NETWORKING

&

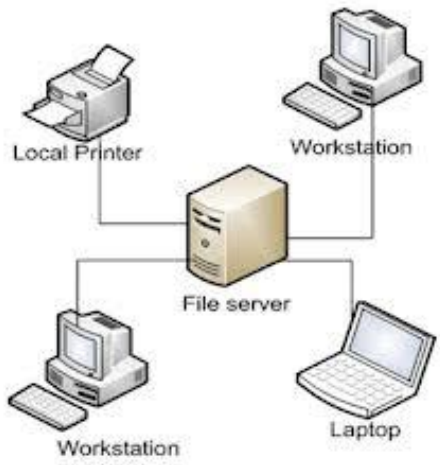
INTERNET ENVIRONMENT

Prepared By: Meghna Bhatt

UNIT: 1 INTRODUCTION TO COMPUTER NETWORK

1. WHAT IS NETWORK?

- A network consists of two or more computers that are linked in order to share resources.
- The computers on a network may be linked through cables, telephone lines, radio waves, satellites, or infrared light beams.
- A network is a collection of computers and other hardware components interconnected by communication channels that allow sharing of resources and information.
- Where at least one process in one device is able to send/receive data to/from at least one process residing in a remote device, then the two devices are said to be in a network.
- **Simply, more than one computer interconnected through a communication medium for information interchange is called a computer network.**
- By using network we can share resources & services.
- The shared resources can be data, printer, a fax modem or services such as database or email services.
- A network is basically a communication system for the computer.



- A resource share
- A path way for data transfer
- A set of rules controlling how to communicate.

NEDDS OF NETWROKS:

- It allows organization to share hardware resources.
- It allows sharing of files/information.
- It supports electronic transfer of text that is email.
- It enables centralized administration & security of resources.
- It allows decentralization of various data & processing function**.**

2. EXPLAIN THE TYPES OF NETWORK.

LOCAL AREA NETWORK (LAN)

- A Local area Network is a group of computers and network communication devices interconnected within a geographically limited area such as campus or building.
- LANs are characterized by
    - They transfer data at high speed.
    - Low error rate.
    - A user administrated private ownership network.
    - Limited geographical area.
    - They exist in limited area-building, campus.
    - Their technology is generally less expensive.

METROPOLITAN AREA NETWORK (MAN)

- A Metropolitan Area Network is a group of computers and network communication devices inter connected within a geographically limited area such as city.

WIDE AREA NETWORK (WAN)

- Wide Area Network interconnects LAN. WAN may be located entirely within a state or country or it may be interconnected around the world.
- WAN are characterized by the following.
    - They exist in a widely scattered geographical area-country continent.
    - They are more susceptible to error due to the large distance data travel.
    - They inter connect multiple LANs.
    - They are more sophisticated technology and complex to build than LAN.

Classification of WAN

- **Enterprise WANs.**
    - An enterprise WAN connects the computer resources of a single organization.
    - An organization with computer operations at several sits build, an enterprise WANS to interconnect the different site.
- **Global WAN:**
    - A global WAN interconnects the networks of several organizations.

3. EXPLAIN NETWORK TOPOLOGIES.

- A topology is map of network, it's a plan for how cabling will interconnect the nodes.
- In other words topology defines the arrangement of nodes, cables & connectivity devices that make up the network.
- The most common topologies are:
    - **Bus**
    - **Ring**
    - **Star**
    - **Mesh**

BUS TOPOLOGY

- In this topology all workstations connected to a common shared cable known as bus.
- Messages are broadcast along the whole bus.
- The connected computers can receive the message and decide whether it is for them or not.
- The basic feature of this topology is that When one computer transmit data then all other computers can listen at the same time.
- The ends of the bus cable in the network are terminated with a terminator.
- A bus topology is commonly used in LAN with Ethernet architecture.
- A failure of a single computer does not affect the performance of the rest of the network.

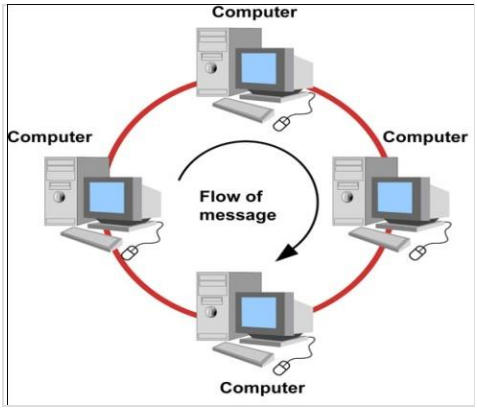
ADVANTAGES:

- Easy to design.
- Simple & flexible.
- Required fewer cables than other topology.
- Easy to add n remove nodes in the BUS.
- A failure of a single computer does not affect the network

DISADVANTAGES:

- Difficult to find error.
- Heavy network traffic in BUS.

RING TOPOLOGY:



- In a ring topology all the computers/workstations/nodes are arrange in such a way that the communication link connect the computers in a ring.
- In this topology any computer can communicate with any other by sending a message around the ring,**but in one direction only**.
- A message goes from one node to another, making a ring.
- Each node takes an active role in transferring the messages.
- This topology is commonly used in LAN with token-ring architecture.

- That is, it is suitable for token passing access method.

ADVANTAGES:

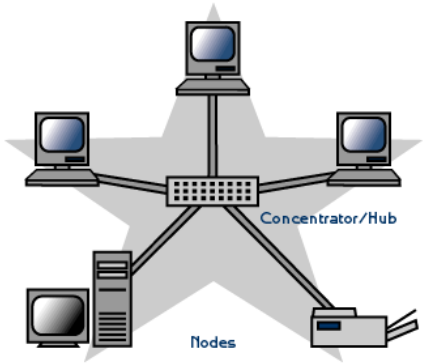
- Required less cable.
- No central access point (server) is required.
- Each computer has equal access to resources.
- Additional components do not affect the performance of network.

DISADVANTAGES:

- Transmission of data becomes slower.
- The addition or removal of computers in a network is difficult.
- It is difficult to find faults in a ring network topology.
- It is not flexible to change the structure of network.
- If any node goes down, the entire ring goes down.
- A break anywhere in a cable brings down the network.

STAR TOPOLOGY:

- In star topology, all workstations/nodes as well as server are connected to a central point such as Hub.
- When a message is going from one computer to another, it is first send to the server, which then re-transmits the message to its direction.
- If a server fails then the entire network does not work.
- A break in cable between a node & center point, brings down the node only.
- In this topology communication link is not shared because each node has its separate link with center point.
- So that it is easy to find out and repair the bad cables.



ADVANTAGES:

- Easy to add or remove nodes.
- Easy to install wires
- Easy to find error.
- Centralized management. It helps in monitoring the network.
- As compared to Bus topology it gives far much better performance
- The failure of a node cannot affect on the entire network.

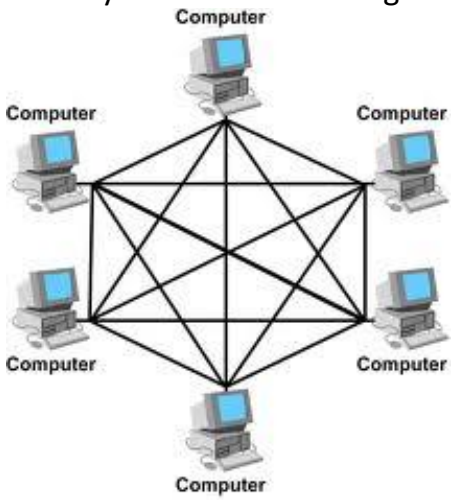
DISADVANTAGES:

- If the hub is failed that the network falls down.
- Required more cable than Bus topology.
- Expensive than Bus topology.

- Too much dependency on hub

MESH TOPOLOGY:

- In this topology each of the network node, computer and other devices are interconnected with one another.
- Every node not only sends its own signals but also relays data from other nodes.
- In fact a true mesh topology is the one where every node is connected to every other node in the network.
- This topology is most commonly used in WAN configurations.



ADVANTAGES:

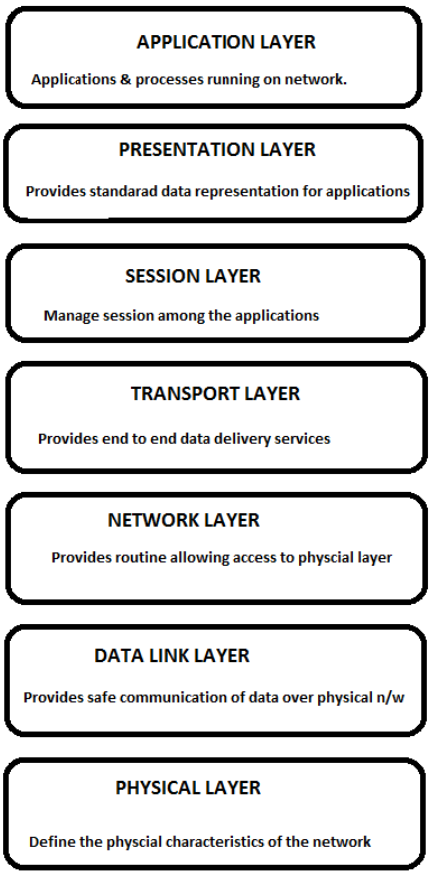
- Data can be transmitted from different devices simultaneously.
- Less traffic.
- If one link is failed than data can be transmit by using another link.
- Easy to find exception & repair the fault.
- Provides more security.

DISADVANTAGES:

- It not easy to set up and maintain.
- Very expensive.
- Required large amount of cables.
- Required larger space.
- Required complex hardware to connect the node.

4. EXPLAIN THE OSI REFERENCE MODEL

- OSI reference model was developed by ISO (international standard organization) as a first step toward international standardisation of program used in various layers.
- OSI- open system interconnection is deal with connecting open system that is the system which is open for all communication with other system.
- OSI contains seven layers; each layer should perform well defined function.
- OSI contains following layers:



PHYSICAL LAYER:

- This layer defines electrical and physical specification for devices.
- It defines the relationship between a device and transmission medium such as cable:
- This layer is responsible for establishment and termination of connection to communication medium.
- It performs modulation or conversion between the representation of digital data in user equipment and the corresponding signal transmitted over communication channels**.**

DATA LINK LAYER:

- This layer provides the functional to transfer between network entities and to detect and possibly correct errors that may occur in the physical layer.
- It provides an error free transformation of data frames from one node to another over physical layers and allows the layers above it.
- This layer defines format of data on the network.
- DLL also handles physical and logical connection to the packet destination using a network interface.

NETWORK LAYER:

- This layer is the in charge of packet addressing, convert logical address into physical address, making it possible to data packets to arrive at their destination.
- This layer is responsible to setting routing.
- This layer determines that how data transmits between the network devices.

TRANSPORT LAYER:

- This layer manages end to end delivery in a network and provides an error checking so that no duplication or errors are occurring in the data transfer across the network.
- It is also provided acknowledgement of successful data transmission and retransmit data if no error free data was transferred.
- This layer ensure that messages are delivered error free in sequence and with no looses and duplication.

SESSION LAYER:

- This layer responsible for establishment, maintenance and termination of session.
- It allows to application processes on different machine to established, maintain use and terminate the session.
- This layer reconnects session if it disconnected. It reports log and upper layer error.
- This layer manages who can transfer the data in certain amount of time and how long.

PRESENTATION LAYER:

- This layer presents the data into a uniform format and masks the different types of data between two dissimilar systems.
- This layer formats data to be presented to application layer.
- This layer may translate data from one format used by application layer into common format at sending station and then translate common format at receiving.

APPLICATION LAYER:

- This layer serves as windows for users and application processes to access network services.
- Application layer makes interface between programs that sending and receiving data.
- It provides the end user services like mail, FTP, DNS etc.

5. WHAT IS INTERNET?

- The term Internet is combination on two words, interconnection and network.
- The internet is a collection of countless small interconnected networks.
- Internet is a worldwide network. Each network could be linked to millions of computers.
- The basic idea behind to develop internet is that we can exchange information between the two computers.
- A simple network is the one that exists between your computer, monitor and the printer.
- A large network can be thought of as a linkup between computers in a building.
- If this network is linked to any other such network say from another, we would have an Internet.
- Today, the Internet is linked and used by

- Universities
- Governments
- Banks
- Companies
- Scientists
- Researchers
- Many other organizations. All of them use it.

ADVANTAGES OF THE INTERNET:

- Internet is the largest most complete learning tool in the world.
- Through the Internet you can find knowledge resources that allow you to study virtually any subject.
- Not only that but you can communicate quickly and effectively with others who are also interested in the same subject.
- A variety of programs have been installed on the Internet to use these services, combine them, or make them easier to use.
- Intranet components are relatively cheap and some are free also.
- Through Intranets people can easily share their ideas and information.
- With e-mail you can send and receive instant electronic messages.
- There is a huge amount of information available on the internet for just about every subject that you can use.
- You can communicate with each other by using Chat facility.
- Many services are provided on the internet like net banking, job searching, purchasing tickets, hotel reservations, guidance services etc.
- Along with getting information on the Internet, you can also shop online. You can purchase or sales things by using internet.
- You can freely download innumerable, software's like utilities, games, music, videos, movies, etc from the Internet.

6. EXPLAIN : TCP/IP

TCP/IP:

- The TCP/IP stands for Transmission Control Protocol Internet Protocol.
- This protocol suite was originally developed by the United States Department of Defense (DoD) to provide service on large internetworks that incorporate a variety of computer types.
- Part of the main purpose of this protocol was for it to be hardware independent.
- In some literature, the TCP/IP protocol suite is referred to as the DoD model.
- In TCP/IP, TCP is an internetwork connection oriented protocol that corresponds to OSI transport layer.
- TCP provides full-duplex end to end connection.

HOW TCP/IP WORKS:

- If a program wishes to send information form node-A to node-B on the same network, TCP will provide the packet sequencing & other services required for end-to-end communications & hence for this, TCP doesn't need an IP.
- But if a program on node-A on network-X wishes to connect with a remote node-B on network-Y then internetwork routing as well conversion would be necessary.

- So IP takes the data packets from TCP and then routs the packets to the IP layer at the remote node for delivery.
- TCP/IP is suitable & used in internet.

INTERNET PROTOCOL:

- The Internet Protocol (IP) is a connectionless protocol that provides datagram service, and IP packets are most commonly referred to as IP datagram.
- IP is a packet switching protocol that performs the addressing and route selection.
- An IP header is appended to packets, which are transmitted as frames by lower-level protocols.
- IP routes packets through internetworks by utilizing routing tables that are referenced at each hop.
- Routing determinations are made by consulting logical and physical network device information, as provided by the Address Resolution Protocol (ARP).
- IP performs packet disassembly and reassembly as required by packet size limitations defined for the Data Link and Physical layers being implemented.
- IP also performs error checking on the header data using a checksum, although data from upper layers is not error checked.

7. SHORT NOTE : ISP

- ISP means Internet Service Provider.
- ISP is an organization that provides an internet connection services like dial-up, leased-line, ISDN, ADSL and cable modem connection.

PARTS OF ISP

- **PHONE NUMBER**
    - ISPs have many phone numbers that can connect the internet in your computer.
- **PRICE**
    - It's generally charge for the hours they provide so you can calculate your internet's bill by calculating your usage required.
- **SOFTWARE**
    - ISP provides many types of software's that you can connect the Internet and this type of software's run on the Windows Operating System like Windows 95, Windows 98, Windows 2000, Windows NT, Windows XP etc.
- **SUPPORT**
    - If you have a problem for Internet connection then ISP gives technical support number that solves the problem and it should be open 24 hours a day and 7 days a week.
- **SPEED**
    - The speed of ISP is very faster because it can work with 28.8 KBPS, 33.6 KBPS and 56 KBPS modems.
- **Generally there are three types of ISP**
    - Online companies
    - National access providers and
    - Local access providers.

