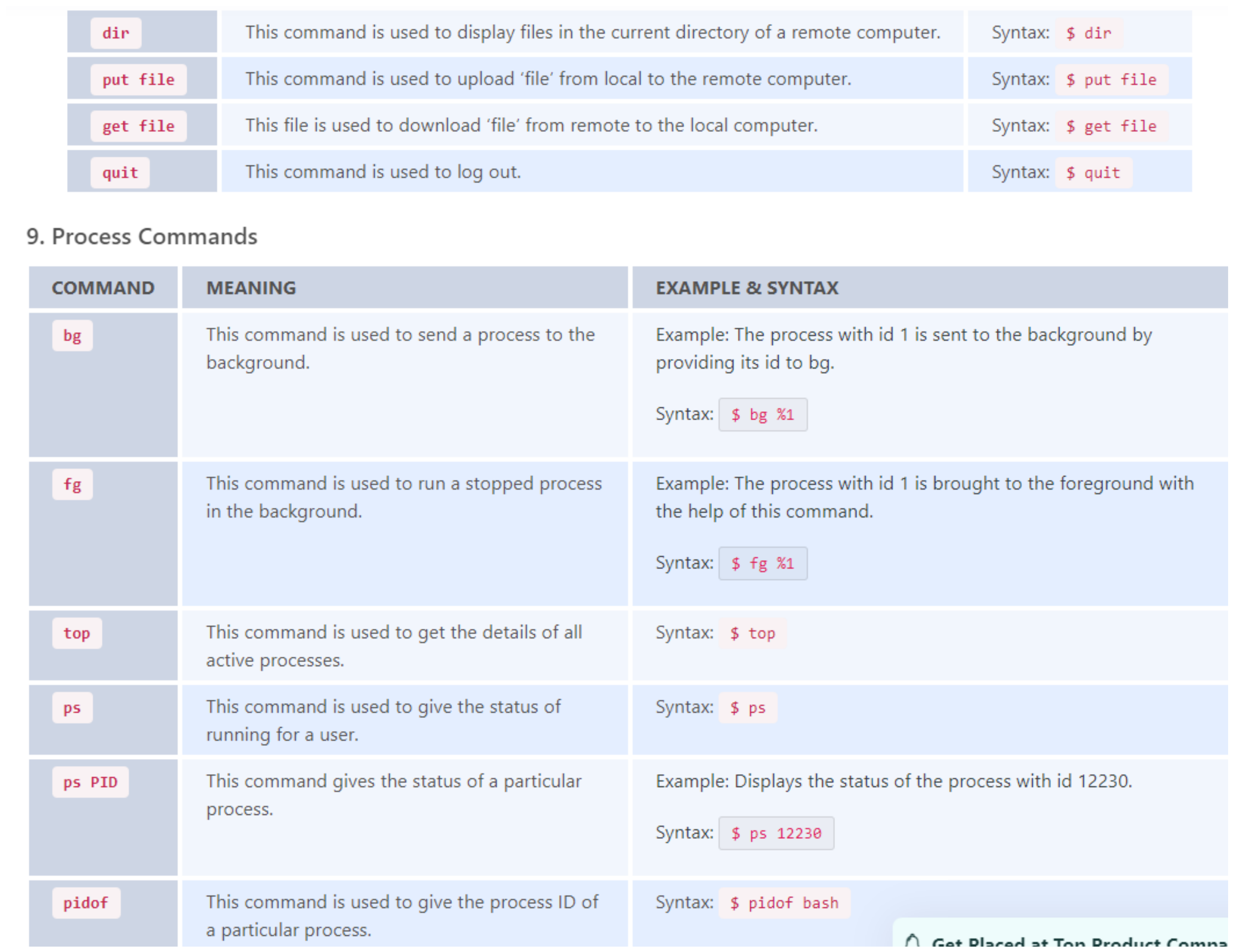
1. r (read) = 4
2. w (write) = 2
3. x (execute) = 1
4. No permissions = 0

`cpu-info`	This command is used to display the information about your CPU. Note that this command is not available by default. It can be used after installation of the necessary package using **sudo apt install cpuinfo**.	Syntax: `$ cpu-info`
`free -h`	This command is used to display the free and used memory. -h is used for converting the information (to be displayed) to human-readable form.	Syntax: `$ free -h`
`lsusb -tv`	List all the USB connected devices.	Syntax: `$ lsusb -tv`
`cat /proc/meminfo`	Gives the information about memory like total and occupied and so on.	Syntax: `$ cat /proc/meminfo`
`du`	This command stands for disk usage and is used to estimate the space usage for a file or directory.	Example: The following command gives the size in human-readable form for the Desktop folder. Syntax: `$ du -h Desktop`

