UNIT - 1

1

INTERNET AND WORLD WIDE WEB

Unit Structure :

1.0 Objective
1.1 Introduction to internet and its applications.
  1.1.1 Email
  1.1.2 Telnet
  1.1.3 FTP
  1.1.4 E–commerce
  1.1.5 Video conferencing
  1.1.6 E business.
1.2 Internet service providers
1.3 Domain name servers
1.4 Internet address
1.5 World wide web and its evolution
1.6 URL
1.7 Browsers
  1.7.1 Internet explorer,
  1.7.2 Netscape navigator,
  1.7.3 opera,
  1.7.4 fire fox,
  1.7.5 chrome,
  1.7.6 Mozilla,
1.8 Search Engine
1.9 Web server
  1.9.1 Apache
  1.9.2 IIS
  1.9.3 Proxy Server
1.10 HTTP protocol.
1.11 Summery
1.12 Unit End exercise

1.0 OBJECTIVE:

After reading through this chapter, you will be able to –

- Understand concept of Internet and World Wide Web, their applications.
- List the services provided by Internet Service providers with examples.
- Define domain name server and list various domains.
- Understand the concept of Internet address.
- Understand the function of a URL and web browsers.

munotes.in

- Use different web browsers.
- Use the search engines to search for required information over the internet.
- Understand the need and use of a web server and proxy server.

1.1 INTRODUCTION TO INTERNET AND ITS APPLICATIONS

Internet is ---

- A computer network consisting of a worldwide network of computers that use the TCP/IP network protocols to facilitate data transmission and exchange.
- The Internet is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide.
- The term Internet actually refers to the combined collection of academic, commercial, and government networks connected over international telecommunication backbones and routed using IP addressing.

The internet has gained popularity rapidly as it is used for various purposes. Few of the main applications of internet are listed below –

Applications of Internet ---

1.1.1 E mail (Electronic mail)

- Electronic mail (also known as email or e-mail) is one of the most commonly used services on the Internet, allowing people to send messages to one or more recipients.
- Email was invented by Ray Tomlinson in1972.
- Electronic mail is a method of exchanging digital messages from an author to one or more recipients.
- Modern email operates across the Internet or other computer networks. Today's email systems are based on a store-and-forward model.
- Email servers accept, forward, deliver and store messages. Neither the users nor their computers are required to be online simultaneously; they need connect only briefly, typically to an email server, for as long as it takes to send or receive messages.
- An email message consists of three components, the message *envelope*, the message *header*, and the message *body*.
- Header contains information about who sent the message, the recipient(s) and the route.
- Header also usually contains descriptive information, such as a subject header field and a message submission date/timestamp.

munotes.in

- Email message body contains text (7bit ASCII) as well as multimedia messages. These processes are declared in Multipurpose Internet Mail Extensions (MIME). MIME is set of RFCs (Request for Comment)
- Network based emails are exchanged over the internet using the SMTP (Simple Mail Transfer protocol).
- In the process of transporting email messages between systems, SMTP communicates delivery parameters using a message *envelope* separate from the message (header and body)itself.
- Email addresses (both for senders and recipients) are two strings separated by the character "@" (the "at sign"): such as user@domain
- The right-hand part describes the domain name involved, and the left-hand part refers to the user who belongs to that domain.
- An email address can be up to 255 characters long and can include the following characters:
  - Lowercase letters from a toz;
  - Digits
  - The characters "."," _" and "-" (full stop, underscore, and hyphen)

In practice, an email address often looks something like this:
fname.lname@provider.domain

1.1.2 Telnet

- Telnet is a network protocol used in any network (internet or LAN) for bidirectional text oriented communication.
- telnet standard was defined in 1973, before which it was considered as a dhocprotocol.
- Original purpose of the telnet protocol was to login to the remote computers on the network.
- Telnet protocol uses 'virtual terminal' to connect to the remote hosts.
- Virtual terminal is a application service that allows host in a multi terminal network to communicate with other hosts irrespective of terminal type or characteristics.
- Telnet uses the TCP protocol for transmission of 8 byte data.
- Most network equipment and operating systems with a TCP/IP stack support a Telnet service for remote configuration (including systems based on Windows NT)
- The term telnet may also refer to the software that implements the client part of the protocol.
- telnet is a client server protocol, which is based upon reliable connection oriented communication transport and basic use of telnet is to make a connection to the TCP protocol.

munotes.in

- Data transferred over telnet is vulnerable as telnet does not use any encryption technique to mask or protect the data.
- Most implementations of Telnet have no authentication that would ensure communication is carried out between the two desired hosts and not intercepted in the middle.
- Commonly used Telnet daemons have several vulnerabilities discovered over the years.
- Extensions to the Telnet protocol provide Transport Layer Security (TLS) security and Simple Authentication and Security Layer (SASL) authentication that address the above issues.
- Few applications of telnet include the 'putty' TCP client which can access a linux server using windows operating system, Absolute telnet (windows client) and RUMBA (terminal emulator).

1.1.3 FTP

- File transfer protocol is a simple and standard network protocols that transfers a file from one host to the other over a TCP network.
- Based on client server architecture.
- Utilizes separate control and data connection for client and server to transmit and receive file(s) over the network.
- It is an application protocol that uses the internet's TCP/IP suite.
- Mainly used to transfer the web pages or related data from the source or creator to a host that acts as a server to make the page or file available to other hosts (uploading) or downloading programs and other files from server to a host.
- FTP protocol can perform over a active or passive connection.
- When a connection is made from the client to server, it is called as control connection and it remains open for duration of session. This connection is responsible for establishing connectivity between client and server.
- Other connection opened by client (passive) or server (active) is called data connection and is used to transfer the data.
- As separate ports are used by client and server for these connections, FTP becomes an put of band protocol.
- Data transfer can take place in following three modes
    - Stream mode : data is sent in a continuous stream where FTP does not do any formatting.
    - Block mode: FTP breaks the data into several blocks (block header, byte count, and data field) and then passes it on to TCP.
    - Compressed mode: Data is compressed using a single algorithm.
- FTP is a old protocol and is basically low in security aspect. Data transferred over FTP is not encrypted and is in clear text format.

Hence the data like usernames, passwords can be read by anyone who can capture the FTPed package. Newer versions of the protocol, however, apply secure shell protocol (SSH) and avoid all the problems faced by FTP.

- Following are few types of FTP protocol with additional features
- Anonymous FTP : Users login using an 'anonymous' account to protect their confidential data.
- Remote FTP: FTP commands executed on a remote FTP server
- FTP with web browser and firewall support.
- Secure FTP (SFTP,FTPS)

1.1.4 ECommerce

- Electronic commerce can be defined as use of electronic communications, particularly via the internet, to facilitate the purchase/ sale of goods and services. E-commerce includes all forms of electronic trading including electronic data interchange (EDI), electronic banking, electronic mail and other online services.
- E transactions are of two categories. – virtual products like policies and actual retail products.
- Most of e transactions of actual products involve physical transportation of goods which are purchased over the electronic media.
- Online retailing has gained a name of Etailing.
- Electronic commerce is generally considered to be the sales aspect of e-business. It also consists of the exchange of data to facilitate the financing and payment aspects of the business transactions.
- Originally, electronic commerce was identified as the facilitation of commercial transactions electronically, using technology such as Electronic Data Interchange (EDI) and Electronic Funds Transfer (EFT).
- Other forms of e commerce were established with the growth and use of credit cards, and air line reservation system going online.
- Electronic commerce of the modern era (post 1990) includes technologies like enterprise resource planning (ERP), data warehousing and data mining.
- The electronic transactions between two businesses like dealer and wholesaler or wholesaler and retailer come in the B2B (business to business) E commerce category.
- Other popular E commerce categories would be business to consumer (B2C) and business to government(B2G)
- Volume of B2B transaction is much higher as compared to the volume of B2C transactions. Reason for this is, many transactions at B2B level lead to finished good and this leads to just one B2C transaction.

- In an example, if a customer buys a product, say a pen, that would be a B2C transaction. But the transaction leading this one, including purchase of plastic, ink, refill, moulds etc would be B2B transaction. Also the sale of the pen to the retailer by the manufacturing company, like cello, is B2Btransaction.
- OtherformofB2Ctransactionsarebusinesstoindividual,where the record of an individual's transaction is maintained.
- C2C is consumer to consumer, or citizen to citizen E commerce. Here customers can perform transaction via a third party. Like a product can be posted on amazon.com and will be sold to another consumer through Amazon.
- C2B E commerce model is reverse of traditional business to consumer approach. This can be explained by a internet blog or a social networking site where author can have a link in his blog article to online sale of a product (promoting the business). This has become possible due to advancements in technology and reduced costs of technology.
- Unique attribute of e commerce is negotiation facility and its immediate results. Also, in E commerce transactions, integration of transactions is automated.

1.1.5 Video Conferencing

- Video conferencing or video teleconference is a set of telecommunication technologies which allow one or more locations to transmit and receive video and audio signals simultaneously.
- This is known as visual collaboration.
- Simple analog video conferencing is achieved by two closed circuit television systems connected with coaxial cables or radio waves.
- This type of communication was established from1968.
- Modern video conferencing is IP based and through more efficient video compression technologies, allowing desktop or PC based video conferencing.
- Video telephony is now popular due to free internet services.
- Core technology used for this is compression of audio and video signals. Hardware and software used for this task is called as codec (coder/ decoder). Compression rate achieved is almost 1:500. The resultant stream of binary data is sent in packet form through digital networks.
- The components required for a videoconferencing system include:
  - Video input : video camera or webcam
  - Video output: computer monitor , television or projector
  - Audio input: microphones, CD/DVD player, cassette player, or any other source of Pre Amp audio outlet.
  - Audio output: usually loudspeakers associated with the display device or telephone
  - Data transfer: analog or digital telephone network, LAN or Internet

- There are basically two types of videoconferencing systems.
  - Dedicated systems: all required components (i.e. software and hardware based codec, control computer and video camera, electrical interfaces) packed in a single console application. They include large group, small group, portable and non portable video conferencing systems.
  - Desktop Systems: add ons to normal computing systems transforming these systems to videoconferencing devices.
- There are following layers in the videoconferencing technology–
  - User interface
  - Conference control
  - Control or signal plane
  - Media plane.
- Videoconferencing has following modes
  - Voice activated switch
  - Continuous presense
- Problems faced by videoconferencing
  - Echo: echo is defined as reflected source wave interference with new wave created by the source. i.e. signal coming out from the source interferes with newly coming source and generating unwanted input signal. This may result into remote party receiving their own sounds again. This can be avoided by using an algorithm called as AEC (Acoustic Echo cancellation).
  - Eye contact : in videoconferencing, due to time delays and parallax, communicators have a feel of the other party avoiding eye contact and can result into issues in professional communication. This can be avoided by using special stereo reconstruction technique.
  - Signal latency: The information transport of digital signals in many steps needs time. In a telecommunicated conversation, an increased latency larger than about 150-300ms becomes noticeable and is soon observed as unnatural and distracting. Therefore, next to a stable large bandwidth, a small total round-trip time is another major technical requirement for the communication channel for interactive videoconferencing.

1.1.6 E business

- E business is conduct of business over the internet, which includes buying and selling of goods and even services.
- In other words it is application of information and communication technologies in support of all activities in business.
- Applications of E business are divided into following categories–

- Internal business systems--
- Customer Relationship Management(CRM)
- Enterprise Resource Planning(ERP)
- Human resource management system(HRMS)
- Enterprise Communication and collaboration
- Content management
- E- mails
- Voicemails
- Web conferencing
- Electronic commerce
- B2B (business to business)
- B2C (business to customer)
- B2E business-to-employee
- B2Gbusiness-to-government
- G2Bgovernment-to-business
- G2G (government-to-government)
- G2C (government-to-citizen)
- C2C (consumer-to-consumer)
- C2B(consumer-to-business)

- A business model is defined as the organization of product, service and information flows, and the source of revenues and benefits for suppliers and customers. The concept of e-business model is the same but used in the online presence.
- Few e business models are–
  - E-shops
  - E-commerce
  - E-procurement
  - E-malls
  - E-auctions
  - Virtual Communities
- E business has more security risks as compared to a regular business, as E business has many more users at a time. Keeping the large information confidential is a difficult task. Also, data integrity, authenticity and storage of data are some challenges faced by Ebusiness.
- Some methods to provide security are physical security as well as encryption in data storage, transmission, antivirus software and firewalls. Digital signature is another way to confirm the ownership of a document.

munotes.in

1.2 INTERNET SERVICE PROVIDER

- An Internet service provider (ISP) is a company that provides access to the Internet, hosts data, or does both. ISP is also known as IAP (internet access provider) Access ISPs connect customers to the Internet using copper, wireless or fibre connections. Hosting ISPs lease server space for smaller businesses and host other people servers (colocation). Transit ISPs provide large tubes for connecting hosting ISPs to access ISPs.

- As internet gained popularity, it was essential to provide internet access to many people or many hosts. Due to the increasing demand to access internet, commercial ISP came into existence in1990.

Technologies used –

For users and small business applications -
- Dial up connection
- DSL (digital subscriber line)
- Broadband wireless connection
- Cable modem
- Fibre optical connection

For medium to large businesses or for other ISPs,
- DSL
- Ethernet
- Metro Ethernet
- Gigabyte Ethernet
- Frame relay
- Satellite Ethernet

- ***ISP connections–***

ISPs which provide connections through phone lines like dial ups, do not seek any information about the caller's (user of the internet) physical location or address. So, caller from any location which is in reach of the ISP, can use the services provided.

Other way of getting connected through ISP is using cable or any other lines. Here, fixed registration of the user at the ISP side is essential.

- ***Services provided –***

ISP host usually provide e mail, FTP and web hosting services. Other services can be like virtual machines, clouds or entire physical servers where clients can run their own softwares.

ISPs often take services from their upstream ISPs. i.e. they work in hierarchy. The ISPs are divided into three categories –

- **Peering :** ISPs taking services from upstream and getting connected to each other to exchange data and to control network traffic through peering points, orinternetex change points. These points help save one

more upstream ISP and cut down on the cost. The ISPs which do not need upstream ISP are called Tier 1 ISP.

 Virtual ISP (VISP) : this is an ISP which purchases services from other ISP and gives them to the end user, without any set up of its own.

 Free ISP: these are ISPs which provide services free of cost to the users and display advertisements till the users are connected. These are called as freenets. These are normally run on no profit basis.

1.3 DOMAIN NAME SERVERS

- ***Domain Name System is–***

DNS is part of a domain name system. It is hierarchical naming system built on a distributed database for resource connected to the internet or a private network. The main purpose of this system is to translate domain names meaningful to humans into names or rather numeric streams which help the corresponding network devices to identify the resource or domain. Domain name system makes it possible to give or allot names to domains or group of networks irrespective of their physical locations.

- ***Domain Name Server is–***

- Domain name system assigns domain name servers for allotting names and mapping these names to IP addresses. In other words domain name servers are nodes of the domain name system which acts like a client server system. Each domain has at least one authoritative DNS server that publishes information about that domain and the name servers of any domains subordinate to it. The top of the hierarchy is served by the root name servers, the servers to query when looking up (resolving) a TLD (Top level domain). Authoritative DNS can either be a master or a slave. Master DNS keeps record of all zone records. Slave DNS uses a automatic update mechanism to maintain copies of records of its master. Every top level domain requires a primary DNS and at least one secondary DNS. Every DNS query must start with recursive queries at the root zone, for authoritative DNS.
- To improve efficiency, reduce DNS traffic across the Internet, and increase performance in end-user applications, the Domain Name System supports DNS cache servers which store DNS query results for a period of time determined in the configuration (time-to-live) of the domain name record in question.
- The client-side of the DNS is called a DNS resolver. It is responsible for initiating and sequencing the queries that ultimately lead to a full resolution (translation) of the resource sought, e.g., translation of a domain name into an IP address.
- A DNS query may be either a non-recursive query or a recursive query

- The resolver, or another DNS server acting recursively on behalf of the resolver, egotiates use of recursive service using bits in the query headers.
- Resolving usually entails iterating through several name servers to find the needed information. However, some resolvers function simplistically and can communicate only with a single name server. These simple resolvers (called "stub resolvers") rely on a recursive name server to perform the work of finding information for them.
- ***Operation of DNS–***
- Domain name resolvers determine the appropriate domain name servers responsible for the domain name in question by a sequence of queries starting with the right-most (top-level) domain label.
- DNS recorsor consults three name servers to resolve one address. The process is as follows–

  - A network host is configured with an initial cache (so called *hints*) of the known addresses of the root name servers. Such a *hint file* is updated periodically by an administrator from a reliable source.
  - A query to one of the root servers to find the server authoritative for the top-level domain.
  - A query to the obtained TLD server for the address of a DNS server authoritative for the second-level domain.
  - Repetition of the previous step to process each domain name label in sequence, until the final step which returns the IP address of the host sought.

1.4 INTERNET ADDRESS

- Internet address follows the TCP/IP suite hence, it is also known as the IP address.
- Internet address has a job of identifying a node on the network. In other words, it is a numeric lable attached to every system (computer or any other device). The basic function of IP address are two–
- Identification of computer or node or device and location addressing.
- The designers of the Internet Protocol defined an IP address as a 32-bit number[1] and this system, known as Internet Protocol Version 4 (IPv4), is still in use today. However, due to the enormous growth of the Internet and the predicted depletion of available addresses, a new addressing system (IPv6), using 128 bits for the address, was developed in 1995,[3] standardized as RFC 2460 in 1998,[4] and is being deployed worldwide since themid-2000s.
- IP addresses are binary numbers, but they are usually stored in text files and displayed in human-readable notations, such as 172.16.254.1 (forIPv4)

munotes.in

- IPV4 address is a 32 bit number, which uses the decimal doted notation consisting of 4 decimal numbers each ranging from 0 to 255 separated by dots. Network administration divides the IP address into two parts. – the most significant 8 bits are called network address portion the remaining bits are known as rest bits or host bits or identifiers and they are used for host numbering in anetwork.
- Although IPV4 provides 4.3 billion addresses, they are exhausted due to high demand and as a result, insufficient addresses available with IANA (Internet assigned numbers authority). The primary address pool of IANA is expected to get exhausted by mid 2011. To permanently address the problem, new version of IP i.e. IPV6 was brought forward, this version moved the size of IP address from 32 bit to 128 bits.
- Both IPV4 as well as IPV6 have reserved addresses for private or internal networks. This is termed as private addressing.
- Both IPV4 and IPV6 have sub netting effect. That mean, IP networks can be divided into smaller groups or subnets. IP addresses two constituents that is network address and host identifier or interface identifier are used for this purpose.
- Internet Protocol addresses are assigned to a host either anew at the time of booting, or permanently by fixed configuration of its hardware or software. Persistent configuration is also known as using a *static IP address*. In contrast, in situations when the computer's IP address is assigned newly each time, this is known as using a*dynamic IP address*

1.5 WORLD WIDE WEB AND ITS EVOLUTION

- The World Wide Web, abbreviated as WWW or W3 and commonly known as the Web, is a system of interlinked hypertext documents accessed via the Internet.
- With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.
- The World-Wide Web was developed to be a pool of human knowledge, and human culture, which would allow collaborators in remote sites to share their ideas and all aspects of a common project.

Evolution of WWW

- In March 1989, Tim Berners-Lee wrote a proposal that referenced ENQUIRE, a database and software project he had built in 1980, and described a more elaborate information management system.
- on November 12, 1990, with help from Robert Cailliau, Tim Berners- Lee published a more formal proposal to build a "Hypertext project" called "World Wide Web" (one word, also "W3") as a "web" of "hypertext documents" to be viewed by "browsers" using a client– server architecture.

- This proposal estimated that a read-only web would be developed within three months and that it would take six months to achieve "the creation of new links and new material by readers, to achieve universal authorshipl" as well as "the automatic notification of a reader when new material of interest to him/her has become available."
- A NeXT Computer was used by Berners-Lee as the world's first web server and also to write the first web browser, World Wide Web, in 1990.
- Tools needed were a working Web the first web browser (which was a web editor as well); the first web server; and the first web pages, which described the project itself.
- On August 6, 1991, Tim Berners-Lee posted a short summary of the World Wide Web project on the alt. hypertext newsgroup.
- This date also marked the debut of the Web as a publicly available service on the Internet. The first photo on the web was uploaded by Berners-Lee in 1992, an image of the CERN house band Les Horribles Cernettes.
- The first server outside Europe was set up at SLAC to host the SPIRES-HEP database in 91 –92.
- The concept of hypertext originated with older projects from the 1960s, such as the Hypertext Editing System (HES) at Brown University by Ted Nelson and Douglas Engelbart.
- Tim Berners Lee introduced the concept of the Universal Document Identifier (UDI), later known as Uniform Resource Locator (URL) and Uniform Resource Identifier (URI); the publishing language Hyper Text Markup Language (HTML); and the Hypertext Transfer Protocol(HTTP).
- In 1993, a graphical browser was developed by a team at the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign (NCSA-UIUC), led by Marc Andreessen. This was the first web browserever.

1.6 URL

- Uniform Resource Locator (URL) is a Uniform Resource Identifier (URI) that specifies where an identified resource is available and the mechanism for retrieving it.
- An example of the use of URLs is the addresses of web pages on the World Wide Web, such as*http://www.example.com/.*
- The format is based on Unix file path syntax, where forward slashes are used to separate directory or folder and file or resource names.
- Conventions already existed where server names could be prepended to complete file paths, preceded by adouble-slash

- Every URL consists of some of the following:
  - The scheme name (commonly called protocol), followed by a colon. The scheme name defines the namespace, purpose, and the syntax of the remaining part of the URL.
  - Domain Name depending upon scheme(alternatively, IP address). The domain name or IP address gives the destination location for the URL.
  - An optional port number; if omitted, the default for the scheme is used
  - Path of the resource to be fetched or the program to be run. The path is used to specify and perhaps find the resource requested. It may be case-sensitive for non window based servers. Eg: *http://www.mudlle.ac.in/news.html*
  - A query string for scripts The query string contains data to be passed to software running on the server. It may contain name/value pairs separated by ampersands, for example *? first_ name= John & last_name=Doe.*
  - Optional fragment identifier that specifies a part or a position within the overall resource or document. When used with HTTP, it usually specifies a section or location within the page, and the browser may scroll to display that part of the page.
- When resources contain references to other resources, they can use relative links to define the location of the second resource.
- relative URLs are dependent on the original URL containing a hierarchical structure against which the relative link is based.
- the ftp, http, and file URL schemes are examples of some that can be considered hierarchical, with the components of the hierarchy being separated by"/"
- A URL is a URI that, "in addition to identifying a resource, provides a means of locating the resource by describing its primary access mechanism.

1.7 BROWSERS

- A web browser or Internet browser is a software application for retrieving, presenting, and traversing information resources on the World Wide Web.
- Web browsers can also be used to access information provided by Web servers in private networks or files in file systems. Some browsers can also be used to save information resources to file systems.
- Primary function of a browser is to identify the URI and brings the information resource to user.

- This process begins when user inputs the URI in the browser. Prefix of the URI describes how to interpret the URI. Most URIs have resource retrieved over Hyper text Transfer Protocol. Some web browsers also support prefixes like FTP.
- Once this is done, the HTML script is passed to the browser's layout engine. To make the script interactive java script support is needed. With this, browser can interpret text, images, video and interactive scripts.
- All major browsers allow users to access multiple information resources at the same time in different windows or in tabs. Major browsers include pop up blockers to prevent windows to open without users consent.
- Most major web browsers have these user interface elements in common: Back and forward buttons to go back to the previous resource and forward again.
    - A history list, showing resources previously visited in a list (typically, the list is not visible all the time and has to be summoned)
    - A refresh or reload button to reload the current resource.
    - A stop button to cancel loading the resource. In some browsers, the stop button is merged with the reload button.
    - A home button to return to the user's homepage
    - An address bar to input the Uniform Resource Identifier (URI) of the desired resource and display it.
    - A search bar to input terms into a search engine
    - A status bar to display progress in loading the resource and also the URI of links when the cursor hovers over them, and page zooming capability.

- The usage share of web browsers is as shown below. (Source: Median values)

- Internet Explorer(43.55%)
- Mozilla Firefox (29.0%; Usage by version number)
- Google Chrome (13.89%)
- Safari(6.18%)
- Opera(2.74%)
- Mobile browsers (4.45%)
- Some special web browsers are listed below–

1.7.1 INTERNET EXPLORER

- Windows Internet Explorer (formerly Microsoft Internet Explorer, commonly abbreviated IE or MSIE) is a series of graphical web browsers developed by Microsoft and included as part of the Microsoft Windows line of operating systems starting in1995.

- It was first released as part of the add-on package Plus! for Windows 95 that year. Later versions were available as free downloads, or in service packs. It was part of later versions of windows operating systems.
- The latest stable release is Internet Explorer 9, which is available as a free update for Windows 7, Windows Vista, Windows Server 2008 and Windows Server 2008R2.
- Internet Explorer uses a componentized architecture built on the Component Object Model (COM) technology. It consists of several major components, each of which is contained in a separate Dynamic- link library (DLL) and exposes a set of COM programming interfaces hosted by the Internet Explorer main executable, 'iexplore.exe'
- Internet Explorer uses a zone-based security framework that groups sites based on certain conditions, including whether it is an Internet- or intranet-based site as well as a user-editable white list. Security restrictions are applied per zone; all the sites in a zone are subject to the restrictions.

1.7.2 NETSCAPE NAVIGATOR

- Netscape Navigator is a proprietary web browser that was popular in the 1990s. It was the most popular web browser till 2002, after which competitor browsers have taken over the market of Netscape.
- Netscape Navigator was based on the Mosaic web browser.
- Netscape announced in its first press release (October 13, 1994) that it would make Navigator available without charge to all non-commercial users, and Beta versions of version 1.0 and 1.1 were indeed freely downloadable in November 1994 and March 1995, with the full version 1.0 available in December1994.
- The first few releases of the product were made available in "commercial" and "evaluation" versions.
- During development, the Netscape browser was known by the code name Mozilla. Mozilla is now a generic name for matters related to the open source successor to Netscape Communicator.

1.7.3 OPERA

- Opera is a web browser and Internet suite developed by Opera Software. The browser handles common Internet-related tasks such as displaying web sites, sending and receiving e-mail messages, managing contacts, chatting on IRC, downloading files via Bit Torrent, and reading webfeeds.
- Opera is offered free of charge for personal computers and mobile phones. This is the most popular mobile phone browser and is not packages in desktop operating system.
- Features include tabbed browsing, page zooming, mouse gestures, and an integrated download manager. Its security features include built-in

phishing and malware protection, strong encryption when browsing secure websites, and the ability to easily delete private data such as HTTP cookies.

- Opera runs on a variety of personal computer operating systems, including Microsoft Windows, Mac OS X, Linux, and FreeBSD
- Opera includes built-in tabbed browsing, ad blocking, fraud protection, a download manager and Bit Torrent client, a search bar, and a web feed aggregator. Opera also comes with an e-mail client called Opera Mail and an IRC chat client builtin.
- Opera has several security features visible to the end user. One is the option to delete private data, such as HTTP cookies, the browsing history, and the cache, with the click of a button. This lets users erase personal data after browsing from a shared computer.
- Opera Mobile is an edition of Opera designed for smart phones and personal digital assistants(PDAs)

1.7.4 MOZILLA FIREFOX

- Mozilla Firefox is a free and open source web browser descended from the Mozilla Application Suite and managed by Mozilla Corporation. As of February 2011[update], Firefox is the second most widely used browser with approximately 30% of worldwide usage share of web browsers.
- To display web pages, Firefox uses the Gecko layout engine, which implements most current web standards.
- The latest Firefox features[15] include tabbed browsing, spell checking, incremental find, live book marking, a download manager, private browsing, location-aware browsing (also known as "geolocation") based exclusively on a Google service.
- Firefox runs on various operating systems including Microsoft Windows, Linux, Mac OS X, FreeBSD, and many other platforms.

1.7.5 CHROME

- Chrome, the web browser by Google, is rapidly becoming popular due to following features-
  - SPEED: Chrome is designed to be fast in every possible way: It's quick in staring up from the desktop, loading web pages and running complex web applications.
  - SIMPLICITY: Chrome's browser window is streamlined, clean and simple. Chrome also includes features that are designed for efficiency and ease of use. For example, you can search and navigate from the same box, and arrange tabs however you wish.
  - SECURITY: Chrome is designed to keep you safer and more secure on the web with built-in malware and phishing protection, auto updates to make sure the browser is up-to-date with the latest

security updates, and more. Learn more about Chrome's security features.

- Chrome is the first browser to incorporate machine translation in the browser itself, without requiring additional plugins orextensions.

1.8 SEARCH ENGINE

- A web search engine is designed to search for information on the World Wide Web and FTP servers. The search results are generally presented in a list of results and are often called hits. The information may consist of web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories.
- The very first tool used for searching on the Internet was Archie.
- The first web robot, the Perl-based World Wide Web Wanderer was built and used by it to generate an index called 'Wandex'. The purpose of the Wanderer was to measure the size of the World Wide Web.
- Around 2000, Google's search engine rose to prominence. The company achieved better results for many searches with an innovation called Page Rank. This iterative algorithm ranks web pages based on the number and Page Rank of other web sites and pages that link there, on the premise that good or desirable pages are linked to more than others.
- Web search engines work by storing information about many web pages, which they retrieve from the html itself. These pages are retrieved by a Web crawler (sometimes also known as a spider) — an automated Web browser which follows every link on the site.
- This information is then analyzed and indexed The contents of each page are then analyzed to determine how it should be indexed . The purpose of an index is to allow information to be found as quickly as possible.

1.9 WEB SERVER

1.9.1 APACHE

- Apache HTTP Server is a free and open-source web server that delivers web content through the internet. It is commonly referred to as Apache and after development; it quickly became the most popular HTTP client on the web.
- The word, Apache, has been taken from the name of the Native American tribe ‘Apache’, famous for its skills in warfare and strategy making.
- Apache is the most widely used Web Server application in Unix-like operating systems but can be used on almost all platforms such as Windows, OS X, OS/2, etc

- It is a modular, process-based web server application that creates a new thread with each simultaneous connection. It supports a number of features; many of them are compiled as separate modules and extend its core functionality, and can provide everything from server side programming language support to authentication mechanism.