INTRANET	• An intranet is a secure and private enterprise network that shares data ,application resources. • An intranet is a private network accessible only to an organization's staff. • It may consist of many interlinked local area networks and also use leased lines in the wide area network. • Intranet is differed than internet; internet provide public network while intranet provide private network.
VSAT	• The complete name of VSAT is**Very Small Aperture Terminals.** • It's a small telecommunication earth station that receives and transmits real-time data via satellite. • A VSAT transmits narrow and broadband signals to orbital satellites. • The data from the satellites is then transmitted to different hubs in other locations. • The majority of VSAT antennas range from 75 cm to 1.2 m. • Data rates, in most cases, range from 4 kbit/s up to 16 Mbit/s.
PORTAL	• A website that serves as a gateway or a main entry point on the internet to a specific field-of-interest or an industry. • A portal provides at least four essential services: (1) search engine(s), (2) email, (3) links to other related sites (4) Personalized content • It may also provide facilities such as chat, members list, free downloads, etc. • Example: MSN, Netcenter, rediff, indiatimes and Yahoo
DNS	• DNS stands for domain name server. • DNS is a host name to IP address translation service. • DNS are the Internet's equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses. This is necessary because, although domain names are easy for people to remember, computers or machines, access websites based on IP addresses.

UNIT: 2 APPLICATION OF INTERNET

1. SHORT NOTE:WWW

- The complete name of www is World Wide Web.
- The World Wide Web is the fastest growing, and in many ways, the most exciting and intriguing part of the Internet.
- The World Wide Web is a globally connected network.
- The World Wide Web (referred to as "Web") is a collection of millions of files stored on thousands of computers (called Web servers) all over the world.
- These files represent text documents, pictures, video, sounds, programs, interactive environments, and just about any other kind of information that has ever been recorded in computer files.
- The Web is probably the largest and most different collection of information.
- What unites these files is a system for linking one file to another and transmitting them across the internet.
- The HTML language allows a file to contain links to related files.
- Such a link (also called a hyperlink) contains the information necessary to location the related file on the internet.
- When you connect to the internet and use a Web browser program, you can read, view, hear, or otherwise interact with the Web without paying attention to whether the information that you are accessing is stored on a computer

2. WRITE A NOTE ON: WEB BROWSER

- A Web browser is a program that your computer runs to communicate with Web servers on the Internet, which enables it to download and display the Web pages that you request.
- At a minimum, a Web browser must understand HTML and display text.
- In recent years Internet users have come to expect a lot more.
- A state-of-the-art Web browser provides a full multimedia experience, complete with pictures, sound, video, and even 3-D imaging.
- Because a Web browser has the ability to interpret or display so many types of files, you often may use a Web browser even when you aren't connected to the Internet.
- There are two types of browsers are mainly used

1. **Text Only Browser**
   - A browser that does not show images.
   - It does not have images turned off; rather it just doesn't display them.
   - A text-only browser is just that, a browser that displays text only.
   - While a user browses web pages, these applications grab all available text including alt attributes, summary tags, etc
   - A good example of this type of browser is Lynx.

2. **Graphical Browser**
   - A browser capable of displaying pictures.
   - A graphical browser is web browsers using software application to display & interact with the various images on web page by the users.
   - Such graphic browsers are first distributed by Mosaic.

- Examples**: Internet explorer, Google chrome, Mozilla firebox, Netscape navigator**

ELEMENTS OF A BROWSER WINDOW

- Most browser windows have the same basic layout. From top to bottom, you find these basicelements:
  - Menu bar
  - Toolbars
  - Address or Location window
  - Viewing window
  - Status bar

3. EXPLAIN THE ELEMENTS OF WEB OR WWW.

- WWW means World Wide Web.
- WWW is software application that publishes the website on the internet.
- WWW is also called Web because a large number of documents are interconnected together that becomes a network.
- WWW contains the following elements.
  - Web server
  - Web browser/ web client
  - Web site
  - Web pages

WEB SERVER

- Web Server is a server that stores more number of websites.
- Web Server connects the Internet and it runs the websites on that internet.
- The responsibility of web server is storing, retrieving and distributing the websites on the internet.
- The web server handles the server side programming, security characteristics and publishing the websites.
- When we writes the website's name on the web browser like Internet Explorer then web browser send the website's name to the web server with the help of HTTP.
- After this process, web server receives the website's name from the web browser and it sends back the website to the web browser with the help of HTTP.
- Here, HTTP is a protocol that can create the links between the web browser and web server.
- Examples:
  APACHE WEB SERVER, INTERNET INFORMATION SERVER (IIS), NOVELL SERVER and LOTUS DOMINO SERVER.

WEB BROWSER / WEB CLIENT

- A Web browser is a program that your computer runs to communicate with Web servers on the Internet, which enables it to download and display the Web pages that you request.
- Web browser or web client is a computer program that requests the websites from the web server on the Internet.

- A web browser is a software application that functions as an interface between a user and an internet.
- Examples:
  MICROSOFT INTERNET EXPLORER, NETSCAPE NAVIGATOR, MOSAIC , GOOGLE CHROM, MOZILA FIREFOX, OPERA

WEB PAGE

- Web page is a page that stores the information of the particular person or an organization.
- We can create the web page with the help of many types of web programming languages like SGML, HTML, DHTML, and PHP and so on.
- With the help of web programming languages, information is stored in a plain text file with the file extension like **.HTML, .DHTML, .PHP** and so on.
- Web page is a combination of text and web programming languages that format the text.
- Generally web page contain three main things : 1) Text 2) Graphics 3) Links

WEB SITE

- A website is a collection of web pages that property of particular person or an organization.
- It means it stores the information that property of the particular person or an organization.
- In website, the beginning of a file that is called Home Page.
- From the home page, you can get to all other pages on their site.
- A very large website may be spread over a number of servers in different locations.
- For example, IBM's website may be spread over a number of servers in different locations around the world.

4. EXPLAIN SEARCH ENGINE IN BRIEF.

- A web search engine is designed to search information on WWW.
- A web search engine is a computer program that allows you to submit a form that contains a word that describing the specific information that locate on the website.
- Web search engine searches all the contents of a website that try to match your search.
- After this process, web search engine returns a list of click able website that contains your search.
- This type of list gives the better matches that appearing at the top.
- A web search engine also allows you to resubmit the search.
- Examples :
  AltaVista, Google, Yahoo, Info seek

5. SHORT NOTE : E-MAIL

- Electronic mail, or email, may be the most heavily used feature of the internet.

- You can use it to send message to anyone who is connected to the internet orconnected to a computer network that has a connection to the internet, such anonline service.
- Millions of people send & receive e-mail every day.
- Email is a great way to keep up with far-flung relatives, friends, co-workers indifferent branches of your company, & colleagues in your field.
- E-mail addressing:-
- To send e-mail to someone, you must know his or her internet e-mail address. Unlike thepostal service, which can often deliver imprecisely address letters the mechanics of the internetrequire an exact e-mail address.
- Internet e-mail addresses look like this:abc@yahoo.com

Host or Domain name:-

- Second part of e-mail address is called host or domain name.
- The host name provides theinternet location of the mailbox, usually the name of a computer owned by a company or internetservice.

How can you receive & send E-mail:-

Receive E-mail: (Incoming messages):-

- Depending on the type of connection that you have, you may download e-mail from the mail server to your computer or you may read your e-mail while it sites on the mail server.
- To read your e-mail you need an e-mail application (also called a mail client or pop client)such as outlook or Eudora.
- A client application works in concert with a server in the case of email, a mail server collects your e-mail & your mail client enables you to read it.
- (mail server receive & store e-mail Messages in mailboxes by using a protocol called post office protocol(POP)

Send E-mail: (outgoing messages):-

- Sending e-mail requires a similar process.
- You write messages on your own computer by using your e-mail application.
- You transfer the messages to an SMTP(Simple Mail Transfer Protocol) server.
- A mail server that accepts outgoing e-mail.
- The SMTP server that takes care of sending your e-mail messages may be a different server that the pop server that collects your e-mail.

6. SHORT NOTE : E-COMMERECE

- E-commerce -- electronic commerce or EC --
- It is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network.
- E-commerce (electronic-commerce) refers to business over the Internet.
- Web sites such as Amazon.com, Buy.com, and eBay are all e-commerce sites.
- The e-commerce field is really large and there are a lot of different models.
- We are going to use two categories:
- A general one based on who the buyers and sellers are.

- Another one with different e-commerce “models”.
- Each business focuses on a type of client, and depending on who they are, we can classify them:
- **B2B (Business-to-Business):**
    - Businesses whose clients are also businesses or organizations.
    - For example, we could think about a construction materials company selling its products to arquitects and interior designers.
- **B2C (Business-to-Consumer):**
    - Businesses that sell their products or services directly to the consumer.
    - This is the usual type and there are thousands of examples of clothes, shoes or electronics stores.
- **C2B (Consumer-to-Business):**
    - sites in which consumers offer products or services and businesses bid on them.
    - We are talking about the traditional websites for freelancers such as Freelancer, Twago, Nubelo or Adtriboo.
- **C2C (Consumer-to-Consumer):**
    - businesses that facilitate the selling of products amongst consumers.
    - The clearest example is eBay or any other second hand website.

ADVANTAGES:-

- Faster buying/selling procedure, as well as easy to find products.
- Buying/selling 24/7.
- More reach to customers, there is no theoretical geographic limitations.
- Low operational costs and better quality of services.
- No need of physical company set-ups.
- Easy to start and manage a business.
- Customers can easily select products from different providers without moving around physically.
- e-commerce is the lowered Advertisement cost
- customers can easily find their product with single click
- customers easily browse multiple e-commerce merchants and find the best prices
- get immediate feedback on prices, features to company
- no geographical limitation

DISADVANTAGES:-

- No guarantee of product quality.
- Customer loyalty becomes a bigger issue as there is a minimal direct customer-company interaction.
- Inability to experience products beforehand leads to more checkout dropouts.
- Anyone can start an online business, which sometimes leads to scam and phishing sites.
- Hackers target web shops more often than you think.
- Mechanical failures can get quite more punishing.
- Many fraud sites bad which eat up customers’ money.

7. SHORT NOTE : M-COMMERCE

- M-commerce is an upgraded version of e-commerce.
- **M-Commerce** also called as **Mobile Commerce** involves the online transactions through the wireless handheld devices such as mobile phone, laptop, palmtop, tablet, or any other personal digital assistant.
- mobile commerce would be the buying and selling of products – or the conduct of commercial transactions and activities – through telecommunication and other mobile devices that run or operate on wireless network technologies.
- It does not require the user to sit at the computer that is plugged in and perform the commercial transactions.
- Through M-Commerce, people can perform several functions such as :
    - pay bills
    - money Transfer
    - mobile browsing
    - marketing & advertisement
    - information services
    - buy and sell goods and services
    - access emails
    - book movie tickets
    - make railway reservations
    - order books
    - read and watch the news
- technology behind m-commerce, which is based on the Wireless Application Protocol (WAP)

ADVANTAGES:

1. Provides Push Notification
2. No Needs For Wifi Connectivity Set Up
3. Provides Convenience Way To Access Ecommerce Services
4. Provides flexibility to use mobile as well as online services together.
5. the users can pay their mobile bills, electricity bills, without standing in the long queues
6. Mobile device is usually dedicated to a specific user so that it is personal. Users can do whatever they want with their handheld devices
7. connect & access commercial facility any time
8. Mobile devices can be used for entertainment purposes, for personal and even for presentations to people and clients.
9. Provide service to store data online
10. M-Commerce site to reach them by giving different and better deals in comparison of their competitor
11. Buyers can have look thousands of items on their cell phones and there is no need of online checkout process.

12. Companies try t reach to the consumer directly through M-Commerce,so users have no need to go far to the store physically and at the end it saves user's time and money

DISADVANTAGES:

1. The Screen of mobile phones is generally small as compared to the computer screen and, therefore, the display of cellular gadgets may not influence the user to make the purchase.
2. Poor connectivity also makes M-commerce failure.
3. Information shared through the wireless medium have higher chances of getting hacked

8. EXPLAIN VARIOUS TYPES OF PAYMENT MATHODS

Digital cash:	• It is also known as e-currency, e-money, electronic cash, electronic currency, digital money, digital currency, cyber currency. • Digital Cash acts much like real cash, except that it's not on paper. • it'sa system in which a person can securely pay for goods or services electronically without necessarily involving a bank to mediate the transaction. • A Digital Cash transaction usually involves three types of users: ○ a Payer or consumer ○ a Payee such as a merchant ○ a financial network like a Bank with whom both Payer and Payee have accounts. • And usually involves three transaction: ○ Withdrawal, the Payer transfer some money from his/her account to her wallet (which could be a computer or smart case) ○ Payment, the Payer transfer the withdrawn money to the Payee's wallet ○ Deposit, the Payee transfers the received money to his/her account. • for example : • **Paypal**
Electronic cheque	• An electronic version or representation of a paper cheque. • An e-Cheque is an electronic document which substitutes the paper check for online transactions. • Digital signatures (based on public key cryptography) replace handwritten signatures. • The account holder writes an e-check (or e-cheque) using a computer or other type of electronic device and transmits the e-cheque to the payee electronically. • Like paper cheques, e-checks are signed by the payer and endorsed by the payee. • Rather than handwritten or machine-stamped signatures, however, e-checks are affixed with digital signatures, using a combination of

	smart cards and digital certificates. • The payee deposits the e-check, receives credit, and the payee's bank clears the e-check to the paying bank
Debit Card	• A debit cards also known as a bank card, plastic card or check card. • It is a plastic payment card that can be used instead of cash when making purchases. • The debits cards are usually linked with customers' bank accounts; It can be savings or current account • It is similar to a credit card, but unlike a credit card, the money comes directly from the user's bank account when performing a transaction • When the card is swiped in a machine or used at online payment gateways, the payment amount is debited from account it's been linked to automatically. • And the customers who have activated the SMS alert service, immediately get a notification regarding the payment. • They can also get email notification if applicable.
Credit Card	• Credit cards have become a common electronic payment instrument. • A convenient payment instrument, reducing the amount of cash that we carry around especially for travel purposes, suitable for conducting purchases over the phone and online • The card issuer (usually a bank) creates a revolving account and grants a line of credit to the cardholder, from which the cardholder can borrow money for payment to a merchant or as a cash advance. • In other words, credit cards combine payment services with extensions of credit • Credit cards are lines of credit. • When you use a credit card, the issuer puts money toward the transaction. • This is a loan you are expected to pay back in full (usually within 30 days), unless you want to be charged interest.

9. EXPLAIN FOLLOWING TERMS

Cyber Law	• Cyber means the use of Internet technologies and computers it includes computers, networks, software, data storage devices, Internet, websites, emails, ATM machines etc. • To protect the cyber crime over Internet, this law is Passed to protect the Internet cyber crime. • This law is approved by the government. Cyber law Includes: ○ Cyber crimes ○ Electronic and Digital Signatures ○ Intellectual Property ○ Data protection and privacy
Firewall	• firewalls work like a filter between your computer/network and the Internet.

	• You can program what you want to get out and what you want to get in. • Everything else is not allowed. • The main goal of a is to protect your personal computer and private network from malicious mischief. • A firewall is a network security device that monitors incoming and outgoing network traffic. • Firewalls have been a first line of defense in network security for over 25 years. • A firewall can be hardware, software, or both.
Cookies	• A cookie is a text file that a Web browser stores on a user's machine. • Cookies are a way for Web applications to maintain application state. • They are designed to hold a modest amount of data specific to a particular client and website, and can be accessed either by the web server or the client computer.
Hackers	• Hacking is identifying weakness in computer systems or networks to exploit its weaknesses to gain access. • Hackers are most often programmers. As such, hackers obtain advanced knowledge of operating systems and programming languages. • They might discover holes within systems and the reasons for such holes. • The term hacker may refer to anyone with technical skills, but it often refers to a person who uses his or her abilities to gain unauthorized access to systems or networks in order to commit crimes. • Example of Hacking: Using password cracking algorithm to gain access to a system
Crackers	• A cracker is someone who breaks into someone else's computer system, often on a network; bypasses passwords or licenses in computer programs; or in other ways intentionally breaches computer security. • A cracker can be doing this for profit, maliciously, for some altruistic purpose or cause, or because the challenge is there.