`gzip fileName`	This command is used to compress a file with gzip compression.	Example: The command to zip file1 using gzip compression is shown on the right. Syntax: `$ gzip file1`
`gunzip fileName.gz`	This command is used to unzip a file that has gzip compression.	Example: The command to unzip fileDemo.gz file with gz compression is shown on the right. Syntax: `$ gunzip fileDemo.gz`
`tar cf myDir.tar myDir`	This command is used to create an uncompressed tar archive.	Example: The command to create an uncompressed tar archive for the directory demoDir is shown on the right. Syntax: `$ tar cf demoDir.tar demoDir`
`tar cfz myDir.tar myDir`	This command is used to create a tar archive with gzip compression.	Example: The command to create gzip tar archive for the directory demoDir is shown on the right. Syntax: `$ tar cfz demoDir.tar demoDir`
`tar xf file`	This command is used to extract the contents of any type of tar archive.	Example: The command to extract the content of demoFile tar archive is shown on the right. Syntax: `$ tar xf demoFile`

`env`	This command displays all the environment variables.	Syntax: `$ env`
`echo $Variable`	This command displays the environment variable.	Example: The command at the right will display the INSTANCE environment variable. Syntax: `$ echo $INSTANCE=`
`unset`	This command removes a variable.	Syntax: `$ unset`

`sudo adduser username`	This command is used to add a user.	Syntax: `$ sudo adduser username`
`sudo passwd -l 'username'`	This command is used to change the password of a user.	Example: Command to change the password for user1 is shown Syntax: `$ sudo passwd -l 'user1'`
`sudo userdel -r 'username'`	This command is used to remove a newly created user.	Example: Command to delete the newly created user1 Syntax: `$ sudo userdel -r 'user1'`
`sudo usermod -a -G GROUPNAME USERNAME`	This command is used to add a user to a particular group.	Example: The command to add user2 to group1 is shown. Syntax: `$ sudo usermod -a -G group1 user2`
`Sudo deluser USER GROUPNAME`	This command is used to remove a user from a group.	Example: The command to delete user1 from group1 is shown. Syntax: `$ sudo deluser user1 group1`
`finger`	This command shows the information of all the users logged in.	Syntax: `$ finger`
`finger username`	This command gives information about a particular user.	Example: The command to get information about the user1 is shown on the right. Syntax: `$ finger user1`



`dir`	This command is used to display files in the current directory of a remote computer.	Syntax: `$ dir`
`put file`	This command is used to upload 'file' from local to the remote computer.	Syntax: `$ put file`
`get file`	This file is used to download 'file' from remote to the local computer.	Syntax: `$ get file`
`quit`	This command is used to log out.	Syntax: `$ quit`

9. Process Commands

COMMAND	MEANING	EXAMPLE & SYNTAX
`bg`	This command is used to send a process to the background.	Example: The process with id 1 is sent to the background by providing its id to bg. Syntax: `$ bg %1`
`fg`	This command is used to run a stopped process in the background.	Example: The process with id 1 is brought to the foreground with the help of this command. Syntax: `$ fg %1`
`top`	This command is used to get the details of all active processes.	Syntax: `$ top`
`ps`	This command is used to give the status of running for a user.	Syntax: `$ ps`
`ps PID`	This command gives the status of a particular process.	Example: Displays the status of the process with id 12230. Syntax: `$ ps 12230`
`pidof`	This command is used to give the process ID of a particular process.	Syntax: `$ pidof bash`

Networking protocols are a set of rules and procedures used by devices to communicate with each other over a network. These protocols define how data is transmitted, received, and processed by the devices. Here are some common networking protocols:

1. Transmission Control Protocol/Internet Protocol (TCP/IP): This is the primary protocol used on the internet and is responsible for data transmission between devices. TCP breaks data into packets and ensures their reliable delivery over the network. IP addresses the routing and delivery of the packets.
2. HyperText Transfer Protocol (HTTP): This protocol is used for transferring data over the World Wide Web. It defines how web clients and servers communicate with each other, and how web pages and other resources are accessed and transferred.
3. File Transfer Protocol (FTP): This protocol is used for transferring files over a network, such as the internet. It provides a way for users to upload and download files between their local computer and a remote server.

4. Simple Mail Transfer Protocol (SMTP): This protocol is used for sending and receiving email over a network. SMTP defines how email messages are formatted, transmitted, and delivered to their destination.
5. Domain Name System (DNS): This protocol is used to translate domain names (such as www.example.com) into IP addresses, which are needed for communication between devices on a network.
6. Simple Network Management Protocol (SNMP): This protocol is used for monitoring and managing network devices, such as routers, switches, and servers. SNMP allows network administrators to collect data about the performance and status of devices on their network.

1. Transmission Control Protocol/Internet Protocol (TCP/IP) - Port 80 (HTTP), Port 443 (HTTPS), Port 25 (SMTP), Port 110 (POP3), Port 143 (IMAP), Port 21 (FTP), Port 22 (SSH), Port 23 (Telnet), Port 53 (DNS)
2. HyperText Transfer Protocol (HTTP) - Port 80
3. File Transfer Protocol (FTP) - Port 20 (FTP data), Port 21 (FTP control)
4. Simple Mail Transfer Protocol (SMTP) - Port 25
5. Domain Name System (DNS) - Port 53
6. Simple Network Management Protocol (SNMP) - Port 161 (SNMP) and Port 162 (SNMP trap)

1. 200 OK - The request was successful and the server has returned the requested data.
2. 201 Created - The request has been fulfilled and a new resource has been created.
3. 204 No Content - The request was successful, but there is no data to return.
4. 301 Moved Permanently - The requested resource has been moved permanently to a new location.
5. 302 Found - The requested resource has been temporarily moved to a new location.
6. 400 Bad Request - The server could not understand the request due to invalid syntax or missing parameters.
7. 401 Unauthorized - The requested resource requires authentication or the user does not have sufficient permissions.