Features of Apache:

- Handling of static files
- Loadable dynamic modules
- Auto-indexing
- Compatible with IPv6
- Supports HTTP/2
- FTP connections
- Bandwidth throttling
- Load balancing
- Session tracking
- URL rewriting
- Geolocation based on IP address and many more

1.9.2 IIS

- Internet Information Services (IIS) is a flexible, general-purpose web server from Microsoft that runs on Windows systems to serve requested HTML pages or files. IIS works through a variety of standard languages and protocols
- An IIS web server accepts requests from remote client computers and returns the appropriate response. This basic functionality allows web servers to share and deliver information across local area networks (LAN), such as corporate intranets, and wide area networks (WAN), such as the internet.
- A web server can deliver information to users in several forms, such as static webpages coded in HTML; through file exchanges as downloads and uploads; and text documents, image files and more.
- Features of IIS:
  - IIS is used to host ASP.NET web applications and static websites
  - It can also be used as an FTP server, host WCF services, and be extended to host web applications. built on other platforms such as PHP
  - An invaluable feature is remote management. IIS can also be managed via the CLI or using Power Shell
  - One of the key feature of IIS is application pool.

munotes.in

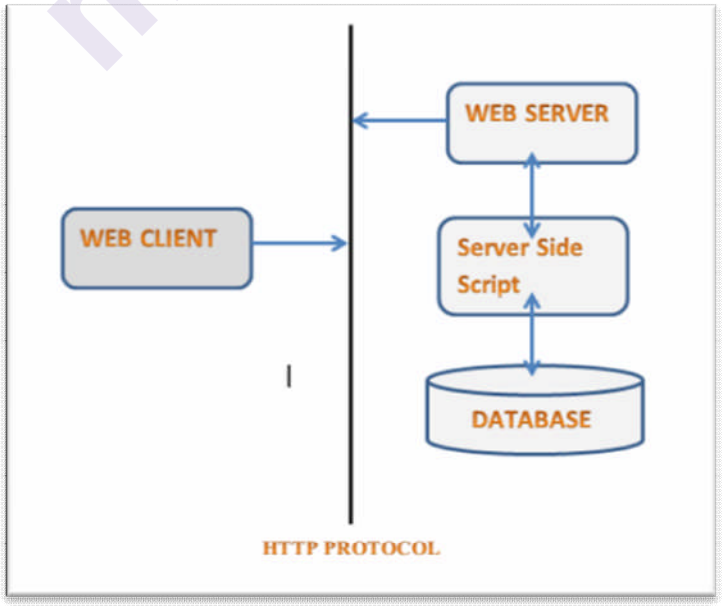
1.9.3 PROXY SERVER

- A proxy server provides a gateway between users and the internet. It is a server, referred to as an "intermediary" because it goes between end-users and the web pages they visit online.
- Proxies provide a valuable layer of security for your computer. They can be set up as web filters or firewalls, protecting your computer from internet threats like malware.
- Benefits of proxy server:
  - Enhanced security
  - Private browsing, watching, listening and shopping
  - Access to location– specific content
  - Prevent employees from browsing inappropriate or distracting sites

1.10 HTTP PROTOCOL

- HTTP stands for Hyper Text Transfer Protocol.
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as Hyper Text Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
- The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients and the Web server acts as a server.

The following diagram shows a very basic architecture of a web application and depicts where HTTP :



munotes.in

The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

Basic Features:

- ✓ HTTP is connectionless protocol
- ✓ HTTP is media independent
- ✓ HTTP is stateless

1.11 EXERCISE

1. Define the internet. What protocol suit does it follow?
2. What is email? How is it sent and received?
3. Describe the three components of email.
4. What is meant by email address? What are the required parts of email address?
5. Are following email addresses valid?
   a. 124sir@idol.com
   b. 11 myname@yahoo.org
   c. Seema_Sathye@ server.co.in
   d. Piyush_mishra@myservices.net.in
6. What is telnet used for?
7. What is virtual terminal? What is it used for?
8. Name applications of telnet.
9. Why is FTP protocol used?
10. Explain different connections that can be used by FTP.
11. What are drawbacks of telnet and FTP?
12. Define E commerce. What are the advantages of ecommerce?
13. Give one example of B2B, B2C and C2C ecommerce.
14. Identify following E commerce category–
    a. Sale of online admission form for a college.
    b. Submission of the above form.
    c. Online resale of a second handcar.
    d. Purchase of raw material by an automobile company.
15. What is visual collaboration?
16. What is codec how does it function?
17. What are different types of video conferencing system?
18. List the components of video conferencing system.
19. Discuss the problems faced by video conferencing system.

munotes.in

20. Define E business.
21. List a few applications of E business.
22. What is a E business model? Give three examples of E business model.
23. What are risks for E business? What are the solutions available for these risks?
24. What is ISP? Explain the role of ISP in an internet connection.
25. Classify following technologies of ISP in business or home connections

a. A dial up connection with speed1Mbps

b. A connection to a LAN using leased cable lines

c. Hosting of personal webpage

d. Use of wi-fi for a laptop.

26. What are different services provided by ISP?
27. Explain what is peering? What is its advantage?
28. What is VISP? Give one example to explain its use.
29. Discuss the concept of free net and its importance.
30. What are domain name servers? What is their function?
31. What is internet address? How is it assigned?
32. Write a note on evolution of www
33. What is a web browser? List and compare different available web browsers

❖❖❖❖

munotes.in

2

HTML5

Unit Structure :

2.0 Objectives
2.1 Introduction
2.2 Why HTML5?
2.3 Formatting text by using tags
2.4 Formatting text by lists
2.5 Formatting of backgrounds
2.6 Creating hyperlinks and anchors
2.7 Style sheets
2.8 CSS formatting text using style sheets
2.9 Formatting paragraphs using style sheets.
2.10 Summary
2.11 References
2.12 Exercise

2.0 OBJECTIVES:

This chapter would make you understand the following concepts: After going through this unit, you will be able to:

- To understandthebasicconceptofHTML5
- To learn and implements the Elements, Attributes of HTML5
- To understand the fundamentals of CSS & CSS3
- To learn and implements the Elements, Attributes of HTML5

2.1 INTRODUCTION

HTML5 is the latest and most enhanced version of HTML. HTML5 is enriched with advance features which make it easy and interactive for developer and users. It has some brand new features which will make HTML much easier. HTML5 is not a programming language, but rather a mark up language.

HTML5 is cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

munotes.in

HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 introduces a number of new elements and attributes that helps in building a modern website.

Features of HTML5

It allows you to play a video and audio file. You can embed audio or video on your web pages without resorting to third-party plugins.

It allows you to draw on a canvas which supports a two-dimensional drawing surface that you can program with JavaScript..

It facilitates you to design better forms and build web applications that work offline.

It provides you advance features for that you would normally have to write JavaScript to do.

New Semantic Elements: These are like <header>, <footer>, and <section>.

HTML5 provides bidirectional communication technology for web applications which is next-generation technology.

2.2 WHY HTML5?

A major focus of HTML5 was to give developers more flexibility, which in turn would lead to more engaging user experiences. There are various benefits of HTML5 and here's why it is the future.

There is a new set of tags that works in enhancing your HTML code, helping make it increasingly meaningful.

Cross-Browser Compatibility

HTML5 is easy to implement and it works with CSS3. Today all browsers support HTML5 tags and even IE6 understands the markup <!doctype html> and will render the pages correctly.

Bring improvements in usability and user experience

HTML5 has several technical enhancements and improved features for web And with HTML5 code, web developer can easily design better applications and dynamic websites which result in better user experience and usability.

Offline Browsing

HTML5 also offers offline browsing, which means that visitors can load certain elements on a web page without an active internet connection. With HTML5 offline caching we can still load core elements of the websites and you can view them offline.

Video and Audio Support

With HTML5 technology, we no longer need to rely upon third-party plugins in order to render audio and videos. You can forget about Flash Player and other third-party media players and plugins. You can make your videos and audio easily accessible with the new HTML5 <video> and <audio> tags.

2.3 FORMATTING TEXT BY USING TAGS

These HTML 5 tags (elements) provide a better document structure.HTML5 Formatting is a method of text formatting for a better appearance and look.

HTML5 allows us to format text without using the CSS.
This list shows all HTML 5 tags in alphabetical order with description.

Tag	Description
<!DOCTYPE>	This tag is used to inform the browser about the version of HTML used in the document.
<article>	This element is used to define an independent piece of content in a document, that may be a blog, a magazine or a newspaper article.
<header>	It defines a header for a section.
<main>	It defines the main content of a document.
<section>	It defines a section in the document.
<figcaption>	It is used to define a caption for a <figure> element.
<figure>	It defines a self-contained content like photos, diagrams etc.
<footer>	It defines a footer for a section.

<!DOCTYPE> tag:

HTML <!DOCTYPE> tag is used to inform the browser about the version of HTML used in the document. It is called as the document type declaration (DTD).

<!DOCTYPE> is not a tag/element, it just an instruction to the browser about the document type.

It is a null element which does not contain the closing tag, and must not include any content within it.

Syntax
<!DOCTYPE html>
An example of HTML document with doctype declaration.

```
<!DOCTYPE html>
<html>
<head>
<title>This is the title</title>
</head>
<body>
This is the content of the document.
</body>
</html>
```

<article> tag :

The HTML <article> tag defines an independent self-contained content in a document, page, application or a site.

The article tag content makes sense on its own. It is independent and complete from other content shown on the page. It is used on News story, blog post, article, Social post, comment etc.

Syntax
<article> ... </article>
The example below shows the <article>tag .

```
<article>
<h1>Introduction to HTML5</h1>
<p>. HTML5 is enriched with advance features which make it easy and
interactive for developer and users..</p>
</article>
```

HTML5 <header> Tag:
The <header> element represents the header of a document or a section. A header should contain title and heading information about the related content.

Syntax
<header> ... </header>
The example below shows the <header>tag .

```
<header>
<h2>Sports</h2>
<p>Cricket sport is World's no.1 sport all over the world</p>
</header>
```

HTML Main Tag:
HTML <main> tag is used to represent the main content of the <body> tag.
The <main> tag is written within <body> tag. It is used to accurately describe the primary content of a page.
The content of the main tag is directly related to the central topic of the document.

Syntax
<main> ... </main>

The example below shows the <main>tag .
<main>
<h1>Introduction to HTML5</h1>
. HTML5 is enriched with advance features which make it easy and interactive for developer and users..
</main>

HTML5 <section> Tag
The<section> tag is used to define sections in a document. When you put your content on a web page, it may contains many chapters, headers, footers, or other sections on a web page.

Syntax
<section> ... </section>
The example below shows the <section>tag .
<section>
<h1>Introduction to HTML5</h1>
. HTML5 is enriched with advance features which make it easy and interactive for developer and users..
</section>

HTML5 <figcaption> Tag
The <figcaption> element is used to provide a caption to an image.
It is an optional tag and can appear before or after the content within the <figure> tag.

Syntax
<figcaption> ... </figcaption>
The example below shows the <section>tag .
<figure>
<imgsrc="discovery.jpg" alt="Space Shuttle">
<figcaption>NASA - Space Shuttle Discovery</figcaption>
</figure>

2.4 FORMATTING TEXT BY LISTS

HTML lists allow web developers to group a set of related items in lists.HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

- **Ordered List or Numbered List (ol)**
- **Unordered List or Bulleted List (ul)**
- **Description List or Definition List (dl)**

munotes.in

Ordered List or Numbered List:

In the ordered HTML lists, all the list items are marked with numbers by default. It is known as numbered list also. The ordered list starts with <ol> tag and the list items start with <li> tag.

```
<ol>
<li>Sachin</li>
<li>Kapil</li>
<li>Rahul</li>
<li>Rohit</li>
</ol>
```

Unordered List or Bulleted List:

In HTML Unordered list, all the list items are marked with bullets. It is also known as bulleted list also. The Unordered list starts with <ul> tag and list items start with the <li> tag.

```
<ul>
<li>Sachin</li>
<li>Kapil</li>
<li>Rahul</li>
<li>Rohit</li>
</ul>
```

Description List or Definition List:

HTML Description list is also a list style which is supported by HTML and XHTML. It is also known as definition list where entries are listed like a dictionary.

The HTML definition list contains following three tags:

<dl> tag defines the start of the list.

<dt> tag defines a term.

<dd> tag defines the term definition (description).

```
<dl>
<dt>HTML5</dt>
<dd>HTML5 is the latest and most enhanced version of HTML.</dd>
</dl>
```

2.5 FORMATTING OF BACKGROUNDS

There are a couple of properties you can use to define color – HTML background-color and HTML color. As the name suggests, the first one is used to change the color of the background. By using the simple color property, you will change the color of the text.

The background-color property provides a color for the background of the text, but not for the whole document. If you wish to change the HTML color for the whole page, you should use the bgcolor attribute in the opening tag of the <body> section:

munotes.in

HTML provides you following two good ways to decorate your webpage background.

- **HTML Background with Colors**
- **HTML Background with Images**

Html Background with Colors:
The bgcolor attribute is used to control the background of an HTML element, specifically page body and table backgrounds.

Syntax
<tagnamebgcolor = "color_value"...>
The example below shows the <bgcolor>attribute.

```
<table bgcolor = "yellow" width = "100%">
<tr>
<td>

This background is yellow
</td>
</tr>
</table>
```

Html Background with Images:
The background attribute can also be used to control the background of an HTML element, specifically page body and table backgrounds.

Syntax
<tagname background = "Image_url"...>
The example below shows the < background > attribute.

```
<table background = "/images/html.gif" width = "100%" height = "100">
<tr><td>

This background is filled up with HTML image.
</td></tr>
</table>
```

2.6 CREATING HYPERLINKS AND ANCHORS:

HTML links are hyperlinks. You can click on a link and jump to another document. The HTML <a> tag defines a hyperlink. To make a hyperlink in an HTML page, use the <a> and tags, which are the tags used to define the links.

Syntax
**Hyperlink Text **
The example below shows the < a > tag.

```
<a href="Next.html">Next Page</a>
```

The HTML anchor tag defines a hyperlink that links one page to another page. It can create hyperlink to other web page as well as files,

location, or any URL. The href attribute is used to define the address of the file to be linked.

If we want to open that link to another page then we can use target attribute of <a> tag. With the help of this link will be open in next page.

Syntax

**Hyperlink Text **

The example below shows the < a > tag.

Home Page

2.7 STYLE SHEETS

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a mark-uplanguage.

CSS describes how HTML elements are to be displayed on screen.

CSS is a formatting language used to provide more customized web pages and make it easier to make multiple pages use the same style.

CSS is a web development technology that stands behind the look-and-feel of every web page.

How to use cascaded style sheets(CSS) with HTML?

There are three primary way to use style sheets

- **Inline style sheets**
- **Embedded style sheets**
- **Linked style sheets**

Inline style sheets:

This approach develop existing HTML, tags within a standard HTML documents & adds a specific style to the information controlled by that tag.

In this method the indentation of a single paragraph using the

style="X" attribute within the tag.

Another method of achieving this is combining the tag

And style="x" attribute.

Here is an inline style example : <spanstyle="font: 15ptArial">

Example of Inline Style Sheets:

<P style="font-size: x-large; color: #ff9900"> Using inline style sheets -or is that inlinestyles?

Embedded style sheets:

This method allows for the control of individual pages.

The rules specified here can be used for the current document.

It uses the <style> tags along with its companion tag</style>. The STYLE element requires the TYPE attribute to tell the browser which style language is being used.

```
<style>p { font-family: Arial, Helvetica, Times New Roman, sans-serif ;}
</style>
```

External or Linked style sheets:

An external style sheet is separate files where you can declare all the styles that you want to use on your website.

You then link to the external style sheet from all your HTML pages.

This means you only need to set the styles for each element once.

You can update style of your website,

```
<style>
p { font-family: Arial, Times new roman, sans-serif ;}
</style>
```

A save the file using the .cssextension

import a CSS file:

```
<link rel="stylesheet" href="external_sheet.css"type="text/css">
@import"external_sheet.css ";
@importurl("external_sheet.css");
```

2.8 FORMATTING TEXT WITH CSS

CSS provides several properties that allows you to define various text styles such as color, alignment, spacing, decoration, transformation very easily and effectively.

The commonly used text properties are: text-align, text-decoration, text-transform, text-indent, line-height, letter-spacing, word-spacing.

Text Color

The color of the text is defined by the CSS color property.

```
body {
color: #434343;
}
```

Text Alignment:

The text-align property is used to set the horizontal alignment of the text.Text can be aligned in four ways: to the left, right, centre or justified.

```
h1 {
text-align: center;
}
p {
```

```
text-align: justify;
}
```

Text Decoration:

The text-decoration property is used to set or remove decorations from text. This property accepts one of the following values: underline, overline, line-through, and none.

```
h1 {
text-decoration: overline;
}
h2 {
text-decoration: line-through;
}
h3 {
text-decoration: underline;
}
```

Text Transformation:

The text-transform property is used to set the cases for a text.Using this property you can change an element's text content into uppercase or lowercase letters.

```
h1 {
text-transform: uppercase;
}
h2 {
text-transform: capitalize;
}
h3 {
text-transform: lowercase;
}
```

Text Indentation:

The text-indent property is used to set the indentation of the first line of text within a block of text. The size of the indentation can be specified using percentage (%), length values in pixels, ems.

```
p {
text-indent: 100px;
}
```

2.9 FORMATTING PARAGRAPHS USING STYLE SHEETS:

Formatting is used to set options for paragraphs such as alignment, indentation, and spacing.

indent: It can be set left and right only. Creates space that represents the distance between the edge of the frame and content.

munotes.in

margin: It can be set top, bottom, left and right. Creates space that represents the distance between the indent and the border.

border: It can be set top, bottom, left and right. A visual effect that represents the distance between the margin and padding.

padding: It can be set top, bottom, left and right. Represents the distance between the border and content.

Single line css using style attribute:

** This is some text in a paragraph. **

Style paragraphs with CSS:

```
<style>
p {
color: navy;
text-indent: 30px;
text-transform: uppercase;
}
</style>
```

2.10 SUMMARY

- HTML5 chapter covers the key web components driving the future of the Web. Harness the power of HTML5 to create web apps and solutions that deliver state-of-the-art media content and interactivity with new Audio, Video, and Canvas elements.
- HTML5 technologies are essential knowledge for today's web developers and designers. New APIs such as Local Storage, Geolocation, Web Workers, and more expand the Web as a platform, allowing for desktop-like applications that work uniformly across platforms

2.11 REFERENCES

- HTML5 Unleashed by Simon Sarris
- HTML and CSS for Beginners with HTML5 by Mark Lass off
- HTML5 Beginner to Professional Step by Step Training Guide by Rajesh kumar.

2.11 EXERCISE

1) What do you understand by HTML tags? How many tags are required to create a web page in HTML5?
2) Explain the various formatting tags in HTML5.
3) Explain the various new tags introduced by HTML5 in Media Elements.

munotes.in

4) What is the correct usage of the following HTML5 semantic elements?
5) How many ways can a CSS be integrated as a web page?
6) What benefits and demerits do External Style Sheets have?
7) What are the limitations of CSS?
8) Explain what are the components of a CSS Style?
9) What do you mean by CSS selector?
10) Which property is used for controlling the image position in the background?
11) What is the main difference between class selectors and id selectors?
12) Which property of a table is used to the appearance of the adjacent borders?
13) How many formats you can display a CSS colour?
14) What do you mean by animation in CSS?
15) What are pseudo elements?

❖❖❖❖

munotes.in

UNIT - 2

3

NAVIGATION

Unit Structure

3.0 Objectives
3.1 Introduction
3.2 Planning site organization
3.3 Creating text based navigation bar
3.4 Creating graphics based navigation bar
3.5 Creating image map
3.6 Redirecting to another URL
3.7 Summary
3.8 Bibliography
3.9 Unit End Exercise

3.0 OBJECTIVES

- The objective of the chapter is as follow
- To get familiar with various Navigation bar
- To understand the creation of image map
- To understand the redirection to another URL

3.1 INTRODUCTION

- The provision of navigation aids in hypertext documents is very important.
- When reading a book, people start at a particular page and read the subsequent pages in order.
- Navigation aids prevent users from losing their place. Perusing a catalogue might not be as linear an activity, but the page numbers, table of contents and the index provide the means to navigate backwards and forwards, and to jump to arbitrary pages.

3.2 PLANNING SITE ORGANIZATION

➔ A website is a collection of related web pages, including multimedia content, typically identified with a common domain name, and published on at least one web server.

munotes.in

Types of web navigation

- The use of website navigation tools allow for a website's visitors to experience the site with the most efficiency and the least incompetence.
- A website navigation system is analogous to a road map which enables webpage visitors to explore and discover different areas and information contained within the website.
- There are many different types of website navigation: Hierarchical website navigation the structure of the website navigation is built from general to specific.
- This provides a clear, simple path to all the web pages from anywhere on the website.
- Global website navigation Global website navigation shows the top level sections/pages of the website.
- It is available on each page and lists the main content sections/pages of the website.

What is the "Look and Feel" of a Website?

- In its most basic terms, the "look and feel" of a website is how the site looks to the user and how it feels when he or she is interacting with it.
- The "look" is defined by the many components of your website.

Site map:

- A site map (or sitemap) is a list of pages of a web site accessible to crawlers or users.
- It can be either a document in any form used as a planning tool for Web design, or a Web page that lists the pages on a website, typically organized in hierarchical fashion.
- Sitemaps make relationships between pages and other content components. It shows shape of information space in overview. Sitemaps can demonstrate organization, navigation, and labeling system.

3.3 CREATING TEXT BASED NAVIGATION BAR

- A text-based navigation bar is the simplest and easiest, and it is also very user-friendly.
- On simple Web pages, text-based navigation bars are usually placed at the top of the page, in a single horizontal line.
- Some Web designers also place a copy at the bottom of each page too, so the visitor does not have to scroll back up to the top of a page to access the links.

munotes.in

➔ A navigation bar is a user interface element within a webpage that contains links to other sections of the website.

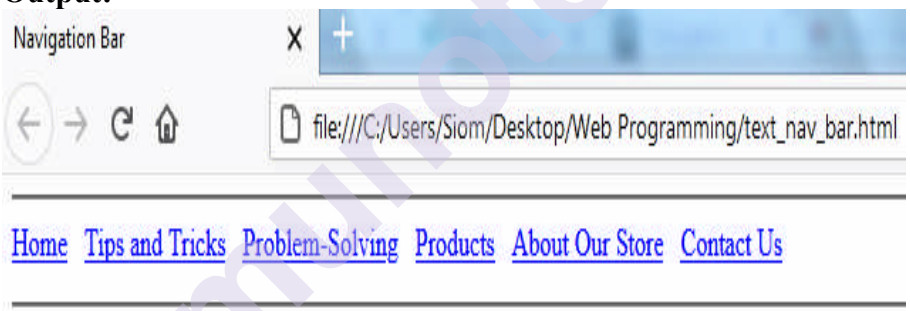
➔ We can use the navigation bar to visit other sections of the website.

Create A Top Navigation Bar:

Example:

```
<html>
        <head>
                <title> Navigation Bar </title>
        </head>
        <body>
        <hr />
                <a href="index.htm">Home</a> 
                <a href="tips.htm">Tips and Tricks</a> 
                <a href="problems.htm">Problem-Solving</a> 
                <a href="products.htm">Products</a> 
                <a href="about.htm">About Our Store</a> 
                <a href="contact.htm">Contact Us</a></p>
        <hr />
        </body>
</html>
```

Output:



Create A Bottom Navigation Bar:

➔To create bottom text navigation bar we just set the style class property

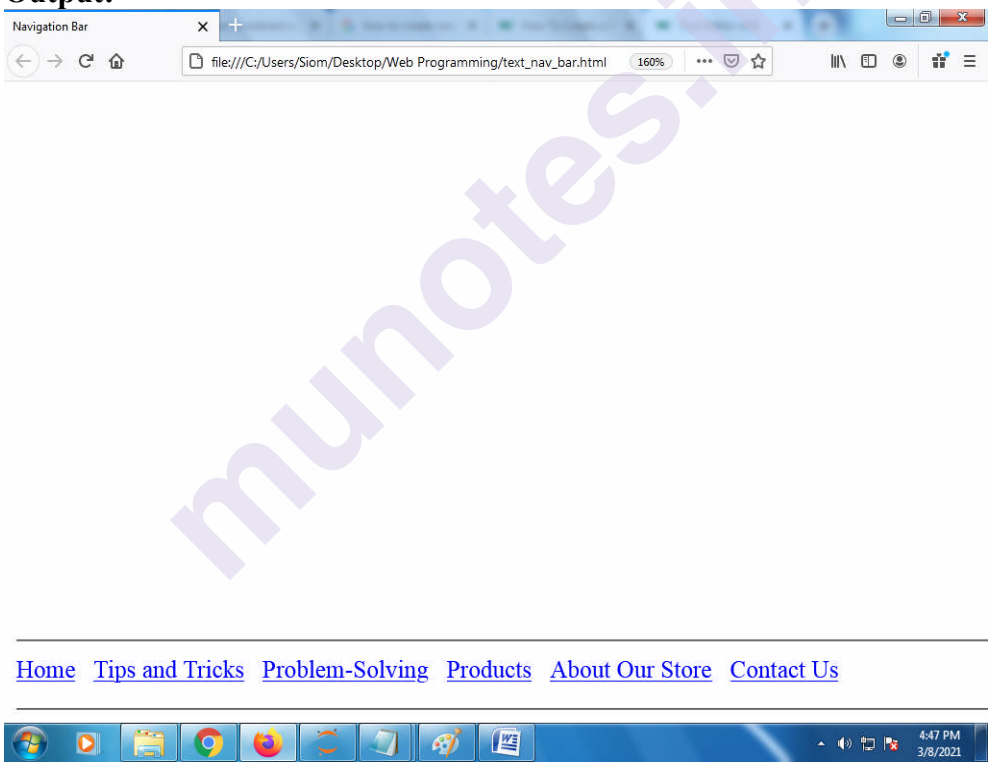
position: fixed;
bottom: 0;
width: 100%;

Example:

```
<html>
<head>
<title> Bottom Navigation Bar </title>
<style>
        .navbar
        {
                position: fixed;
                bottom: 0;
                width: 100%;
        }
```

```
</style>
</head>
<body>
<nav>
        <div class="navbar">
        <hr />
                <a href="index.htm">Home</a> 
                <a href="tips.htm">Tips and Tricks</a> 
                <a href="problems.htm">Problem-Solving</a> 
                <a href="products.htm">Products</a> 
                <a href="about.htm">About Our Store</a> 
                <a href="contact.htm">Contact Us</a></p>
        <hr />
        </div>
        </nav>
</body>
</html>
```

Output:



munotes.in

3.4 CREATING GRAPHICS BASED NAVIGATION BAR

- ➔ Text hyperlinks are clear and unambiguous, but not all that attractive.
- ➔ We might prefer to create a navigation bar that uses buttons or other graphics instead of text links.
- ➔ We can create the graphics yourself in a graphics-editing program. Follow these guidelines:

- Keep the size of each button small (150 pixels wide at the most).
- Make each button the same size and shape. They should vary only in the text on them.
- Save each button as a separate file in GIF or JPG format.

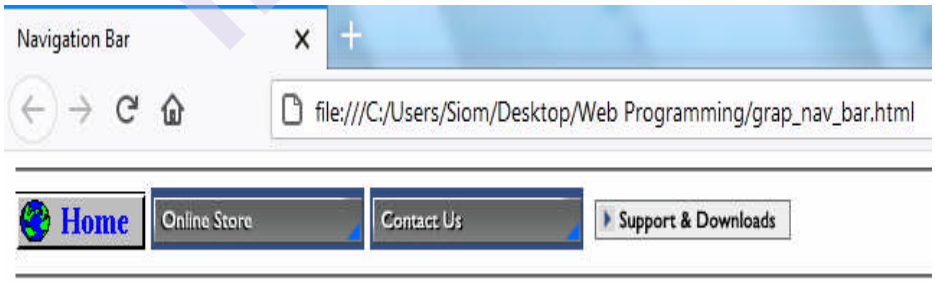
➔ There are thousands of sites with free graphical buttons you can download.

➔ Make several copies of a button you like, and then use a text tool in a graphics-editing program to place different text on each copy.

Example:

```
<html>
<head>
<title> Navigation Bar </title>
</head>
<body>
<nav>
        <hr>
        <p style="margin:0px">
        <a href="index.htm"><imgsrc="images/btn_home.gif"
style="border: none"></a>
        <a href="tips.htm"><imgsrc="images/btn_store.gif"
style="border:none"></a>
        <a href="problems.htm"><imgsrc="images/btn_contactus.gif"
style="border: none"></a>
        <a href="products.htm"><imgsrc="images/btn_support.gif" style=
"border: none"></a>
<hr>
</nav>
</body>
</html>
```

Output:



3.5 CREATING IMAGE MAP

What is image mapping:

➔ In image mapping an image is specified with certain set of coordinates inside the image which act as hyperlink areas to different destinations.

➔ It is different from an image link since in image linking, an image can be used to serve a single link or destination whereas in a mapped

image, different coordinates of the image can serve different links or destinations.

Elements required in Mapping an Image :

There are three basic html elements which are required for creating a mapped image.

1. **Map:** It is used to create a map of the image with clickable areas.
2. **Image:** It is used for the image source on which mapping is done.
3. **Area:** It is used within the map for defining clickable areas.

Steps to create a mapped image:

1) Determining Image size:

➔ Determining the size of the image is very important because if the size of the image is changed then the area coordinates will also require updation.

```
<imgsrc="world-map-151576_960_720.png"width="960"
height="492"  alt="World Map" usemap="#worldmap">
```

2) **Creating a map for overlaying the image:**

```
<map name="worldmap">
```

3) Determine the coordinates of the areas that you want to map:

➔It can be done in three shapes which are rectangle, circle and polygon.
➔Coordinates can be found easily by using MS-Paint.

```
<area shape="rect" coords="184, 36, 272, 158" alt="north america"
```

Example:

```
<! DOCTYPE html>
<html>
<h3>Mapping an Image
<body>
<p>Click on the different continents of the map to know about them.</p>
<imgsrc="world-map-151576_960_720.png"  width="960"
height="492" alt="World Map" usemap="#worldmap">
<map name="worldmap">
<area shape="rect" coords="184, 36, 272, 158" alt="north america"
href="https://en.wikipedia.org/wiki/North_America">
<area shape="rect" coords="282, 215, 354, 367" alt="south america"
href="https://en.wikipedia.org/wiki/South_America">
<area shape="rect" coords="506, 151, 570, 333" alt="africa"
href="https://en.wikipedia.org/wiki/Africa">
<area shape="rect" coords="618, 42, 791, 162" alt="asia"
href="https://en.wikipedia.org/wiki/Asia">
```

```
<area shape="rect" coords="509, 44, 593, 110" alt="europe"
href="https://en.wikipedia.org/wiki/Europe">
<area shape="rect" coords="786, 288, 862, 341" alt="australia"
href="https://en.wikipedia.org/wiki/Australia_(continent)">
<area shape="rect" coords="249, 463, 760, 488" alt="antartica"
href="https://en.wikipedia.org/wiki/Antarctica">
</map>
</body>
</html>
```

Output:

3.6 REDIRECTING TO ANOTHER URL

- ➔ To redirect from an HTML page to another page you can use the **<meta>** tag.
- ➔ It is the client-side redirection; the browsers request the server to provide another page.
- ➔ Also, use the http-equiv attribute to provide an HTTP header for the value of the content attribute.
- ➔ The value in the content attribute is the number of seconds, you want the page to redirect after.
- ➔ For immediate loading, you can set it to 0. Some browsers don't correctly render the refresh tag.
- ➔ There the user may see a message before the next page loading.
- ➔ Some old browsers don't correctly refresh when you add such quick links. In that case, you should add an anchor link to let the user follow to next page.

munotes.in

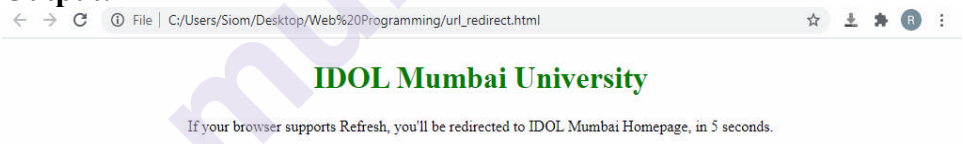
Syntax:

<meta http-equiv = "refresh" content = " time ;url = l

Example:

```
<! DOCTYPE html>
<html>
<head>
<title>HTML Redirect</title>
<meta http-equiv="refresh"
content="5; url = https://old.mu.ac.in/distance-open-learning/" />
</head>

<body>
<h1 style="text-align: center; color: green ;">
        IDOL Mumbai University
</h1>

<p style="text-align: center;">
        If your browser supports Refresh,
        You'll be redirected to IDOL Mumbai
Homepage, in 5 seconds.
</p>
</body>
</html>
```

Output:



Supported Browsers: The browsers supported by **HTML Redirect** are listed below:

- Google Chrome
- Internet Explorer
- Firefox
- Safari
- Opera

3.7 SUMMARY

➔ The **html <nav> element** represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents.