UNIT: 3 HTML

1. WHAT IS HTML?

- HTML means **HYPER TEXT MARKUP LANGUAGE.**
- It is a standard language for creating/designing web pages
- HTML was invented by **Tim Berners Lee.**
- HTML is a web programming language that can be used to create the website on the internet.
- HTML contains the tags that decide how a web page looks when it displayed on web browser.
- HTML tags also make the possible hyperlinks that connect the information on the website.
- There are two types of tags like
    - Start tag / Opening tag
    - End tag / Closing tag.
- End tag is different from start tag in which end tag has / before them.
- The group of start tag and end tag that is called container tags or pair tags.
- A tag in which there is only start tag but no end tag so it is called empty tag or singular tag.
- The text between start tag and end tag that is called element content.
- We can write the HTML tags either in small or in capital.
- We can write the HTML program in text editor like Notepad, WordPad and so on.
- We can save the HTML program with .html or .htm extension.
- Every HTML program runs on the web browser like Internet Explorer, Netscape Navigator and so on.

```
<HTML>
<HEAD>
        <TITLE> MY FIRST WEB DOCUMENT</TITLE>
</HEAD>
<BODY>
        <H1> WEB PAGE DEVELOPING USING HTML</H1>
</BODY>
</HTML>
```

2. WHAT IS TAG? EXPLAIN ITS TYPE.

- Tags are instructions that are embedded into the text of the documents.
- An HTML tags is a signal to a browser that it should do something other than just throw text upon the screen.
- By convention all HTML tags begin with open angle brackets and end with closing angle brackets.
- In short, any HTML page would typically contain the keywords for creatingHTML page is known as Tags.
- HTML tags can be of two types.
    - **PAIRED TAG**
        - These tagsare also known as container or bracketed tag.

- The tags that are grouped in opening & closing tag pair are referred as paired tag.
- A tag is said to be a companion tag.
- In this tag one tag is used for starting or opening & the other is used to terminate the effect of tag.
- Examples: <HTML>,<B>, <I>, <OL>, <UL> etc.

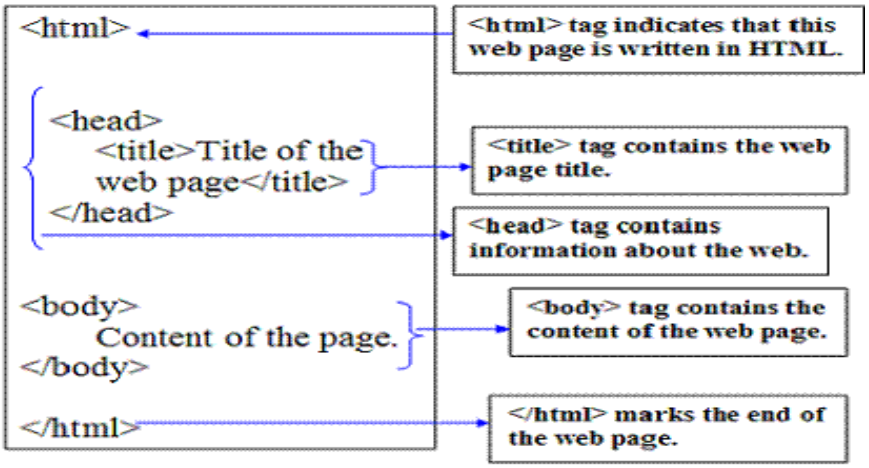
- **SINGULAR TAG**
  - It is also called stand alone tag or empty tag.
  - A stand alone tag does not have a companion tag.
  - Examples:
, <HR>, <LI> etc

RULES OF USING TAG:

- Tab must be written in angular bracketed. (<>).
- Paired tag must be closed using / (forward slash).
- Tag is not case sensitive.
- Tag and its attribute must be separated by space.
- Attribute and its value must be separated by = sign.

3. EXPLAI THE STRUCTURE OF HTML DOCUMENT

- An HTML document has two main parts:
- head. : The head element contains title and meta data of a web document.
- body. : The body element contains the information that you want to display on a web page.



<HTML>

- HTML document begins and ends with HTML tag i.e. <HTML></HTML>
- Here <HTML> indicates the browser that it is a HTML document and </HTML> tells the browser that HTML document is completed.

<HEAD>

- Header Tag i.e. <HEAD></HEAD>
- Header Tag does not contain any text; it only contains the Title Tag in it.

<TITLE>

- Title tag i.e. <TITLE></TITLE>
- Anything written between this tags is not displayed on the screen but it is used to identify the Webpage.

<BODY>

- Body tag i.e.<BODY></BODY>

- This is the main part of HTML document.
- The content which is to be displayed on screen as webpage should be written here.
- Body Tag contains the text as well as various tags but only the text will be displayed on Webpage.

EXAMPLE:

```
<HTML>
<HEAD>
        <TITLE>   MY FIRST WEB PAGE </TITLE>
</HEAD>
<BODY>
          HTML IS HYPER TEXT MARK-UP LANGUAGE
</BODY>
</HTML>
```

4. EXPLAIN
 AND <NOBR> AND <P>.

- This tag is also known as singular tag.
- The line breaks tag
 forces text to begin on a new line.

<NOBR>

- The <NOBR> element forces no break. That means all the text between
- The <NOBR> and </NOBR> elements cannot have line breaks inserted.

<P>

- The <p> tag defines a paragraph.
- Browsers automatically add some space (margin) before and after each <p> element. The margins can be modified with CSS (with the margin properties).
- It contains 2 attribute: align and class.
- Align:
    - Default value of align is align=left
    - We can set alignment of paragraph either left, right or center.

5. EXPLAIN <FONT>.

- The font can be varied as per the customized needs for design value using the <FONT> and </FONT> tag pair.

ATTRIBUTES

- **SIZE**
    - You can increase or decrease the size of the font.
    - The size should be between –7 and +7.
- **COLOR**
    - You can also specify a color for a particular selection of text
- **FACE**
    - In addition, HTML also allows the designer to use different font faces for their text.
    - Each face is specified as the font name used in your computer.

EXAMPLE:

<FONT FACE=”ARIAL” COLOR=”RED” SIZE=3>

6. EXPLAIN <BODY> WITH ATTRIBUTES.

- It is the largest part of the document.
- The body of an HTML document contains all the text and images that make up the page, together with the entire HTML elements that provide the formatting of the page.

ATTRIBUTES OF BODY TAG

- **BACKGROUND:**
  - This attribute is used to set an image in the background of the file that will be tiled across the browser window.

EXAMPLE:
<BODY BACKGROUND=”MYFILE.JPG”></BODY>

- **TEXT**
  - THIS ATTRIBUTE IS USED TO CONTROL THE COLOUR OF ALL THE NORMAL TEXT IN THE DOCUMENT.

EXAMPLE
<BODY TEXT=”CYAN”> CONTENT FOR DOCUMENT HERE </BODY>

- **LINK, VLINK, AND ALINK**
  - This attributes allow control over the link text colouring.
  - Linkdenotes the link, Vlink stands for visited link, and Alink standsfor active link.
  - The default colouring of these are:
    - Link = blue
    - Vlink = purple
    - Alink = red

EXAMPLE
<BODY LINK=”#rrggbb” VLINK=”#RRGGBB”ALINK=”#RRGGBB”> document content here </BODY>

- **BGCOLOR**
  - This attribute gives the specified background color to the document.

EXAMPLE
<BODY BGCOLOR=”pink”> document contents</BODY>

- **MARGIN**
  - This attribute allows the user to set the top, left hand margin of the document.

EXAMPLE
<BODY LEFTMARGIN=”15”>content of document</BODY>

7. EXPLAIN HEADERS TAG IN BRIEF.

- <h1>.....<h6>: These tags are known as headers tag.
- There are six levels of these tags.
- According to each level one tag is used.
- <h1> is used to display largest size heading &<h6> shows smallest size heading.

EXAMPLE:

```
<HTML>
<BODY>
<H1>GEETANJALI COLLEGE</H1>
<H2>GEETANJALI COLLEGE</H2>
<H3>GEETANJALI COLLEGE</H3>
<H4>GEETANJALI COLLEGE</H4>
<H5>GEETANJALI COLLEGE</H5>
<H6>GEETANJALI COLLEGE</H6>
</BODY>
</HTML>
```

==========

OUTPUT:

GEETANJALI COLLEGE
GEETANJALI COLLEGE
GEETANJALI COLLEGE
GEETANJALI COLLEGE
GEETANJALI COLLEGE
GEETANJALI COLLEGE

8. EXPLAIN <HR> WITH ALL ATTRIBUTES.

- <HR> (horizontal ruler) tag is also known as singular tag.
- The <HR> tag inserts horizontal lines in the web page

ATTRIBUTES

- **SIZE**
  - The SIZE attribute gives thickness to the horizontal ruler value and should be given in pixels.
- **ALIGN**
  - It is possible to specify the horizontal alignment of the rule.
  - Possible values: left, right, or center.
  - Default value: Center.
- **WIDTH**
  - With the WIDTH attributes, one can specify an exact width in pixels, or a relative width measured in percent of document width.
- **NOSHADE**
  - For those times when a solid bar is required, the NOSHADE attribute specifies that the horizontal rule should not be shaded at all.
- **COLOR**
  - This attribute allows you set colour for line
  - we can use colour name, RGB() or hexadecimal colour code

EXAMPLE

```
<HR SIZE=5 WIDTH=80% COLOR=”BLACK ALIGN=”RIGHT”>
```

9. EXPLAIN TEXT FORMATTING TAG AVILABLE IN HTML.

- The following HTML tags are used to format the appearance of the text on your web page.

<H1> TO <H6>

- There are 6 levels of headings available, from h1 for the largest and most important heading, down to h6 for the smallest heading.

<B>- BOLD

- The text in between the tags will be bold, and stand out against text around it, the same as in a word processor.

<I>-Italic

- Also working the same way as a word processor, italics display the text at a slight angle.

<U>-Underline

- Again, the same as underline in a word processor

<STRIKE>-Strike out

- Puts a line right through the centre of the text, crossing it out. Often used to show that text is old and no longer relevant. Also works by using <s></s> instead.

<PRE>-Preformatted Text

- Any text between the pre tags, including spaces, carriage returns and punctuation, will appear in the browser as it would in a text editor

<TT>-Typewriter Text

- The text appears to have been typed by a typewriter, in a fixed-width font.

<STRONG> - STRONG

- The HTML element defines strong text, with added semantic "strong" importance.

<BIG>- BIG

- Make text bigger than normal text

<SMALL>- SMALL

- Instead of having to set a font size, you can use the small tag to render text slightly smaller than the text around it.

<SUB>- SUBSCRIPT

- The HTML <SUB> element defines subscripted text.

<SUP>- SUPERSCRIPT

- The HTML <SUP> element defines superscripted text.

EXAMPLE

```
<HTML>
<BODY>
<H1> FORMATTING TAG</H1>
<B> GEETANJALI</B><BR>
<I GEETANJALI </I><BR>
<U> GEETANJALI </U><BR>
<STRONG> GEETANJALI </STRONG><BR>
<BIG GEETANJALI </BIG><BR>
<SMALL GEETANJALI </SMALL><BR>
<TT> GEETANJALI </TT><BR>
```

```
<PRE>
                    GEETANJALI
                                        COLLEGE</PRE><BR>
<STRIKE> GEETANJALI </STRIKE><BR>
<BLINK> GEETANJALI </BLINK><BR>
1<SUP>ST</SUP> CLASS<BR>
H<SUB>2</SUB>O<BR>
</BODY>
</HTML>
```

==========

OUTPUT

FORMATTING TAG
GEETANJALI
GEETANJALI
GEETANJALI
GEETANJALI
GEETANJALI
GEETANJALI
GEETANJALI
COLLEGE
~~***GEETANJALI***~~
GEETANJALI
1ST CLASS
H_2O

10. EXPLAIN <IMG> WITH ALL ATTRIBUTES.

- Image can be added inside documents using the <IMG> tag.
- All browsers support GIF and JPEG file formats for images and either of them can be used.

ATTRIBUTES

- **SRC**
  - This attribute is used for placing the image file inside your document.
- **ALIGN**
  - When you want to fix the position of the image inside the document then this attribute is used.
  - Possible values: Left, Center, Right, Top, Middle, Bottom.
- **BORDER**
  - You can set the border of the image using Border attribute.
- **WIDTH**
  - When you want to increase or decrease the width of the image then you can use this attribute.
- **HEIGHT**
  - When you want to increase or decrease the height of the image then this attribute is used.
- **HSAPCE**
  - This attribute places the horizontal space in image file.

- **VSPACE**
  - This attribute places the vertical space in image file.
- **ALT**
  - You can specify ALT text, which is text that is displayed in place of a graphic if the graphic cannot be displayed in a Web browser.

EXAMPLE

```
<IMG SRC = "CAKE.JPG" ALIGN = "CENTER" ALT="MY CAKE" BORDER=3 HEIGHT=80
        WIDTH=50% HSAPCE=30 VSPACE=30>
```

11. EXPLAIN LISTING TAG AVAILABLE IN HTML.

- HTML offers authors several mechanisms for specifying lists of information.
- All lists must contain one or more list elements.
- Lists may contain:
  - **Unordered information.**
  - **Ordered information.**
  - **Definitions.**
- There are 2 tags used with list : list header and list item

HEADING IN A LIST

- <LH>…. </LH> - LIST HEADER
- <LH> tag for a listed item, used to display heading of the list.

LIST ITEM

- <LI>. …. </LI> - LIST ITEM
- Used to specify items of a list on a web page need to be included in the list should be enclosed within this tag.

ORDERED LIST

- <OL>….. </OL>
- In an ordered list, item have an order that is signified by a number. Hence it is also called a numbered list.
- An ordered list should start with the <OL> element, which is
- immediately followed by a <LI>(list item) element

ATTRIBUTES:

- **TYPE**
  - Controls the numbering schema.
  - Possible values
    - Alphabet capital
    - Alphabet small
    - Roman capital
    - Roman small
    - Number
- **START**
  - Alter the numbering sequence can be set to any numeric value.

```
<HTML>
<BODY>
<LH>ORDERED LIST</LH>
```

```
<OL>
        <LI>RED
        <LI>GREEN
        <LI>BLUE
</OL>
<LH>TYPES OF COMPUTER LIST</LH>
<OL TYPE="A">
        <LI>MICRO
        <LI>MINI
        <LI>MAINFRAME
</OL>
<LH>TEXT EDITOR</LH>
<OL START="501">
        <LI>NOTEPAD
        <LI>WORDPAD
<LI>WORD
</OL>
</BODY>
</HTML>
```

==========

OUTPUT:

ordered List

1. red
2. green
3. blue

types of computer List

a. micro
b. mini
c. mainframe

text editor

501. notepad
502. wordpad
503. word

UNORDERED LIST

- <UL>…</UL>
- Unordered Lists are also called unnumbered list. It is similar to the Ordered List.
- The Unordered List element is used to present a list of items, which are typically separated by white space and/or marked by bullets

ATTRIBUTE:

- **TYPE**
    - The TYPE attribute can be added to the <UL> element.
    - Possible values: disc, circle, square
    - Default value: **disc**

```
<HTML>
<BODY>
<LH>UNDERED LIST</LH>
<UL>
        <LI>GREEN
```

```
        <LI>BLUE
</UL>
<LH>TYPES OF COMPUTER LIST</LH>
<UL TYPE="CIRCLE">
        <LI>MICRO
        <LI>MINI
        <LI>MAINFRAME
</UL>
```

==========

OUTPUT:

Undered List

- red
- green
- blue

types of computer List

- micro
- mini
- mainframe

12. EXPLAIN <MARQUEE> TAG WITH ITS ATTRIBUTES.

- A marquee displays a scrolling text message.
- When we want to display text which is continuously moved on screen ; marquee will be used.
- By default marquee text will be starts from right side and scroll continuously until the screen will be off.
- **DIRECTION**
  - To specify the direction that text in a marquee moves Left or Right.
  - Possible values : Right / Left
- **BEHAVIOR**
  - This attribute indicates, how text to be moved on the screen.
  - Possible values : Alternate / Scroll / Slide
- **BGCOLOR**
  - You can specify the background color of the marquee text.
- **ALIGN**
  - To align a marquee with surrounding text.
  - Possible values: Top / Middle / Bottom
- **HEIGHT**
  - To specify the size of a marquee, whether height is given in pixels or as a percent of the browser window.
- **WIDTH**
  - To specify the total speed of a marquee, whether width is given inpixels or as a percent of the browser window.
- **LOOP**
  - To specify how many times the text effect in a marquee is repeated, torepeat the marquee a fixed number of times.