8. 403 Forbidden - The requested resource is forbidden and the server refuses to fulfill the request.
9. 404 Not Found - The requested resource could not be found on the server.
10. 500 Internal Server Error - An unexpected error occurred on the server while processing the request.
11. 503 Service Unavailable - The server is currently unable to handle the request due to maintenance or overload.

These status codes are commonly used in HTTP, but other protocols and applications may have their own status codes with different meanings.

HTTPS (Hypertext Transfer Protocol Secure) is a protocol for secure communication over the internet. It is an extension of the HTTP protocol, which is used for transmitting data between a client (such as a web browser) and a server (such as a website).

HTTPS works by encrypting the data transmitted between the client and server using SSL/TLS (Secure Sockets Layer/Transport Layer Security) encryption. This helps to prevent eavesdropping, tampering, and other types of data interception that can occur over an insecure network.

When a user accesses a website using HTTPS, their browser establishes a secure connection with the server and verifies its identity using a digital certificate. The certificate contains information about the server's identity and is issued by a trusted third-party certificate authority.

Once the connection is established, data is transmitted using encryption and can only be decrypted by the intended recipient. This helps to protect sensitive information such as passwords, credit card numbers, and other personal data from being intercepted by hackers or other third parties.

Many websites, particularly those that handle sensitive data such as financial transactions or personal information, use HTTPS to ensure the security and privacy of their users.

When you type a website address (also known as a URL) into your browser, several things happen at the backend to display the website on your screen. Here's a simplified overview of the process:

1. Domain name resolution: Your browser first looks up the domain name of the website (such as www.example.com) in a domain name system (DNS) server to get the IP address of the server hosting the website.
2. TCP/IP connection: Once the IP address is obtained, your browser establishes a TCP/IP connection with the server.
3. HTTP request: Your browser sends an HTTP request to the server, requesting the webpage or resource that you want to view.
4. Server processing: The server receives the request and processes it, retrieving the requested webpage or resource from its files or database.
5. HTTP response: The server sends an HTTP response back to your browser, containing the requested webpage or resource.
6. Browser rendering: Your browser receives the response and renders the webpage or resource, displaying it on your screen.
7. Additional requests: If the webpage contains additional resources (such as images, scripts, or stylesheets), your browser may make additional HTTP requests to retrieve them.
8. Caching: Your browser may cache some of the resources to speed up future requests to the same website.

This is a simplified overview of the process, and there may be additional steps involved depending on the complexity of the website and the technologies used.

A router is a networking device that forwards data packets between computer networks. Routers are used to connect networks and facilitate communication between devices on different networks, such as connecting a home network to the internet.

Routers typically have multiple network interfaces, including WAN (wide area network) interfaces such as Ethernet ports or DSL modems, and LAN (local area network) interfaces such as Wi-Fi or Ethernet ports. They use a routing table to determine the best path for data packets to take through the network, based on their destination IP address.

When a device on a network wants to send data to another device on a different network, it sends the data packet to the router. The router then examines the packet's destination IP address and looks up the best route to forward the packet through the network. This may involve forwarding the packet to another router on the network, or sending it to the internet via the WAN interface.

Routers may also provide additional features such as network address translation (NAT), which allows multiple devices on a LAN to share a single public IP address, and firewall functionality, which can help protect the network from unauthorized access or attacks.

In home or small office environments, routers are often combined with other networking functions such as switches, Wi-Fi access points, and modems into a single device called a "router." These devices are commonly used to connect multiple devices to the internet and provide basic networking functions for home or small office networks.

An IP address, short for Internet Protocol address, is a unique numerical identifier assigned to every device connected to a computer network that uses the Internet Protocol for communication.

IP addresses are used to identify and locate devices on a network, enabling them to communicate with each other and exchange data. IP addresses are made up of four sets of numbers, each ranging from 0 to 255, separated by periods (for example, 192.168.0.1).

There are two types of IP addresses:

1. IPv4: This is the older version of the IP protocol and is still widely used today. IPv4 addresses consist of 32 bits and are written in dotted decimal notation, as described above.
2. IPv6: This is the newer version of the IP protocol, designed to address the growing need for IP addresses as more devices connect to the internet. IPv6 addresses consist of 128 bits and are written in hexadecimal notation (for example, 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

IP addresses can be static, meaning they are assigned permanently to a device, or dynamic, meaning they are assigned temporarily by a network administrator or a service provider. Dynamic IP addresses are often used in home networks and by Internet Service Providers (ISPs) to conserve IP addresses.

IP addresses are essential for the functioning of the internet and are used by devices to locate and communicate with each other.