➔ Common examples of navigation sections are menus, tables of contents, and indexes.

3.8 BIBLIOGRAPHY

1.https://learning.oreilly.com/library/view/microsoft-html5-step/9780735656543/ch10s03.html

2.https://www.geeksforgeeks.org/html-mapping-image/

3.https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav

3.9 UNIT END EXERCISE

1. Explain how to create text based top navigation bar in details.
2. Create text based bottom navigation bar of four link.
3. Write a short note on graphical based navigation bar.
4. Explain image mapping concept in details with examples.
5. Explain how to redirect from one url to another url.

❖❖❖❖

munotes.in

4

LAYOUTS

Unit Structure :

4.0 Objectives
4.1 Introduction
4.2 HTML5 Semantic Tags
4.3 Creating Divisions
4.4 Creating HTML5 Semantic Layout
4.5 Positioning and Formatting Divisions
4.6 Summary
4.7 Bibliography
4.8 Unit End Exercise

4.0 OBJECTIVES

- Understand tags and the new HTML semantic tags in breaking a page into sections.
- Learn to place the division in a specified location within your page.
- Create a division-based layout to your web page
- Create a HTML5 Semantic layout to your web page

4.1 INTRODUCTION

A web page being rendered in the browser consists of many things - logo, informative text, pictures, hyperlinks, navigational structure and more. HTML5offers a set of mark-up elements that allow you to create a structured layout for web pages. These elements are often termed as Semantic Mark-up because they convey their meaning and purpose clearly to the developer and to the browser.

4.2 HTML5 SEMANTIC TAGS

Semantic HTML tags are those that clearly describe their meaning in a human- and machine-readable way.

Tags such as **<header>, <footer>** and **<article>** are all considered semantic because they accurately describe the purpose of the tags and the type of content that is inside them.

munotes.in

List of new semantic tags:
The semantic tags added in HTML5 are:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>

<article>: It contains independent content which doesn't require any other context.

Example: Blog Post, Newspaper Article etc.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Article Tag</title>
    <style>
        h1 {
                Color: #006400;
                font-size:50px;
                Text-align: left;
                }
         p {
                font-size:25px;
                text-align: left;
                margin-top: -40px;
                }
    </style>
  </head>
  <body>
    <article>
      <h1>IDOL </h1>
      <p>A Distance and Open Learning School for students</p>
    </article>
  </body>
</html>
```

Output:

IDOL

A Distance and Open Learning School for students

In this way we can use other semantic tags for accomplishment of our purpose.

<aside>: It is used to place content in a sidebar i.e. aside the existing content. It is related to surrounding content.

<details> and <summary>: "details" defines additional details that the user can hide or view. "summary" defines a visible heading for a "details" element.

<figure> and <figcaption>: These are used to add an image in a web page with small description.

<header>: As the name suggests, it is for the header of a section introductory of a page. There can be multiple headers on a page.

<footer>: Footer located at the bottom of any article or document, they can contain contact details, copyright information etc. There can be multiple footers on a page.

<main>: It defines the main content of the document. The content inside main tag should be unique.

<section>: A page can be split into sections like Introduction, Contact Information, Details etc and each of these sections can be in a different section tag.

<nav>: It is used to define a set of navigation links in the form of navigation bar or nav menu.

<mark>: It is used to highlight the text.

4.3 CREATING DIVISIONS

HTML provides you a lot of ways to layout your page better. Web designers today are geared towards a more appealing way of separating content into sections which is called the division-based layout.

What is tag?

- ➔ The tag is used to establish separate divisions or areas of your page.
- ➔ The tag is used to group block-elements to format them with CSS.
- ➔ The **div tag** is known as Division tag.
- ➔ The div tag make divisions of content in the web page like (text, images, header, footer, navigation bar, etc).
- ➔ The div tag has both open(**)** and closing (**)** tag

- ➔ It is mandatory to close the tag.
- ➔ Most websites have put their content in multiple columns, multiple columns can be created by using ** ** tags.

Advantages of the Division-based Layout:

- ➔ You can place layout styles externally, and then make the style changes to many pages at once simply by modifying the style sheet.
- ➔ It reduces the number of lines of code needed to produce a page.

The id attribute:

- ➔ The id attribute can be used with any tag element. When used along with the tag, it will identify and define an area of your page.
- ➔ As a unique identifier, the id's value should only be used once on your page.
- ➔ Start the **id** value with a letter followed by any of these: letters, number digits (1- 9), dashes (-) and underscore (_). Don't use space or special characters.
- ➔ The value is case-sensitive.

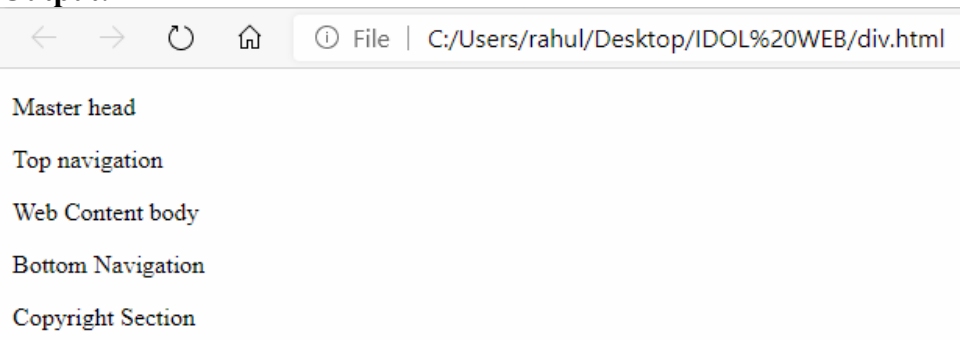
** **is different from ** **

Example: Creating Divisions in Web Page

```
<!DOCTYPE html>
<html>
<head>
<title>Web Page Divisions</title>
</head>
<body>
<div id="masterhead">
        <p>Master head </p>
</div>
<div id="top-navigation">
        <p>Top navigation </p>
</div>
        <div id="web-contentbody">
        <p>Web Content body </p>
</div>
<div id="bottom-navigation">
        <p>Bottom Navigation </p>
</div>
<div id="copyright-section">
        <p>Copyright Section </p>
</div>
</body>
</html>
```

munotes.in

Output:



The output does not show a layout. You can clearly see how your page is divided into different sections.

4.4 CREATING HTML5 SEMANTIC LAYOUT

A web page being rendered in the browser consists of many things - logo, informative text, pictures, hyperlinks, navigational structure and more.

HTML5offers a set of markup tags that allow you to create a structured layout for web pages.

These tags are often termed as Semantic Markup because they convey their meaning and purpose clearly to the developer and to the browser.

HTML5 includes a set of markup tags have meaningful names so that just by looking at these elements you get a clear idea about their content. These semantic tags of HTML5 we already discussed, now we focusing how to make layout by using it.

The following figure shows a sample page layout designed using these elements: This figure shows a typical arrangement of various elements. Note that their exact location on a page is purely dependent on the layout. For example, the <aside> element can be placed on the left-hand side of the <section> or even above or below it.

munotes.in

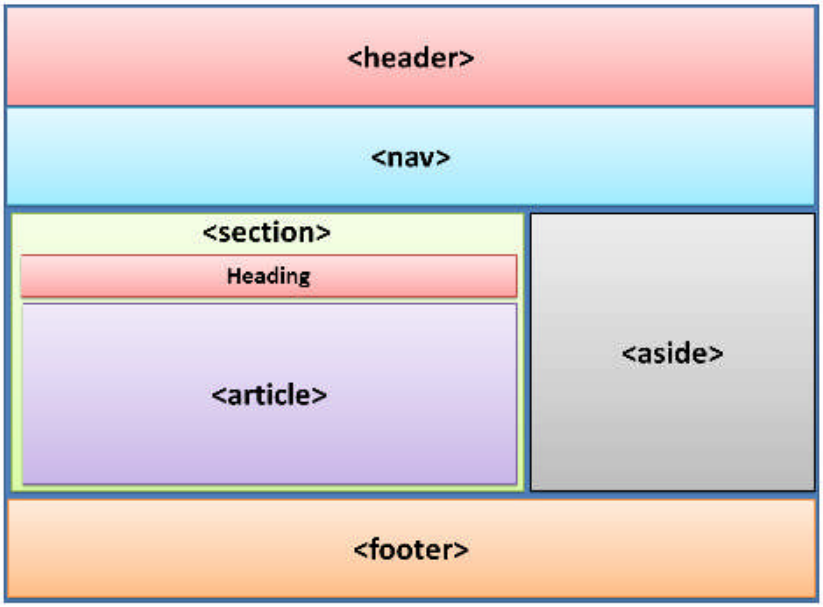


Figure 4.1 Semantic tags layout

Example: Complete Semantic layout of web page

```
<!DOCTYPE html>
<html>
<head>
<title> HTML5 Semantic Layout</title>
<link href="StyleSheet.css" rel="stylesheet" />
</head>
<body>
<header>
<h1>This is page heading</h1>
</header>
<nav>
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">About Us</a></li>
<li><a href="#">Contact Us</a></li>
</ul>
</nav>
<article>
<h1>This is article heading</h1>
<p>
        Hi! I am in article section

</p>
</article>
<aside>
<figure>
<imgsrc="Images/laptop.jpeg" height="100px" width="60px" />
<figcaption>Figure caption goes here</figcaption>
```

munotes.in

```
</figure>
<p>
        I aside tag
</p>
</aside>
<section>
<h1>This is a section heading</h1>
<p>
        Hello world! Hello world! Hello world!

</p>
</section>
<footer>
<hr />
      Copyright (C) 2021. All rights reserved.
</footer>
</body>
</html>
```

Creating CSS for above Example Web Page:

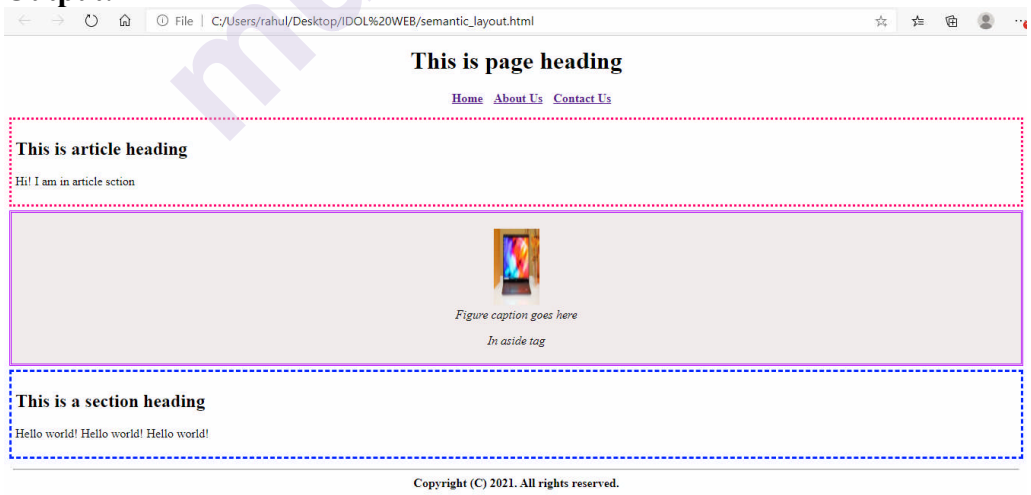
The example web page that you created in the preceding section has **stylesheet.css** linked with it.

```
article
{
   padding:5px;
   border: dotted 3px #ff006e;
   margin-top:5px;
}

header
{
   padding:0px;
   text-align: center;
}

aside
{
   margin-top:5px;
   background-color: #f0eaea;
   padding:5px;
   text-align: center;
   font-style: italic;
   border: double 3px #b200ff;
}
```

munotes.in

```
section
{
   padding:5px;
   border: dashed 3px #0026ff;
   margin-top:5px;
}

footer
{
   padding:5px;
   text-align: center;
   font-weight: bold;
}

nav
{
   text-align: center;
}
ul li
{
   display: inline;
   padding-left:5px;
   padding-right:5px;
   text-align: left;
   font-size:16px;
   font-weight: bold;
}
```

Output:



munotes.in

4.5 POSITIONING AND FORMATING DIVISIONS

If you insert an additional **<aside>**division to your web page layout, and place your division beside another either to the right or left, you can easily use the float style rule.

Positioning Style –

If you want to place a division on a specific spot of your page, you can use the position style rule.

There are Three possible values for a position style rule:

1. Static - position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page.

2. **Absolute value– position: absolute;**

This value specifies a fixed position with the respect to its parent element or tag. Usually, the tag as the parent element.

3. Relative value – position: relative,

This value is compensated from the element's natural position. Other elements on the page are not affected, even if the new position cause overlap.

4. Fixed value - position: fixed;

This value specifies a fixed position within the browser window that doesn't change even when the display is scrolled up or down.

Position Style:

You must use each of these values in combination with top, right, bottom, and /or left style rule that specifies the location to which the position style rules refers.

Example: To position a section at exactly 100px from the top of the page and 200px from the left side, the syntax should be:

```
<section style= "top:100px; left:200px; position: absolute">
```

Horizontal Lines

Horizontal Lines can be useful as a visual divider between sections of text in a web page.

You can add a horizontal line after your header then add another horizontal line before your footer.

This will clearly show your division in your web page.

You can simply add the horizontal line <hr> (horizontal rule) tag where you want the line to appear.

munotes.in

4.6 SUMMARY

This chapter covered semantic mark-up elements of HTML5 that can be used in the proper layout of the web pages. Unlike the element, the semantic elements are intended to be used for a specific purpose. The elements we covered in this article include <header>, <nav>, <footer>, <section>, <article> and <aside>.as well as we covered the div tags with positioning.

4.7 BIBLIOGRAPHY

1. https://www.freecodecamp.org/news/semantic-html5-elements/
2. https://www.geeksforgeeks.org/html5-semantics/
3.https://www.slideshare.net/grayzon21/html5-create-divisions-in-a-web

4.8 UNIT END QUESTIONS

1. Create the structure of from pre-designed websites using HTML5 semantic elements.
2. ExplainHTML5 Semantic Tags in details.
3. Explain positioning in divisions with examples.
4. What is division tags and explain how to use it?
5. Explain difference between tag and semantic tag in details with examples.

❖❖❖❖

munotes.in

5

TABLES

Unit Structure

5.0 Objectives
5.1 Introduction
5.2 Creating simple table
5.3 Specifying the size of the table
5.4 Specifying the width of the column
5.5 Merging table cells
5.6 Using tables for page layout
5.7 Applying table borders
5.8 Applying background and foreground fills
5.9 Changing cell padding
5.10 Spacing and alignment
5.11 Summary
5.12 Bibliography
5.13 Unit End Exercise

5.0 OBJECTIVES

- To create basic tables
- Understand some arguments with table Height, width, border, Cell text alignment.
- To gain basic familiarity with HTML tables.
- Understand how to apply background and foreground to tables.

5.1 INTRODUCTION

You may want to consider using HTML tables in your website. In addition to creating HTML tables to present data in rows and columns, 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. Coding HTML tables into your web page is fairly easy since you need only understand a few basic table codes.

munotes.in

5.2 CREATING SIMPLE TABLES

The basic structure of an HTML table consists of the following tags:

Table tags: <table></table>

Row tags: <tr></tr>

Cell tags: <td></td>

- Constructing an HTML table consists of describing the table between the beginning table tag, <table>, and the ending table tag, </table>.
- Between these tags, you then construct each row and each cell in the row.
- To do this, you would first start the row with the beginning row tag, <tr>, and then build the row by creating each cell with the beginning cell tag, <td>, adding the data for that cell, and then closing the cell with the ending cell tag, </td>.
- When you finish all of the cells for a row, you would then close the row with the ending row tag, </tr>.
- Then, for each new row, you would repeat the process of beginning the row, building each cell in the row, and closing the row.

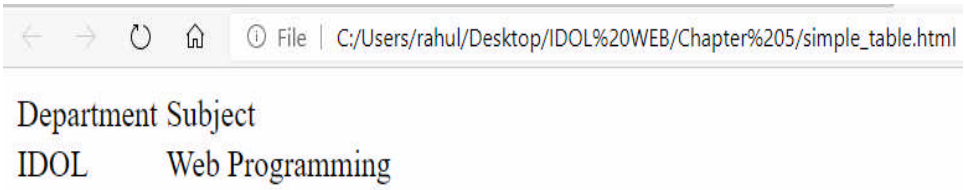
Example:
The following table is an example of a basic table with two rows and two columns of data.

Html code:

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
        <body>
                <table>
                        <tr>
                                <td>Department</td>
                                <td>Subject</td>
                        </tr>
                        <tr>
                                <td>IDOL</td>
                                <td>Web Programming</td>
                        </tr>

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

munotes.in

Output:



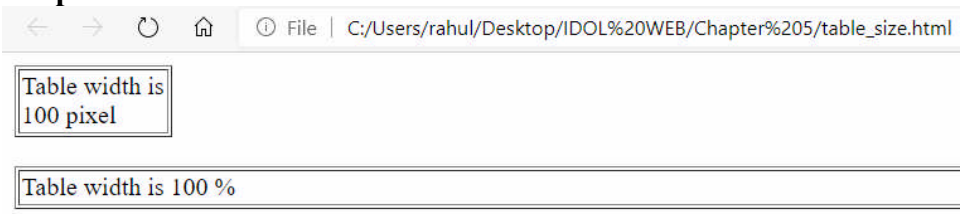
5.3 SPECIFYING THE SIZE OF THE TABLE

- The width attribute specifies the width of a table. The width can be set either as an absolute value in pixels, or as in percentage (%).
- If the width attribute is not set, it will take up the space of longest single word in each cell.
- Table width in pixels:<table width=100>
- Table width in percentage (%): <table width=100%>

Example:
Html code:

```
<html>
<body >
<table width=100>
<tr>
<td>
      Table width is 100 pixels
</td>
</tr>
</table>
<br>
<table width=100%>
<tr>
<td>
      Table width is 100 %
</td>
</tr>
</table>
</body>
</html>
```

Output:



- The above HTML code display two tables, one is 100-pixel width and another one is 100% width.

- First table is only 100-pixel width in any changes in browser window state, while other table will always stretch the full width of the window.

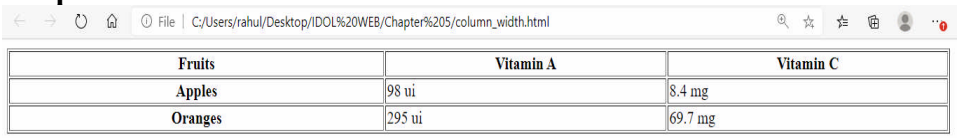
5.4 SPECIFYING THE WIDTH OF THE COLUMN

- We use the <col> element to specify the width of column.
- This element is to be placed in the <table> tag
- we use the style attribute to define the width of the columns.
- The <col> element is a self-closing element and we need one for every columns of table.

Example:
Html code

```
<html>
        <head>
        </head>
        <body>
                <table border="1" width="100%">
                        <col style="width:40%">
                        <col style="width:30%">
                        <col style="width:30%">
                        <tr>
                                <th>Fruits</th>
                                <th>Vitamin A</th>
                                <th>Vitamin C</th>
                        </tr>
                        <tr>
                                <th>Apples</th>
                                <td>98 ui</td>
                                <td>8.4 mg</td>
                        </tr>
                        <tr>
                                <th>Oranges</th>
                                <td>295 ui</td>
                                <td>69.7 mg</td>
                        </tr>
                </table>
        </body>
</html
```

Output:

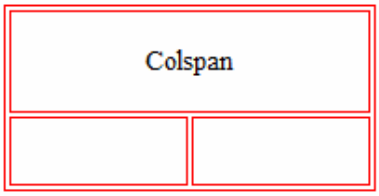


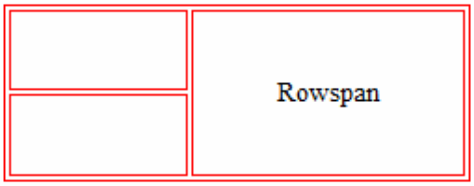
Fruits	Vitamin A	Vitamin C
Apples	98 ui	8.4 mg
Oranges	295 ui	69.7 mg

munotes.in

5.5MERGING TABLE CELLS

- We can merge two or more table cells in a column by using the **colspan** attribute in a <td> HTML tag (table data).
- To merge two or more row cells, use the **rowspan** attribute.





Colspan:

- The colspan attribute defines the number of columns a cell should span (or merge) horizontally.
- That is, we merge two or more Cells in a row into a single Cell.

<td colspan=2 >

The above code will merge two Cells as one Cell horizontally.

Before Colspan

First Cell	Second Cell
Third Cell	Forth Cell

After Colspan

Merged	
Third Cell	Forth Cell

HTML code:

```
<html>
        <body >
                <table border=1 >
                        <tr>
                                <td colspan=2 >Merge</td>
                        </tr>
                        <tr>
                                <td>Third Cell</td>
                                <td>Forth Cell </td>
                        </tr>
                </table>
        </body>
</html>
```

- Colspan (Column Span) merged Cells horizontally that is from left to right.
- The default value of Colspan is 1.

Rowspan:

- The rowspan attribute specifies the number of rows a cell should span vertically.

munotes.in

- That is, we merge two or more Cells in the same column as a single Cell vertically.

<td rowspan=2 >

The above code will merge two Cells as one Cell vertically.

Before Rowspan

First Cell	Second Cell
Third Cell	Forth Cell

After Rowspan

First Cell	Merged
Third Cell	

HTML code:

```
<html>
        <body >
        <table border=1 >
                <tr>
                        <td>First Cell</td>
                        <td rowspan=2 >Merged</td>
                </tr>
                <tr>
                        <td valign=middle>Third Cell </td>
                </tr>
        </table>
        </body>
</html>
```

➔Rowspan merged Cells vertically, that is from top to bottom.

5.6 USING TABLES FOR PAGE LAYOUT

- HTML tables are most often used to define the layout of an entire Web page.
- If we want to design a page that displays text in newspaper style columns, or separates the page into distinct sections, we'll find tables an essential and useful tool.

Table Layout of a Web Page

a sample table layout of a Web page.

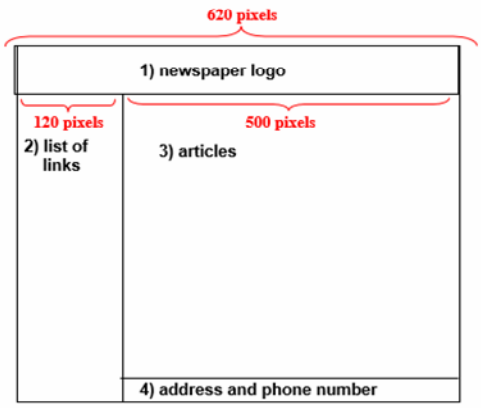


Figure 5.1 table layout

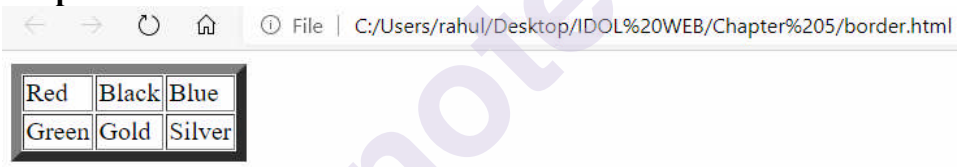
5.7 APPLYING TABLE BORDERS

- By default, tables that you create do not have visible borders (a line around the table that visually defines the table).
- Borders can help to make your table stand out more and adds visual interest.
- After creating an HTML table, you should add a border to it,

Add Table Borders Using HTML5

```
<table border=6px>
        <tr>
        <td>Red</td>
        <td>Black</td>
        <td>Blue</td>
        </tr>
        <tr>
        <td>Green</td>
        <td>Gold</td>
        <td>Silver</td>
        </tr>
</table>
```

Output:



5.8 APPLYING BACKGROUND AND FOREGROUND FILLS

- Each table, row, and cell is its own distinct area, and each can have its own background.
- For example, you might want to apply a different color background to a heading row to make it stand out, or change the color of every other line in a listing to help visitors track a line across the table.
- Table elements support the **bgcolor** attribute.
- To specify a background color for all of the cells in a table, all of the cells in a row, or for individual cells, by adding the bgcolor attribute to either the **<table>**, **<tr>**, **<td>**, or **<th>** tags as follows:
  <table bgcolor= "color">
  <tr bgcolor= "color">
  <td bgcolor= "color">
  <th bgcolor= "color">
- ***color*** is either a color name or hexadecimal color value

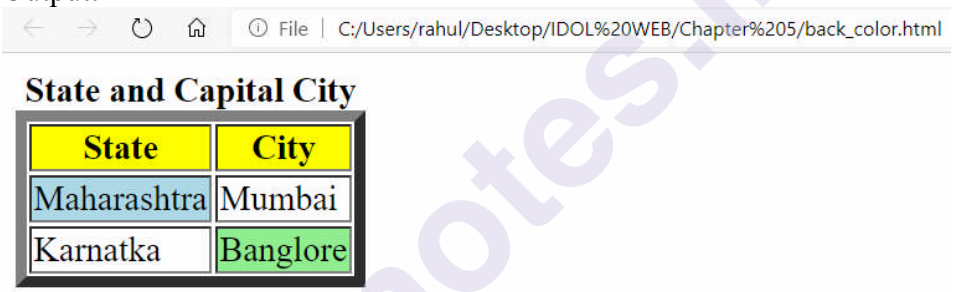
munotes.in

Specifying Table, Row, and Cell Colors:

Html code

```
<html>
<table border="5" bgcolor="white">
        <caption><b>State and Capital City</b></caption>
        <tr bgcolor="yellow">
                <th > State</th>
                <th> City</th>
        </tr>
        <tr >
                <td bgcolor="lightblue" > Maharashtra</td>
                <td> Mumbai</td>
        </tr>
        <tr >
                <td > Karnataka</td>
                <td bgcolor="lightgreen"> Bangalore</td>
        </tr>
</table
</html>
```

Output:



The bordercolor Attribute:

- By default, table borders are displayed in two shades of gray that create a three-dimensional effect.
- The syntax for the bordercolor attribute is:

<table bordercolor= "color">

color is an HTML color name or hexadecimal color value

Applying a Table Background:

- Add a background image to your tables using the **background** attribute.
- A background can be applied to the entire table or to a cell.

```
<table background=" parch.jpg">
```

Applying a Table foreground:

- Foreground means we can apply text font color by using **style** attribute

```
<table style="color: blue;">
<trstyle="color: green;">
<tdstyle="color: red;">
```

munotes.in

Example:

```
<html>
<table border="7" bordercolor="lightblue" background="parch.jpg" >
        <caption><b>State and Capital City</b></caption>
        <tr style="color: blue;">
                <th > State</th>
                <th> City</th>
        </tr>
        <tr >
                <td > Maharashtra</td>
                <td> Mumbai</td>
        </tr>
        <tr >
                <td > Karnataka</td>
                <td style="color: red;"> Bangalore</td>
        </tr>
</table
</html>
```

Output:

5.9 CHANGING CELL PADDING

- To control the space between the table text and the cell borders, add the **cellpadding** attribute to the table tag.
- The syntax for this attribute is:
  <table cellpadding="value">
  - ***value*** is the distance from the table text to the cell border, as measured in pixels
  - the default cell padding value is 1 pixel
- Cell padding refers to the space within the cells.

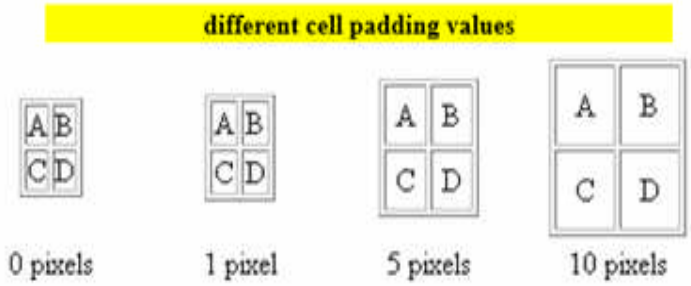


Figure 5.2 Cell padding values

5.10 PACING AND ALIGNMENT

- The **cellspacing** attribute controls the amount of space inserted between table cells.
- The syntax for specifying the cell space is:

  <table cellspacing= "value">
  - ***value*** is the width of the interior borders in pixels
  - the default cell spacing is 2 pixels
  - Cell spacing refers to the space between the cells.

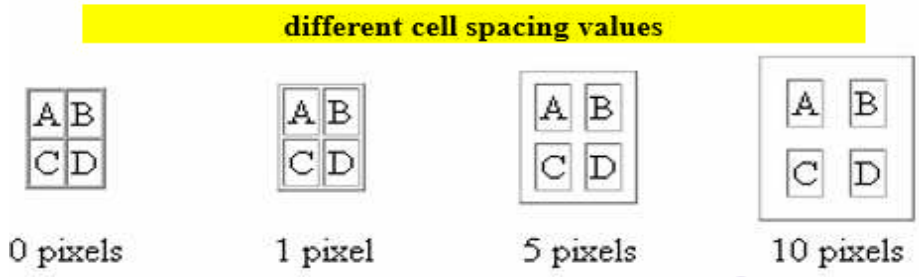
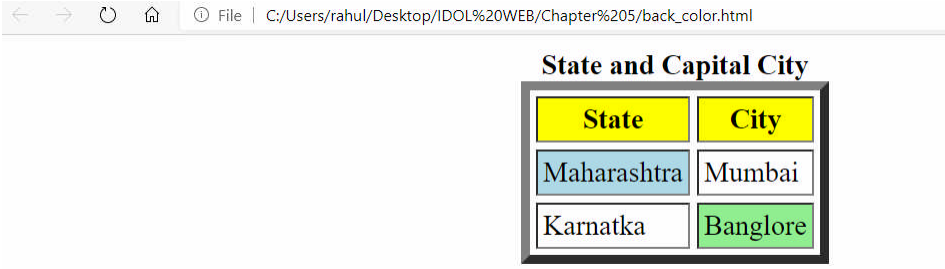


Figure 5.3 Cell spacing values

Example of Cell Padding, pacing and alignment:

```
<html>
<table border="5" bgcolor="white"cellspacing="4" cellpadding="3" align="center">
        <caption><b>State and Capital City</b></caption>
        <tr bgcolor="yellow">
                <th > State</th>
                <th> City</th>
        </tr>
        <tr >
                <td bgcolor="lightblue" > Maharashtra</td>
                <td>Mumbai</td>
        </tr>
        <tr >
                <td > Karnataka</td>
                <td bgcolor="lightgreen"> Bangalore</td>
        </tr>
</table>
</html>
```

Output:



5.11 SUMMARY

- Use the HTML <table> element to define a table.
- Use the HTML <tr> element to define a table row.
- Use the HTML <td> element to define a table data.
- Use the HTML <th> element to define a table heading.
- Use the HTML <caption> element to define a table caption.
- Use the CSS border property to define a border.
- Use HTML <table width> to define size of table
- Use HTML <table cellpadding>,<table cellspacing> to define space within cell and between cell.

5.12 BIBLIOGRAPHY

1.https://its.temple.edu/creating-tables-html

2.http://www.corelangs.com/html/tables/width-height.html

3.https://htmlreference.io/tables/

4.http://www.corelangs.com/html/tables/colspan-rowspan.html

5.https://www.teachucomp.com/add-table-borders-using-html5/#:~:text=To%20assign%20a%20border%2C%20insert,and%20not%20the%20individual%20cells.

5.13 UNIT END QUESTIONS

1. What is the use of rowspan and colspan attribute?
2. What is the importance of table tag in web
3. What are the attributes that can be used in table tag
4. Explain cell padding and cell spacing in details with example.
5. Explain background and foreground attributes of table with examples.

munotes.in

6

FORMS AND MEDIA

Unit Structure

6.0 Objectives
6.1 Introduction
6.2 Creating basic form
6.3 Using check boxes and option buttons
6.4 Creating lists
6.5 Additional input types in HTML5
6.6 Audio and Video in HTML5
6.7 HTML multimedia basics
6.8 Embedding video clips
6.9 Incorporating audio on web page
6.11 Summary
6.12 Bibliography
6.13 Unit End Exercise

6.0 OBJECTIVES

1. To gain familiarity with what web forms are, what they are used for, how to think about designing them, and the basic HTML elements we'll need for simple cases.
2. To understand the newer input type values available to create native form controls, and how to implement them using HTML.
3. To learn how to embed video and audio content into a webpage, and add captions/subtitles to video.

6.1 INTRODUCTION

The first article in our series provides you with your very first experience of creating a web form, including designing a simple form, implementing it using the right HTML form controls and other HTML elements, adding some very simple styling via CSS, and describing how data is sent to a server. We'll expand on each of these subtopics in more detail later on in the module.

munotes.in

6.2 CREATING BASIC FORMS

What are web forms?

- **Web forms** are one of the main points of interaction between a user and a web site or application.
- Forms allow users to enter data, which is generally sent to a web server for processing and storage.
- A web form's HTML is made up of one or more **form controls**, plus some additional elements to help structure the overall form — they are often referred to as **HTML forms**.
- The controls can be single or multi-line text fields, dropdown boxes, buttons, checkboxes, or radio buttons, and are mostly created using the <input> element.

Designing your form:

- Before starting to code, it's always better to step back and take the time to think about your form.
- Designing a quick mockup will help you to define the right set of data you want to ask your user to enter.
- We'll build a simple contact form. Let's make a rough sketch.

- Our form will contain three text fields and one button.
- We are asking the user for their name, their e-mail and the message they want to send.
- Hitting the button will send their data to a web server.

Implementing our form HTML:

- Creating the HTML for our form. We will use the following HTML elements: <form>, <label>, <input>, <textarea>, and <button>.

- All forms start with a <form> element, like this:

  <form action="/my-handling-form-page" method="post">

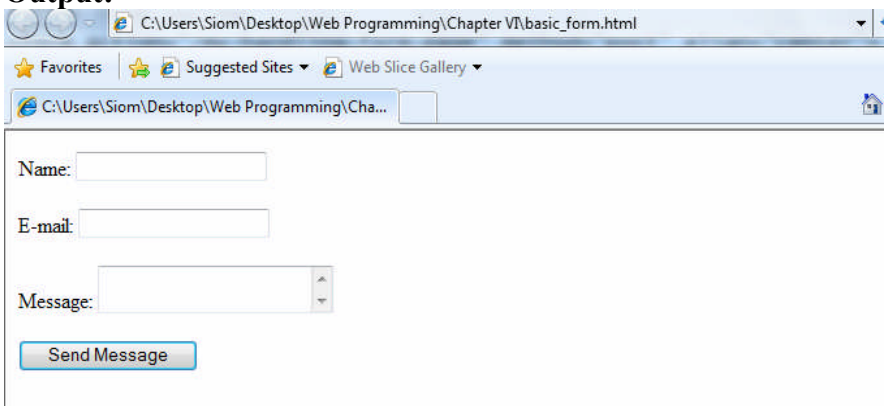
  </form>

- This element formally defines a form.
- The action attribute defines the location (URL) where the form's collected data should be sent when it is submitted.
- The method attribute defines which HTTP method to send the data with (usually get or post).
- The data entry portion contains three text fields, each with a corresponding <label>:
- The input field for the name is a single-line text field.
- The input field for the e-mail is an input of type email: a single-line text field that accepts only e-mail addresses.
- The input field for the message is a <textarea>; a multiline text field.

Example: creating basic form

```
<form action="/my-handling-form-page" method="post" align="center">

<label for="name">Name :< /label>
<input type="text" id="name" name="user_name"><br><br>

<label for="mail">E-mail :< /label>
<input type="email" id="mail" name="user_email"><br><br>

<label for="msg">Message :< /label>
<textarea id="msg" name="user_message"></textarea><br><br>

<button type="submit">Send Message</button>
</form>
```

Output:



munotes.in

6.3 USING CHECK BOXES AND OPTION BUTTONS

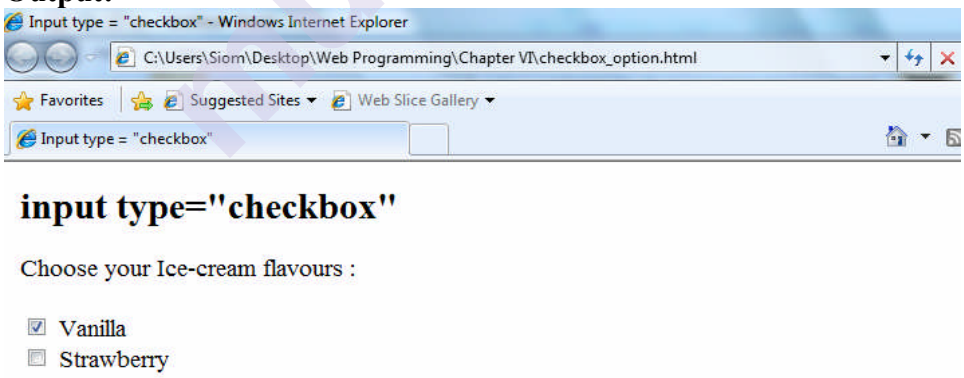
<input type="checkbox">:

- The <input type="checkbox"> defines a checkbox.
- The checkbox is shown as a square box that is ticked (checked) when activated.
- Checkboxes are used to let a user select one or more options of a limited number of choices.

Example:

```
<html>
<head>
<title> Input type = "checkbox" </title></head>
<body>
        <h2> input type="checkbox" </h2>
        <p>Choose your Ice-cream flavours :</p>
        <div>
        <input type="checkbox" id="Vanilla" name="Vanilla"checked>
        <label for="Vanilla">Vanilla</label>
        </div>
        <div>
        <input type="checkbox" id="horns" name="Strawberry">
        <label for="Strawberry">Strawberry</label>
        </div>
</body>
</html>
```

Output:



Option Buttons:

- Option buttons are sometimes called Radio Buttons, and they force the user into choosing only one item in a list, such as a Male/Female option, or selecting a payment method.
- The Option button HTML look like this:

```
<INPUT TYPE="Radio" Name="Gender" Value="Male">
```

Male
<INPUT TYPE="Radio" Name="Gender"
Value="Female">Female

- After typing the INPUT tag, the word TYPE comes next. For Option Buttons, the type is "Radio".
- The NAME is definitely needed here, and note that the NAME for both in our code above is "Gender".
- We use the same name for each group of option buttons we are adding to your form.
- So if you wanted payment option buttons, the code might be this:

  <INPUT TYPE="Radio" Name="payment" Value="CC">
  Credit Card
  <INPUT TYPE="Radio" Name="payment" Value="DC">
  Debit Card
  <INPUT TYPE="Radio" Name="payment" Value="PP">
  PayPal

- This time, each radio button has the name "payment".
- The reason you keep the same name for each group of option buttons is simply to distinguish one group of option buttons from another.
- The VALUE attribute is quite useful.
- When the user submits the form to us using the Submit button, these VALUES are going to be returned.
- If we just got Radio1 and Radio2, you won't know (or won't remember, more likely) which option the user has selected.
- Manipulating values with scripts is also a lot easier if the Value is the same as the text the user sees.
- If you want to have a default option button selected, use the word "Checked":

  <INPUT TYPE="Radio" Name="payment" Value="CC">
  Credit Card
  <INPUT TYPE="Radio" Name="payment" Value="DC">
  Debit Card
  <INPUT TYPE="Radio" Name="payment"
  Value="PP" **Checked**>PayPal

- Attaching a label to each button is very useful for your visitors.
- When the label is clicked it will select the option button it is named for:

Example:

<html>

<head>

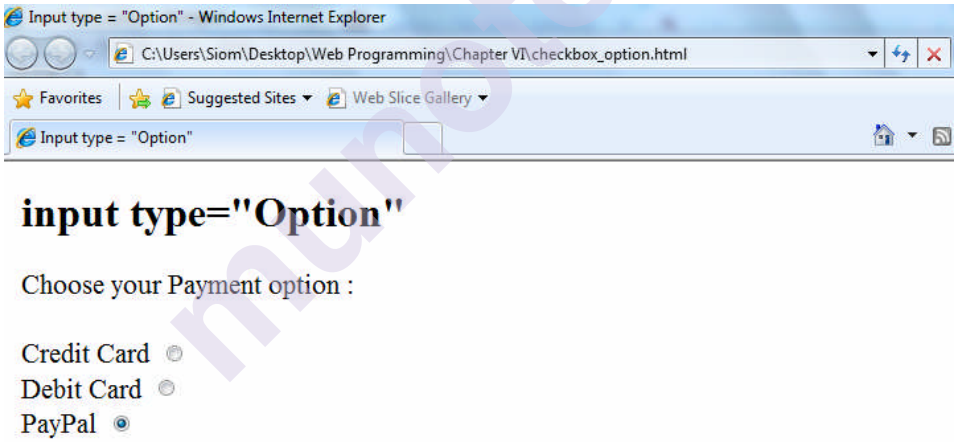
<title> Input type = "Option" </title></head>

<body>

<h2> input type="Option" </h2>

```
<p>Choose your Payment option :</p>
<div>
<LABEL FOR="R1">Credit Card</LABEL>
<INPUT TYPE="Radio" Name="payment" ID="R1"
                    Value="Credit Card">
</div>
<div>
<LABEL FOR="R2">Debit Card</LABEL>
<INPUT TYPE="Radio" Name="payment" ID="R2" Value="Debit
Card">
</div
<div>
<LABEL FOR="R3">PayPal</LABEL>
<INPUT TYPE="Radio" Name="payment" ID="R3" Value="PayPal"
Checked>
</div>
</body>
</html>
```

Output:



6.4 CREATING LIST

HTML offers three ways for specifying lists of information.
All lists must contain one or more list elements.

The types of lists in HTML are :

- **ul:** An unordered list. This will list items using plain bullets.
- **ol:** An ordered list. This will use different schemes of numbers to list your items.
- **dl:** A definition list. This arranges your items in the same way as they are arranged in a dictionary.

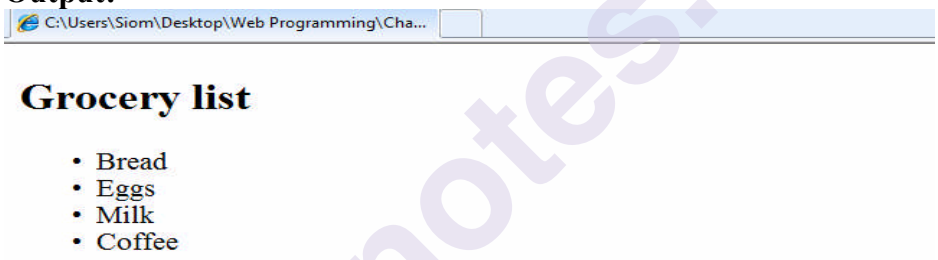
The Unordered HTML List:

- An unordered list starts with the “ul” tag.
- Each list item starts with the “li” tag.
- The list items are marked with bullets i.e small black circles by default.

Example:

```
<!DOCTYPE html>
<html>
<body>
 <h2>Grocery list</h2>
  <ul>
        <li>Bread</li>
         <li>Eggs</li>
        <li>Milk</li>
        <li>Coffee</li>
</ul>
  </body>
</html>
```

Output:



The HTML Ordered List:

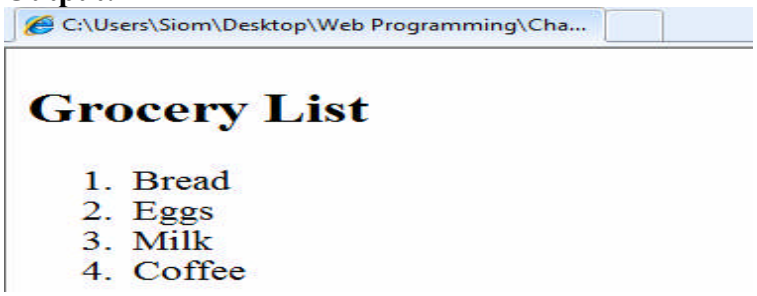
- An ordered list starts with the “ol” tag.
- Each list item starts with the “li” tag.
- The list items are marked with numbers by default.

Example:

```
<! DOCTYPE html>
<html>
<body>
  <h2>Grocery List</h2>
  <ol>
  <li>Bread</li>
  <li>Eggs</li>
  <li>Milk</li>
  <li>Coffee</li>
</ol>

</body>
</html>
```

munotes.in

Output:



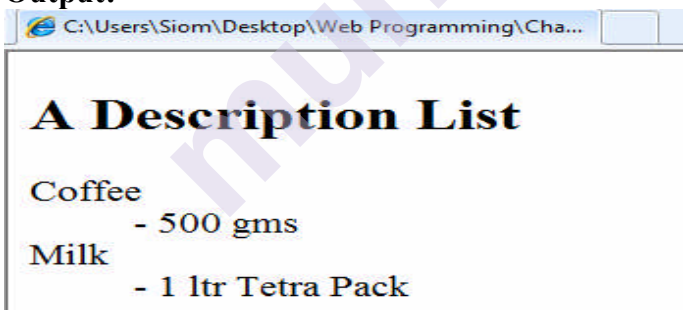
The HTML Description List:

- A description list is a list of terms, with a description of each term.
- The <dl> tag defines the description list, the <dt> tag defines the term name, and the <dd> tag describes each term.

Example:

```
<! DOCTYPE html>
<html>
<body>
<h2>A Description List</h2>
<dl>
        <dt>Coffee</dt>
        <dd>- 500 gms</dd>
        <dt>Milk</dt>
        <dd>- 1 ltr Tetra Pack</dd>
</dl>
</body>
</html>
```

Output:



6.5 ADDITIONAL INPUT TYPES IN HTML5

We now look at the functionality of newer form controls which were added in HTML5 to allow collection of specific types of data.

E-mail address field:

<input type="email" id= "email"name="email">

- This type of field is set using the value email for the type attribute:

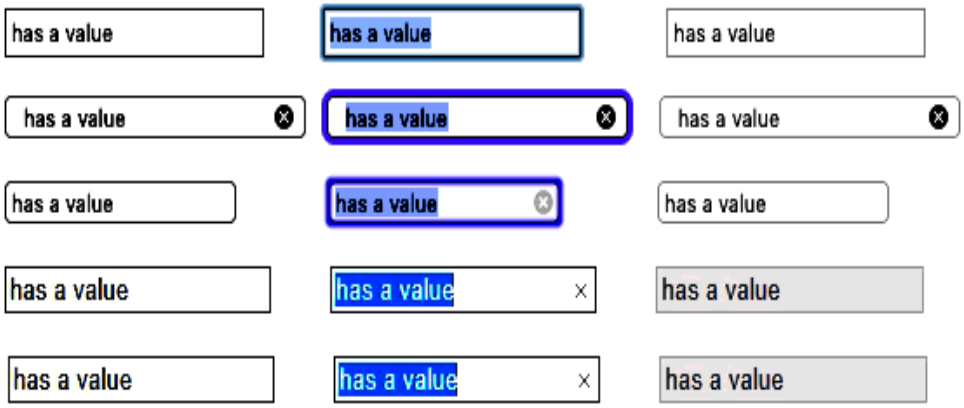
munotes.in

- When this type is used, the user is required to type a valid email address into the field.
- Any other content causes the browser to display an error when the form is submitted. You can see this in action in the below screenshot.

Search field:

- Search fields are intended to be used to create search boxes on pages and apps.
- This type of field is set by using the value search for the type attribute:

  <input type=”search” id= “search” name=” search”>
- The main difference between a text field and a search field is how the browser styles its appearance.
- Often, search fields are rendered with rounded corners; they also sometimes display an "Ⓧ", which clears the field of any value when clicked.
- The below screenshots show a non-empty search field in Firefox 71, Safari 13, and Chrome 79 on macOS, and Edge 18 and Chrome 79 on Windows 10.
- Note that the clear icon only appears if the field has a value, and, apart from Safari, it is only displayed when the field is focused.



munotes.in

Phone number field

A special field for filling in phone numbers can be created using tel as the value of the type attribute:

< input type=”tel”id= “tel”name=” tel”>

- When accessed via a touch device with a dynamic keyboard, most devices will display a numeric keypad when type="tel" is encountered, meaning this type is useful whenever a numeric keypad is useful, and doesn't just have to be used for telephone numbers.

URL field

A special type of field for entering URLs can be created using the value url for the type attribute:

< input type=”url” id= “url”name=” url”>

- It adds special validation constraints to the field.
- The browser will report an error if no protocol (such as http:) is entered, or if the URL is otherwise malformed.

Numeric field:

- Controls for entering numbers can be created with an <input> type of number.
- This control looks like a text field but allows only floating-point numbers, and usually provides buttons in the form of a spinner to increase and decrease the value of the control.
- Let's look at some examples. The first one below creates a number control whose value is restricted to any value between 1 and 10, and whose increase and decrease buttons change its value by 2.

  < input type=”number” id= “number”name=” number” min= “1” max= “10” step= “2” >
- The number input type makes sense when the range of valid values is limited, for example a person's age or height.

Date and time pickers:

- Gathering date and time values has traditionally been a nightmare for web developers.
- HTML date controls are available to handle this specific kind of data, providing calendar widgets and making the data uniform.
- A date and time control is created using the <input> element and an appropriate value for the type attribute, depending on whether you wish to collect dates, times, or both.
- Here's a live example that falls back to <select> elements in non-supporting browsers:

Choose a date and time for your party: mm/dd/yyyy --:-- -- ✓

- <input type="datetime-local"> creates a widget to display and pick a date with time with no specific time zone information.
  < input type= “datetime-local” id= “datetime”name=” datetime”>

6.6 AUDIO AND VIDEO IN HTML5

- We are comfortable with adding simple input types to a webpage, now we start adding video and audio players to your HTML documents!
- In this topic we'll look at doing just that with the <video> and <audio> elements;

The <video> element:
The <video> element allows you to embed a video very easily.

```
<video controls
        <source  src= “sample.mp4”  type= “video /mp3>
        <p> your browser doesn’t support HTML5 video. Here is a <a
href= “sample.mp4”> link to the video</a> instead.<p>

</video>
```

The features of note are:
src
In the same way as for the <img> element, the src (source) attribute contains a path to the video you want to embed.

controls
Users must be able to control video and audio playback (it's especially critical for people who have epilepsy.) You must either use the controls attribute to include the browser's own control interface

The paragraph inside the <video> tags
This is called fallback content — this will be displayed if the browser accessing the page doesn't support the <video> element, allowing us to provide a fallback for older browsers.

The <audio> element:
The <audio> element works just like the <video> element, with a few small differences as outlined below. A typical example might look like so:

```
<audio controls
        <source src= “Sleep.mp3” type= “audio/mp3>
        <p>your browser doesn’t support HTML5 audio. Here is a <a
href= “Sleep.mp3”> link to the audio </a>instead. <p>
</audio>
```

munotes.in

- This takes up less space than a video player, as there is no visual component — you just need to display controls to play the audio. Other differences from HTML video are as follows:
  - The <audio> element doesn't support the width/height attributes — again, there is no visual component, so there is nothing to assign a width or height to.
  - It also doesn't support the poster attribute — again, no visual component

6.7 HTML MULTIMEDIA BASICS

- The term media refers to various means of communicating and disseminating information, such as text, images, graphics, audio, video, and animation.
- All these mediums of communication are collectively termed as multimedia.

What is Multimedia?

- Multimedia comes in many different formats.
- It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.
- Web pages often contain multimedia elements of different types and formats.
- A combination of video and audio files can also be used in websites to gain popularity in terms of viewership or provide information and entertainment to the users.
- HTML helps you to add multimedia files on your website by providing various multimedia tags.
- These tags include AUDIO, VIDEO, EMBED, and OBJECT.
- The AUDIO tag is used to display the audio file on the Web page, whereas the VIDEO tag is used to display the video files on the Web page.
- The EMBED and OBJECT tags display the multimedia files on a Web page as well as embed the files from other websites.

Browser Support:

- The first web browsers had support for text only, limited to a single font in a single color.
- Later came browsers with support for color, fonts, images, and multimedia!

munotes.in

Multimedia Formats:
Multimedia elements (like audio or video) are stored in media files.

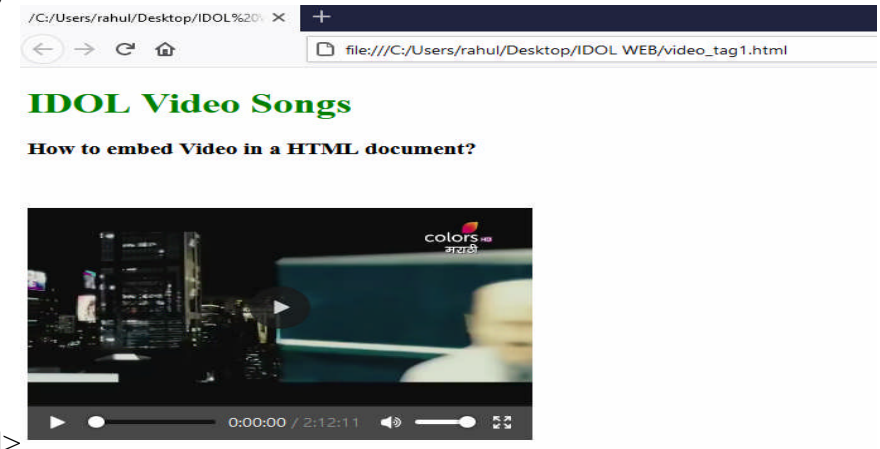
- The most common way to discover the type of a file, is to look at the file extension.
- Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

6.8 EMBEDDING VIDEO CLIPS

- HTML5 has a **native video element** that supports **three video formats** (MP4, WebM and Ogg), making it much easier to embed videos in a webpage.
- We can define the external source for the video using a **file** or a **URL**.
- In HTML5, we can embed a video in your webpage using a pair of <video> tags.
- It is also mandatory to define the **source** for the video.
- we can do it by using a simple src attribute, but it is recommended to choose the <source> tags for that:

Example:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color: green;">
      IDOL Video Songs
</h1>
<h3>
      How to embed Video in a HTML document?
</h3>
<video controls width= "400" height= "300">
<source src="aapla.mp4" type= "video/mp4">
</video>
</body>
```




```
</html>
```

- The HTML5 video player supports three formats, but not all of them have the same level of browser support.
- This means you can add sources in **different formats** to ensure the user can see the video.
- It is always recommended to add **HTML5 video controls** to your player.
- Using special buttons in the player window, the user can manually start and stop the video, skip to specific place using the slider, or toggle between window and full screen video display.
- To add video controls, include the **controls** attribute.

Player Dimensions:

- To define the **size** of your player, you can use the height and width attributes.
- The video will keep the same aspect ratio.
- In the HTML5 video example above, the values are set to 300 for the height and 400 for the width.

6.9 INCORPORATING AUDIO ON WEB PAGE

- Since the release of HTML5, audio can be added to webpages using the <audio> tag.
- Previously audio could be only played on webpages using web plugins like Flash.
- The <audio> tag is an inline element which is used to embed sound files into a web page.
- It is a very useful tag if you want to add audios such as songs, interviews, etc on your webpage.

Syntax:

```
<audio>

<source src="sample.mp3" type="audio/mpeg">

</audio>
```

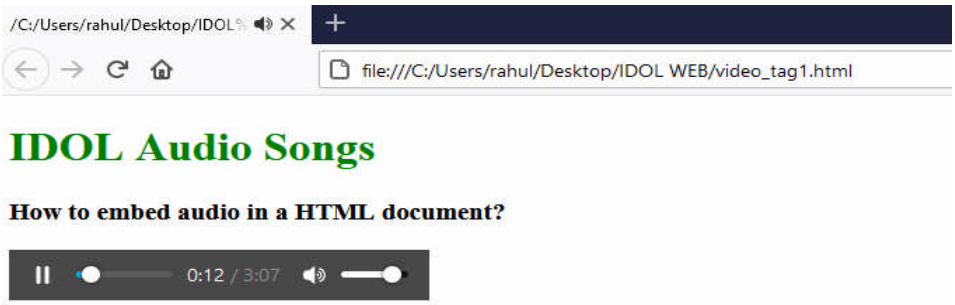
Example:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color: green;">
      IDOL Audio Songs
</h1>
<h3>
      How to embed audio in a HTML document?
</h3>
<audio controls>
```

munotes.in

```
<source src="Test.mp3" type="audio/mp3">
<source src="Test.ogg" type="audio/ogg">
</audio>
</body>
</html>
```

Output:



6.10 SUMMARY

- In this chapter we have studied basic input types of html and new HTML5 input type as well as controls.
- We build the form by using controls and input types.
- We focusing on multimedia to embed video and audio in to web pages.

6.11 BIBLIOGRAPHY

1. https://developer.mozilla.org/en-US/docs/Learn/Forms/Your_first_form
2. https://www.w3schools.com/tags/att_input_type_checkbox.asp
3. https://www.homeandlearn.co.uk/WD/wds8p5.html
4. https://www.tutorialspoint.com/html5/html5_audio_video.htm
5. https://www.bitdegree.org/learn/html5-video-player
6. https://www.geeksforgeeks.org/how-to-embed-audio-element-in-a-html-document/
7. https://codescracker.com/html/html-multimedia-working.htm

6.12 UNIT END QUESTIONS

1. Explain basic html input types?
2. Create Basic Html forms with four fields?
3. Explain HTML5 input types and controls in details?
4. Write short notes on following:
   a. audio tag
   b. video tag
5. Explain how to embed video in web page?
6. Explain how to incorporate audio in web page?

❖❖❖❖

munotes.in

Unit -III

7

JAVASCRIPT

Unit Structure :

7.1 Objective
7.2 Introduction
7.2.1 Client side JavaScript
7.2.2 Server side JavaScript
7.2.3 JavaScript objects
7.2.4 JavaScript security
7.3 Operators
7.3.1 Assignment
7.3.2 Arithmetic
7.3.3 Comparison
7.3.4 Modulus
7.3.5 Increament
7.3.6 Decreament
7.3.7 Unary negation
7.3.8 Logical operators
7.3.9 Short circuit evaluation
7.3.10 String operators
7.3.11 Special operators
7.3.12 Conditional operators
7.3.13 Comma operators
7.3.14 Delete
7.3.15 New
7.3.16 This
7.3.17 Void
7.4 Statements
7.4.1 Break
7.4.2 Comment
7.4.3 Continue
7.4.4 Delete
7.4.5 Do while
7.4.6 Export
7.4.7 For
7.4.8 For in

munotes.in

7.4.9 Function
7.4.10 Ifelse
7.4.11 Import
7.4.12 Labelled
7.4.13 Return
7.4.14 Switch
7.4.15 Var
7.4.16 While
7.4.17 With
7.5 Core java script
7.5.1 Array
7.5.2 Boolean
7.5.3 Object
7.5.3.1Date
7.5.3.2 Math
7.5.4 Number
7.5.5 Object
7.5.6 String
7.5.7 reg Exp

7.0 OBJECTIVE

After reading this chapter you will be able to –

- Write small javascrips with operators and functions.
- Use of variables, data types, control statements.
- Identify client and server side scripts
- Use javascript objects and their inbuilt properties.

7.1INTRODUCTION

- Javascript is object based scripting language based on c++. Scripting language is a light weight programming language which easy to learn and understand. It is generally used for small applications.
- Javascript was basically designed to add interactivity in HTML pages and is directly embedded into HTML.
- Javascript is free to use by anyone.
- Javascript is interpreted language ie is not precompiled before execution. As javascript is interpreted, it is platform independent.
- Java is a full fledged complex programming language developed by sun micro system. Javascript is developed by netscape communications and is no sub language of java.

munotes.in

- Javascript is originally a scripting language developed by European Computer Manufacturer's association (ECMA)
- Javascript that runs at the client side (ie at the client's browser) is client side java script (CCJS) and javascript that runs at the server is serverside java script (SSJS)
- Javascript being object oriented, uses number of built in javascript as well as objects can be created.
- Every object has properties and methods. Property is value(s) associated with an object. Methods are actions associated with an object.
- Example:<scripttype="text/javascript">

  document. write("This message is written by JavaScript");

  </script>

  in above example, 'document' is object and write() is a method of

  document object.
- Javascript runs in a web browser, and when a script written by a third party is executed on the browser, there is a risk of running a spyware or a virusprogram.
- Hence, each time javascript is loaded on the browser implements a security policy designed to minimise the risk of such unknown code.
- Security policy is set of rules governing what scripts can do under which circumstances.
- Modern javascript security is based upon Java. Scripts downloaded are isolated from the operating system and then executed. This is known as the 'sandbox' model. Some scripts are often stored randomly here and there. And hence, many times obtain more power than expected by design or by accident.
- Scripts in general are given limited access and more access is only given with the user consent. Taking a consent for every execution is not a practical solution.
- Scripts from 'trusted' source are many times excluded from this consent procedure.
- A policy called 'same origin' does not block scripts coming from the same origin as trusted scripts. This same origin check is performed on all methods of windows object, also on embedded and externally linked objects.

munotes.in

7.3 JAVASCRIPT OPERATORS

Operator	Name	Description	Example	Result	Note
=	Assignment	Assigns value to a variable.	y = 5; x = y = 5;	y =5 x=5 y=5 and	Primitive data types get direct values where as functions are pointers to the variables.
+=	Plus equal to		x += y	x =10	Same as operator equal to. That is, x+=y means that x=x+y and soon.
-=	Minus to equal		x -= y	x = 0	
*=	Multiply equal to		x *= y	x = 25	
/=	Divide to equal		x /= y	x =1	
%=	Modulo equal to		x %= y	x = 0	
Arithmetic					
+	Addition	Adds values	x = y + 2;	x=7	
-	Subtraction	Subtracts second from first value	x = y – 2;	x=3	
*	Multiplication	Multiplies two values	x = y * 2;	x=10	
/	Division	Divides	x = y / 2;	x=2.5	
%	Modulus	Divides gives remainder and the	x = y % 2;	x=1	
++	Increment	Increments value by 1	x = ++y;	x=6	
--	Decrement	Decrements value by 1	x = --y;	x=4	
Logical					
&&	And	Logically ands	(x<10 y>1) &&	TRUE	As value of x as of now is 4 and that of y is 5
||	Or	Logically Ors	(x==5 y==5) ||	TRUE	As value of x <> 5 but that of y =5

mubnotes.in

!	Not	Negation	!(x==y)	TRUE	As x and y are not equal.
String					
+	Concatenation	Connects two strings.	If x = “5” and y= “5” then	x+y=“55”	
Conditional					
?	Variable =(condition)? value1:value2		If a = 0, b = 7 a=(b= =10)?5:8	a =8, b =7	If condition is satisfied, first value is assigned, else the second value is assigned.
Short circuit evaluation					
&&	Sand	Short circuit and	If x = 0; p = 66 x=p && p.getvalue()	x = 66	If p is not null, then assign value of p to x.
||	SOr	Short circuit or	If x=0, default=5 x=default|| 10	x=5	If there is a default, assign default; else assign the given value.
,	Comma		x=4, y= 5;	x=4 y=5	Executes all expressions from left to right.
Delete		Used in functions	Delete <object>	Object undefined	Deletes the properties from objects and array elements from arrays making themundefined.
This	Can be used in following contexts – • Asglobal • Function context Simplecall As object method Prototype chain • As aconstructor • As getter orsetter.		this.window return this; return this.value		

munotes.in

New		Creates new instance of the object	var x= new Fn()	x becomes new instance of function Fn and gets all its properties and methods.	
Void		Evaluates expression and return s undefined	Void expr	Undefined .	

Javascript Operators Examples 1.

JSOperatorEg.html - Notepad

File Edit Format View Help