- **SCROLLAMOUNT**
  - In this attribute, type the increment number, in pixels, that the text inthe marquee should move.
- **SCROLLDELAY**
  - To specify the speed that text in a marquee moves, in the Scrolldelayattribute type the delay, in milliseconds, before the marquee text begins tomove.

EXAMPLE:
<MARQUEE> GEETANJALI COLLEGE</MARQUEE>
<MARQUEE BGCOLOR=YELLOW> GEETANJALI COLLEGE</MARQUEE>

13. EXPLAIN <TABLE> TAG WITH EXAMPLE.

- HTML allow to present data in rows and columns, or in the form of table.
- You can also create HTML tables to organize information on your web page.
- The process of creating an HTML table is similar to the process that you used to create your web page and any elements that you may have already included in your page, such as links or frames.

CREATING A BASIC TABLE

- The basic structure of an HTML table consists of the following tags:
  - Table tags: <TABLE></TABLE>
  - Row tags: <TR></TR>
  - Cell tags: <TD></TD>

<TABLE>

- This tag is used to create table.
- Table start with <TABLE>, and the ending with </TABLE>.

ALLTRIBUTES OF TABLE:

BGCOLOR	Using this attribute we can specify the background colour for table
BORDER	Using this attribute we can specify the thickness of border for table
ALIGN	This attribute used to display table in left, right or center align.
WIDTH	sets the width of your table as a percentage of the screen.
CELLPADDING	Adjusts the vertical dimension of the cells.
CELLSPACING	sets the space or border around the cells
VALIGN	Command will vertically align the data in a cell top, bottom or middle.
BORDERCOLOR	Specify border colour for table.

<TR>

- This tag is used to construct each row and each cell in the row.
- To do this, you would first start the row with the beginning row tag, <TR>

ALLTRIBUTES OF <TR>

BGCOLOR	Using this attribute we can specify the background colour for each row
ALIGN	This attribute used to display row data in left, right or center align.
VALIGN	Vertically align the data in a cell top, bottom or middle.

<TD>

- To build any cell <TD> tag is used.

ALLTRIBUTES OF <TR>

BGCOLOR	Using this attribute we can specify the background colour for each cell
ALIGN	This attribute used to display cells data in left, right or center align.
VALIGN	Vertically align the data in a cell top, bottom or middle.

```
<HTML>
<BODY>
<TABLE BORDER=1 WIDTH="50%">
<TR>
        <TH>FIRSTNAME</TH>
        <TH>LASTNAME</TH>
        <TH>PER</TH>
</TR>
<TR>
    <TD>AVNI</TD>
    <TD>SHAH</TD>
    <TD>50</TD>
</TR>
<TR>
         <TD>EKTA</TD>
         <TD>BHATT</TD>
        <TD>94</TD>
</TR>
<TR>
        <TD>MIRA</TD>
        <TD>TANK</TD>
        <TD>80</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

==========

OUTPUT

Firstname	Lastname	Per
avni	shah	50
ekta	bhatt	94
mira	tank	80

14. EXPLAIN IMAGE MAP IN HTML.

- Image map is a facility provided by HTML which allows the users to open a linked document by clicking on a particular.
- We can specify co-ordinates on image & by clicking on that area we can open a link document.
- There are two types of image map:

- Client side image map
- Server side image map

- **Server side image map:**
  - Server-side image maps enable the web browser to send positional information to the server about where the user clicks within an image.
- **client side image map:**
  - Client-side image created by using 2 main tags: <MAP>&<AREA>
  - When we create client side image map use USEMAP attribute with <img> to define that this image used as a map &its a client side map

<MAP>	The <map> tag is used to define a client-side image-map	
	Name	by using this attribute we can define name of map
<AREA>	It defines the clickable areas in the image map. It contain three attributes:	
	Shape	• Specifies the shape of the area. • Three possible values are: rect, circle, poly
	Coords	Specifies the coordinates of the area
	href	Specifies the hyperlink target for the area

EXAMPLE:

```
<HTML>
<BODY>
<IMG SRC="FILE:///C:/WINDOWS/WEB/WALLPAPER/NATURE/IMG5.JPG"
WIDTH="145" HEIGHT="126" ALT="PLANETS" USEMAP="#MAP1">

<MAP NAME="MAP1">
    <AREA SHAPE="RECT" COORDS="0,0,82,126" HREF="FORM.HTM">
    <AREA SHAPE="CIRCLE" COORDS="90,58,3"  HREF="TABLE.HTM">
</MAP>
</BODY>
</HTML>
```

15. EXPLAIN <FRAMESET> WITH ALL ATTRIBUTES.

- HTML frames allow to present documents in multiple views, which may be independent windows or sub windows.
- By using frame we can display multiple documents in a single window.
- Generally it is created in <head> section.
- The <frameset> tag is used to define a frameset.
- The <frameset> element holds one or more <frame> elements.
- Each <frame> element can hold a separate document.

ATTRIBUTES OF <FRAMESET>

ROWS	This attribute defines the number of horizontal subspaces in a frameset. Value can de define in either % or pixels
COLS	This attribute defines the number of vertical subspaces in a frameset. Value can de define in either % or pixels

ATTRIBUTES OF <FRAME>

SRC	o This attribute defines the path of file that we want to display in frame.
MARGINHEIGHT	o This attribute defines the number of vertical subspaces in a frameset. o Value can de define in either % or pixels
MARGINWIDTH	o This attribute defines the number of horizontal space in a frameset. o Value can de define in either % or pixels
FRAME BORDER	o This attribute defines the border will be displayed in frame or not o it contain two values : **0 or 1** o default : **1**
BORDERCOLOR	o This attribute defines colour for frame. o to use bordercolor attribute the frame border must be set to **1** o you can use colour name, hexadecimal colour code or RGB()
SCROLLING	o This attribute specify that whether the scrollbar appeared on frame or not o this property contain 3 values: **YES, NO** and **AUTO** o default value: **AUTO**
NORESIZE	o this attribute prevents frame window to be resized.

```
<HTML>
<FRAMESET  COLS="50%,*">
    <FRAME SRC="">
    <FRAME SRC="">
</FRAMESET>
</HTML>
```

16. EXPLAIN <FORM> IN HTML.

<FORM>

- The <form> tag is used to create an HTML form for user input.
- A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more.
- Forms are used to pass data to a server.
- generally form can be created by using five tags:
    - **<FORM>**
    - **<INPUT>**

- **<TEXTAREA>**
- **<OPTION>**
- **<SELECT>**

ATTRIBUTE	DESCRIPTION
ACTION	○ Specifies where to send the form-data when a form is submitted ○ specify URL of file that you want to display when user send form
METHOD	○ Specifies itty using which method form will be submitted to server ○ two methods are used: **GET OR POST** ○ default value : **GET**

<INPUT>

- The <input> tag is used to select user information.
- An input field can vary in many ways, depending on the type attribute.
- An input field can be a text field, a checkbox, a password field, a radio button, a button, and more.

ATTRIBUTE	DESCRIPTION
TYPE	Specifies the type of an input element We can use following types with type attributes:

TEXT	Used to create a single line box
PASSWORD	Use to create password box . the value that we enter will be displayed in the form of symbol / bullet
HIDDEN	Used to create hidden field
RADIO	Use when we want to select a single option from multiple options
CHECKBOX	Use when we want to select a multiple option from multiple options
SUBMIT	Used to submit form
RESET	Used to display form with its default value. Generally used to clear data
FILE	Use to select file for upload. Provide one text box & browse button. The value select displayed in textbox

VALUE	Specifies the value of an input element used with text, radio, <OPTION> , hidden, password, checkbox
SIZE	Specifies the width of an input field
NAME	Specifies a name for an input element
DISABLED	Specifies that an input element should be disabled when the page loads
MAXLENGTH	Specifies the maximum length (in characters) of an input field (for type="text" or type="password")
CHECKED	Specifies that an input element should be preselected when the page loads (for type="checkbox" or type="radio")

<SELECT>

- The <select> tag is used to create a drop-down list or combo box
- The <option> tags inside the select element define the available options in the list.

ATTRIBUTE	DESCRIPTION
DISABLED	○ Specifies that a drop-down list should be disabled
MULTIPLE	○ Specifies that multiple options can be selected
NAME	○ Specifies the name of a drop-down list
SIZE	○ Specifies the number of visible options in a drop-down list

<OPTION>

- The <option> tag defines an option in a select list.
- The option element goes inside the select element.

ATTRIBUTE	DESCRIPTION
LABEL	○ Specifies a shorter label for an option
SELECTED	○ Specifies that an option should be selected by default
VALUE	○ Specifies the value to be sent to a server when a form is submitted

<TEXTAREA>

- The <textarea> tag defines a multi-line text input control.
- A text area can hold an unlimited number of characters, and the text renders in a fixed-width font
- The size of a <textarea> can be specified by the cols and rows attributes

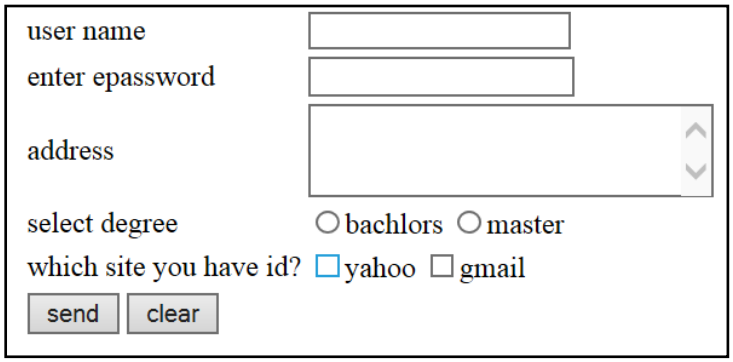
ATTRIBUTE	DESCRIPTION
COLS	○ Specifies the visible width of a text-area
ROWS	○ Specifies the visible number of rows in a text-area
NAME	○ Specifies name of textarea box

EXAMPLE:

```
<HTML>
<BODY>
<FORM  METHOD="GET">
<TABLE>
<TR>
        <TD>USER NAME</TD>
        <TD><INPUT TYPE="TEXT" NAME="T1">
</TR>
```

```
<TR>
        <TD>ENTER EPASSWORD</TD>
        <TD><INPUT TYPE="PASSWORD" NAME="T1">
</TR>
<TR>
        <TD>ADDRESS</TD>
        <TD><TEXTAREA NAME="ADD" ROWS=3 COLS=25></TEXTAREA></TD>
</TR>
<TR>
    <TD>SELECT DEGREE</TD>
    <TD><INPUT TYPE=RADIO NAME="G1"  VALUE="BACHLORS">BACHLORS
        <INPUT TYPE=RADIO NAME="G1"  VALUE="MASTER">MASTER
</TR>
<TR>
        <TD>WHICH SITE YOU HAVE ID?</TD>
        <TD><INPUT TYPE=CHECKBOX NAME="R1"  VALUE="YAHOO">YAHOO
        <INPUT TYPE=CHECKBOX NAME="RR1"  VALUE="GMAIL">GMAIL
</TR>
<TR>
        <TD><INPUT TYPE="SUBMIT" VALUE="SEND">
        <INPUT TYPE="RESET" VALUE="CLEAR"></TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

==========OUTPUT==========



17. EXPLAIN HYPER LINK.

- The HTML Anchor Element (<a>) defines a hyperlink to a location on the same page or any other page on the Web.
- It can also be used (in an obsolete way) to create an anchor point—a destination for hyperlinks within the content of a page
- **ATTRIBUTES OF <A>**

HREF	This attribute Specifies the URL of the page the link goes to.
TARGET	This attribute defines that Specifies where to open the linked document
NAME	Specifies the name of an anchor

EXAMPLE

```
<HTML>
<BODY>
    <A HREF=”A1.HTML”> READ MY FILE</A>
</BODY>
</HTML>
```

18. WHAT IS THE DIFFRENCE BETWEEN HTML & HTML 5.

HTML	HTML5
Allows the JavaScript to run only in the browser interface but not originally in the browser background.	The JavaScript can actually run in the background of the browser itself through JS web worker API.
Does not allow audio <audio> and video <video> tags.	Does allow audio and video controls and the tags.
One cannot draw various shapes (circles, triangles, rectangles, and others) in older HTML versions.	HTML5 allows one to draw various shapes (circles, triangles, rectangles, and others).
Older HTML versions are less mobile-friendly.	HTML5 language is more mobile-friendly.
Supports vector graphics only with the help of programs like Flash, Silver-light, and VML.	HTML 5 allows the use of virtual vector graphics without the help of programs like Flash, Silver-light, and VML.
Doctype declaration is too long and complicated in HTML.	Doctype declaration is quite simple and easy in HTML 5.
Character encoding is long and complicated in HTML.	Character encoding is pretty simple and easy in HTML 5.
It uses cookies to store temporary data.	It uses SQL databases and application cache to store offline data.
Old elements still exist in older HTML.	strike, frame, frameset, font, center, tt, big, dir, acronym, basefont, noframes, applet, isindex and various other deprecated elements are dropped completely in HTML 5.
It does not allow drag and drop effects.	HTML5 allows drag and drop effects.

19. EXPLAIN DOCUMENT STRUCTURE OF HTML 5

- The following tags have been introduced for better structure −

SECTION	○ This tag represents a generic document or application section. ○ It can be used together with h1-h6 to indicate the document structure.
ARTICLE	○ This tag represents an independent piece of content of a document, such as a blog entry or newspaper article.
ASIDE	○ This tag represents a piece of content that is only slightly related to the rest of the page.
HEADER	○ This tag represents the header of a section.
FOOTER	○ This tag represents a footer for a section and can contain information about the author, copyright information, etc
NAV	○ This tag represents a section of the document intended for navigation.
DIALOG	○ This tag can be used to mark up a conversation.
FIGURE	○ This tag can be used to associate a caption together with some embedded content, such as a graphic or video.

SYNTAX:

```
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">
        <title>...</title>
</head>
<body>
                <header>...</header>
                <nav>...</nav>
        <article>
<section>
                ...
</section>
</article>
<aside>...</aside>
<figure>...</figure>
<footer>...</footer>
</body>
</html>
```

20. EXPLAIN STANDARD ATTRIBUTES OF HTML 5

ATTRIBUTE	OPTIONS	FUNCTION
ALIGN	right, left, center	○ Horizontally aligns tags
BACKGROUND	URL	○ Places an background image behind an element

BGCOLOR	numeric, hexidecimal, RGB values	○ Places a background color behind an element
CLASS	User Defined	○ Classifies an element for use with Cascading Style Sheets.
CONTENTEDITABLE	true, false	○ Specifies if the user can edit the element's content or not.
CONTEXTMENU	Menu id	○ Specifies the context menu for an element.
DRAGGABLE	true,false, auto	○ Specifies whether or not a user is allowed to drag an element.
HEIGHT	Numeric Value	○ Specifies the height of tables, images, or table cells.
HIDDEN	hidden	○ Specifies whether element should be visible or not.
ID	User Defined	○ Names an element for use with Cascading Style Sheets.
SPELLCHECK	true, false	○ Specifies if the element must have it's spelling or grammar checked.
STYLE	CSS Style sheet	○ Specifies an inline style for an element.
TABINDEX	Tab number	○ Specifies the tab order of an element.
TITLE	User Defined	○ "Pop-up" title for your elements.
VALIGN	top, middle, bottom	○ Vertically aligns tags within an HTML element.
WIDTH	Numeric Value	○ Specifies the width of tables, images, or table cells.

21. EXPLAIN <CANVAS> ELEMENT OF HTML 5

- HTML5 element <canvas> gives you an easy and powerful way to draw graphics using JavaScript.
- It can be used to draw graphs, make photo compositions or do simple animations.
- Here is a simple <canvas> element which has only two specific attributes
  - **width**
  - **height**
  - **id**
  - **name**

- **class**

- You can easily find that <canvas> element in the DOM using getElementById()method

EXAMPLE

```
<!DOCTYPE HTML>
<html>
<head>
<style>
        #mycanvas{border:1px solid red;}
</style>
</head>
<body>
    <canvas id="mycanvas" width="100" height="100"></canvas>
</body>
</html>
```

==========OUTPUT==========

22. EXPLAIN <INPUT> ELEMENT OF HTML 5

- HTML5 input elements introduced several new values for the type attribute.
- These are listed below.

TYPE	DESCRIPTION
DATETIME	○ A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601 with the time zone set to UTC.
DATETIME-LOCAL	○ A date and time (year, month, day, hour, minute, second, fractions of a second) encoded according to ISO 8601, with no time zone information.
DATE	○ A date (year, month, day) encoded according to ISO 8601.
MONTH	○ A date consisting of a year and a month encoded according to ISO 8601.
WEEK	○ A date consisting of a year and a week number encoded according to ISO 8601.
TIME	○ A time (hour, minute, seconds, fractional seconds) encoded according to ISO 8601.
NUMBER	○ This accepts only numerical value. The step attribute specifies the precision, defaulting to 1.
RANGE	○ The range type is used for input fields that should contain a value from a range of numbers.

EMAIL	○ This accepts only email value. This type is used for input fields that should contain an e-mail address. If you try to submit a simple text, it forces to enter only email address in email@example.com format.
URL	○ This accepts only URL value. This type is used for input fields that should contain a URL address. If you try to submit a simple text, it forces to enter only URL address either in http://www.example.com format or in http://example.com format.

23. HOW TO EMBED VIDEO USING HTML 5

- The HTML5 <audio> and <video> tags make it simple to add media to a website.
- You need to set src attribute to identify the media source and include a controls attribute so the user can play and pause the media.
- The current HTML5 draft specification does not specify which video formats browsers should support in the video tag.
- But most commonly used video formats are −
  - **Ogg** − Ogg files with Thedora video codec and Vorbis audio codec.
  - **mpeg4 –** MPEG4 files with H.264 video codec and AAC audio codec.

ATTRIBUTES:

ATTRIBUTE	DESCRIPTION
AUTOPLAY	○ This boolean attribute if specified, the video will automatically begin to play back as soon as it can do so without stopping to finish loading the data.
AUTOBUFFER	○ This boolean attribute if specified, the video will automatically begin buffering even if it's not set to automatically play.
CONTROLS	○ If this attribute is present, it will allow the user to control video playback, including volume, seeking, and pause/resume playback.
HEIGHT	○ This attribute specifies the height of the video's display area, in CSS pixels.
LOOP	○ This boolean attribute if specified, will allow video automatically seek back to the start after reaching at the end.
PRELOAD	○ This attribute specifies that the video will be loaded at page load, and ready to run. Ignored if autoplay is present.
POSTER	○ This is a URL of an image to show until the user plays or seeks.
SRC	○ The URL of the video to embed. This is optional; you may instead use the <source> element within the video block to specify the video to embed
WIDTH	○ This attribute specifies the width of the video's display area, in CSS pixels.

EXAMPLE

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

<video  width="300" height="200" controls autoplay>
<source src="/html5/foo.ogg" type="video/ogg" />
<source src="/html5/foo.mp4" type="video/mp4" />
      Your browser does not support the video element.
</video>

</body>
</html>
```

24. HOW TO EMBED VIDEO USING HTML 5

- HTML5 supports <audio> tag which is used to embed sound content in an HTML or XHTML document as follows.
- The current HTML5 draft specification does not specify which audio formats browsers should support in the audio tag.
- But most commonly used audio formats are
  - **ogg**
  - **mp3**
  - **wav**

ATTRIBUTES:

ATTRIBUTE	DESCRIPTION
AUTOPLAY	○ This boolean attribute if specified, the audio will automatically begin to play back as soon as it can do so without stopping to finish loading the data.
AUTOBUFFER	○ This boolean attribute if specified, the audio will automatically begin buffering even if it's not set to automatically play.
CONTROLS	○ If this attribute is present, it will allow the user to control audio playback, including volume, seeking, and pause/resume playback.
LOOP	○ This boolean attribute if specified, will allow audio automatically seek back to the start after reaching at the end.
PRELOAD	○ This attribute specifies that the audio will be loaded at page load, and ready to run. Ignored if autoplay is present.
SRC	○ The URL of the audio to embed. This is optional; you may instead use the <source> element within the video block to specify the video to embed

EXAMPLE

```
<!DOCTYPE HTML>
<html>
<body>
        <audio controls autoplay>
        <source src="/html5/audio.ogg" type="audio/ogg" />
        <source src="/html5/audio.wav" type="audio/wav" />
                Your browser does not support the audio element.
        </audio>
</body>
</html>
```

25. EXPLAIN LINE IN CANVAS.

- To draw a line using HTML5 Canvas we have to use following methods
- We can use following methods with lineTo().

beginPath()	This method used to declare that we are about to draw a new path
lineTo()	This method used to position the context point (i.e. drawing cursor)
moveTo()	This method used to draw a straight line from the starting position to a new position.
stroke()	This method used to make the line visible,
lineWidth()	This method used to specify the width of line
strokeStyle()	This method used to specify color for your line

Example:

```
<html>
<body>
<canvas id="mycanvas" height="200" width="500">
</canvas>
<script>
        var c1=document.getElementById("mycanvas");
        var c2=c1.getContext("2d");
        c2.beginPath();
        c2.moveTo(0,0);
        c2.lineTo(100,150);
        c2.stroke();
</script>
</body>
</html>
```

26. EXPLAIN CIRCLE IN CANVAS.

- To draw a circle using HTML5 Canvas we have to use following methods.
- To create a circle with arc():

Syntax: arc(x , y, r, startangle,endangle, *counter clockwise)*

Parameter	Description
X	The x-coordinate of the center of the circle
Y	The y-coordinate of the center of the circle
R	The radius of the circle
startangle	The starting angle, in radians (0 is at the 3 o'clock position of the arc's circle)
endangle	The ending angle, in radians
counterclockwise	• Optional. • Specifies whether the drawing should be counterclockwise or clockwise. • False is default.

- Set start angle to 0 and end angle to 2*Math.PI.
- We can use following methods with arc()

beginPath()	This method used to declare that we are about to draw a new path
fillStyle()	This method used to fill color for your circle
fill()	This method Fill color specified by fillStyle()
stroke()	This method used to make the line visible,
lineWidth()	This method used to specify the width of line
strokeStyle()	This method used to specify color for your line

Example:

```
<html>
<body>
<canvas id="mycanvas" height="500" width="1000">
</canvas>
<script>
```

```
        var c1=document.getElementById("mycanvas");
        var c2=c1.getContext("2d");
        c2.beginPath();
        c2.arc(100,50,20,0,1.5* Math.PI);
        c2.stroke();
</script> </body> </html>
```

27. EXPLAIN RECT() IN CANVAS.

- To draw a circle using HTML5 Canvas we have to rect() will be used.

Syntax: rect (x, y, width, height)

Parameter	description
X	The x-coordinate of the upper-left corner of the rectangle
Y	The y-coordinate of the upper-left corner of the rectangle
width	The width of the rectangle, in pixels
height	The height of the rectangle, in pixels

SOME OTHER METHODS

fillStyle()	This method used to fill color for rectangle
fill()	This method Fill color specified by fillStyle()
stroke()	This method used to make the rectangle visible
lineWidth()	This method used to specify the width of line
strokeStyle()	This method used to specify color for your line

Example:

```
<html>
<body>
<canvas id="c" height=400 width=800></canvas>
<script>
        var a=document.getElementById("c");
        var b=a.getContext("2d");
        b.rect(0,0,200,100);
        b.stroke();
</script>
</body>
```

```
</html>
```

28. EXPLAIN LINEAR GRADIENT IN CANVAS.

- To draw a circle using HTML5 Canvas we have to rect() will be used.

UNIT: 4 CSS& CSS 3

1. WHAT IS CSS? EXPLAIN ITS TYPES

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- It is a powerful process for adding styles in the Web pages.
- CSS is a style language that defines layout of HTML documents.
- For example, CSS covers fonts, colors, margins, lines, height, width, background images, advanced positions and many other things
- HTML is used to structure content while CSS is used for formatting structured content.
- A CSS provides attributes that create dynamic effects.
- The advantage of a CSS includes the ability to make changes to all documents from a single location
- Using CSS, we can give the special effects and controls over the appearance, layout and behavior of the web page.

SYNTAX:

```
<STYLE type= "text/css">
    Tag {attribute: value; attribute: value; attribute: value; ……}
</STYLE>
```

- The Style tag is represented by the <STYLE> and </STYLE> tags.
- <STYLE> written between the <HEAD> and </HEAD> tags.
- Between the <STYLE> and </STYLE>, HTML tags and specific style attributes are listed.
- The CSS syntax is made up of three parts: a selector, a property and a value.
- The CSS syntax is made up of three parts: a selector, a property and a value.
- The selector is normally the HTML tag and the property is the attribute and each property can take a value and the property and value are separated by a colon and enclosed by curly braces.
- If the value is multiple words then put quotes around the value.
- If you wish to specify more than one property then you must separate each property with a semi-colon.

THREE TYPES OF CSS

- CSS comes in three types:
    - At the top of a web page document (internal)
    - Right next to the text it decorates (inline)
    - In a separate file (external)

INTERNAL STYLE SHEET:

- Internal styles are placed in <HEAD> section of a particular web page using <style> tag.
- These styles can be used only for the web page in which they are embedded.

- Therefore you need to create over and over again for each web page you wish to style.

EXTERNAL STYLE SHEET:

- External style sheets are separate files full of CSS instructions (with the file extension .css).
- With an external style sheet, you can change the look of an entire Web site by changing one file.
- Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

INLINE STYLE SHEET:

- The Inline style is specific to the tag itself.
- The inline style uses the HTML "*style*" attribute to style a specific tag.
- This is not recommended, as every CSS change has to be made in every tag that has the inline style applied to it.
- The Inline style is good for one an individual CSS change that you do not use repeatedly through the site.

2. EXPLAIN FONT PROPERTY OF CSS

- The Font properties allow you to change the appearance of the font like style, size, boldness and look of font.
- Main four attributes of font property are:
  - **FONT-FAMILY**
  - **FONT-STYLE**
  - **FONT-WEIGHT**
  - **FONT-SIZE**

FONT-FAMILY

- It sets the font family names like busterd, arial, verdana etc. for a text.

SYNTAX
font-family: Verdana, sans-serif;

FONT-STYLE

- You can set the style of text in a element with the font-style property
- Possible values are:
  - Normal
  - italic

SYNTAX
font-style: value;

FONT-WEIGHT

- You can control the weight of text in an element with the font-weight property:
- Possible values are:
  - lighter
  - normal
  - 100
  - 200
  - 600
  - 700
  - 800
  - 900

- 300
- 400
- 500
- bold
- bolder

SYNTAX
font-weight: value;

FONT-SIZE

- You can set the size of the text used in an element by using the font-size property.
- POSSIBLE VALUES ARE:
  - xx-large
  - x-large
  - larger
  - large
  - medium
  - small
  - smaller
  - x-small
  - xx-small
  - length
  - % (percent)

SYNTAX
font-size: value;

3. EXPLAIN TEXT PROPERTY.

- You can to control the appearance of text.
- It is possible to increase or decrease space between characters in text, align, indent color etc.

Color

- You can set the color of text with the following:

SYNTAX:
color: value;

- Possible values are:
  - color name – example:(red, black…)
  - hexadecimal number – example:(#ff0000, #000000)
  - RGB color code – example:(rgb(255, 0, 0), rgb(0, 0, 0))

Letter Spacing

- You can adjust the space between letters in the following manner.
- Setting the value to 0, prevents the text from justifying.
- You can use negative values.

SYNTAX:
letter-spacing: value;

- Possible values are
  - normal
  - length

Text Align

- You can align text with the following:

SYNTAX:
text-align: value;

- Possible values are
    - left
    - right
    - center
    - justify

Text Decoration

- You can decorate text with the following:

SYNTAX:
text-decoration: value;

- Possible values are
    - none
    - underline
    - overline
    - line through
    - blink

Text Indent

- You can indent the first line of text in an (X)HTML element with the following:

SYNTAX:
text-indent: value;

- Possible values are
    - length
    - percentage

Text Transform

- You can control the size of letters in an (X)HTML element with the following:

SYNTAX:
text-transform: value;

- Possible values are
    - none
    - capitalize
    - lowercase
    - uppercase

Word Spacing

- You can adjust the space between words in the following manner. You can use negative values.

SYNTAX:
word-spacing: value;

- Possible values are
    - normal

- length

4. EXPLAIN: BACKGROUND PROPERTY

Background

- You can style the background of an element in one declaration with the background property.

SYNTAX:
background: #ffffffurl(path_to_image) top left no-repeat fixed;

- Values:
  - attachment
  - color
  - image
  - position
  - repeat

Background Color

- You can specifically declare a color for the background of an element using the background-color property.

SYNTAX:
background-color: value;

- Values:
  - color name
  - hexadecimal number
  - RGB color code
  - transparent

Background Image

- You can set an image for the background of an element using the background-image property.

SYNTAX:
background-image: url(path_to_image);

- Values:
  - url
  - none

Background Position

- You can position an image used for the background of an element using the background-position property.

SYNTAX:
background-position: value;

-

  Values:
  - top left
  - top center
  - top right
  - center left
  - center center
  - center right
  - bottom left
  - bottom center
  - bottom right

Background Repeat

- You can set if an image set as a background of an element is to repeat (across=x and/or down=y) the screen using the background-repeat property.

SYNTAX:
background-repeat: value;

- Values:
  - no-repeat
  - repeat
  - repeat-x
  - repeat-y

5. EXPLAIN: LIST PROPERTY

- You can set and change list appearance

List Style

- You can control the appearance of ordered and unordered lists in one declaration with the list-style property

SYNTAX:
list-style: value value;

- Values:
  - image
  - position
  - type

List Style Image

- You can use an image for the bullet of unordered lists with the list-style property
- If you use animage, it is a good idea to declare the list-style-type also in case the user has images turned off.

SYNTAX:
list-style-image: url(path_to_imagEXAMPLEif, jpg or png);

List Style Position

- You can control the position of ordered and unordered lists with the list-style-position property

SYNTAX:

list-style-position: value;

- Values
  - inside
  - outside

List Style Type

- You can control the type of bullet ordered and unordered lists use with the list-style-type property

SYNTAX:
list-style-type: value;

- Values
  - disc
  - circle
  - square
  - decimal
  - lower-roman
  - upper-roman
  - lower-alpha
  - upper-alpha
  - none

6. EXPLAIN: MARGIN PROPERTY

You can set margin property to define the space around the text.

SYNTAX:
margin-top/ margin-left/ margin-right/ margin-bottom: length percentage or auto;

- **Values:**
  - Margin-top:
    - Set the top margin either in percentage or length (px)
  - Margin-bottom
    - Set the bottom margin either in percentage or length (px)
  - Margin-right
    - Set the right margin either in percentage or length (px)
  - Margin-left
    - Set left margin either in percentage or length (px)

7. EXPLAIN: PADDING

- The CSS padding properties allow you gives the space between border and text

ATTRIBUTES	VALUES
PADDING - TOP	○ It sets the top space either in percentage (%) or in length (px) ○ web browser sets a top space by using percentage (%) or

	numerical value (px) ○ negative values are not allowed in this type of property.
PADDING - BOTTOM	○ It sets the bottom space either in percentage (%) or in length (px) ○ web browser sets a bottom space by using percentage (%) or numerical value (px) ○ negative values are not allowed in this type of property
PADDING - LEFT	○ It sets the left space either in percentage (%) or in length (px) ○ web browser sets a left space by using percentage (%) or numerical value (px) ○ negative values are not allowed in this type of property.
PADDING - RIGHT	○ It sets the right space either in percentage (%) or in length (px) ○ web browser sets a right margin by using percentage (%) or numerical value (px) ○ Negative values are not allowed in this type of property.

8. EXPLAIN BORDER PROPERTY OF CSS

- The CSS border properties allow you to the specify the style and color of an element's border so by using CSS border properties, we can create borders with nice effects..
- There are 11 types of attributes in border properties like
  - Border – style
  - Border – color
  - Border – width
  - Border – top – width
  - Border – bottom – width
  - Border – left – width
  - Border – right – width
  - Border – top
  - Border – bottom
  - Border – left
  - Border – right

ATTRIBUTES	VALUES
BORDER-STYLE	○ It sets the style of border like solid, double, groove, ridge, inset, outset, dotted, dashed etc so the web browser displays the border around the text in different type of style like solid, double, grooved, ridged, inset, outset, dotted, dashed .
BORDER-COLOR	○ It sets the color of border. ○ We can use either in a color name or in a color code

BORDER-WIDTH	○ It sets the width of border ○ we can set width like thin, medium, thick, etc so the web browser displays the width of border around the text
BORDER-TOP-WIDTH	○ It sets the width of top border like thin, medium, thick, etc. ○ the web browser displays the width of top border around the text
BORDER-BOTTOM-WIDTH	○ It sets the width of bottom border like thin, medium, thick, etc ○ the web browser displays the width of bottom border around the text
BORDER-LEFT-WIDTH	○ It sets the width of left border like thin, medium, thick, etc ○ the web browser displays the width of left border around the text
BORDER-RIGHT-WIDTH	○ It sets the width of right border like thin, medium, thick ○ the web browser displays the width of right border around the text
BORDER-TOP	○ It sets the width, color and style of top border ○ the web browser displays the top border around the text
BORDER-BOTTOM	○ It sets the width, color and style of bottom border ○ The web browser displays the bottom border around the text
BORDER-LEFT	○ It sets the width, color and style of left border so the web browser displays the left border around the text
BORDER-RIGHT	○ It sets the width, color and style of right border ○ the web browser displays the right border around the text

CSS3

1. WHAT IS CSS3?

- Cascading Style Sheets is a style sheet language used for describing the look and formatting of a document written in a mark up language.
- CSS3 is a latest standard of css earlier versions(CSS2).
- The CSS3 include following new things which is not available in css2
  - Media Queries
  - Namespaces
  - Selectors Level 3
  - Color
- Css3 add some more collaboration which is known as its modules

2. EXPLAIN BACKGROUND PROPERTY OF CSS3

- In css3 we can use multiple backgrounds which is not possible in older version of css.
- Css3 introduced Multi Background property

Multibackground :

- This property used to add one or more images at a time without HTML code, We can add images as per our requirement.
- It supports following values:
- Background-image
  - Used to setting all the background image properties in one section
  - You have to specify url for background-image.
  - url must be separated by comma.

SYNTAX:
background-image: url(path_to_image), url(path_to_image);

Background Position

- You can position an image used for the background of an element using the background-position property.

SYNTAX:
background-position: value;

- Values:
  - top left
  - top center
  - top right
  - center left
  - center center
  - center right
  - bottom left
  - bottom center
  - bottom right

Background Repeat

- You can set if an image set as a background of an element is to repeat (across=x and/or down=y) the screen using the background-repeat property.

SYNTAX:
background-repeat: value;

- Values:
  - no-repeat
  - repeat
  - repeat-x
  - repeat-y

Background –Size

- By using this property you can set up size for background-image

SYNTAX:
background-size: value;

3. EXPLAIN GRADIENT PROPERTY OF CSS3

- Gradients display the combination of two or more colours.
- CSS3 supports two types of Gradient

- Linear Gradient
- Radial Gradient

- To use gradient property use following prefix in browser:

PREFIX	BROWSER
-webkit-	for Webkit browsers (chrome, Safari)
-moz-	for FireFox
-ms-	for Internet Explorer
-o-	for Opera

Linear gradients

- Linear gradients are used to arrange two or more colors in linear formats like
  - top to bottom
  - left to right
  - diagonal

SYNTAX:
Background: linear-gradient(direction, color1, color2…);

- you can use colour name, hexadecimal colour code as well as rgb() as color value

EXAMPLE:
#bck{ background:-webkit- linear-gradient(left,red,magenta,yellow);

- **diagonal:**
  - To create diagonal background effect we have to specify direction.
  - The value of direction can be to bottom, to top, to right, to left, to bottom right.