```
<body>
<script type="text/JavaScript">
<!--
var two = 2
var ten = 10

document.write("two plus ten = ")
var result = two + ten
document.write(result)
document.write("<br />")


document.write("ten * ten = ")
result = ten * ten
document.write(result)
document.write("<br/>")

document.write("ten / two = ")
result = ten / two
document.write(result)
//-->
</script>
</body>
```

munotes.in

Output

jsOperatorEg.html

file:///E:/arch

Customize Links » Other bookmarks

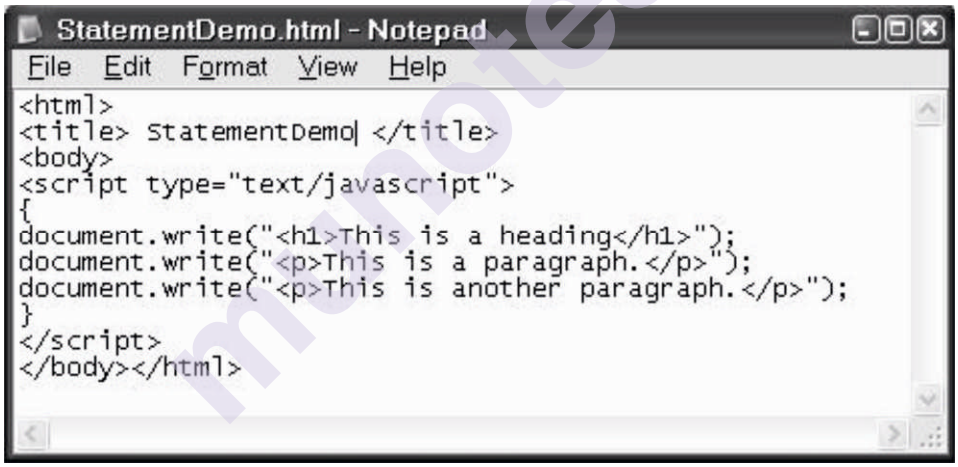
two plus ten = 12

ten * ten = 100

ten / two = 5

7.4 JAVASCRIPT STATEMENTS

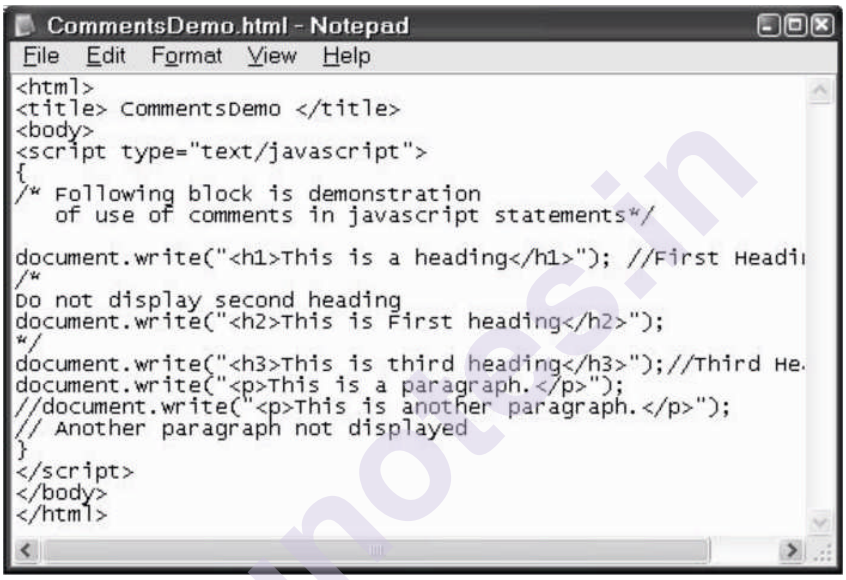
- JavaScript is a sequence of statements to be executed by the browser. Browser executes the statements in the same order as they are written.
- JavaScript is case sensitive with all syntax, variable and function names.
- The semicolon at the end of line is optional (according to the JavaScript standard), and the browser is supposed to interpret the end of the line as the end of the statement. However, semicolon at the end of line is good programming practice. Also, it enables us to write multiple statements on the same line.
- JavaScript statements can be grouped together in blocks. Blocks start with a left curly bracket {, and end with a right curly bracket }. The purpose of a block is to make the sequence of statements execute together.
- A block is normally used to group the statements in a function or condition.
- A general example of block is–



munotes.in

JavaScript Comments

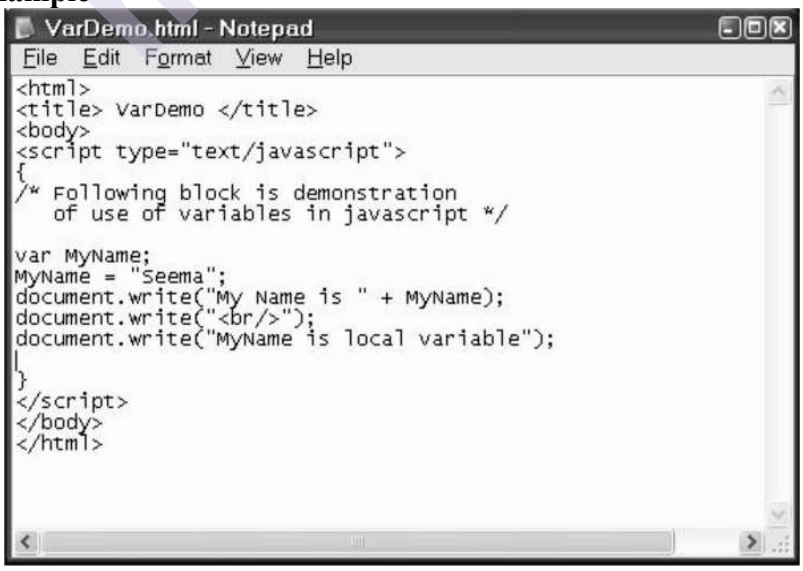
- JavaScript comments can be used to make the code more readable.
- Comments can be added to explain the Java Script.
- Comments can be added at end of a line.
- Single line comments start with//.
- Multi line comments start with /* and end with*/.
- The comment is used to prevent the execution of a single code line or a code block. This can be suitable for debugging
- Following example demonstrates use of comments in java script code.



Javascript Variables (Var statement)

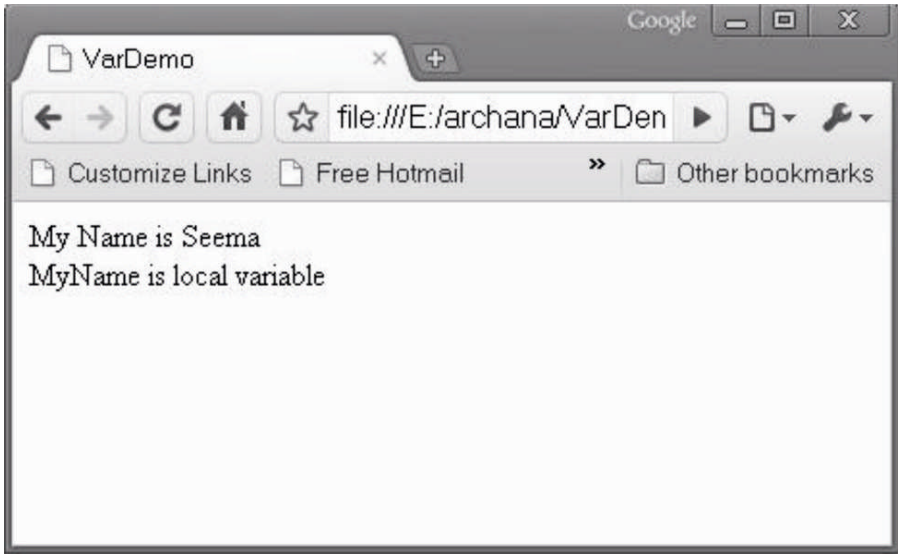
- Variables are "containers" for storing information. This information can be values or expressions.
- A variable can have a short name, like x, or a more descriptive name, like My Name.

- Rules for JavaScript variable names:
  - Variable names are case sensitive (y and Y are two different variables)
  - Variable names must begin with a letter or the underscore character
- Creating variables in JavaScript is most often referred to as "declaring" variables.
- You declare JavaScript variables with the var keyword like varx;
- After the declaration shown above, the variables are empty (they have no values yet). However, you can also assign values to the variables when you declare them like var x = 10; After the execution of this statement, the variable x will hold the value10
- A variable declared within a JavaScript function becomes **LOCAL** and can only be accessed within that function. (the variable has localscope).
- You can have local variables with the same name in different functions, because local variables are only recognized by the function in which they are declared.
- Local variables are destroyed when you exit the function.
- Variables declared outside a function become **GLOBAL**, and all scripts and functions on the web page can access it.
- Global variables are destroyed when you close thepage.
- If you declare a variable, without using "var", the variable always becomes GLOBAL.
- If you assign values to variables that have not yet been declared, the variables will automatically be declared as global variables.
- All javascript operators can be used with variables
- **Example –**




```
<html>
<title> VarDemo </title>
<body>
<script type="text/javascript">
{
/* Following block is demonstration
   of use of variables in javascript */

var MyName;
MyName = "Seema";
document.write("My Name is " + MyName);
document.write("<br/>");
document.write("MyName is local variable");

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

munotes.in



Conditional statements

- Conditional statements are used to perform different actions based on different conditions.
- In JavaScript we have the following conditional statements:
- **If statement** - This statement is used to execute some code only if a specified condition is true.
- **Syntax:**

```
if (condition)
{
code to be execute difconditionis
true
}
```

- **If...elsestatement**-This statement is used to execute some code if the condition is true and another code if the condition is false
- Syntax:

```
if (condition)
{
code to be executed if condition is true
}
else
{
code to be executed if condition is not true
}
```

- **If...else if..... else statement -** This statement is used to select one of many blocks of code to be executed
- Syntax:

```
if (condition1)
{
code to be executed if condition1 is true
}
```

munotes.in

else if (condition2)

{

code to be executed if condition2 is true

}

else

{

code to be executed if neither condition1 nor condition2 istrue

}

Example

IfElseElseifDemo - Notepad

File Edit Format View Help

```
<html>
<title> Conditional Statements Demo</title>
<body>
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time<10)
  {
  document.write("<b>Good morning</b>");
  }
else if (time>10 && time<16)
  {
  document.write("<b>Good day</b>");
  }
else
  {
  document.write("<b>Hello World!</b>");
  }
</script>
<p>
This example demonstrates the if..else if...else statement.
</p>

</body>
</html>
```

Output

Conditional Statements Der

file:///C:/archana/web%20design/ifelseelseifdemo.html

Hello World!

This example demonstrates the if..else if...else statement.

- **switch statement -**This statement is another way to select one of many blocks of code to be executed.
- **Syntax-**

switch(n)

{

case 1:

execute code block 1 break;

case 2:

execute code block 2 break;

default:

code to be executed if n is different from case 1 and 2

}

Example:

SwitchDemo - Notepad

File Edit Format View Help

```
<html>
<title> Conditional Statements Demo - switch statement</title>
<body>
<script type="text/javascript">
/*You will receive a different greeting based
on what day it is. Note that Sunday=0,
Monday=1, Tuesday=2, etc.*/

var d=new Date();
document.write("Today is " + d + ("<br\>"));
var theDay=d.getDay();

switch (theDay)
{
case 4:
   document.write("Through with Thursday");
case 5:
  document.write("Finally Friday");
  break;
case 6:
  document.write("Super Saturday");
  break;
case 0:
  document.write("Sleepy Sunday");
  break;
default:
  document.write("I'm looking forward to this weekend!");
}
</script>
</body>
</html>
```

Output

Conditional Statements Demo - switch statement - Windows Internet Explorer

C:\Archana\Web Design\

Windows Live Bing What's New Sign in

Secure Search McAfee

Favorites Internet Explorer cannot d... Web Slice Gallery

Conditional Statements Dem... Page Safety Tools

Today is Tue Aug 16 19:11:18 UTC+0530 2011
I'm looking forward to this weekend!

Done Computer | Protected Mode: Off 105%

munotes.in

Loop statements

- Loops execute a block of code a specified number of times, or while a specified condition is true.
- Often when you write code, you want the same block of code to run over and over again in a row. Instead of adding several almost equal lines in a script we can use loops to perform a task like this.
- In JavaScript, there are two different kind of loops:

o for - loops through a block of code a specified number of times. It can be used only when it is known in advance, how many times we have to run the loop.

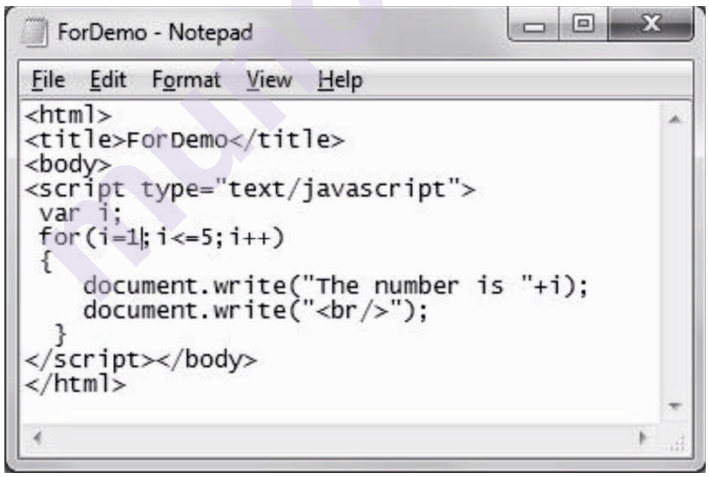
o while - loops through a block of code while a specified condition is true.

- For Loop Syntax:

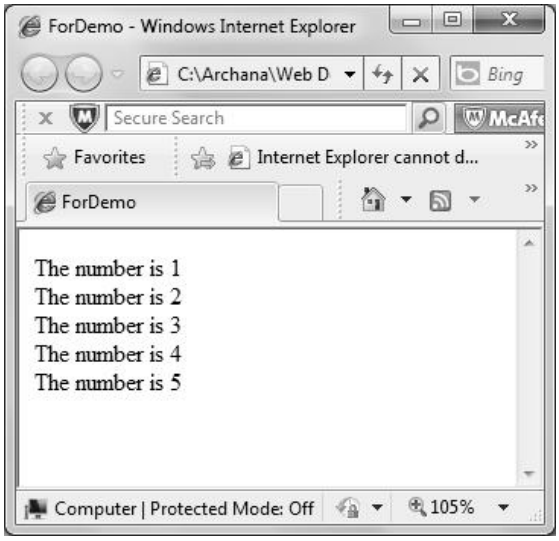
```
for (variable=startvalue;variable<=endvalue;variabl
e=variable+increment)
{
code to be executed
}
```

- **Example:**

Javascript given below will print 5 numbers. Each time, value of the variable is incremented by 1.



munotes.in



- While loop syntax

```
while(var<=end value)
{
code to be executed
}
```

- While loop can be used with any comparison operator.
- **Do While** loop is a variation to the while loop. In this case block will be executed at least once, as the statements are executed before the condition is tested. Syntax for do while is as follows–

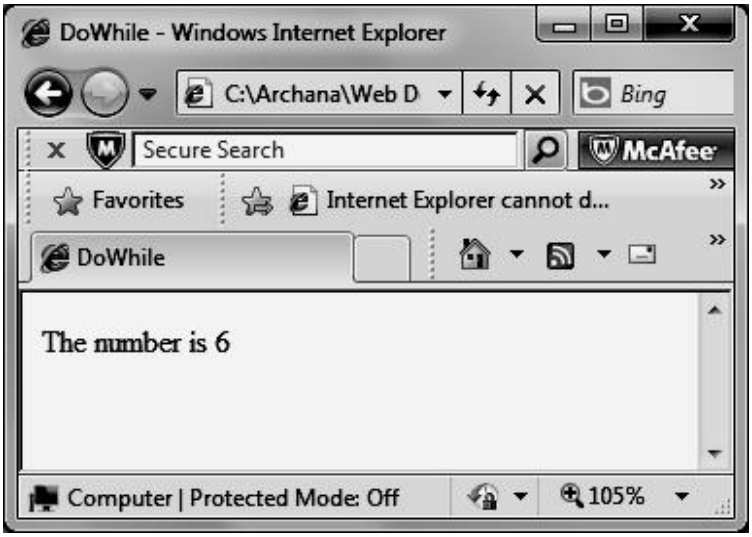
```
do
{
code to be executed
}
while (var<=endvalue);
```

- Consider the example where number is printed after incrementing itby1.This is performed while the number is less than orequalto

5. Script and the outputs with while and do while loop are as given below

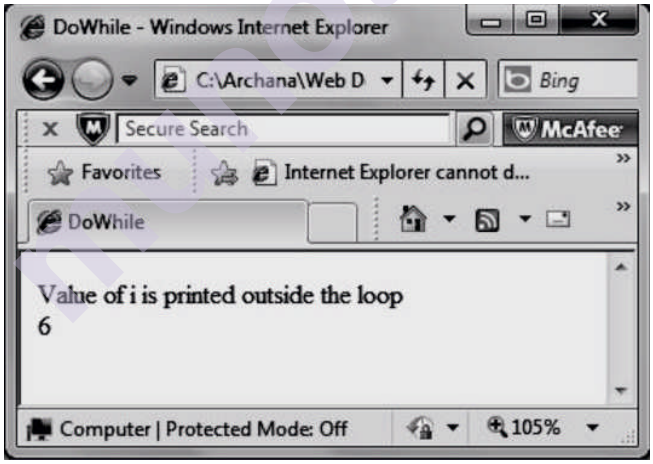
```
<html>
<body>
<script type="text/javascript">var i=6;
do
{
document.write("The number is " + i); document.write("<br />");
i++;
}
while (i<=5);

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

raunotes.in



- Changes in script with just while loop and corresponding output--

```
while (i<=5)
{
document.write("The number is " + i); document.write("<br />");
i++;
}
document.write("value of i is printed outside the loop");
document.write("<br>\" + i);
```



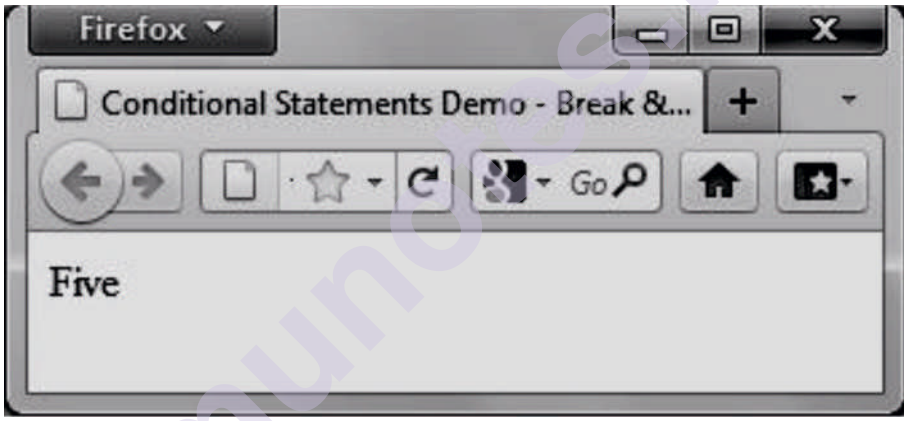
Break and Continue statement in javascript

- The break statement will break the loop and continue executing the code that follows after the loop (if any).
- The continue statement will break the current loop and continue with the next value.

Source of: file:///C:/Archana/Web%20Design/BreakDemo.html - Mozilla Firefox

File Edit View Help

```
<html>
<title> Conditional Statements Demo - Break & Continue</title>
<body>
<script type="text/javascript">
var i=5;
switch (i)
{
case 0:
 document.write("the number is zero");
case 1:
 document.write("one");
 break;
case 2:
 break;
case 5:
 document.write("Five");
 break;
default:
 document.write("Default number is 100");
 break;
}
</script>
</body>
</html>
```



Function

- Function is a segment of program that performs a given task.
- A function contains code that will be executed by an event or by a call to the function.
- You may call a function from anywhere within a page (or even from other pages if the function is embedded in an external JavaScript file).
- Functions can be defined both in the <head> and in the <body> section of a document.
- However, to assure that a function is read/loaded by the browser before it is called, it should be put functions in the<head>section.
- Syntax of the function is as follows–

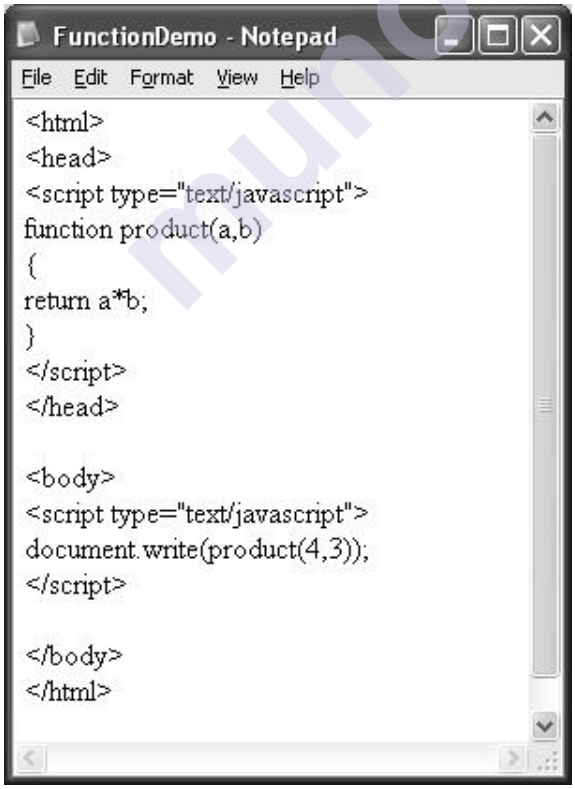
```
function  name(var1,var2,...,varX)
{
some code
}
```

- Function always includes parenthesis after the name of function'()'

- Function calls are case sensitive as javascript is case sensitive.
- The return statement is used to specify the value that is returned from the function.

  So, functions that are going to return a value must use the return statement.
- If a variable is declared using "var", within a function, the variable can only be accessed within that function.
- The variable is destroyed once function call is over. These variables are called local variables. Local variables can have the same name in different functions, because each is recognized only by the function in which it is declared.
- If a variable is declared outside a function, all the functions on your page can access it.
- The lifetime of these variables starts when they are declared, and ends when the page is closed.
- Following is an example of function. Function products computes and returns product of variables a and b.
- Basic advantage of using a function is reusability. Same task can be performed again and again simply by calling the function which performs the task.

Example:

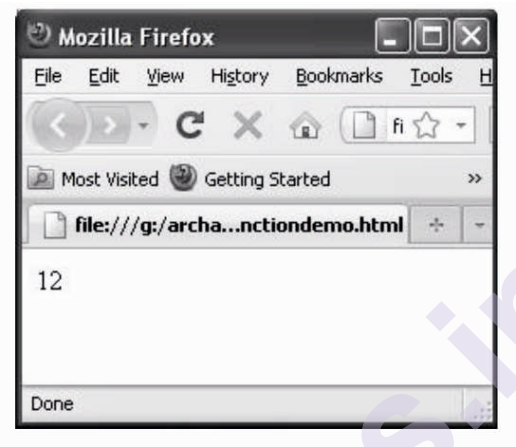


munotes.in

7.5 COREJAVASCRIPT

Data Types are classified as primitive data types and composite data types. Composite Data types -

- Numbers - are values that can be processed and calculated. The numbers can be either positive or negative. Javascript integer can have three base values – 10 (decimal), 8(octal) or 16(hexadecimal). Number can be integer or floating point number.



- Strings - are a series of letters and numbers enclosed in single or double quotation marks. Strings are used for text to be displayed or values to be passed along.
- Some characters that you may want in a string may not exist on the keyboard, or may be special characters that can't appear as themselves in a string.
- In order to put these characters in a string, you need to use an escape sequence to represent the character. An escape sequence is a character or numeric value representing a character that is preceded by a backslash (\ ) to indicate that it is a special character.

Some escaped characters are as follows:

\b Backspace.

\t Tab. Tabs behave erratically on the Web and are best avoided, but sometimes you need them.

\n New line (\u0000a). Inserts a line break at the point specified. It is a combination of the carriage return (\r) and the form feed(\f).

\" Double quote.

\' Single quote, or an apostrophe, such as incan\'t.

\\ The backslash, since by itself it is a command.

- Boolean (true/false) - lets you evaluate whether a condition meets or does not meet specified criteria.
- Null - is an empty value. null is not the same as 0 -- 0 is a real, calculable number, whereas null is the absence of any value. An empty string is distinct from null value.

- NAN – Some javascript functions return a special value called not a number.

Primitive Data Types –

Object –

- An *object* is a collection of named values, called the *properties* of that object. Functions associated with an object are referred to as the *methods* of that object.
- Properties and methods of objects are referred to with a dot(.) notation that starts with the name of the object and ends with the name of the property. For instance image. src.
- Normally in objects there are only two nodes, the object and the property, but sometimes the properties can have properties of their own, creating an object tree.
- For instance,document.form1.namefield.
- Objects in JavaScript can be treated as associative arrays. This means that image.src and image['src'] are equivalent.
- JavaScript has many predefined objects, such as a Date object and a Math object. These are used much as function libraries are used in a language like C.
- They contain a collection of useful methods that are predefined and ready for use in any JavaScript code you may write.

Date Object –

- The Date object is used to work with dates andtimes.
- Date objects are created with newDate().
- There are four ways of instantiating adate:

var d = new Date();

var d = new Date(milliseconds); var d = new Date(dateString);

var d = new Date(year, month, day, hours, minutes, seconds, milliseconds);

- Some javascript predefined methods are–

- **getDate()** Returns the day of the month (from1-31)
- **getDay()** Returns the day of the week (from0-6)
- **getFullYear()** Returns the year (fourdigits)
- **getHours()** Returns the hour (from0-23)
- **getMilliseconds()** Returns the milliseconds (from0-999)
- **getMinutes()** Returns the minutes (from0-59)
- **getMonth()** Returns the month (from0-11)
- **getSeconds()** Returns the seconds (from0-59)
- **getTime()** Returns the number of milliseconds since midnight Jan 1,1970
- **getTimezoneOffset()** Returns the time difference between GMT and local time, inminutes

- **getYear()** Deprecated. Usethe
- **getFullYear()** methodinstead
- **parse()** Parses a date string and returns the number of milliseconds since midnight of January 1,1970
- **setDate()** Sets the day of the month (from1-31)
- **setHours()** Sets the hour (from0-23)
- **setMilliseconds()** Sets the milliseconds (from0-999)
- **setMinutes()** Set the minutes (from0-59)
- **setMonth()** Sets the month (from0-11)
- **setSeconds()** Sets the seconds (from0-59)
- **setTime()** Sets a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970
- **toString()** Converts a Date object to astring
- **toTimeString()** Converts the time portion of a Date object to astring
- **toUTCString()** Converts a Date object to a string, according to universaltime
- **valueOf()** Returns the primitive value of a Dateobject

Math object –

- The Math object allows you to perform mathematical tasks.
- The Math object includes several mathematical constants and methods.
- round() – rounds the number to nearest integer value.
- random() - returns a random number between 0 and1.
- max() - returns the number with the highest value of two specified numbers.
- min() - returns the number with the lowest value of two specified numbers.

Array –

- An *Array* is an ordered collection of data values.
- In JavaScript, an array is just an object that has an index to refer to its contents. In other words, the fields in the array are numbered, and you can refer to the number position of the field.
- The array index is included in square brackets immediately after the array name. In JavaScript, the array index starts with zero, so the first element in an array would be array Name[0], and the third would be array Name[2].
- JavaScript does not have multi-dimensional arrays, but you can nest them, which is to say, an array element can contain another array.
- You access them listing the array numbers starting for the outmost array and working inward. Therefore, the third element (position 2) of or inside the ninth element (position 8) would bearrayName[8][2].

munotes.in

7.6 SUMMARY

- JavaScript is light weighted programming language used for small applications.
- Javascrit is object based programming language made inC++.
- Javascript can be executed either on client side browser or on server side browser.
- Javascripts can be downloaded and run and carry a threat of virus attack. Security of javascript depends upon the security of java language. The process of executing the javascript code after isolating it from the operating system is called 'sandbox' model.
- Different types of javascript operators help perform various arithmetic, logical, string and other types of functions.
- Browser sequentially executes javascript statements of various types such as conditional and loop
- statements. Some of loop statements are while, do while, switch statement etc.
- Number, Boolean, string are few data types used in JavaScript.

7.7 EXERCISE

1. Define javascript. How is itused?
2. Explain the difference between client side and server side JavaScript.
3. Explain how virus threat from javascript is avoided?
4. write a javascript to find greatest of three given numbers
5. write a javascript to print the month if numeric value for the month is given (eg January for 1 and so on)
6. Write a javascript to print all prime numbers from 1 to 100 in reverse order.
7. Write a javascipt to print reverse of a number.
8. Explain what is an object in javascript. Describe the date and math object with at least two properties each.
9. explain the difference between String"" and null value
10. Discuss what is a function? Does it always return a value? Give example of at least one function.
11. What is an array? How is it used in JavaScript?
12. Explain how a date value can be converted into a string value and vice versa.

munotes.in

8

DOCUMENT AND ITS ASSOCIATED OBJECTS & EVENT AND EVENT HANDLERS

Unit Structure :

8.0 Objective
8.1 Document and associated objects
8.1.1 Document
8.1.2 Link
8.1.3 Area
8.1.4 Anchor
8.1.5 Image
8.1.6 Applet
8.1.7 Layer
8.2 Events and Event Handlers
8.2.1 General information about events
8.2.2 Defining event handlers
8.3 Event
8.3.1 on Abort
8.3.2 on Blur
8.3.3 on Change
8.3.4 on Click
8.3.5 on DblClick
8.3.6 on Drag Drop
8.3.7 on Error
8.3.8 on Focus
8.3.9 on Key Down
8.3.10 on Key Press
8.3.11 on Key Up
8.3.12 on Load
8.3.13 on Mouse Down
8.3.14 on Mouse Move
8.3.15 on Mouse Out
8.3.16 on Mouse Over
8.3.17 on Mouse Up
8.3.18 on Move
8.3.19 on Reset

munotes.in

8.3.20 on Resize
8.3.21 on Select
8.3.22 on Submit
8.3.23 on Unload

8.1 OBJECTIVE

After reading this chapter you will be able to -

- Understand the use of document object and other objects associated with the document object.
- Understand and use the properties and methods and to use common properties and methods.
- Understand and use objects such as area, anchor, link etc. and properties and methods associated with these objects.
- Write java scripts using the above properties and methods.
- Understand the concept of event and event handlers.
- Understand the use of each event associated with the document object.

8.1 DOCUMENT AND ASSOCIATEDOBJECTS

8.1.1 Document

- Each HTML document loaded into a browser window becomes a Document object.
- The Document object provides access to all HTML elements in a page, from within a script.
- The Document object is also part of the Window object, and can be accessed through the 'window. Document' property.
- Document object is part of document object model.
- This model has a fixed hierarchy, where topmost object in the hierarchy is Browser itself.
- After browser window object and inside window, as shown below, comes the document object.
- Relation of document object to the window object can be depicted in the fig given below –

munotes.in

|-> Document
|-> Anchor
|-> Link
|-> Images
|->Tags
|->Form

|-> Text-box
|-> Text Area
|-> Radio Button
|-> Check Box
|->Select
|->Button

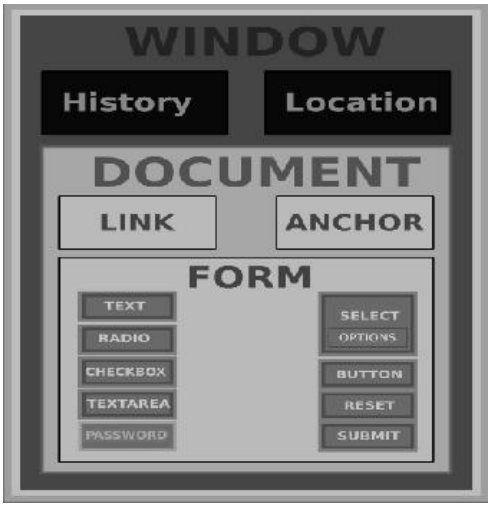


Fig 8.1 Document Object and Window object

Fig 8.2 Hierarchy in Document Object

- Document object has following properties and methods

Properties –

- **cookie** Returns all name/value pairs of cookies in the document
- **document Mode** Returns the mode used by the browser to render the document
- **domain** Returns the domain name of the server that loaded the document
- **last Modified** Returns the date and time the document was last modified
- **ready State** Returns the (loading) status of the document
- **referrer** Returns the URL of the document that loaded the current document
- **title** Sets or returns the title of the document
- **URL** Returns the full URL of the document

Methods –

- **close()** Closes the output stream previously opened with document. open()
- **get Element ById()** Accesses the first element with the specifiedid
- **get Elements By Name()** Accesses all elements with a specified name
- **get Elements By Tag Name()** Accesses all elements with a specified tagname
- **open()** Opens an output stream to collect the output from document. write() or document. writeln()
- **write()** Writes HTML expressions or JavaScript code to a document
- **writeln()** Same as write(), but adds a new line character after each statement Yes

munotes.in

Following examples demonstrates use of some properties and methods of document object.

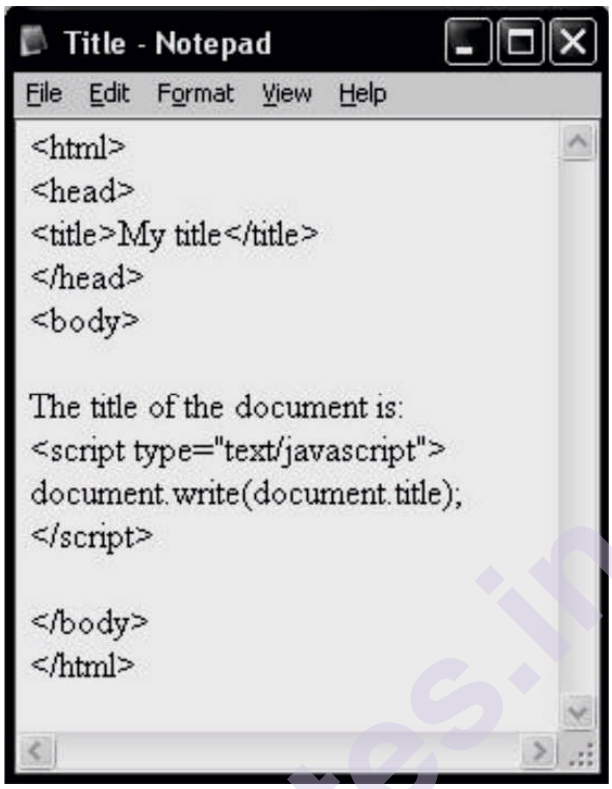


Fig 8.3 Script and output Demonstrating write method and title property of Document object.

OpenMethod - Notepad

File Edit Format View Help