```
EXAMPLE:
#bck1
{
   height: 200px;
   background-color: red;
   background-image: linear-gradient(to bottom right, red, yellow);
}
```

Radial gradients

- Radial gradients allow you to display two or more colours that appear at center.

SYNTAX: Background: radial-gradient(color1, color2…);

- you can use colour name, hexadecimal colour code as well as rgb() as color value
- by using repeating keyword we can repeat colour effect in radial property

EXAMPLE:
#bck1{ background:-webkit- radial-gradient(magenta 40%,yellow 30%);

4. EXPLAIN DROP SHADOW PROPERTY OF CSS3.

- This property Used to add shadow effects to elements.
- This property attaches one or more shadows to an element.

SYNTAX:

box-shadow: value;none |h-offset v-offset blur spread color |inset |initial |inherit;

- We can use following values:
- none (default)
- **h-offset**
  - Required.
  - The horizontal offset of the shadow.
  - A positive value puts the shadow on the right side of the box.
  - A negative value puts the shadow on the left side of the box
- **v-offset**
  - Required.
  - The vertical offset of the shadow.
  - A positive value puts the shadow below the box
  - A negative value puts the shadow above the box
- **Blur**
  - Optional.
  - The blur radius.
- **spread**
  - Optional.
  - The spread radius.
  - A positive value increases the size of the shadow, a negative value decreases the size of the shadow
- **Color**
  - Optional.
  - The colour of the shadow.
- **Initial**
  - The initial keyword is used to set a CSS property to its default value.

EXAMPLE:

```
#a1
{
Border:1 px solid red;box-shadow: 10px 20px 20px pink;
}
#a2
{
border:5px solid red;
background-color:yellow;
box-shadow:25px 25px 50px 20px purple inset;
height:100;width:200;
}
```

5. EXPLAIN 2D TRANSFORMS PROPERTY OF CSS3.

- 2D transforms are used to re-change the element structure as translate, rotate, scale, and skew.
- It provides following methods –
  - translate()
  - rotate()
  - scale()

- skewX()
- skewY()

translate()

- This method used to move object form current position.
- It uses two parameters: x-axis, y-axis.

EXAMPLE:
#a1{Border:1 px solid red; transform: translate(10px,20px);

rotate()

- The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.
- Specify degree value.

EXAMPLE:
.a1{Border:1 px solid red; height:100px;width:100px; transform: rotate(45deg);

scale()

- The scale () method increases or decreases the size of an element .
- Specify two value of scale
- First parameter specifies width and second for height.
- It will multiply height & width with the given number.

EXAMPLE:
.a2{border:1px solid red; height:150px;width:150px; transform:scale(2,1);}

skewX()

- This method skews an element along the X-axis by the given angle.
- Specify degree as a value.

EXAMPLE:
.a2{border:1px solid red; height:150px;width:150px; ransform:skewX(25deg);}skewY()

skewY()

- This method skews an element along the Y-axis by the given angle.
- Specify degree as a value.

EXAMPLE:
.a2{border:1px solid red; height:150px;width:150px; transform:skewY(25deg);}

skew ()

- This method skews an element along the X-axis & Y-axis by the given angle.
- Specify degree as a value.

EXAMPLE:
.a2{border:1px solid red; height:150px;width:150px; transform:skew (25deg,15deg);}

6. EXPLAIN 3D TRANSFORM PROPERTY OF CSS3

- CSS allows you to format your elements using 3D transformations.
- We can use following methods for 3D transformation
  - rotateX()
  - rotateY()

- rotateZ()

rotateX():

- The rotateX() method rotates an element around its X-axis at a given degree.
- Specify degree as a value

EXAMPLE:
.a1{transform:rotateY(45deg);}

rotateY():

- The rotateX() method rotates an element around its Y-axis at a given degree
- Specify degree as a value

EXAMPLE:
.a2{transform:rotateY(45deg);}

rotateZ():

- The rotateX() method rotates an element around its Y-axis at a given degree
- Specify degree as a value

EXAMPLE:
.a3{transform:rotateZ(145deg);}

7. EXPLAIN POSITION PROPERTY OF CSS3.

- The position property specifies the type of positioning method used for an element
  - Static
  - Absolute
  - Relative
  - Fixed

static	Default value. Elements render in order, as they appear in the document flow
absolute	The element is positioned relative to its first positioned (not static) ancestor element
fixed	The element is positioned relative to the browser window
relative	The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position

8. EXPLAIN TRANSITION PROPERY OF CSS3.

- CSS transitions allows you to change property values smoothly (from one value to another), over a given duration.
- To create a transition effect, you must specify two things:
  - the CSS property you want to add an effect to
  - the duration of the effect

CSS Transition Properties

- **transition-delay**

- Specifies a delay for the transition effect
- Specify value in second

- **transition-duration**
  - Specifies how many seconds or milliseconds a transition effect takes to complete
- **transition-timing-function**
  - Specifies the speed curve of the transition effect
  - We can set following values:

SYNTAX:
transition-timing-function: linear|ease|ease-in|ease-out|ease-in-out

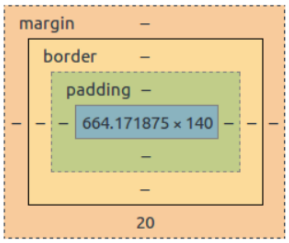
Value	Description
ease	Default value. Specifies a transition effect with a slow start, then fast, then end slowly
linear	Specifies a transition effect with the same speed from start to end
ease-in	Specifies a transition effect with a slow start
ease-out	Specifies a transition effect with a slow end
ease-in-out	Specifies a transition effect with a slow start and end

EXAMPLE:

```
p
 {
   width: 100px;
   height: 100px;
   background: red;
   transition-delay: 2s;
   transition-duration:5s;
   transition-timing-function:ease-in;
}
p:hover
{
width:300px;
background:blue;
}
```

9. EXPLAIN BOX SIZING PROPERTY OF CSS3

- This property is basically used to change height & width of element.
- Border property calculate like this:



Box Model

Width = width + padding-left + padding-right + border-left + border-right
Height = height + padding-top + padding-bottom + border-top + border-bottom

- To use box sizing we have to use four properties:
  - Height
  - Width
  - Padding
  - Box-sizing:
    - Calculate height , width ,padding & border but ***not margin***

EXAMPLE:

```
#a1
{
Border:1 px solid red;
height:100px;
width:100px;
padding:20px;
box-sizing: border-box;
}
```

10. EXPLAIN MEDIA QUERY.

- Media queries are a feature of CSS that enable webpage content to adapt to different screen sizes and resolutions.
- They are a fundamental part of responsive web design and are used to customize the appearance of websites for multiple devices.
- Media queries may be inserted within a webpage's HTML or included in a separate .CSS file referenced by the webpage.
- Media queries can be used to check many things, such as:
  - width and height of the viewport
  - width and height of the device
  - orientation
  - resolution

SYNTAX:

```
@media not|only mediatype and (expressions)
 {
   CSS-Code;
}
```

EXAMPLE:

```
<style>
body {
   background-color: pink;
}

@media screen and (min-width: 480px) {
   body {
      background-color: lightgreen;
   }
}
```

UNIT: 5 JAVA SCRIPT

1. WHAT IS JAVA SCRIPT? EXPLAIN ITS ADVANTAGES.

- JavaScript is a scripting language which is used to create web site on internet.
- It's a lightweight programming language.
- It is interpreted by the browser engine when the web page is loaded.
- JavaScript was created by "**Brendan Eich** "at Netscape.
- It was first introduced in December 1995 under the name of Live Script.
- Java script is best used in Netscape Navigator web browser.
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license
- It's a case sensitive language.
- It provides an easy development process for program.
- We can write java script program in notepad & also add in html file.
- We can run java script program on any browsers like internet explorer, Netscape navigator etc.
- JavaScript is a cross-platform, object-oriented scripting language.
- JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements.
- <SCRIPT> tag is used to write program of java script.
- We can create java script program in <HEAD> part or <BODY> part using <SCRIPT> tag.
- In java script each line end with (;)

<SCRIPT> TAG

- The script tag has two purposes:
  - It identifies a block of script in the page.
  - It loads a script file.
- The <script> tag is used to define a client-side script, such as a JavaScript.
- The <script> element either contains scripting statements.
- We can also add an external script file through the src attribute.
- It contain main three attributes:
  - **language="JavaScript"**
    - This attribute has been deprecated.
    - It was used to select other programming languages and specific versions of JavaScript.
  - **type="text/JavaScript"**
    - This attribute is optional because default programming language in all browsers has been JavaScript.
  - **src="path of external JavaScript file"**
    - allows you to add external java script file

SYNTAX:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE=”JAVASCRIPT” >
            SCRIPTING CODE
</SCRIPT>
</BODY>
</HTML>
```

OR

```
<HTML>
<BODY>
<SCRIPT TYPE=”text/JavaScript” >
          SCRIPTING CODE
</SCRIPT>
</BODY>
</HTML>
```

==========

EXAMPLE:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE=”JAVASCRIPT” >
      document.write(‘hello’);
</SCRIPT>
</BODY>
</HTML>
```

2. EXPLAIN THE FEATURES/CHARACTERISTICS OF JAVA SCRIPT.

- **JavaScript is interpreted, not compiled**.
  - Some programming languages must be compiled or translated into machine code, before they can be executed.
  - JavaScript, on the other hand, is an interpreted language: so we can run program directly.
- **JavaScript can react to events** -
  - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
- **JavaScript can read and write HTML elements -**
  - A JavaScript can read and change the content of an HTML element
- **JavaScript can be used to validate data -**
  - A JavaScript can be used to validate form data before it is submitted to a server, this will save the server from extra processing
- **Can be used on client side as well as on server side**
  - As java script has access to Document object model of browser, you can actually change the structure of web pages at runtime.
  - On the other hand, java script could be used on the server side as well.
- **Java script is a case sensitive language**
  - Case sensitive means function name “sum” is same as function name “SUM”

- **Comments**
  - Single-line comments start with a double-slash (//).
  - Multi-line comments begin with a slash-asterisk (/*) and end with an asterisk-slash (*/)
- **JavaScript can perform actions**
  - JavaScript can perform actions on various objects in an HTML document, such as frames, buttons, links, and other objects.

3. EXPLAIN DATA TYPE OF JAVASCRIPT.

- Data type is a specification that shows what kinds of data a variable can hold.
- In java script when we store data inside the variable according to the type of data, data type will be automatically defined.
- JavaScript provides different data types to hold different types of values.
- There are two types of data types in javascript.
  - Primitive data type
  - Non-primitive (reference) data type

PRIMITIVE DATA TYPE

- There are five types of primitive data type supported by java script:

Data Type	Description
String	Represents Sequence Of Characters E.G. "Hello"
Number	Represents Numeric Values E.G. 100
Boolean	Represents Boolean Value Either False Or True
Undefined	Represents Undefined Value
Null	Represents Null I.E. No Value At All

NON PRIMITIVE DATA TYPE

Data Type	Description
Object	Represents Instance Through Which We Can Access Members
Array	Represents Group Of Similar Values
RegExp	Represents Regular Expression

4. LIST AND EXPLAIN OPERATORS AVILABLE IN JAVASCRIPT.

Arithmetic: perform arithmetic on numbers or variables	Assignment: assign values to variables	Comparison and Logical:	Logical	String Operators: Used to add (concatenate) strings.	Conditional Operator: Evaluates an expression for a true or false
+	=	==	&& (Logical AND)	+	(? :)
-	+=	===	|| (Logical OR)		typeof
*	-=	!=	! (Logical NOT)		ne
/	*=	!==			
%	/=	>			
++	%=	<			
--		>=			
		<=			

5. SHORT NOTE: JAVA SCRIPT VARIABLE.

- variable is one type of container which is used to hold value
- It is mainly used to hold values that we want to use during the script.
- Before you use a variable in a JavaScript program, you must declare it.
- Variables are declared with the (***var***) keyword.

```
<scripttype="text/JavaScript">
var name ="meghna";
var money;
money=5000;
                document.write (money);
</script>
================output==============================================
======
5000
```

RULES FOR VARIABLE:

- In java script variables are case sensitive
- Variable name must begin with alphabet or underscore sign.
- Generally variable can have short name.
- You cannot use any special character as variable name
- Any java script keyword cannot be used as variable name.

- When we Store a value in a variable is called variable initialization.
- **(=)** sign is an assignment operator which is used to assign value in variable.
- Java script does not allow the data type of the variable to be declared when a variable is created.
- The same variable can be used to hold different types of data.

JAVASCRIPT VARIABLE SCOPE

- The scope of a variable is the section of your program in which it is defined.
- JavaScript variables have only two scopes.
    - **Global Variables** –
        - A global variable has global scope which means it can be defined anywhere in your JavaScript code.
    - **Local Variables** –
        - A local variable will be visible only within a function where it is defined.
        - Function parameters are always local to that function.

6. EXPLAIN DIALOG BOX AVAILABLE IN JAVA SCRIPT.

- JavaScript supports three important types of dialog boxes.
    - Alert
    - Prompt
    - Confirm
- These dialog boxes can be used to raise and alert, or to get confirmation on any input or to have a kind of input from the user.
- This dialog boxes display as separate windows that contain one or two buttons.

ALERT DIALOG BOX:

- An alert dialog box is mostly used to give a warning message to the users
- It will display message in a separate window.
- This dialog button contain only one button: **OK**
- When use press OK button it will display information provided by the user.

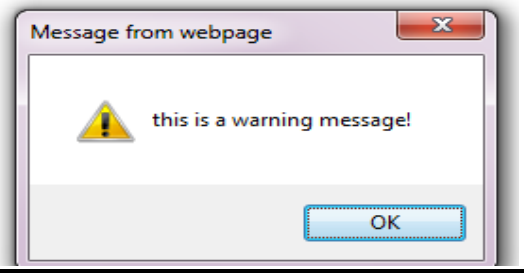
SYNTAX:

alert(“message”);

EXAMPLE:

```
<script type="text/JavaScript">
alert ("this is a warning message!");
</script>
```

========== OUTPUT ==========



PROMPT DIALOG BOX:

- The prompt dialog box is very useful when you want to pop-up a text box to get user input.
- It enables you to interact with the user.
- The user needs to fill in the field and then click OK.

- This dialog box is displayed using a method called prompt() which takes two parameters:
  - **A label which you want to display in the text box**
  - **A default string to display in the text box.**
- This dialog box has two buttons: OK and Cancel.
- If the user clicks the OK button, the window method prompt() will return the entered value from the text box.
- If the user clicks the Cancel button, the window method prompt() returns null.

SYNTAX:

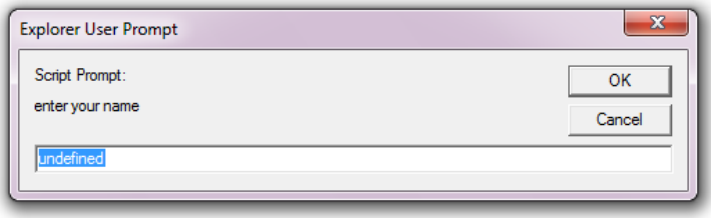
prompt("message", "default message")

EXAMPLE:

```
<script type="text/JavaScript">
    var name=prompt("enter your name");
    document.write("your name is: "+name);
</script>
```

==========

OUTPUT:==========



CONFIRM DIALOG BOX

- A confirm dialog box is mostly used to take user's consent on any option.
- It displays a dialog box with two buttons: OK and CANCEL.
- If the user clicks on the OK button, the window method confirm () will return true.
- If the user clicks on the Cancel button, then confirm() returns false

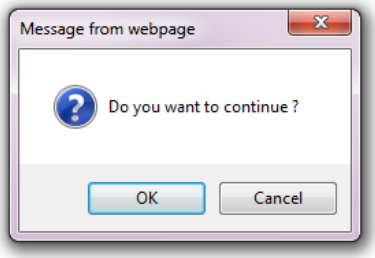
SYNTAX:

```
if(confirm("message")
{
        true block
}
else
{
    false block
}
```

EXAMPLE:

```
<script type="text/JavaScript">
var c = confirm("Do you want to continue ?");
if( c == true )
{
document.write ("User wants to continue!");
}
else
```

```
{
document.write ("User does not want to continue!");
 }
</script>
==================================
OUTPUT:==================================
```



7. EXPLAIN CONDITIONAL STATEMENT OF JAVA SCRIPT.

- JavaScript supports conditional statements which are used to perform different actions based on different conditions.
- It is also known as decision making statements.
- JavaScript supports the following forms of if...else statement –
  - if statement
  - if...else statement
  - if...else if... statement
  - Switch statement

IF STATEMENT:

- if you want to execute some code if a condition is true at that time IF statement / Simple IF is used.

SYNTAX:

```
if (condition)
{
    code to be executed if condition is true
}
```

EXAMPLE:

```
<script type="text/JavaScript">
var age = prompt("enter your age.:");
if( age >=18 )
{
document.write ("You are eligible for vote!");
}
</script>
```

IF… ELSE STATEMENT:

- if you want to execute some code if a condition is true and some other code if condition becomes false at that time IF …. Else statement is used.

SYNTAX:

```
if (condition)
{
        true block
}
else
{
        false block
}
```

EXAMPLE:

```
<script type="text/JavaScript">
var age = prompt("enter your age.:");
if( age >=18 )
{
document.write ("You are eligible for vote!");
}
else
{
document.write ("You are not eligible for vote!");
}
</script>
```

IF...ELSE IF... STATEMENT

- The if...else if... statement allows JavaScript to make a correct decisionout of several conditions.

SYNTAX

```
if (condition 1)
{
  Statement if condition 1 is true
}
else if (condition 2)
{
  Statement if condition 2 is true
}
else if (condition 3)
{
  Statement condition 3 is true
}
else
{
Statement if no condition is true
}
```

EXAMPLE:

```
<script type="text/JavaScript">
var book = prompt('enter book name:');
if( book == "wsl" )
```

```
{
        document.write("<b>wsl book</b>");
}
else if( book == "vb" )
{
document.write("<b>vb book</b>");
}
else if( book == "c" )
{
        document.write("<b>c book</b>");
}
else
{
        document.write("<b>unknown book</b>");
 }
</script>
```

SWITCH STATEMENT

- SWITCH statement allows JavaScript to make a correct decisionout of several conditions.

SYNTAX:

```
switch (expression)
{
case label1:
        code to be executed if expression = label1
        break
case label2:
        code to be executed if expression = label2
        break
default:
        code to be executed
        if expression is different
}
```

EXAMPLE:

```
<script type="text/JavaScript">
var book = prompt('enter book name:');
switch (book)
{

case "WSL":
        document.write("select WSL book");
        break;
case "VB":
        document.write("select VB book");
        break;
case "C":
        document.write("select C book");
```

```
        break;
default:
        document.write("unknown book!");
}
</script>
```

8. EXPLAIN LOOPING STRUCTURE.

- In some situation when you need to perform an action over and over again.
- In such situations, you would need to write loop statements.
- Java script support 3 types of looping statement:
    - while
    - do.. while
    - for

WHILE LOOP:

- The most basic loop in JavaScript is the while loop.
- It's an entry control looping statement that is used to repeat statement multiple times.
- The purpose of a while loop is to execute a statement or code block repeatedly as long as an expression is true.
- Once the expression becomes false, the loop terminates.

SYNTAX:

```
while(expression)
{
        Statementif expression istrue
}
```

EXAMPLE:

```
<script type="text/JavaScript">
var n =1;
while (n<=5)
{
        document.write(n+"<BR>");
        n=n+1;
}
</script>
```

==========

OUTPUT:

1
2
3
4
5

DO….WHILE LOOP:

- It's an exit control looping statement.

- 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
{
        Statementif expression istrue
}
while(expression);
```

EXAMPLE:

```
<script type="text/JavaScript">
var n =1;
do
{
        document.write(n+"<BR>");
        n=n+1;
} while (n<=5);
</script>
```

==========

OUTPUT:

1
2
3
4
5

FOR LOOP:

- The 'for' loop is the most compact form of looping. It includes the following three important parts –
- **initialization**
  - The loop initialization where we initialize our counter to a starting value.
  - The initialization statement is executed before the loop begins.
- **Test condition:**
  - The test statement which will test if a given condition is true or not.
  - If the condition is true, then the code given inside the loop will be executed, otherwise the control will come out of the loop.
- **Iteration:**
  - The iteration statement where you can increase or decrease your counter.

SYNTAX:

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

```
}
```

EXAMPLE:

```
<script type="text/JavaScript">
var n;
for(n=1;n<=5;n++)
{
        document.write(n+"<BR>");
}
</script>
```

==========

OUTPUT:

1
2
3
4
5

9. WHAT IS ARRAY? EXPLAIN VARIOUS TYPES OF ARRAY

- The Array object lets you store multiple values in a single variable.
- It stores a collection of multiple values of the same type.
- We can create an array using NEW keyword.
- Array use index number for storing set of values & index always begin with 0.

SYNTAX:

```
varvarname = new Array(elements of array );
varvarname= new Array(length);
```

EXAMPLE:

```
<script type="text/JavaScript">
var c=new Array(3);
c[0]="red";
c[1]="green";
c[2]="blue";
document.write(c+"<BR>");
document.write(c[2]);
</script>
```

==========

OUTPUT:

red,green,blue
blue

10. SHORT NOTE: USER DEFINE FUNCTION

- A function is a block of code that will be executed when "someone" calls it.
- In JavaScript, we can define our own functions, called user defined functions.
- A user define function execute block of code when it is called.
- We can create user define function using***function***keyword.

- This type of function is case sensitive.
- Function name must begin with alphabet.
- UDF is used between <script>…..</script> in the head section of the document.
- in java script function are two types:
    a. user define function with argument
    b. User define function without using parameter as an argument.

SYNTAX:

```
Function <function name> (parameter)
{
        Code
}
```

EXAMPLE:

```
<html>
<head>
<script type="text/JavaScript">
functionmsg()
{
      alert("Hello Welcome to my site!");
}
</script>
</head>
<body>
<script type="text/JavaScript">
      msg ();
<script>
</body>
</html>
```

11. EXPLAIN DOCUMENT OBJECT.

- Document is an object that allows you to place all elements of html like: HEAD , BODY

Document Object Properties

Alink	The color of active links.
Bgcolor	Sets the background color attribute set in the <body> tag.
Fgcolor	The text color attribute set in the <body> tag.
Domain	The domain name of the document server.
FileCreatedDate	Returns the date and time the document was created
lastModified	Returns the date and time the document was last modified
Forms	Find out how many <form> elements there are in the document
Images	Find out how many <img> elements there are in the document:
Links	Returns a collection of all <a> and <area> elements in the document that have a href attribute

Linkcolor	The color of HTML links in the document. It is specified in the <body> tag.
Vlinkcolor	The color of visited links as specified in the <body> tag/
URL	The location of the current document.
Title	The name of the current document as described between the header TITLE tags.
Cookie	Used to identify the value of a cookie.

Document Object methods

Close()	Closes an output stream and forces data to display.
Open()	Opens a stream to collect the output of write or writeln methods.
Write()	Writes any number of expressions in the document window.
Writeln()	Any number of expressions in the document window and follows them with a newline character.

We can use following events with document object.

EVENTS

- onafterupdate
- onbeforeupdate
- onClick
- ondblclick
- ondragstart
- onerrorupdate
- onhelp
- onkeydown
- onkeypress
- onkeyup
- onmousedown
- onmousemove
- onMouseOut
- onMouseOver
- onmouseup
- onreadystatechange
- onrowenter
- onrowexit
- onselectstart

12. EXPLAIN NAVIGATOR OBJECT.

- The JavaScript navigator object is the object representation of the client internet browser or web navigator program that is being used.
- This object is the top level object to all others.

allCodeName	return name of the browser's code
AppMinorVersion	Return The minor version number of the browser.
AppName	Return The name of the browser such as "Microsoft Internet Explorer" or "Netscape Navigator".
appVersion	Return The version of the browser which may include a compatibility value and operating system name.
browserLanguage	Return the current browser language
cookieEnabled	Return Boolean value that specified whether cookes are enabled in browser

Platform	A description of the operating system platform.
systemLanguage	Returns the default language used by OS
userAgent	Describes the browser associated user agent header.
userLanguage	Returns the OS natural language setting.
Online	Determines whether the browser is online or not and return Boolean value

Methods

Method	**Description**
javaEnabled()	Specifies whether or not the browser has Java enabled
taintEnabled()	Specifies whether the browser has data tainting enabled

13. SHORT NOTE: HISTORY OBJECT

- The JavaScript history object represents an array of URLs visited by the user.
- By using this object, you can load previous, forward or any particular page.
- The history object is the window property, so it can be accessed by:

window. history Orhistory

PROPERTY

property	**Description**
length	Returns the length of the history URLs. The property returns at least 1, because the list includes the currently loaded page.

METHODS

METHOD	**DESCRIPTION**
forward()	loads the next page
back()	loads the previous page
go()	loads the given page number.

14. SHORT NOTE: COOKIES

- Some website stores the information in small text file on your computer so this type of file is called a cookie.
- A cookie stored information such as personal information like user name, password, home address, work address, telephone number, city name and so on.
- The purpose of cookie is to identify the user.
- A cookie file size cannot exceed more than **4KB.**
- A cookie file is easily use and modify by the user.
- When user requests a page, then request is sent to the server with the help of HTTP.

- After this process, server receives the request from the user and server returns the request to the user with the help of HTTP.
- When a user requests a page for the first time, cookies can be stored in the browser by a set-cookie entry.
- The set-cookie field includes information to be stored in the cookie along with several information like name, expire date, path, domain and security.
- In future, when a user request a page then the browser send the stored cookies information to the server.
- Cookies are a plain text data record of 5 variable-length fields –

NAME	DESCRIPTION
NAME = value	○ It specifies the name of the cookie.
EXPIRES = date	○ It specifies the expiry date of the cookie. ○ After this date, the cookie will no longer to be stored by the user. ○ The form of date is DD-MON-YY HH:MM:SS
PATH = path	○ It specifies the path of the URL for the cookie. ○ If the URL matches both the PATH and DOMAIN then the cookie is sent to the server. ○ The default value is the current web page.
DOMAIN = domain	○ It specifies the domain name of the URL for the cookie. ○ The default value is the current domain name.
SECURE	○ It specifies the cookie should be transmitted over a secure link.

EXAMPLE:

```
<html>
<head>
<script type="text/JavaScript">
functionnewcookie()
{
    document.cookie = name=geetanjali";
}
</script>
</head>
<body onload="newcookie()">
<script>
        vartr=document.cookie;
        document.write( "STRING: =" + str);
</script>
</body>
</html>
```

==========

Output
STRING: =name=geetanjali

15. EXPLAIN IN DETAIL: WINDOW OBJECT

- The window object represents an open window in a browser.
- Window object is a top-level object, it contains other objects like 'document', 'history' etc. within it.

Name	Description
closed	Specify whether a window is closed or not.
defaultStatus	Retrieves the default message displayed in the window's status bar at the bottom.
document	Contains information about the current document.
innerHeight	Specifies the height of the window in pixels.
innerWidth	Specifies the width of the document in pixels.
location	Specify the information of the current URL.
locationbar	Test whether the location bar is visible is or not.
menubar	Refers the browser window's menu bar.
name	Get the unique name of a specified window.
outerHeight	Specifies the window's outside boundary in pixel
outerWidth	Specifies window's outside boundary in pixel
pageXOffset	Refers the current x-position of a page in a window.
scrollbars	Refers the scroll bars of the browser window.
status	Specifies a string displayed in the browser status bar at the bottom of the window.
statusbar	Refers the status bar of the browser window.
toolbar	Refers the toolbar of the browser window.

Javascript Window Objects Methods

Name	Description

alert	Displays an Alert dialog box with a message and an OK button.
back	Takes the browser to the previous URL in the current history list.
blur	Removes focus from a specific window or frame.
close	Closes the specified window.
confirm	Displays a Confirm dialog box with the specified message and two buttons OK and Cancel.
find	Search the specified text string in the contents of the specified window.
focus	Gives focus to a specific window.
forward	Takes the browser to the next URL in the current history list.
home	Returns the browser to the user's specified home page.
open	Opens a new web browser window.
print	Prints the contents of the window.
prompt	Displays a Prompt dialog box including a message and an input field.
setTimeout	Evaluates an expression or calls a function after a specified number of milliseconds.