```
<html>
<head>
<script type="text/javascript">
function createDoc()
  {
  var doc=document.open("text/html","replace");
  var txt="<html><body>Example of creating a document object for writting text.</body></html>";
  doc.write(txt);
  doc.close();
  }
</script>
</head>

<body>
<input type="button" value="New document" onclick="createDoc()" />
</body>
</html>
```

Fig 8.4 Script Demonstrating the Document. open() method to open a text stream

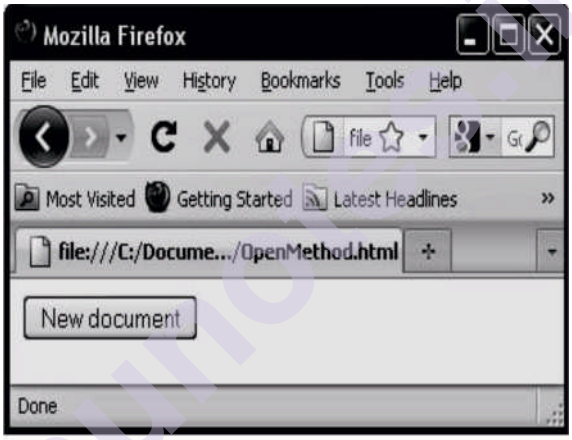


Fig 8.5 Output of the script Before and After Clicking the 'New Document' button.

8.1.2 Link

- The Link object represents an HTML link element.
- The link element must be placed inside the head section of an HTML document, and it specifies a link to an external resource.
- In other words, link tag defines how a click on object can redirect or take the browser window to a new location specified by the address specified.
- A common use of the <link> tag is to link to external style sheets.
- Some of the properties of javascript link objects are as given below—

charset: -Sets or returns the character encoding of a linked document

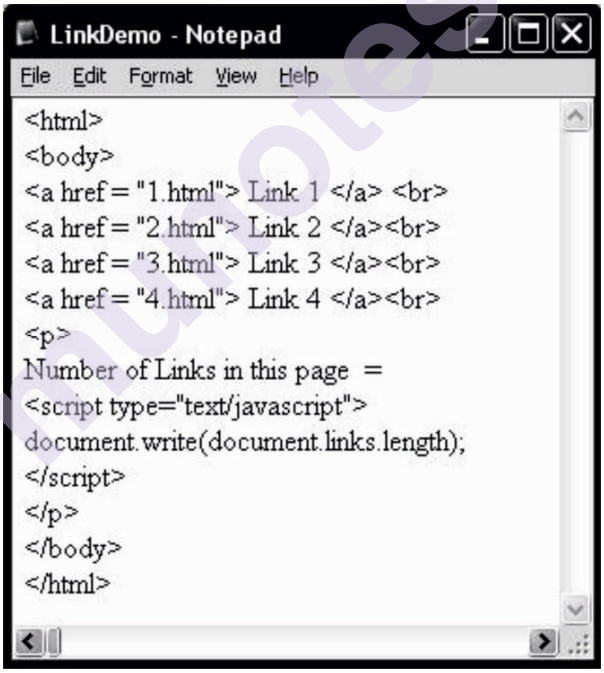
href: -Sets or returns the URL of a linked document

hreflang:- Sets or returns the language code of the linked document

media: - Sets or returns the media type for the link element

type: - Sets or returns the content type of the linked document

- In addition to above properties, link object supports all standard properties like id, length, chartypeetc.



munotes.in

Fig 8.6 Script and Outputs Demonstrating length property of link

8.1.3 Area

- The Area object represents an area inside an HTML image-map (an image-map is an image with clickable areas).
- For each <area> tag in an HTML document, an Area object is created.
- In addition to standard properties and methods, javascript area object supports following properties –

alt: Sets or returns the value of the alt attribute of an area

cords: Sets or returns the value of the coords attribute of an area

hash: Sets or returns the anchor part of the href attribute value

host: Sets or returns the hostname: port part of the href attribute value

hostname: Sets or returns the hostname part of the href attribute value

href: Sets or returns the value of the href attribute of an area

noHref: Sets or returns the value of the no href attribute of an area

pathname: Sets or returns the pathname part of the href attribute value

port: Sets or returns the port part of the href attribute value

protocol: Sets or returns the protocol part of the href attribute value

search: Sets or returns the query string part of the href attribute value

shape: Sets or returns the value of the shape attribute of an area

target: Sets or returns the value of the target attribute of an area

munotes.in

- Following example shows an image map and returns the shape of an area marked in the image map.

PlanetMap - Notepad

File Edit Format View Help

```
<html>
<body>
<img src="planets.gif"
width="145" height="126"
usemap="#planetmap" />

<map name="planetmap">
<area id="venus" shape="circle" coords="124,58,8"
alt="Venus"
href="venus.htm" />
</map>
<p>The shape for the "Venus" area is:
<script type="text/javascript">
document.write(document.getElementById("venus").shape);
</script>
</p>

</body>
</html>
```

The shape for the "Venus" area is: circle

Fig 8.7 Script and output Demonstrating Area object.

8.1.4 Anchor

- The Anchor object represents an HTML hyperlink.
- For each <a> tag in an HTML document, an Anchor object is created.
- An anchor allows you to create a link to another document (with the href attribute), or to a different point in the same document (with the name attribute).
- You can access an anchor by using get Element ById(), or by searching through the anchors[] array of the Document object.
- In addition to standard properties and methods, javascript anchor object supports following properties–

charset Sets or returns the value of the charset attribute of a link **href** Sets or returns the value of the href attribute of a link **hre flang** Sets or returns the value of the hre flang attribute of a link **name** Sets or returns the value of the name attribute of alink

rel Sets or returns the value of the rel attribute of a link

rev Sets or returns the value of the rev attribute of a link **target** Sets or returns the value of the target attribute of a link **type** Sets or returns the value of the type attribute of alink

- Following example shows anchor tag used in a webpage.

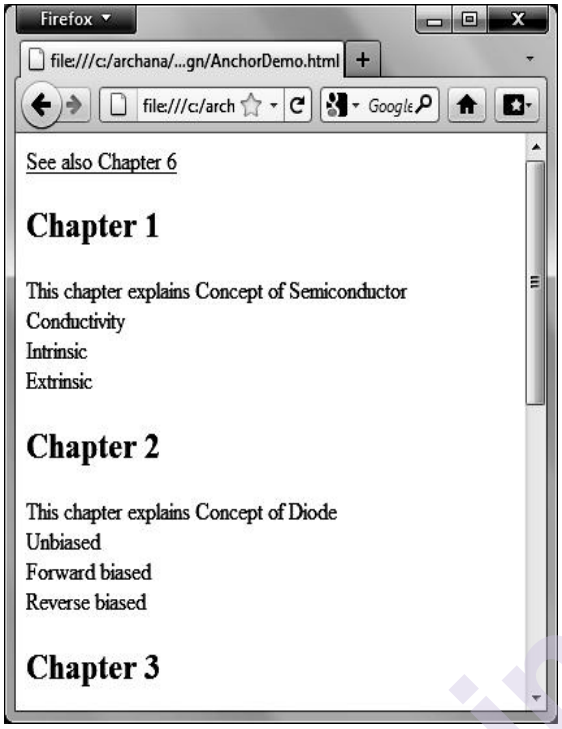
AnchorDemo - Notepad

File Edit Format View Help

```
<html>
<body>
<p><a href="#C6">See also Chapter 6</a></p>
<h2>Chapter 1</h2>
<p>This chapter explains Concept of Semiconductor <br>
Conductivity<br>
Intrinsic<br>
Extrinsic<br>

</p>
<h2>Chapter 2</h2>
<p>This chapter explains Concept of Diode<br>
Unbiased<br>
Forward biased<br>
Reverse biased<br>
</p>
<h2>Chapter 3</h2>
<p>This chapter explains Transistors<br>
NPN<br>
PNP<br> </p>
<h2>Chapter 4</h2>
<p>This chapter explains Amplifier<br>
Single Stage<br>
Mulistage<br>
</p>

<h2>Chapter 5</h2>
<p>This chapter explains Feedback Circuits</p>

<h2><a id="c6" name="C6">Chapter 6</a></h2>
<p>This chapter explains Oscillators </p>
<script type="text/javascript">
document.write("Return name of anchor: ");
document.write(document.getElementById("c6").name);
</script>
</body>
</html>
```

Fig 8.8 Script For demonstrating anchor object

munotes.in



align : Sets or returns the value of the align attribute of an image

alt : Sets or returns the value of the alt attribute of an image

border : Sets or returns the value of the border attribute of an image

complete : Returns whether or not the browser is finished loading an image

height : Sets or returns the value of the height attribute of an image

hspace : Sets or returns the value of the hspace attribute of an image

longDesc : Sets or returns the value of the longdesc attribute of an image

lowsrc : Sets or returns a URL to a low-resolution version of an image

name : Sets or returns the name of an image

src : Sets or returns the value of the src attribute of an image

useMap : Sets or returns the value of the use map attribute of an image

vspace : Sets or returns the value of the vspace attribute of an image

width : Sets or returns the value of the width attribute of an image

munotes.in

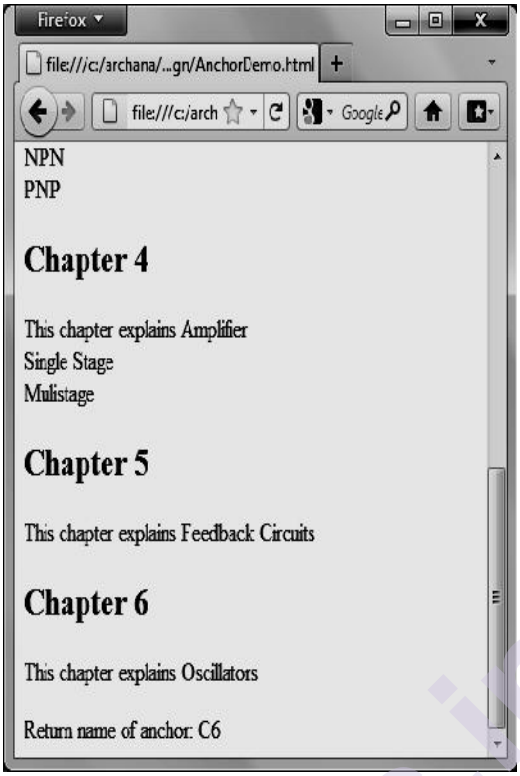


Fig 8.9 Before and after using the anchor link.

- After the anchor is used URL of browser changed to 'file:///c:/AnchorDemo.html#C6', as c6 is used as an anchor.

8.1.5 Image

- The Image object represents an embedded image.
- For each <img> tag in an HTML document, an Image object is created.
- Notice that images are not technically inserted into an HTML page, images are linked to HTML pages. The <img> tag creates a holding space for the referenced image.
- Image object, in addition to the standard properties and methods, supports following properties and events.

- **<u>Image ObjectEvents</u>**

<u>on abort</u> Loading of an image is interrupted
<u>on error</u> An error occurs when loading an image
<u>on load</u> An image is finished loading

- Following is the example of some properties of image object and javascript functions used to change these properties.
- Function changesize changes the height and width of the image object and function add Border adds border to the image.
- Similarly, we can also change the image by using the srcattribute of imagetag.

- Alt attribute gives the text to be displayed, if the image can not be displayed.
- Also, a lower version of image to speed up loading, can be shown using a lowsrc option attribute.

Example:

imageDemo - Notepad

File Edit Format View Help

```
<html>
<head>
<script type="text/javascript">
function addBorder()
{
document.getElementById("compgirl").border="2";
}

function changeSize()
{
document.getElementById("compgirl").height="250";
document.getElementById("compgirl").width="300";
}
</script>
</head>
<body>

<img id="compgirl" src="imagesample.gif" alt="girl on computer" width="107" height="98" />
<br /><br />
<input type="button" onclick="addBorder()" value="Add border" />
<p>
The value of the alt attribute is:
<script type="text/javascript">
document.write(document.getElementById("compgirl").alt);
</script>
</p>
<input type="button" onclick="changeSize()" value="Change size of image" />

</body>
</html>
```

Fig 8.10 source code for demonstrating some image properties and attributes.

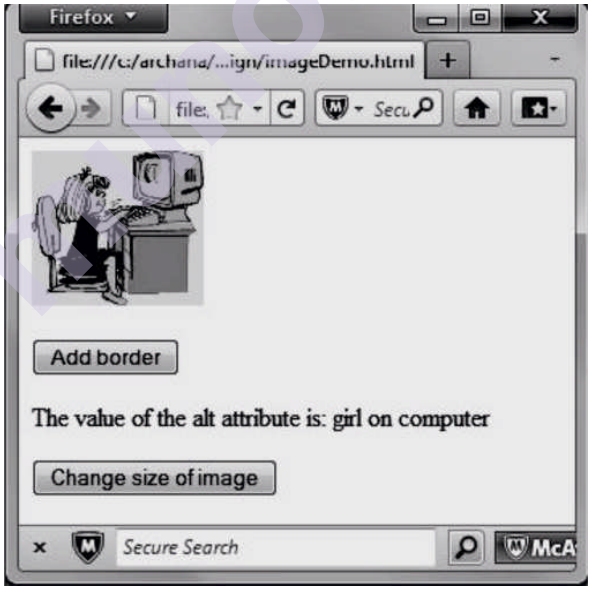


Fig 8.11 Before click of the command button objects

Fig 8.12 After click of the command buttons, border added and a increased sized image.

8.1.6 Applet

- Applet tag was used in earlier version of HTML (HTML4) to embed anjavaappletina browser. It is not supported in HTML

  5.0 and is replaced by the object tag.
- The <object> tag defines an embedded object within an HTML document. It is used to embed multimedia (like audio, video, Java applets, ActiveX, PDF, and Flash) in your webpages.
- You can also use the <object> tag to embed another webpage into your HTML document.
- You can use the <param> tag to pass parameters to plugins that have been embedded using the <object>tag.
- An object element must appear inside the body element. The text between the <object> and </object> is an alternate text, for browsers that do not support this tag.
- At least one of the "data" and "type" attributes MUST be defined where data specifies a URL that refers to the object's data and type Specifies the MIME type of data specified in the data attribute. **(**Multipurpose Internet Mail Extensions (**MIME**) is an Internet

standard that extends the format of email to support Text in character sets other than ASCII, Non-text attachments, Message bodies with multiple parts, Header information in non-ASCII character sets.

8.2 EVENTS AND EVENTHANDLERS

8.2.1 General information about events

- Events are actions that can be detected by JavaScript.
- Every element on a web page has certain events which can trigger a Java Script.
- For example, we can use the on Click event of a button element to indicate that a function will run when a user clicks on the button.
- Examples of events:
  - A mouse click
  - A web page or an image loading
  - Mousing over a hot spot on the webpage
  - Selecting an input field in an HTML form
  - Submitting an HTML form
  - A keystroke
- Events are normally used in combination with functions, and the function will not be executed before the event occurs!

8.2.2 Defining event handlers

- They are JavaScript code that are not added inside the <script> tags, but rather, inside the html tags, that execute JavaScript when something happens, such as pressing a button, moving your mouse over a link, submitting a formetc.
- The basic syntax of these event handlersis:
  name_of_handler="JavaScript code here"
- **For example:**

```
<a
href="http://google.com"onClick="alert('hello
```

- When events are associated with functions, the functions are written in the head section within the <script> tag and are called from the event handlers.

Example :

```
<html>
<body>
<h1 onclick="this.innerHTML='Welcome to
EventHandlers'">Click on this text</h1>
</body>
</html>
```

- This code will generate following output–

Fig 8.12 Event Handler and after event is called.

8.3 EVENT

Following is the list of events used by various javascript objects and when are these events triggered.

Attribute	The event occurs when...
onabort	Page is not finished loading
onblur	An element loses focus
onchange	The content of a field changes
onclick	Mouse clicks an object
ondblclick	Mouse double-clicks an object
ondragdrop	A user drops an object
onerror	An error occurs when loading a document or an image
onfocus	An element gets focus
onkeydown	A keyboard key is pressed
onkeypress	A keyboard key is pressed or held down
onkeyup	A keyboard key is released
onload	A page or image is finished loading
onmousedown	A mouse button is pressed
onmousemove	The mouse is moved
onmouseout	The mouse is moved off an element

onmouseover	The mouse is moved over an element
onmouseup	A mouse button is released
onmove	The position of top left corner of an object is moved.
onresize	A window or frame is resized
onreset	Reset button on the form is clicked.
onselect	Text is selected
onsubmit	Validate all form fields before submitting it.
onunload	The user exits the page

- Following is an example which shows onmouseover event to give different messages for different parts of animagemap.

eventDemo Notepad

File Edit Format View Help

```
<html>
<head>
<script type="text/javascript">
function writeText(txt)
{
document.getElementById("desc").innerHTML=txt;
}
</script>
</head>

<body>
<img src ="planets.gif" width ="145" height ="126" alt="Planets" usemap="#planetmap" />

<map name="planetmap">
<area shape ="rect" coords ="0,0,82,126"
onmouseover="writeText('The Sun and the gas giant planets like Jupiter are by far the largest objects in
our Solar System.')"
href ="sun.htm" target ="_blank" alt="Sun" />

<area shape ="circle" coords ="90,58,3"
onmouseover="writeText('The planet Mercury is very difficult to study from the Earth because it is
always so close to the Sun.')"
href ="mercur.htm" target ="_blank" alt="Mercury" />

<area shape ="circle" coords ="124,58,8"
onmouseover="writeText('Until the 1960s, Venus was often considered a twin sister to the Earth
because Venus is the nearest planet to us, and because the two planets seem to share many
characteristics.')"
href ="venus.htm" target ="_blank" alt="Venus" />
</map>

<p id="desc">Mouse over the sun and the planets and see the different descriptions.</p>

</body>
</html>
```

Fig 8.13 Script to demonstrate on mouse over event and its event handler

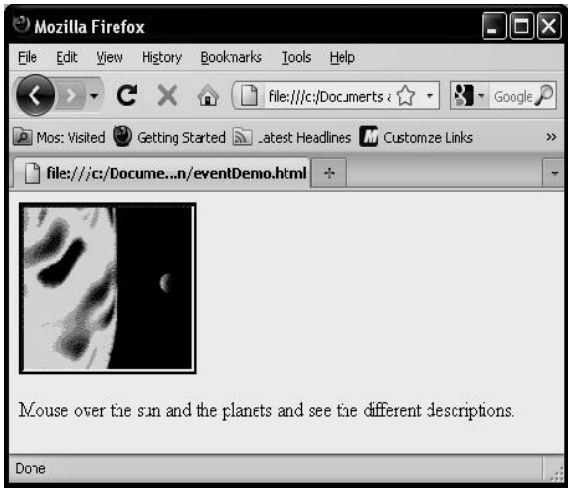


Fig 8.14 Newly Loaded page

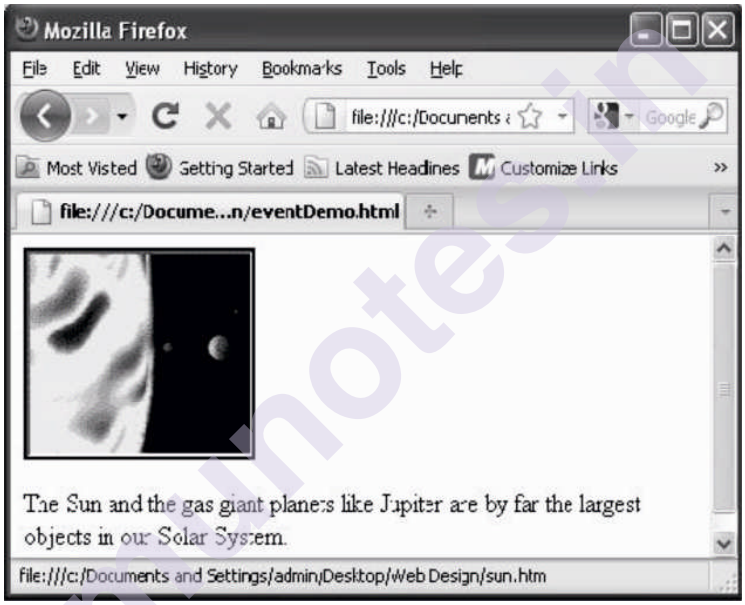


Fig 8.15 cursor on the sun.

Fig 8.16 Cursor pointing to mercury

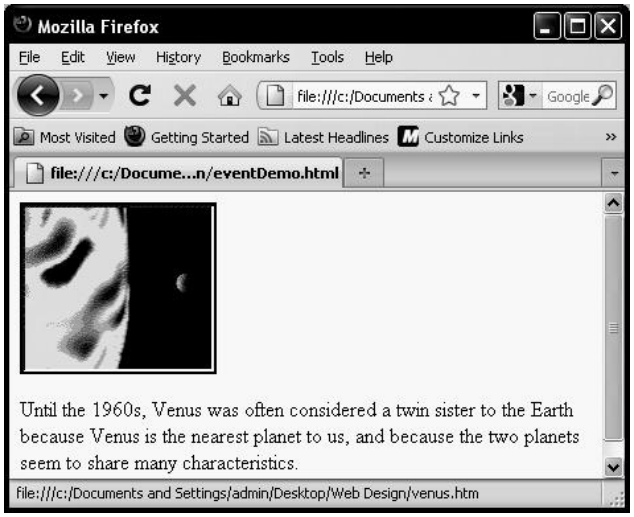


Fig 8.17 Cursor pointing venus.

8.4 EXERCISES

1. Define document object. What are the objects associated with document.
2. which are the different methods associated with document object.
3. write a javascript to write a message "Welcome to the Document object" using the write method of document object.
4. list the objects associated with form object.
5. Explain what is link object? What is the use of link object? State any two properties of this object.
6. List various properties of Area tag. What is the use of shape property of area object? Explain how is it used with an example.
7. What is anchor? How is it used in a document object?
8. Define an event. What is event handler? Explain how event handler is used.
9. Explain how a function can be used as event handler?
10. Write a java script with at least three functions to demonstrate use of on mouse move and on mouse over event.
11. Write a javascript to display a message "welcome to scripting" when a button labled "hello" is clicked. Display a message "Thank you for using scripting" when the same button is double clicked.
12. Write a javascript code to display a text "Image can not be displayed in current browser setting" if a image used in script can not be disaplyed.

❖❖❖❖

UNIT - IV

9

BASICS OF PHP

Unit Structure :

9.0 Objectives
9.1 Introduction
9.2 Why PHP and My SQL?
9.3 Server-side scripting
9.4 PHP syntax and variables
9.5 Comments, types
9.6 Summary
9.7 References
9.8 Unit End Exercise

9.0 OBJECTIVES:

After reading through this chapter, you will be able to --

- To understand how server-side programming works on the web.
- PHP Basic syntax for variable types and calculations.
- To understand the basic PHP and My SQL
- To understand the types, comments of the PHP

9.1 INTRODUCTION

- PHP started out as a small open-source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.
- PHP is stands for "PHP: Hypertext Preprocessor".
- PHP is one of the most widely used server-side scripting language for web development. Popular websites like Facebook, Yahoo, Wikipedia etc., are developed using PHP.
- PHP is so popular because it's very simple to learn, code and deploy on server, hence it has been the first choice for beginners since decades.
- In this chapter we will be covering all the important concepts of Php language from basics to advanced and will also share some ready-to-use, useful code sinppets for beginners to kickstart their web development project.

- **Why PHP?**

Following are some reasons why PHP is popular now days

• Compatible with almost all servers used nowadays:

A web server is an information technology that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web. There exist many types of web servers that servers use. Some of the most important and well-known are: Apache HTTP Server, IIS (Internet Information Services), lighttpd, Sun Java System

Web Server etc. As a matter of fact, PHP is compatible with all these web servers and many more.

• PHP will run on most platforms:

Unlike some technologies that require a specific operating system or are built specifically for that, PHP is engineered to run on various platforms like Windows, Mac OSX, Linux, Unix etc)

• PHP supports such a wide range of databases:

An important reason why PHP is so used today is also related to the various databases it supports (is compatible with). Some of these databases are: DB++, dBase, Ingres, Mongo, MaxDB, MongoDB, mSQL, Mssql, MySQL, OCI8, PostgreSQL, SQLite, SQLite3 and so on.

• PHP is free to download and open source:

Anyone can start using PHP right now by downloading it from php.net. Millions of people are using PHP to create dynamic content and database-related applications that make for outstanding web systems. PHP is also open source, which means the original source code is made freely available and may be redistributed and modified.

• Easy to learn & large community:

PHP is a simple language to learn step by step. This makes it easier for people to get engaged in exploring it. It also has such a huge community online that is constantly willing to help you whenever you're stuck (which actually happens quite a lot).

9.2 WHY PHP AND MYSQL?

➔PHP being the server-side programming language and MYSQL an open-source relational database management system when combined together is capable of delivering highly unique solutions. One of the main reasons the businesses choose PHP MYQL Application Development is its simplicity and ease of use.

➔PHP programming language is indeed one of the best solutions for developing websites and web-based applications. However, PHP can't store data by its own thereby a reliable database for storing information securely and being able to easily retrieve it whenever needed is required.

➔MySQL is a first choice of PHP developers. As an open-source Relational Database Management System (RDBMS) that uses SQL language, MySQL database helps to automate data retrieving and provide great support in PHP MySQL web application development.

Listed below are some of the advantages of PHP MYSQL Web Development:

Dynamic: Since PHP is a server-side scripting language it creates dynamic pages with customized features. PHP provides a user-friendly and interactive website or web application and also enables visitors to freely interact while producing a very flexible and dynamic content.

Ease of use: PHP is very easy to learn as compared to the other programming languages. The PHP syntax can easily be parsed. With its stability, PHP is sure to solve many problems with ease.

HTML embedded codes: PHP is no doubt a stable and cross-platform compatible language. And because of its ability to decode HTML, there is no need to have separate coding for PHP. This property comes with several other benefits such as:

- PHP can easily be incorporated into a code generated by WYSIWYG editors
- PHP can reduce the cost while increasing the efficiency of the websites or web applications
- With PHP, one does not have to rewrite every line of HTML in a programming language

Ideal for ecommerce development: PHP can easily be customized as per the business specific needs of the clients and can execute all the demands for ecommerce development.

Cost effective: PHP web development is very cost-effective and never cost an extra dime. With a free license, you can be sure that no one will ask you to pay extra after developing the website. It is worth knowing that Apache/PHP/MYSQL combo runs perfectly well on a low cost, low end hardware that you can ever imagine for ISS/ASP/SQL Server.

Talking about What are the benefits of using PHP MySQL Development Services? The another big advantage of PHP is its interoperability with multiple operating systems and servers. Portability being one of the biggest concerns for businesses, PHP solves the portability issues with one or more use of the operating systems. Businesses can save money and leverage their existing resources rather than investing large sum of money in purchasing the proprietary products.

munotes.in

9.3 SERVER-SIDE SCRIPTING

Three are the main areas where PHP scripts are used:

• Server-side scripting:

This is the most used and main target for PHP. You need three things to make this work the way you need it. The PHP parser (CGI or server module), a web server and a web browser. You need to run the web server where. You can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on your home machine if you are just experimenting with PHP programming.

• Command line scripting:

You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.

• Writing desktop applications:

PHP may not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way.

9.4 PHP SYNTAX AND VARIABLES

A PHP script is executed on the server, and the plain HTML result is sent back to the browser.

Basic PHP Syntax:

A PHP script can be placed anywhere in the document. A PHP script starts with <?php and ends with?>:

```
<?php
// PHP code goes here
?>
```

The default file extension for PHP files is ".php". A PHP file normally contains HTML tags, and some PHP scripting code.

Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:

Example:

```
<!DOCTYPE html>
<html>
<body>
<h1>My first PHP page</h1>
<?php
echo "Hello IDOL!";
```

```
?>
</body>
</html>
```
Output:
My first PHP page
Hello IDOL!

Note: PHP statements end with a semicolon (;).

PHP Case Sensitivity:
➔In PHP, keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.

In the example below, all three echo statements below are equal and legal.

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

<?php
ECHO " Hello Mumbai!<br>";
echo " Hello Mumbai!<br>";
EcHo " Hello Mumbai!<br>";
?>
</body>
</html>
```
Output:
Hello Mumbai!
Hello Mumbai!
Hello Mumbai!