16. EXPLAIN STRING FUNCTION OF JAVA SCRIPT.

FUNCTION NAME	DESCRIPTION	EXAMPLE	OUTPUT
Big() **SYNTAX:** **string.big( )**	String to be displayed in a big font as if it were in a BIG tag.	**<script type="text/javascript">** **varstr = new String("Hello world");** **alert(str.big());** **</script>**	**HELLO WORLD**
Small() **SYNTAX:** **string.small( )**	string to be displayed in a small font as if it were in a <SMALL>	**<script type="text/javascript">** **varstr = new String("Hello world");** **alert(str.small());** **</script>**	**HELLO WORLD**
Bold() **SYNTAX:** **string.bold( )**	string to be displayed in a bold font	**<script type="text/javascript">** **varstr = new String("Hello world");** **alert(str.bold());**	**HELLO WORLD**

	as if it were in a <B>	**</script>**	
Italics() **SYNTAX:** **string.italics( )**	string to be displayed in a italic font as if it were in a <I>	**<script type="text/javascript">** **varstr = new String("Hello world");** **alert(str.italics());** **</script>**	***HELLO WORLD***
Strike() **SYNTAX:** **string.strike( )**	string to be displayed in a strike out font as if it were in a <STRIKE>	**<script type="text/javascript">** **varstr = new String("Hello world");** **alert(str.strike());** **</script>**	~~**HELLO WORLD**~~
Fontsize() **SYNTAX:** **string.fontsize** **(size)**	Display string in the specified size as if it were in a <font size="size"> tag.	**<script type="text/javascript">varstr = new String("Hello world");** **alert(str.fontsize(3));** **</script>**	**HELLO WORLD**
Fontcolor() SYNTAX: string.fontcolor (color)	Display string in the specified size as if it were in a <font Color="red" > tag.	**<script type="text/javascript">varstr = new String("Hello world");** **alert(str.fontcolor('red'));** **</script>**	**HELLO WORLD**
Link() SYNTAX: string.link (hrefname)	creates an HTML hypertext link that requests another URL.	**<script type="text/javascript">varstr = new String("Hello world");** **var URL = "http://www.google.com";** **alert(str.link(URL));** **</script>**	
Charat() SYNTAX: string.charAt (index);	returns the character from the specified index.	**<script type="text/javascript">varstr = new String("hello");** **document.writeln(str.charAt(0));** **</script>**	**h**
Concat() SYNTAX: string.concat	Merge two or more strings and	**<script type="text/javascript">** **var str1 = new String("hello");** **var str2 = new String("world");**	**helloworld**

(string2, string3 ….);	returns a new single string.	**var str3 = str1.concat(str2); document.write(str3); </script>**	
Indexof() SYNTAX: string.indexOf (searchValue [, fromIndex])	returns the index within the calling String object of the first occurrence of the specified value, starting the search at fromIndex	**<script type="text/javascript"> var str1 = new String("This is string one"); var index = str1.indexOf("one"); document.write(index); </script>**	**15**
Match() SYNTAX: str.matcngh (“string”);	We can match the string with given string	**<script> document.write("compterconcepts".match("concepts")); </script>**	**concepts**
Search() SYNTAX: string.search (“string”);	We cansearch the string with given string	**<script> document.write("compterconcepts".search("t")); </script>**	**4**
Slice() SYNTAX: String.slice (start,end)	Returns the string from the given string by using start & end parameter	**<script type="text/javascript"> var a="computer"; document.write(a.substr(5,7)); </script>**	**ter**
Substr() SYNTAX: string.substr (start[, length]);	returns the characters in a string beginning at the specified location through the specified number of characters.	**<script type="text/javascript"> var a="computer"; document.write(a.substr(5,3)); </script>**	**ter**
Tolowercase	Converts	**<script>**	**computer**

() **SYNTAX:** **String.** **Tolowercase** **()**	string into lowercase.	**document.write("COMPUTER".** **tolowercase());** **</script>**	
Touppercase **()** **SYNTAX:** **String.** **Touppercase** **()**	Converts string into lowercase.	**<script>** **document.write("compter".** **touppercase());** **</script>**	**COMPUT** **ER**
Length() **SYNTAX:** **String.length** **()**	Returns the length of given string	**<script>** **document.write("compter".** **length());** **</script>**	**8**

17. EXPLAIN DATE FUNCTION OF JAVA SCRIPT.

FUNCTION NAME	DESCRIPTION	EXAMPLE	OUTPUT
date() **syntax:** **date()**	Returns today's date and time.	**<script>** **document.write(date());** **</script>**	**Tue Sep 08 2015 00:58:23 GMT-0700**
getDate() **SYNTAX:** **Date.getDate()**	Returns today's date and time.	**<script>** **document.write(new Date().getDate());** **</script>**	**8**
getDay() **SYNTAX:** **Date.getDay()**	Returns the day of the week for the specified date according to local time. Returns an integer corresponding to the day of the week: 0 for Sunday, 1 for Monday and so on	**<script type="text/javascript">** **vardt = new Date("December 25,2015 23:15:00");** **document.write("getDay() : " + dt.getDay());** **</script>**	**6**
getMonth() **SYNTAX:** **Date.getMonth() ;**	Returns current month of the year. Returns months value between 0 to 11	**<script type="text/javascript">** **vardt = new Date("December 25,2015 23:15:00");** **document.write(dt.getMonth());** **</script>**	**08**
getYear( ) **SYNTAX:**	Returns current year.	**<script type="text/javascript">** **vardt = new Date("December**	**2015**

Date.getYear()		**25,2015 23:15:00"); document.write(dt.getYear()); </script>**	
getFullyear () SYNTAX: Date.getFullYear()	Returns current full year.	**<script type="text/javascript"> vardt = new Date("December 25,2015 23:15:00"); document.write(dt.getFullyear()); </script>**	**2015**
getHours() SYNTAX: Date.getHours()	Return the hour, according to local time: values between 0 to 23	**<script type="text/javascript"> vardt = new Date("December 25,2015 23:15:00"); document.write(dt.getHours()); </script>**	**23**
getMinutes() SYNTAX: Date.getMinutes()	Returns the minutes in the specified date according to local time. values between 0 to 59	**<script type="text/javascript"> vardt = new Date("December 25,2015 23:15:00"); document.write(dt.getMinutes()); </script>**	**15**
getSeconds() SYNTAX: Date.getSeconds()	Returns the seconds in the specified date according to local time. values between 0 to 59	**<script type="text/javascript"> vardt = new Date("December 25,2015 23:15:10"); document.write(dt.getMinutes()); </script>**	**10**
getMilliseconds() SYNTAX: Date. getMilliseconds()	Returns the milliseconds in the specified date according to local time.	**<script type="text/javascript"> vardt = new Date( ); document.write(dt.getMilliseconds()); </script>**	**765**
setDate() SYNTAX: Date.setDate(dayValue)	sets the day of the month for a specified date according to local time.	**<script type="text/javascript"> vardt = new Date("Aug 28, 2015 23:30:00"); dt.setDate(24); document.write(dt); </script>**	**Aug 24, 2015 23:30:00**
setMonth() SYNTAX: Date.setMonth (monthValue[, dayValue])	sets the month for a specified date according to local time.	**<script type="text/javascript"> vardt = new Date("Aug 28, 2015 23:30:00"); dt.setMonth(2); document.write(dt); </script>**	**Fri Mar 28 2008 23:30:00 GMT+0530**
setYear() SYNTAX:	sets the year for a specified date	**<script type="text/javascript"> vardt = new Date("Aug 28, 2008**	**Mon Aug 28 2015**

Date.setYear (yearValue)	according to universal time.	**13:30:00"); dt.setYear(2015); document.write(dt); </script>**	**13:30:00 GMT+0530**
setFullyear() SYNTAX: Date.setSeconds()	sets the year for a specified date according to universal time.	**<script type="text/javascript"> vardt = new Date("Aug 28, 2008 13:30:00"); dt.setfullYear(2015); document.write(dt); </script>**	**Mon Aug 28 2015 13:30:00 GMT+0530**
setHours() SYNTAX: Date.setHours (hoursValue [, minutesValue [, secondsValue[, msValue]]])	sets the hours for a specified date according to local time.	**<script type="text/javascript"> vardt = new Date("Aug 28, 2015 23:30:00"); dt.setHours(02); document.write(dt); </script>**	**Thu Aug 28 2015 02:30:00 GMT+0530**
setMinutes() SYNTAX: Date.setMinutes (minutesValue[, secondsValue[, msValue]])	sets the minutes for a specified date according to local time.	**<script type="text/javascript"> vardt = new Date("Aug 28, 2015 23:30:00"); dt.setMinutes(45); document.write(dt); </script>**	**Thu Aug 28 2015 23:45:00 GMT+0530**
setSeconds() SYNTAX: Date.setSeconds (secondsValue[, msValue])	sets the seconds for a specified date according to local time	**<script type="text/javascript"> vardt = new Date("Aug 28, 2008 23:30:00"); dt.setSeconds(20); document.write(dt); </script>**	**Thu Aug 28 2008 23:51:20 GMT+0530**

18. EXPLAIN MATH FUNCTION OF JAVA SCRIPT.

FUNCTION NAME	DESCRIPTION	EXAMPLE	OUT PUT
abs() SYNTAX: Math.abs(x) ;	Returns the absolute value of a number.	**<script type="text/javascript"> document.write(math.abs(-5) </script>**	**5**
ceil() SYNTAX: Math.ceil(x)	returns the smallest integer greater than or equal to a number.	**<script type="text/javascript"> document.write(math.ceil(45.25) </script>**	**46**
floor() SYNTAX: Math.floor(x)	returns the largest integer less than or equal to a number.	**<script type="text/javascript"> document.write(math.floor(45.25));**	**45**

		</script>	
Pow() **SYNTAX:** **Math.pow(base, exponent) ;**	returns the base to the exponent power, that is, base exponent	**<script type="text/javascript">** **document.write(math.pow(5,2));** **</script>**	**25**
random( ) **SYNTAX:** **Math.random()**	returns a random number between 0 and 1	**<script type="text/javascript">** **document.write(math.random());** **</script>**	**0.473631626692924475**
round() **SYNTAX:** **Math.round(x)**	returns the value of a number rounded to the nearest integer.	**<script type="text/javascript">** **document.write(math.round(45.25));** **</script>**	**45**
max() **SYNTAX:** **Math.max(value1, value2, ... valueN) ;**	Returns the maximum no from given numbers	**<script type="text/javascript">** **document.write(math.max(45,40,41));** **</script>**	**45**
min() **SYNTAX:** **Math.min(value1, value2, n... valueN) ;**	Returns the maximum no from given numbers	**<script type="text/javascript">** **document.write(math.min(45,40,41));** **</script>**	**40**

19. EXPLAIN ARRAY FUNCTION OF JAVA SCRIPT.

FUNCTION NAME	DESCRIPTION	**EXAMPLE**	**OUTPUT**
join() **SYNTAX:** **array.join (separator);**	joins all the elements of an array into a string.	**<script type="text/javascript">** **var a=new array(3);** **a[0]=”red”;a[1]=”green”;a[2]=”blue”;** **document.write(a.join(“and”));** **</script>**	**red and green and blue**
reverse() **SYNTAX:** **Array.reverse( )**	By using this function we can print elements of array in reverse order.	**<script type="text/javascript">** **var a=new array(3);** **a[0]=”red”;a[1]=”green”;a[2]=”blue”;** **document.write(a.reverse());** **</script>**	**blue green red**
pop() **SYNTAX:**	Remove & return last element of array	**<script type="text/javascript">** **var a=new array(3);**	**blue**

Array.pop( )		**a[0]=”red”;a[1]=”green”;a[2]=”blue”; document.write(a.pop()); </script>**	
push() SYNTAX: Array.push (value)	Add one more element to the end of array & return new length	**<script type="text/javascript"> var a=new array(3); a[0]=”red”;a[1]=”green”;a[2]=”blue”; document.write(a.push(‘pink’)); </script>**	**4**
shift() SYNTAX: Array.shift( )	Remove & returns the first element of an array.	**<script type="text/javascript"> var a=new array(3); a[0]=”red”;a[1]=”green”;a[2]=”blue”; document.write(a.shift()); </script>**	**red**
sort() SYNTAX: Array.sort	Sorts elements of an array alphabetical order	**<script type="text/javascript"> var a=new array(3); a[0]=”red”;a[1]=”green”;a[2]=”blue”; document.write(a.sort()); </script>**	**blue green red**

20. EXPLAIN FOLLOWING EVENTS WITH EXAMPLE.

- JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.
- When the page loads, it is called an event. When the user clicks a button, that click too is an event.
- Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users or data to be validated

EVENT NAME	DESCRIPTION	SUPPORTED BY JAVASCRIPT OBJECTS
onclick	Occurs when a user clicks the left button of his mouse. You can put your validation, warning etc., against this event type.	**Button, Checkbox, Radio, Reset, Submit**
ondblclick	Occurs when a user double clicks the left button of his mouse. You can put your validation, warning etc., against this event type.	**Document, Link**
onfocus	This event occurs when input element get focus	**Button, Checkbox,**

		Frame, Password, Radio, Reset, Submit, Text, Textarea, Layer
onblur	This event Triggers when the window loses focus	**Button, Checkbox, Frame, Password, Radio, Reset, Submit, Text, Textarea**
onchange	This event Triggers when an element changes	**Select, Text, Textarea**
onkeyup	This event is occurred when a user release the key	**Document, Image, Link, Textarea**
onkeydown	This event is occurred when a user presses the key	**Document, Image, Link, Textarea**
onkeypress	This event is fired when a us.er press the key from the keyboard and release a key	**Document, Image, Link, Textarea**
onmouseover	These two event types will help you create nice effects with images or even with text as well. The onmouseover event triggers when you bring your mouse over any element	**Document, Button, Link**
onmouseout	These two event types will help you create nice effects with images or even with text as well. The onmouseout event triggers when you bring your mouse out any element	**Document, Button, Link**
onmousemove	This event triggers when use moves mouse pointer	**Document, Button, Link**
onsubmit	This event is occurs when a form is submitted	**Form**
onreset	This event is occurs when user reset form by pressing reset button of the form	**Form**
onload	This event occurs when the document loads or user enters the page.	**Image, Layer, Window**
onunload	This event occurs when the document unloads or user leaves the page.	**Image, Layer, Window**
onselect	This event is fired when user select text in a textbox or text area	**Text, Textarea**

Remote log in	• It's a facility that allows a user to connect to a host computer via a network or direct telecommunications link, and to interact with that host computer as if the user terminal were directly connected to that host computer
telnet	• Telnet is a protocol that allows you to connect to remote computers (called hosts) over a TCP/IP network (such as the internet). • Using telnet client software on your computer, you can make a connection to a telnet server (i.e., the remote host). • Once your telnet client establishes a connection to the remote host, your client becomes a virtual terminal, allowing you to communicate with the remote host from your computer.
HTTP	• HTTP stands for Hypertext Transfer Protocol. • It is the set of rules for transferring files on the World Wide Web. • As soon as a Web user opens their Web browser, the user is indirectly making use of HTTP.
FTP	• FTP stands for File Transfer Protocol. • It is a standard network protocol used for the transfer of computer files between a client and server on a computer network. • It may be authenticated with user names and passwords.
URL	• URL stands for Uniform Resource Locator. • It is also known as a Universal Resource Locator (URL) or Web address. • It specify address of a resource on the Internet. • A URL indicates the location of a resource as well as the protocol used to access it. • A URL contains the following information: • The protocol used to a access the resource • The the location of the server (whether by IP address or domain name) • The port number on the server (optional) • The location of the resource in the directory structure of the server
IP address	• An Internet Protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.[1][2] An IP address serves two principal functions: host or network interface identification and location addressing. • There are two versions of IP protocol • IPV4 ○ Internet Protocol version 4 (IPv4) defines an IP address as a 32-bit number

	1. IPV6 ○ Internet Protocol version 6 (IPv6), using 128 bits for the IP address
Smart card	**1.** A plastic card with a built-in microprocessor, used typically to perform financial transactions.