Note: However all variable names are case-sensitive.
Look at the example below; only the first statement will display the value of the $color variable! This is because $color, $COLOR, and $coLOR are treated as three different variables:

```
<!DOCTYPE html>
<html>
<body>
<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>
</body>
</html>
```

munotes.in

Output:
My car is red
My house is
My boat is

Overview of variables:
➔The variables in PHP are used for saving the value of the program. These values can be used later whenever needed by the programmer. The latest value assigned to the variable is its value. All the variables are preceded by a $ sign. User does not need to declare the variable before the assignment. The = operator is used for assigning the values to the variable.

➔A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

Rules for PHP variables:
- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive ($age and $AGE are two different variables)

Output Variables:
The PHP echo statement is often used to output data to the screen.
The following example will show how to output text and a variable:

```
<!DOCTYPE html>
<html>
<body>
<?php
$txt = "SIOM";
echo "welcome to $txt!";
?>
</body>
</html>
```

Output:
welcome to SIOM!

Scope of variables:
➔The variables in PHP are available to a specific limit in the coding. There are four types of variable scope defined as mentioned below.

1. Local variables

The local variables are the one defined inside the function. The variable declared outside the function are corresponding to another function.

```
<?php
   $a = 10;
   function add()
   {
  $a = 1;
```

```
print "Value of $a is: <br>";
  }
add();
print "Value of $a here is:<br>";
?>
```

Output:
Output for the above code is as shown below:
Value of $a is 1
Value of $a here is 10

2. Global variables
The global variables are accessed anywhere in the program. The variable is explicitly declared to be global in a function. The GLOBAL keyword is used before the variable name.

```
<?php
   $var1 = 20;
   function display()
   {
      GLOBAL $var1;
      $var1 ++;
      print " Variable is: $var1;
   }
   display();
?>
```

Output:
The result generated when the code is executed is:
Variable is: 20

3. Static variables
The static variables maintain its value even when the user exits the function. The STATIC keyword is used to define the static variable.

```
<?php
   function increase()
   {
      STATIC $counter = 0;
      $counter ++;
      print $counter;
      print "<br/>";
   }
   increase();
   increase();
?>
```

Output:
The output generated when the code is executed is:
1
2

munotes.in

9.5 TYPES, COMMENTS

PHP has a total of eight data types which we use to construct our variables –

- **Integers** − are whole numbers, without a decimal point, like 4195.
- **Doubles** − are floating-point numbers, like 3.14159 or 49.1.
- **Booleans** − have only two possible values either true or false.
- **NULL** − is a special type that only has one value: NULL.
- **Strings** − are sequences of characters, like 'PHP supports string operations.'
- **Arrays** − are named and indexed collections of other values.
- **Objects** − are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- **Resources** − are special variables that hold references to resources external to PHP (such as database connections).

The first five are *simple types*, and the next two (arrays and objects) are compound - the compound types can package up other arbitrary values of arbitrary type, whereas the simple types cannot.

- **Integers:**

They are whole numbers, without a decimal point, like 4195. They are the simplest type .they correspond to simple whole numbers, both positive and negative. Integers can be assigned to variables, or they can be used in expressions,

```
$int_var = 12345;
$another_int = -12345 + 12345;
```

- **Doubles:**

They like 3.14159 or 49.1. By default, doubles print with the minimum number of decimal places needed. For example,

```
<?php
  $many = 2.2888800;
  $many_2 = 2.2111200;
  $few = $many + $many_2;

  print("$many + $many_2 = $few <br>");
?>
```

Output:

2.28888 + 2.21112 = 4.5

munotes.in

- **Boolean**

They have only two possible values either true or false. PHP provides a couple of constants especially for use as Booleans: TRUE and FALSE, which can be used like so

```
if (TRUE)
  print("This will always print<br>");
else
  print("This will never print<br>");
```

- **Interpreting other types as Booleans**

Here are the rules for determine the "truth" of any value not already of the Boolean type −

- If the value is a number, it is false if exactly equal to zero and true otherwise.
- If the value is a string, it is false if the string is empty (has zero characters) or is the string "0", and is true otherwise.
- Values of type NULL are always false.
- If the value is an array, it is false if it contains no other values, and it is true otherwise. For an object, containing a value means having a member variable that has been assigned a value.
- Valid resources are true (although some functions that return resources when they are successful will return FALSE when unsuccessful).
- Don't use double as Booleans.

Each of the following variables has the truth value embedded in its name when it is used in a Boolean context.

```
$true_num = 3 + 0.14159;
$true_str = "Tried and true"
$true_array[49] = "An array element";
$false_array = array();
$false_null = NULL;
$false_num = 999 - 999;
$false_str = " ";
```

NULL:

➔NULL is a special type that only has one value: NULL. To give a variable the NULL value, simply assign it like this –
$my_var = NULL;

➔The special constant NULL is capitalized by convention, but actually it is case insensitive; you could just as well have typed –
$my_var = null;

A variable that has been assigned NULL has the following properties −

It evaluates to FALSE in a Boolean context.

It returns FALSE when tested with IsSet() function.

Strings:

➔They are sequences of characters, like "PHP supports string operations". Following are valid examples of string

$string_1 = "This is a string in double quotes";

$string_2 = 'This is a somewhat longer, singly quoted string';

$string_39 = "This string has thirty-nine characters";

$string_0 = " "; // a string with zero characters

Singly quoted strings are treated almost literally, whereas doubly quoted strings replace variables with their values as well as specially interpreting certain character sequences.

```
<?php
  $variable = "name";
  $literally = 'My $variable will not print!';
  print($literally);
  print "<br>";
  $literally = "My $variable will print!";
  print($literally);
?>
```

Output:

My $variable will not print!

My name will print!

9.6 SUMMARY

➔PHP consists of a scripting language and an interpreter. Like other scripting languages, PHP enables web developers to define the behaviour and logic they need in a web page.

➔These scripts are embedded into the HTML documents that are served by the web server.

➔This chapter covered PHP is a server-side scripting language that is embedded in HTML. It is used to manage dynamic web pages.

➔This chapter we are more focuses on variables, types, syntax in php

munotes.in

9.7 REFERENCES

- https://www.studytonight.com/php/introduction-to-php
- https://www.brainvire.com/
- https://assets.ctfassets.net/
- https://www.w3schools.com/
- https://www.go4expert.com/
- https://www.tutorialspoint.com/

9.8 UNIT END EXERCISE

1. Write a program to print “Hello PHP” using php variable?
2. Write a program to print “Welcome to the PHP World” using some part of the text in variable & some part directly in echo.
3. Write a program to print 2 php variables using single echo statement.
4. Write a program to print “Welcome to my PHP World” using echo only?

munotes.in

10

CONTROL STRUCTURE AND LOOPING

Unit Structure :

10.0 Objectives
10.1 Introduction
10.2 Control structures
10.3 Branching
10.4 Looping
10.5 Termination
10.6 Summary
10.7 References
10.8 Unit End Exercise

10.0 OBJECTIVES

After reading through this chapter, you will be able to --

1. To understand the control structures in PHP.
2. Understand the branching concept.
3. To understand the loop control in PHP.
4. Understand the termination concept.

10.1INTRODUCTION

In simple terms, a control structure allows you to control the flow of code execution in your application. Generally, a program is executed sequentially, line by line, and a control structure allows you to alter that flow, usually depending on certain conditions. Control structures are core features of the PHP language that allow your script to respond differently to different inputs or situations. This could allow your script to give different responses based on user input, file contents, or some other data.

10.2 CONTROL STRUCTURES

Code execution can be grouped into categories as shown below

- **Sequential** – this one involves executing all the codes in the order in which they have been written.
- **Decision** – this one involves making a choice given a number of options. The code executed depends on the value of the condition.

munotes.in

A control structure is a block of code that decides the execution path of a program depending on the value of the set condition.

Following are the some of the control structures that PHP supports.

10.3 BRANCHING

PHP IF Else:
If… then... else is the simplest control structure. It evaluates the conditions using Boolean logic
When to use if… then… else

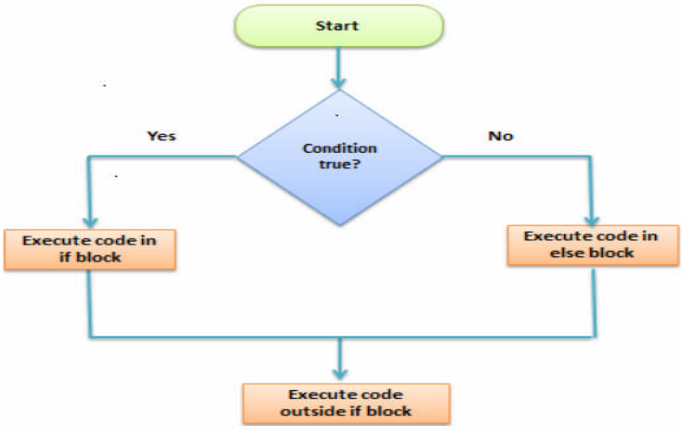
- You have a block of code that should be executed only if a certain condition is true
- You have two options, and you have to select one.
- If… then… else if… is used when you have to select more than two options and you have to select one or more

Syntax The syntax for if… then… else is;

```
<?php
if (condition is true) {
block one
                    }
else{
block two
}
?>
```

HERE,

- "**if (condition is true)**" is the control structure
- "**block one**" is the code to be executed if the condition is true
- **{…else…}** is the fallback if the condition is false
- "**block two**" is the block of code executed if the condition is false
  How it works The flow chart shown below illustrates how the if then… else control



Example The code below uses “if… then… else” to determine the larger value between two numbers.

```
<?php
$first_number = 7;
$second_number = 21;
if ($first_number> $second_number){
echo "$first_number is greater than $second_number";
}
else{
echo "$second_number is greater than $first_number";
}
?>
```

Output:
21 is greater than 7

PHP Switch Case:
Switch… case is similar to the if then… else control structure.
It only **executes** a single block of code depending on the **value** of the condition.

If no condition has been met then the default block of code is executed.

It has the following basic syntax.

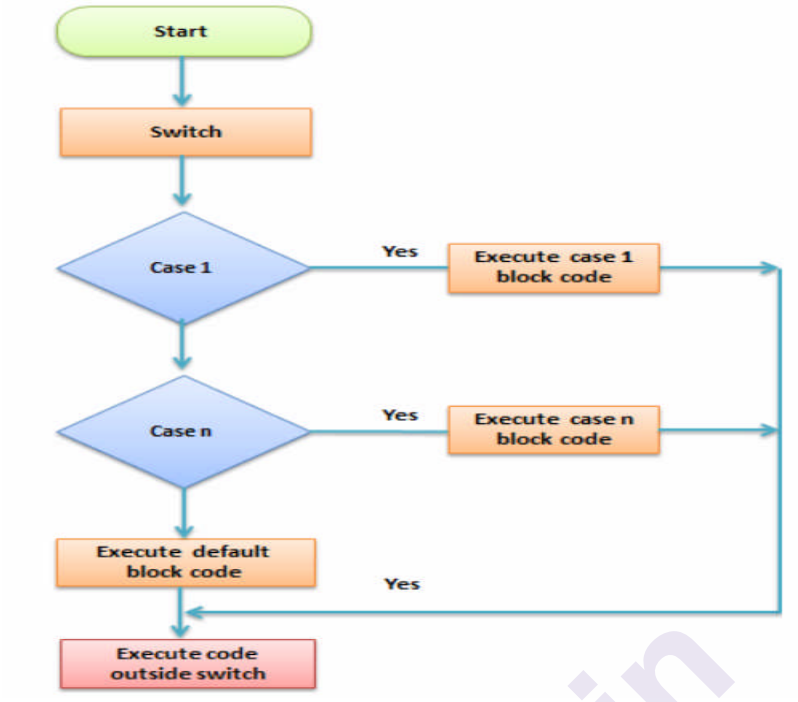
```
<?php
switch(condition){
case value:
//block of code to be executed
break;
case value2:
//block of code to be executed
break;
default:
//default block code
break;
}
?>
```

HERE,

- **“switch(…){…}”** is the control structure block code
- **“case value: case…”** are the blocks of code to be executed depending on the value of the condition
- **“default:”** is the block of code to be executed when no value matches with the condition

How it works The flow chart shown below illustrates how the switch control

munotes.in



Example: The code below uses the switch control structure to display a message depending on the day of the week.

```
<?php
$today = "wednesday";
switch($today){
case "sunday":
echo "pray for us sinners.";
break;
case "wednesday":
echo "out for dinner";
break;
case "saturday":
echo "take care as you go out tonight.";
break;
default:
echo "have a nice day at work";
break;
}
?>
```

Output: out for dinner

Summary

- Control structures are used to control the execution of the program
- The if then... else is when you have more than route block of code to execute depending on the value of the condition
- Switch… case is used to when you have a number of block codes, and you only have to execute one of them depending on the value of the set case.

munotes.in

10.4 PHP LOOP: FOR, FOREACH, WHILE, DO WHILE

A Loop is an Iterative Control Structure that involves executing the same number of code a number of times until a certain condition is met.

PHP For Loop

The above code outputs "21 is greater than 7" For loops For... loops execute the block of code a specifiednumber of times. There are basically two types of for loops;

- for
- for… each.

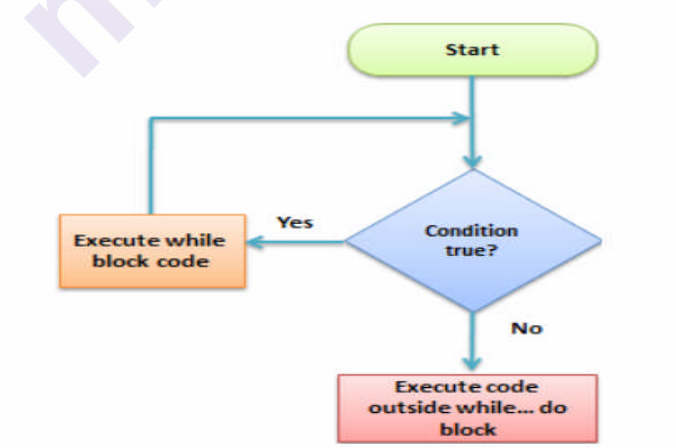
Let's now look at them separately. **For loop** It has the following basic **syntax**

```
<?php
for (initialize; condition; increment){
//code to be executed
}
?>
```

HERE,

- **"for…{…}"** is the loop block
- "**initialize**" usually an integer; it is used to set the counter's initial value.
- **"condition"** the condition that is evaluated for each php execution. If it evaluates to true then execution of the for... loop is terminated. If it evaluates to false, the execution of the for... loop continues.
- **"increment"** is used to increment the initial value of counter integer.

The flowchart shown below illustrates how for loop in php works



Example: The code below uses the "for… loop" to print values of multiplying 10 by 0 through to 10

```
<?php
for ($i = 0; $i< 10; $i++){
```

```
$product = 10 * $i;
echo "The product of 10 * $i is $product <br/>";
}
?>
```

Output:

The product of 10 x 0 is 0
The product of 10 x 1 is 10
The product of 10 x 2 is 20
The product of 10 x 3 is 30
The product of 10 x 4 is 40
The product of 10 x 5 is 50
The product of 10 x 6 is 60
The product of 10 x 7 is 70
The product of 10 x 8 is 80
The product of 10 x 9 is 90

PHP For Each loop:

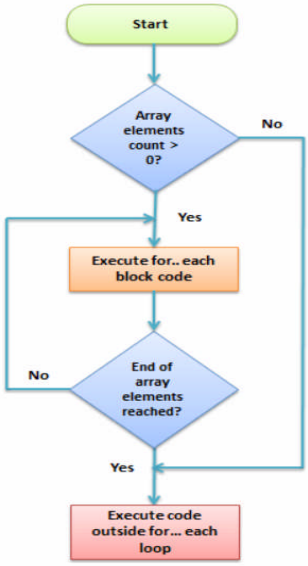
The php foreach loop is used to iterate through array values. It has the following basic syntax

```
<?php
foreach($array_variable  as $array_values){
block of code to be executed
}
?>
```

HERE,

- **"foreach(…){…}"** is the foreach php loop block code
- **"$array_data"** is the array variable to be looped through
- **"$array_value"** is the temporary variable that holds the current array item values.
- "block of code…" is the piece of code that operates on the array values

How it works The flowchart shown below illustrates how the for… each… loop works



Example The code below uses for… each loop to read and print the elements of an array.

```
<?php
$animals_list = array("Lion","Wolf","Dog","Leopard","Tiger");
foreach($animals_list as $array_values){
echo $array_values . "<br>";
}
?>
```

Output:
Lion
Wolf
Dog
Leopard
Tiger

Let's look at another example that loops through an **associative array**. An associative array uses alphanumeric words for access keys.

```
<?php
$persons = array("Mary" => "Female", "John" => "Male", "Mirriam" =>
"Female");
foreach($persons as $key => $value){
echo "$key is $value"."<br>";
}
?>
```

Note: The names have been used as array keys and gender as the values.
Output:
Mary is Female
John is Male
Mirriam is Female

While Loop:
PHP While loop
They are used to execute a block of code a repeatedly until the set condition gets satisfied

When to use while loops

- While loops are used to execute a block of code until a certain condition becomes true.
- You can use a while loop to read records returned from a database query.

Types of while loops

- **Do… while** - executes the block of code at least once before evaluating the condition
- **While…** - checks the condition first. If it evaluates to true, the block of code is executed as long as the condition is true. If it evaluates to false, the execution of the while loop is terminated.

munotes.in

While loop

It has the following syntax

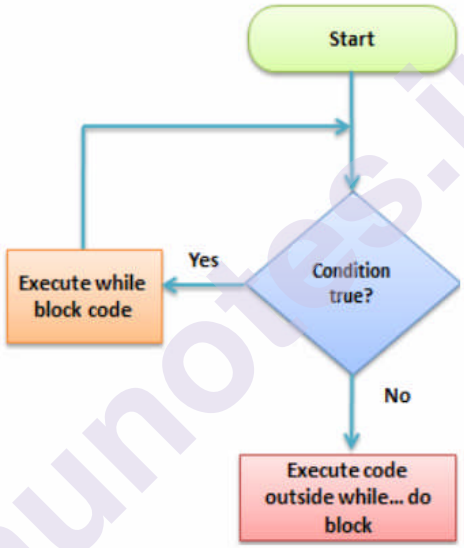
```
<?php
while (condition){
block of code to be executed;
}
?>
```

HERE,

- **"while(…){…}"** is the while loop block code
- **"condition"** is the condition to be evaluated by the while loop
- **"block of code…"** is the code to be executed if the condition gets satisfied

How it works

- The flow chart shown below illustrates how the while… loop works



The example below uses the while… loop to print numbers 1 to 10.

```
<?php
$i = 0;
while ($i<10){
echo $i + 1 . "<br>";
$i++;
}
?>
```

Output:

1
2
3
4
5
6
7

8
9
10

PHP Do While:

The difference between While… loop and Do… while loop is do… while is executed at-least once before the condition is evaluated.

Let's now look at the basic syntax of a do… while loop

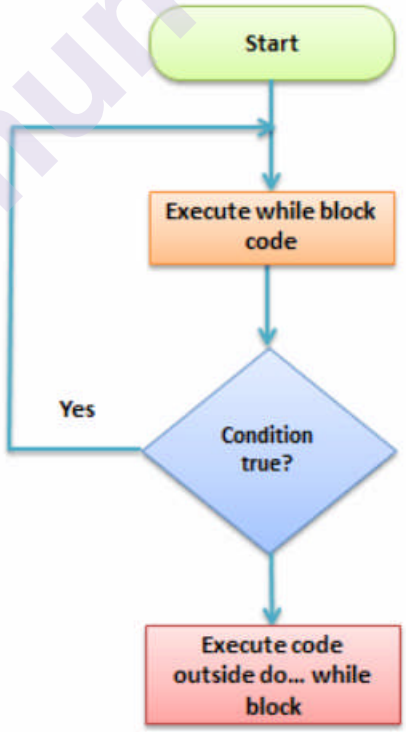
```
<?php
do{
block of code to be executed
}
?>
while(condition);
```

HERE,

- **"do{…} while(…)"** is the do… while loop block code
- **"condition"** is the condition to be evaluated by the while loop
- **"block of code…"** is the code that is executed at least once by the do… while loop

How it works

- The flow chart shown below illustrates how the while… loop works



munotes.in

Example
We are now going to modify the while… loop example and implement it using the do… while loop and set the counter initial value to 9.

The code below implements the above modified example

```
<?php
$i = 9;
do{
   echo "$i is"." <br>";
}
while($i< 9);
?>
```

Output:
9
Note the above example outputs 9 only.

This is because the do… while loop is executed at least once even if the set condition evaluates to false.

Summary

- The for… loop is used to execute a block of a specified number of times
- The foreach… loop is used to loop through arrays
- While… loop is used to execute a block of code as long as the set condition is made to be false
- The do… while loop is used to execute the block of code at least once then the rest of the execution is dependent on the evaluation of the set condition

munotes.in

10.5 TERMINATION (PHP BREAK, CONTINUE AND EXIT)

Break:

- A break statement can be used to terminate or to come out from the loop or conditional statement unconditionally.
- It can be used in switch statement to break and come out from the switch statement after each case expression.
- Whenever, break statement is encounter within the program then it will break the current loop or block.
- A break statement is normally used with if statement.
- When certain condition becomes true to terminate the loop then break statement can be used.

The following program demonstrates the use of break statement. Loop will be terminated as soon as the counter value becomes greater than 5.

```
<?php
for( $i = 1; $i<= 10 ; $i++ )
  {
      if ($i> 5)
break;    // terminate loop
      echo "$i"."</br>" ;
  }
?>
```

- *Break* can also be used with an optional numeric argument to specify how many nested enclosing structures are to be broken out of.
- The default value is *1*, means only the immediate enclosing structure is broken out of.

Continue:

- A continue statement can be used into the loop when we want to skip some statement to be executed and continue the execution of above statement based on some specific condition.
- Similar to break statement, continue is also used with if statement.
- When compiler encounters continue, statements after continue are skipped and control transfers to the statement above continue.

The following example uses the continue statement to print upper and lower a to z alphabets

```
<?php
  /* program to print upper and lower a to z alphabets using continue */
  for ( $i = 65 ; $i<= 122 ; $i++ )   // loop through ASCII value for a to z
  {
if($i>= 91 && $i<= 96)
continue ;       // skip unnecessary special characters.
        echo "| $i "."</br>" ;   // print character equivalent for ASCII value.
  }
?>
```

Exit:

- An *exit* statement is used to terminate the current execution flow.
- As soon as exit statement is found, it will terminate the program.

- It can be used to output a message and terminate the current script: for example exit("Good Bye!");
- It can also be used with error code. For example: exit(1), exit(0376).
- the following program demonstrates the use of exit statements.

```
<?php
  $filename = 'sample.txt' ;
  $file = fopen($filename, 'r')  // open file for reading
     or exit("unable to open file ($filename)");
?>
```

10.6 SUMMARY

➔ In this chapter we have studied control structure in PHP that can be grouped into two categories like sequential and decision.

➔PHP Control Structures. Any PHP script is built out of a series of statements. A statement can be an assignment, a function call, a loop, a conditional statement of even a statement that does nothing (an empty statement). Statements usually end with a semicolon.

➔In this chapter, we have learn the fundamentals of PHP programming. You'll start with the basics, learning how PHP works and writing simple PHP loops and functions.

➔Control structures are used to control the execution of the program. The if then... else is when you have more than route block of code to execute depending on the value of the condition

➔ Switch… case is used to when you have a number of block codes, and you only have to execute one of them depending on the value of the set case

10.7 REFERENCES

https://code.tutsplus.com/
https://www.guru99.com/
http://www.hexainclude.com/
https://www.brainvire.com/
https://assets.ctfassets.net/
https://www.w3schools.com/
https://www.go4expert.com/
https://www.tutorialspoint.com/

munotes.in

10.8 UNIT END EXERCISE

1. Write a simple calculator program in PHP using switch case (operation's like addition, substruction, multiplication, division)
2. Write a PHP program to check if a person is eligible to vote or not.

Conditions:

- Minimum age required for vote is 18.
- You can use PHP Functions.
- You can use Decision Making Statements.

3. Write a PHP program to check whether a number is positive, negative or zero.
4. Write a Program to display count, from 5 to 15 using PHP loop as given below.
5. Write a program to calculate factorial of a number using for loop in php.

munotes.in

11

FUNCTIONS AND METHODS

Unit Structure :

11.0 Objectives
11.1 Functions
11.2 Passing information with PHP GET, POST
11.3 Formatting form variables
11.4 Super global arrays
11.5 Strings and string functions
11.6 Summary
11.7 References
11.8 Unit End Exercise

11.0 OBJECTIVES

After reading through this chapter, you will be able to–

1. Understand the functions in PHP.
2. Understand the passing information with GET & POST methods in PHP.
3. Concept of super global arrays in PHP.
4. Understand the string and string functions in PHP.

11.1FUNCTIONS

The real power of PHP comes from its functions. PHP has more than 1000 built-in functions, and in addition you can create your own custom functions.

PHP Built-in Functions:

PHP has over 1000 built-in functions that can be called directly, from within a script, to perform a specific task.

PHP User Defined Functions:

Besides the built-in PHP functions, it is possible to create your own functions.

- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute automatically when a page loads.
- A function will be executed by a call to the function.

munotes.in

Create a User Defined Function in PHP:
A user-defined function declaration starts with the word function:

Syntax:

```
Function Name() {
  code to be executed;
}
```

Note: A function name must start with a letter or an underscore. Function names are NOT case-sensitive.

In the example below, we create a function named "write Msg()". The opening curly brace ({) indicates the beginning of the function code, and the closing curly brace (}) indicates the end of the function. The function outputs "Welcome to IDOL!". To call the function, just write its name followed by brackets ():

```
<!DOCTYPE html>
<html>
<body>
<?php
function writeMsg() {
  echo "Welcome to IDOL!";
}
writeMsg();
?>
</body>
</html>
```

Output:
Welcome to IDOL!

munotes.in

PHP Function Arguments:

- Information can be passed to functions through arguments. An argument is just like a variable.
- Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.
- The following example has a function with one argument ($fname). When the family Name() function is called, we also pass along a name (e.g. Rahul), and the name is used inside the function, which outputs several different first names, but an equal last name:

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

<?php
function familyName($fname) {
  echo "$fname Dixit.<br>";
}

familyName("Rahul");
familyName("Hemant");
familyName("Ajay");
familyName("Rajesh");
familyName("Mukesh");
?>

</body>
</html>
```

Output:
Rahul Dixit.
Hemant Dixit.
Ajay Dixit.
Rajesh Dixit.
Mukesh Dixit.

The following example has a function with two arguments ($fname and $year):

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

<?php
function familyName($fname, $year) {
  echo "$fname Navale. Born in $year <br>";
}

familyName("Rahul","1975");
familyName("Hemant","1978");
familyName("Mukesh","1983");
?>

</body>
</html>
```

Output:
Rahul Navale. Born in 1975
Hemant Navale. Born in 1978
Mukesh Navale. Born in 1983

munotes.in

PHP is a Loosely Typed Language:

In the example above, notice that we did not have to tell PHP which data type the variable is.

PHP automatically associates a data type to the variable, depending on its value. Since the data types are not set in a strict sense, you can do things like adding a string to an integer without causing an error.

In PHP 7, type declarations were added. This gives us an option to specify the expected data type when declaring a function, and by adding the strict declaration, it will throw a "Fatal Error" if the data type mismatches.

In the following example we try to send both a number and a string to the function without using strict.

```
<?php
function add Numbers(int $a, int $b) {
  return $a + $b;
}
echo add Numbers(5, "5 days");
// since strict is NOT enabled "5 days" is changed to int(5), and it will return 10
?>
```

Output:
10

To specify strict we need to set declare(strict_types=1);. This must be on the very first line of the PHP file.

In the following example we try to send both a number and a string to the function, but here we have added the strict declaration

```
<?php declare(strict_types=1); // strict requirement
function add Numbers(int $a, int $b) {
  return $a + $b;
}
echo addNumbers(5, "5 days");
// since strict is enabled and "5 days" is not an integer, an error will be thrown
?>
```

Output:PHP Fatal error

PHP Default Argument Value:

The following example shows how to use a default parameter. If we call the function setHeight() without arguments it takes the default value as argument:

```
<?php declare(strict_types=1); // strict requirement ?>
<!DOCTYPE html>
<html>
<body>
```

```
<?php
function setHeight(int $minheight = 50) {
  echo "The height is : $minheight<br>";
}
setHeight(350);
setHeight();
setHeight(135);
setHeight(80);
?>
</body>
</html>
```

Output:
The height is : 350
The height is : 50
The height is : 135
The height is : 80

PHP Functions - Returning values
To let a function return a value, use the return statement:

```
<?php declare(strict_types=1); // strict requirement ?>
<!DOCTYPE html>
<html>
<body>
<?php
function sum(int $x, int $y) {
  $z = $x + $y;
  return $z;
}
echo "5 + 10 = " .sum(5,10) . "<br>";
echo "7 + 13 = " .sum(7,13) . "<br>";
echo "2 + 4 = " .sum(2,4);
?>
</body>
</html>
```

Output:
5 + 10 = 15
7 + 13 = 20
2 + 4 = 6

PHP Return Type Declarations:
➔PHP 7 also supports Type Declarations for the return statement. Like with the type declaration for function arguments, by enabling the strict requirement, it will throw a "Fatal Error" on a type mismatch.

➔To declare a type for the function return, add a colon (:) and the type right before the opening curly ({)bracket when declaring the function.

➔In the following example we specify the return type for the function

```
<?php declare(strict_types=1); // strict requirement
function addNumbers(float $a, float $b) : float {
  return $a + $b;
}
echo addNumbers(1.2, 5.2);
?>
```

Output:
6.4

You can specify a different return type, than the argument types, but make sure the return is the correct type

```
<?php declare(strict_types=1); // strict requirement
function addNumbers(float $a, float $b) : int {
  return (int)($a + $b);
}
echo addNumbers(1.2, 5.2);
?>
```

Output:
6

Passing Arguments by Reference:

➔In PHP, arguments are usually passed by value, which means that a copy of the value is used in the function and the variable that was passed into the function cannot be changed.

➔When a function argument is passed by reference, changes to the argument also change the variable that was passed in. To turn a function argument into a reference, the & operator is used.

➔ Following example uses a pass-by-reference argument to update a variable.

```
<!DOCTYPE html>
<html>
<body>
<?php
function add_five(&$value) {
  $value += 5;
}
$num = 2;
add_five($num);
echo $num;
?>
</body>
</html>
```

Output:
7

munotes.in

11.2 PASSING INFORMATION WITH PHP GET, POST

➔There are two ways the browser client can send information to the web server.

- The GET Method
- The POST Method

➔Before the browser sends the information, it encodes it using a scheme called URL encoding. In this scheme, name/value pairs are joined with equal signs and different pairs are separated by the ampersand.

name1=value1&name2=value2&name3=value3

➔ Spaces are removed and replaced with the + character and any other no alphanumeric characters are replaced with a hexadecimal value. After the information is encoded it is sent to the server.

The GET Method:

➔The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the? character.

➔The GET method produces a long string that appears in your server logs, in the browser's Location: box.

➔The GET method is restricted to send upto 1024 characters only.

➔Never use GET method if you have password or other sensitive information to be sent to the server.

➔GET can't be used to send binary data, like images or word documents, to the server.

➔The data sent by GET method can be accessed using QUERY_STRING environment variable.

➔The PHP provides $_GET associative array to access all the sent information using GET method.

➔ The following example by putting the source code in test.php script.

```
<?php
if( $_GET["name"] || $_GET["age"] ) {
     echo "Welcome ". $_GET['name']. "<br />";
     echo "You are ". $_GET['age']. " years old.";
```

munotes.in

```
exit();
  }
?>
<html>
<body>

<form action = "<?php $_PHP_SELF ?>" method = "GET">
        Name: <input type = "text" name = "name" />
        Age: <input type = "text" name = "age" />
<input type = "submit" />
</form>

</body>
</html>
```

Output:

Name: [ ] Age: [ ] Submit

The POST Method:

➔The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.

- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides **$_POST** associative array to access all the sent information using POST method.
- The following example by putting the source code in test.php script.

```
<?php
if( $_POST["name"] || $_POST["age"] ) {
     if (preg_match("/[^A-Za-z'-]/",$_POST['name'] )) {
        die ("invalid name and name should be alpha");
     }
     echo "Welcome ". $_POST['name']. "<br />";
     echo "You are ". $_POST['age']. " years old.";
exit();
  }
?>
<html>
<body>
```

```
<form action = "<?php $_PHP_SELF ?>" method = "POST">
        Name: <input type = "text" name = "name" />
        Age: <input type = "text" name = "age" />
<input type = "submit" />
</form>

</body>
</html>
```

Output:

Name: [ ] Age: [ ] Submit

11.3 FORMATTING FORM VARIABLES

➔This shows how to keep the values in the input fields when the user hits the submit button.

PHP - Keep The Values in The Form:

➔To show the values in the input fields after the user hits the submit button, we add a little PHP script inside the value attribute of the following input fields: name, email, and website.

➔In the comment textarea field, we put the script between the <textarea> and </textarea> tags. The little script outputs the value of the $name, $email, $website, and $comment variables.

➔ Then, we also need to show which radio button that was checked. For this, we must manipulate the checked attribute.

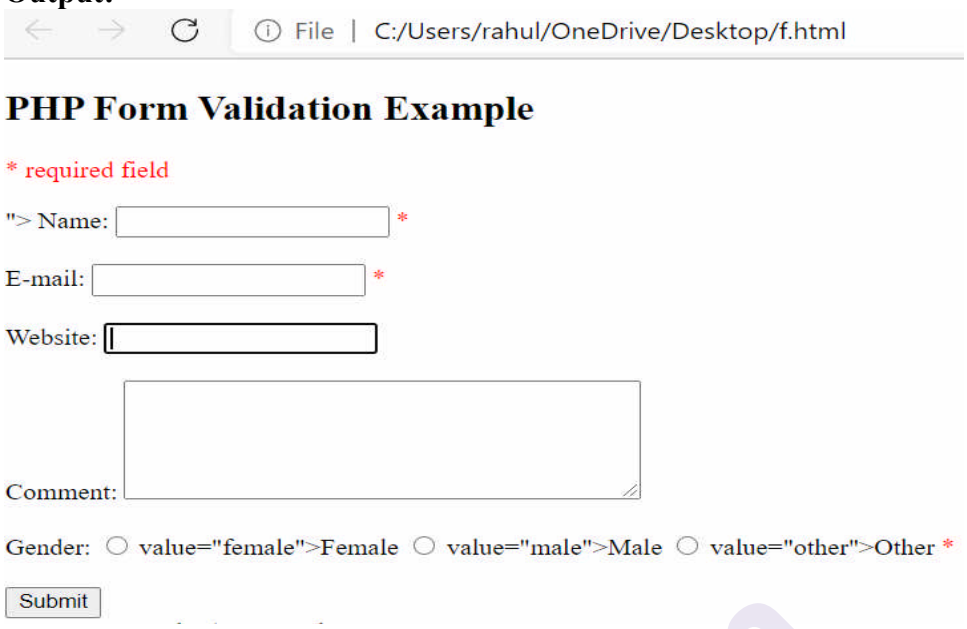
```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  if (empty($_POST["name"])) {
    $nameErr = "Name is required";
  } else {
    $name = test_input($_POST["name"]);
    // check if name only contains letters and whitespace
```

```
    if (!preg_match("/^[a-zA-Z-' ]*$/",$name)) {
      $nameErr = "Only letters and white space allowed";
    }
  }


  if (empty($_POST["email"])) {
    $emailErr = "Email is required";
  } else {
    $email = test_input($_POST["email"]);
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
      $emailErr = "Invalid email format";
    }
  }

  if (empty($_POST["website"])) {
    $website = "";
  } else {
    $website = test_input($_POST["website"]);
    // check if URL address syntax is valid (this regular expression also
allows dashes in the URL)
    if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-
9+&@#\/%?=~_|!:,.;]*[-a-z0-9+&@#\/%=~_|]/i",$website)) {
      $websiteErr = "Invalid URL";
    }
  }

  if (empty($_POST["comment"])) {
    $comment = "";
  } else {
    $comment = test_input($_POST["comment"]);
  }

  if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
  } else {
    $gender = test_input($_POST["gender"]);
  }
}

function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?>

<h2>PHP Form Validation Example</h2>
```

```
<p><span class="error">* required field</span></p>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
  Name: <input type="text" name="name" value="<?php echo
$name;?>">
<span class="error">* <?php echo $nameErr;?></span>
<br><br>
  E-mail: <input type="text" name="email" value="<?php echo
$email;?>">
<span class="error">* <?php echo $emailErr;?></span>
<br><br>
  Website: <input type="text" name="website" value="<?php echo
$website;?>">
<span class="error">><?php echo $websiteErr;?></span>
<br><br>
  Comment:<textareaname="comment" rows="5" cols="40"><?php
echo $comment;?></textarea>
<br><br>
  Gender:
<input type="radio" name="gender" <?php if (isset($gender) &&
$gender=="female") echo "checked";?> value="female">Female
<input type="radio" name="gender" <?php if (isset($gender) &&
$gender=="male") echo "checked";?> value="male">Male
<input type="radio" name="gender" <?php if (isset($gender) &&
$gender=="other") echo "checked";?> value="other">Other
<span class="error">* <?php echo $genderErr;?></span>
<br><br>
<input type="submit" name="submit" value="Submit">
</form>

<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>

</body>
</html>
```

munotes.in

Output:



11.4 SUPERGLOBAL ARRAYS

➔Some predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

The PHP superglobal variables are:

- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET
- $_FILES
- $_ENV
- $_COOKIE
- $_SESSION

PHP $GLOBALS:

➔$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script also from within functions or methods.

➔PHP stores all global variables in an array called $GLOBALS[index]. The index holds the name of the variable.

➔The example below shows how to use the super global variable $GLOBALS.

```
<html>
<body>
<?php
$x = 75;
```

```
$y = 25;
function addition() {
  $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
addition();
echo $z;
?>
</body>
</html>
```

Output:
100

In the example above, since z is a variable present within the $GLOBALS array, it is also accessible from outside the function

PHP $_SERVER:

➔$_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

➔The example below shows how to use some of the elements in $_SERVER.

```
<html>
<body>

<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
</body>
</html>
```

Output:
/demo/demo_global_server.php
35.194.26.41
35.194.26.41
showphp.php?filename=demo_global_server

Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/88.0.4324.190 Safari/537.36
/demo/demo_global_server.php

munotes.in

The following table lists the most important elements that can go inside $_SERVER

Element/Code	Description
$_SERVER['PHP_SELF']	Returns the filename of the currently executing script
$_SERVER['SERVER_NAME']	Returns the name of the host server
$_SERVER['HTTP_HOST']	Returns the Host header from the current request
$_SERVER['HTTP_REFERER']	Returns the complete URL of the current page
$_SERVER['SCRIPT_NAME']	Returns the path of the current script

PHP $_REQUEST:

➔PHP $_REQUEST is a PHP super global variable which is used to collect data after submitting an HTML form.

➔ The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag.

➔In this example, we point to this file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice. Then, we can use the super global variable $_REQUEST to collect the value of the input field.

```
<html>
<body>
<form method="post" action="<?php echo
$_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
<input type="submit">
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = htmlspecialchars($_REQUEST['fname']);
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>
</body>
</html>
```

Output:

Name: [ ] Submit

PHP $_POST:

➔PHP $_POST is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.

➔The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag.

➔In this example, we point to the file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice. Then, we can use the super global variable $_POST to collect the value of the input field.

```
<html>
<body>
<form method="post" action="<?php echo
$_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
<input type="submit">
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   // collect value of input field
   $name = $_POST['fname'];
   if (empty($name)) {
      echo "Name is empty";
   } else {
      echo $name;
   }
}
?>
</body>
</html>
```

Output:

Name: [ ] Submit

PHP $_GET:

➔PHP $_GET is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".

➔$_GET can also collect data sent in the URL.

➔ Assume we have an HTML page that contains a hyperlink with parameters

```
<html>
<body>
<a href="test_get.php?subject=PHP&web=gmail.com">Test $GET</a>
</body>
</html>
```

➔ When a user clicks on the link "Test $GET", the parameters "subject" and "web" are sent to "test_get.php", and you can then access their values in "test_get.php" with $_GET.

➔The example below shows the code in "test_get.php"

```
<html>
<body>
<a href="test_get.php?subject=PHP&web=gmail.com">Test $GET</a>
</body>
</html>
```

Output:
Test $GET

11.5 STRINGS AND STRING FUNCTIONS

➔A string is a collection of characters. String is one of the data types supported by PHP.

➔The string variables can contain alphanumeric characters. Strings are created when,

- You declare variable and assign string characters to it
- You can directly use PHP Strings with echo statement.
- PHP String functions are language construct, it helps capture words.

PHP Create Strings Using Single quotes with Example:
➔Creating PHP Strings Using Single quotes: The simplest way to create a string is to use single quotes.

➔Let's look at an example that creates a simple string in PHP.

```
<?php
   var_dump('You need to be logged in to view this page');
?>
```

Output:

string(42) "You need to be logged in to view this page"

➔If the single quote is part of the string value, it can be escaped using the backslash.The code below illustrates how to escape a single quote.

```
<?php
echo 'I \'ll be back after 20 minutes';
?>
```

Output:
I'll be back after 20 minutes

PHP Create Strings Using Double quotes with Example:
➔The double quotes are used to create relatively complex strings compared to single quotes.

➔Variable names can be used inside double quotes and their values will be displayed.

Let's look at an example.

```
<?php
$name='Pune';
echo "$name is in Maharashtra";
?>
```

HERE,
The above example creates a simple string with the value of Pune.

The variable name is then used in the string created using double quotes and its value is interpolated at run time.

Output:
Pune is in Maharashtra

Basic PHP String Functions:
Below are the some basic string functions which are most commonly used in PHP scripts.

1.Getting length of a String:

➔PHP has a predefined function to get the length of a string. Strlen() displays the length of any string.

➔It is more commonly used in validating input fields where the user is limited to enter a fixed length of characters.

Syntax:
Strlen(string);

Example:

```
<?php
echo strlen("Welcome to IDOL");//will return the length of given string
?>
```

2. Counting of the number of words in a String:
➔ Another function which enables display of the number of words in any specific string is str_word_count(). This function is also useful in validation of input fields.

Syntax
Str_word_count(string)
Example:
Example
<?php
echo str_word_count("Welcome to Pune");//will return the number of words in a string
?>

Output: 3

3. Reversing a String:

➔Strrev() is used for reversing a string. You can use this function to get the reverse version of any string.

Syntax:
Strev(string)
Example:
<?php
echo strrev("Welcome to pune");// will return the string starting from the end
?>

Output:
enup ot emocleW

4. Finding Text Within a String:

➔Strpos() enables searching particular text within a string. It works simply by matching the specific text in a string.

➔If found, then it returns the specific position. If not found at all, then it will return "False". Strops() is most commonly used in validating input fields like email.

Syntax:
Strpos(string,text);
Example
<?php
echo strpos("Welcome to Sinhgad","Sinhgad");
?>

Output:
7

5. Replacing text within a string

➔Str_replace() is a built-in function, basically used for replacing specific text within a string.

Syntax:
Str_replace(string to be replaced,text,string)

munotes.in

Example:
```
<?php
echo str_replace("Pune", "the programming world", "Welcome to Pune");
?>
```

Output: Welcome to the programming world

6. Converting lowercase into Title Case

➔Ucwords() is used to convert first alphabet of every word into uppercase.

Syntax: Ucwords(string)

Example:
```
<?php
echo ucwords("welcome to the php world");
?>
```

Output:
Welcome To The Php World

7. Converting a whole string into UPPERCASE

➔Strtoupper() is used to convert a whole string into uppercase.

Syntax: Strtoupper(string);

Example
```
<?php
echo strtoupper("welcome to sinhgad");// It will convert all letters of string into uppercase
?>
```

Output: WELCOME TO SINHGAD

8. Converting whole String to lowercase

➔Strtolower() is used to convert a string into lowercase.

Syntax: Strtolower(string)

Example:
```
<?php
echo strtolower("WELCOME TO PUNE");
?>
```
Output:
welcome to pune

munotes.in

11.6 SUMMARY

➔ In this chapter we studied PHP user defined functions. Besides the built-in PHP functions, it is possible to create your own functions.

➔A function is a block of statements that can be used repeatedly in a program. A function will not execute automatically when a page load. A function will be executed by a call to the function.

➔There are two ways the browser client can send information to the web server that is GET and POST methods that we studied in this chapter.

➔Also, we studied string in PHP various methods of string and functions

11.7 REFERENCES

https://www.cloudways.com/
https://www.guru99.com/
https://www.studytonight.com/php/introduction-to-php
https://www.brainvire.com/
https://assets.ctfassets.net/
https://www.w3schools.com/
https://www.go4expert.com/
https://www.tutorialspoint.com/

11.8 UNIT END EXERCISE

1. Write a PHP program to calculate area of rectangle by using PHP Function.
2. Write a PHP program to find factorial of a number using recursive function.
3. Write a PHP program to reverse the string.
4. Write a PHP program to find the length of the string.
5. Write a PHP program to count the words in the string.
6. Write a PHP program to convert all the characters inside the string into uppercase.(use built-in string function).

❖❖❖❖

munotes.in

12

REGULAR EXPRESSION AND ARRAYS

Unit Structure :

12.0 Objectives
12.1 Introduction
12.2 Regular Expressions
12.3 Arrays
12.4 Number handling
12.5 Basic PHP errors
12.6 Summary
12.7 References
12.8 Unit End Exercise

12.0 OBJECTIVES

After reading through this chapter, you will be able to –

1. Understand the regular expressions.
2. Understand the concept of arrays in PHP.
3. Understand the number handling in PHP.
4. Understand the basic PHP errors.

12.1INTRODUCTION

➔**PHP Regular Expression** also known as regex are powerful pattern matching algorithm that can be performed in a single expression.

➔ Regular expressions use arithmetic operators such as (+,-,^) to create complex expressions. They can help you accomplish tasks such as validating email addresses, IP address etc.

Why use regular expressions:

➔PHP Regular expressions simplify identifying patterns in string data by calling a single function. This saves us coding time.

➔When validating user input such as email address, domain names, telephone numbers, IP addresses, Highlighting keywords in search results.

➔When creating a custom HTML template. Regex in PHP can be used to identify the template tags and replace them with actual data.

munotes.in

12.2 REGULAR EXPRESSIONS

➔Regular Expressions, commonly known as "**regex**" or "**RegExp**", are a specially formatted text strings used to find patterns in text.

➔Regular expressions are one of the most powerful tools available today for effective and efficient text processing and manipulations.

➔ For example, it can be used to verify whether the format of data i.e. name, email, phone number, etc. entered by the user was correct or not, find or replace matching string within text content, and so on.

➔PHP supports Perl style regular expressions via its preg_ family of functions. Why Perl style regular expressions? Because Perl (Practical Extraction and Report Language) was the first mainstream programming language that provided integrated support for regular expressions and it is well known for its strong support of regular expressions and its extraordinary text processing and manipulation capabilities.

➔Let's begin with a brief overview of the commonly used PHP's built-in pattern-matching functions before delving deep into the world of regular expressions.

Function	What it Does
preg_match()	Perform a regular expression match.
preg_match_all()	Perform a global regular expression match.
preg_replace()	Perform a regular expression search and replace.
preg_grep()	Returns the elements of the input array that matched the pattern.
preg_split()	Splits up a string into substrings using a regular expression.
preg_quote()	Quote regular expression characters found within a string.

Regular Expression Syntax

➔Regular expression syntax includes the use of special characters. The characters that are given special meaning within a regular expression, are . * ? + [ ] ( ) { } ^ $ | \.

Character Classes:

➔ Square brackets surrounding a pattern of characters are called a character class e.g. [abc]. A character class always matches a single character out of a list of specified characters that means the expression [abc] matches only a, b or c character.

➔Negated character classes can also be defined that match any character except those contained within the brackets. A negated character class is

defined by placing a caret. (^) symbol immediately after the opening bracket, like this [^abc].

➔You can also define a range of characters by using the hyphen (-) character inside a character class, like [0-9]. Let's look at some examples of character classes:

RegExp	What it Does
[abc]	Matches any one of the characters a, b, or c.
[^abc]	Matches any one character other than a, b, or c.
[a-z]	Matches any one character from lowercase a to lowercase z.
[A-Z]	Matches any one character from uppercase a to uppercase z.
[a-Z]	Matches any one character from lowercase a to uppercase Z.
[0-9]	Matches a single digit between 0 and 9.
[a-z0-9]	Matches a single character between a and z or between 0 and 9.

The following example will show you how to find whether a pattern exists in a string or not using the regular expression and PHP preg_match() function.

```
<html>
<head>
<title>Match a Pattern against a String Using RegExp in PHP</title>
</head>
<body>
<?php
$pattern = "/ca[kf]e/";
$text = "He was eating cake in the cafe.";
if(preg_match($pattern, $text)){
   echo "Match found!";
} else{
   echo "Match not found.";
}
?>
</body>
</html>
```

Output:
Match found!

Similarly, you can use the preg_match_all() function to find all matches within a string.

```
<html>
<head>
<title>Find all Occurrences of a Pattern in a String Using RegExp in PHP</title>
```

```
</head>
<body>

<?php
$pattern = "/ca[kf]e/";
$text = "He was eating cake in the cafe.";
$matches = preg_match_all($pattern, $text, $array);
echo $matches . " matches were found."
?>

</body>
</html>
```

Output: 2 matches were found.

Predefined Character Classes:

➔Some character classes such as digits, letters, and whitespaces are used so frequently that there are shortcut names for them. The following table lists those predefined character classes:

Shortcut	What it Does
.	Matches any single character except newline \n.
\d	matches any digit character. Same as [0-9]
\D	Matches any non-digit character. Same as [^0-9]
\s	Matches any whitespace character (space, tab, newline or carriage return character). Same as [\t\n\r]
\S	Matches any non-whitespace character. Same as [^ \t\n\r]
\w	Matches any word character (definned as a to z, A to Z,0 to 9, and the underscore). Same as [a-zA-Z_0-9]
\W	Matches any non-word character. Same as [^a-zA-Z_0-9]

The following example will show you how to find and replace space with a hyphen character in a string using regular expression and PHP preg_replace() function

```
<html>
<head>
<title>Find and Replace Characters in a String Using RegExp in PHP</title>
</head>
<body>

<?php
$pattern = "/\s/";
$replacement = "-";
$text = "Earth revolves around\nthe\tSun";
```

```
// Replace spaces, newlines and tabs
echo preg_replace($pattern, $replacement, $text);
echo "<br>";
// Replace only spaces
echo str_replace(" ", "-", $text);
?>

</body>
</html>
```

Output:
Earth-revolves-around-the-Sun
Earth-revolves-around the Sun

Repetition Quantifiers:

➔if you want to match on more than one character? For example, let's say you want to find out words containing one or more instances of the letter p, or words containing at least two p's, and so on.

➔This is where quantifiers come into play. With quantifiers you can specify how many times a character in a regular expression should match. The following table lists the various ways to quantify a particular pattern

RegExp	What it Does
p+	Matches one or more occurrences of the letter p.
p*	Matches zero or more occurrences of the letter p.
p?	Matches zero or one occurrences of the letter p.
p{2}	Matches exactly two occurrences of the letter p.
p{2,3}	Matches at least two occurrences of the letter p, but not more than three occurrences of the letter p.
p{2,}	Matches two or more occurrences of the letter p.
p{,3}	Matches at most three occurrences of the letter p

The regular expression in the following example will splits the string at comma, sequence of commas, whitespace, or combination thereof using the PHP preg_split() function.

```
<html>
<head>
<title>Split a String Using RegExp in PHP</title>
</head>
<body>

<?php
$pattern = "/[\s,]+/";
```

```
$text = "My favourite colors are red, green and blue";
$parts = preg_split($pattern, $text);

// Loop through parts array and display substrings
foreach($parts as $part){
   echo $part . "<br>";
}
?>
</body>
</html>
```

Output:
My
favourite
colors
are
red
green
and
blue

Position Anchors:
➔There are certain situations where you want to match at the beginning or end of a line, word, or string.

➔To do this you can use anchors. Two common anchors are caret (^) which represent the start of the string, and the dollar ($) sign which represent the end of the string.

RegExp	What it Does
^p	Matches the letter p at the beginning of a line.
p$	Matches the letter p at the end of a line.

The regular expression in the following example will display only those names from the names array which start with the letter "J" using the PHP preg_grep() function.

```
<html>
<head>
<title>Match Strings Beginning with Specific Characters Using RegExp in PHP</title>
</head>
<body>

<?php
$pattern = "/^s/";
```

```
$names = array("sinhgad", "rajgad", "shivneri");
$matches = preg_grep($pattern, $names);

// Loop through matches array and display matched names
foreach($matches as $match){
   echo $match . "<br>";
}
?>
</body>
</html>
```

Output:
sinhgad
shivneri

12.3 ARRAYS

➔An array is a data structure that stores one or more similar type of values in a single value. For example, if you want to store 100 numbers then instead of defining 100 variables it's easy to define an array of 100 length.

➔There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

- **Numeric array** − An array with a numeric index. Values are stored and accessed in linear fashion.
- **Associative array** − An array with strings as index. This store element values in association with key values rather than in a strict linear index order.
- **Multidimensional array** − An array containing one or more arrays and values are accessed using multiple indices

Numeric Array:
➔These arrays can store numbers, strings and any object but their index will be represented by numbers. By default, array index starts from zero.

➔Example
Following is the example showing how to create and access numeric arrays. Here we have used array() function to create array.

```
<html>
<body>
<?php
      /* First method to create array. */
      $numbers = array( 1, 2, 3, 4, 5);
foreach( $numbers as $value ) {
         echo "Value is $value <br />";
      }
```

```
        /* Second method to create array. */
        $numbers[0] = "one";
        $numbers[1] = "two";
        $numbers[2] = "three";
        $numbers[3] = "four";
        $numbers[4] = "five";
foreach( $numbers as $value ) {
          echo "Value is $value <br />";
        }
     ?>
</body>
</html>
```

Output:

Value is 1
Value is 2
Value is 3
Value is 4
Value is 5
Value is one
Value is two
Value is three
Value is four
Value is five

Associative Arrays:

➔The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index.

➔Associative array will have their index as string so that you can establish a strong association between key and values.

➔To store the salaries of employees in an array, a numerically indexed array would not be the best choice. Instead, we could use the employee's names as the keys in our associative array, and the value would be their respective salary.

Example:

```
<html>
<body>
<?php
        /* First method to associate create array. */
        $salaries = array("Rahul" => 2000, "qadir" => 1000, "zara" => 500);
        echo "Salary of Rahul is ". $salaries['Rahul'] . "<br />";
        echo "Salary of qadir is ".  $salaries['qadir']. "<br />";
        echo "Salary of zara is ".  $salaries['zara']. "<br />";

        /* Second method to create array. */
        $salaries['Rahul'] = "high";
        $salaries['qadir'] = "medium";
```

```
        $salaries['zara'] = "low";
        echo "Salary of Rahul is ". $salaries['Rahul'] . "<br />";
        echo "Salary of qadir is ".  $salaries['qadir']. "<br />";
        echo "Salary of zara is ".  $salaries['zara']. "<br />";
    ?>
</body>
</html>
```

Output:
Salary of Rahul is 2000
Salary of qadir is 1000
Salary of zara is 500
Salary of Rahul is high
Salary of qadir is medium
Salary of zara is low

Multidimensional Arrays:

➔A multi-dimensional array each element in the main array can also be an array. And each element in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

➔Example: In this example we create a two-dimensional array to store marks of three students in three subjects

```
<html>
<body>
<?php
        $marks = array(
          "Rahul" => array (
            "physics" => 35,
            "maths" => 30,
            "chemistry" => 39
          ),
          "qadir" => array (
            "physics" => 30,
            "maths" => 32,
            "chemistry" => 29
          ),
          "zara" => array (
            "physics" => 31,
            "maths" => 22,
            "chemistry" => 39
          )
        );
        /* Accessing multi-dimensional array values */
        echo "Marks for Rahul in physics : " ;
        echo $marks['Rahul']['physics'] . "<br />";
        echo "Marks for qadir in maths : ";
        echo $marks['qadir']['maths'] . "<br />";
        echo "Marks for zara in chemistry : " ;
        echo $marks['zara']['chemistry'] . "<br />";
```

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

Output:
Marks for Rahul in physics : 35
Marks for qadir in maths : 32
Marks for zara in chemistry : 39

12.4 NUMBER HANDLING

➔One thing to notice about PHP is that it provides automatic data type conversion.

➔So, if you assign an integer value to a variable, the type of that variable will automatically be an integer. Then, if you assign a string to the same variable, the type will change to a string.

➔This automatic conversion can sometimes break your code.

PHP Integers:
➔An integer is a number without any decimal part.2, 256, -256, 10358, -179567 are all integers. While 7.56, 10.0, 150.67 are floats.

➔So, an integer data type is a non-decimal number between -2147483648 and 2147483647. A value greater (or lower) than this, will be stored as float, because it exceeds the limit of an integer.

Here are some rules for integers:

- An integer must have at least one digit
- An integer must not have a decimal point
- An integer can be either positive or negative
- Integers can be specified in three formats: decimal (10-based), hexadecimal (16-based - prefixed with 0x) or octal (8-based - prefixed with 0)

PHP has the following functions to check if the type of a variable is integer:

- is_int()
- is_integer() - alias of is_int()
- is_long() - alias of is_int()

Example
Check if the type of a variable is integer:

```
<html>
<body>
<?php
```

```
// Check if the type of a variable is integer
$x = 5985;
var_dump(is_int($x));
echo "<br>";
// Check again...
$x = 59.85;
var_dump(is_int($x));
?>
</body>
</html>
```

Output:
bool(true)
bool(false)

PHP Floats:

➔A float is a number with a decimal point or a number in exponential form. 2.0, 256.4, 10.358, 7.64E+5, 5.56E-5 are all floats.

➔The float data type can commonly store a value up to 1.7976931348623E+308 and have a maximum precision of 14 digits.

➔PHP has the following functions to check if the type of a variable is float:
is_float()
is_double() - alias of is_float()

Example

Check if the type of a variable is float:

```
<html>
<body>
<?php
// Check if the type of a variable is float
$x = 10.365;
var_dump(is_float($x));
?>
</body>
</html>
```

Output:
bool(true)

PHP Infinity:

➔A numeric value that is larger than PHP_FLOAT_MAX is considered infinite.

➔PHP has the following functions to check if a numeric value is finite or infinite:
is_finite()
is_infinite()

However, the PHP var_dump() function returns the data type and value:

Example
Check if a numeric value is finite or infinite:

```
<html>
<body>
<?php
// Check if a numeric value is finite or infinite
$x = 1.9e411;
var_dump($x);
?>
</body>
</html>
```

Output:
float(INF)

PHP NaN:
➔NaN stands for Not a Number.NaN is used for impossible mathematical operations.

➔PHP has the following functions to check if a value is not a number:
is_nan()

However, the PHP var_dump() function returns the data type and value:

Example
Invalid calculation will return a NaN value:

```
<html>
<body>
<?php
// Invalid calculation will return a NaN value
$x = acos(8);
var_dump($x);
?>
</body>
</html>
```

Output: float(NAN)

PHP Numerical Strings:
➔The PHP is_numeric() function can be used to find whether a variable is numeric. The function returns true if the variable is a number or a numeric string, false otherwise.

Example:
Check if the variable is numeric:

```
<html>
<body>
<?php
// Check if the variable is numeric
$x = 5985;
```

```
var_dump(is_numeric($x));
echo "<br>";
$x = "5985";
var_dump(is_numeric($x));
echo "<br>";
$x = "59.85" + 100;
var_dump(is_numeric($x));
echo "<br>";
$x = "Hello";
var_dump(is_numeric($x));
?>
</body>
</html>
```

Output:
bool(true)
bool(true)
bool(true)
bool(false)

PHP Casting Strings and Floats to Integers:
➔Sometimes you need to cast a numerical value into another data type.

The (int), (integer), or intval() function are often used to convert a value to an integer.

Example
Cast float and string to integer:

```
<html>
<body>
<?php
// Cast float to int
$x = 23465.768;
$int_cast = (int)$x;
echo $int_cast;
echo "<br>";
// Cast string to int
$x = "23465.768";
$int_cast = (int)$x;
echo $int_cast;
?>
</body>
</html>
```

Output:
23465
23465

munotes.in

12.5 BASIC PHP ERRORS

➔A PHP Error occurs when something is wrong in the PHP code. The error can be as simple as a missing semicolon, or as complex as calling an incorrect variable.

➔To efficiently resolve a PHP issue in a script, you must understand what kind of problem is occurring.

➔**The four types of PHP errors are:**
1. Warning Error
2. Notice Error
3. Parse Error
4. Fatal Error

Warning Error:
➔A **warning error in PHP** does not stop the script from running. It only warns you that there is a problem, one that is likely to cause bigger issues in the future.

➔The most common causes of warning errors are:
-Calling on an external file that does not exist in the directory
-Wrong parameters in a function

Example:
```
<?php
echo "Warning error"";
include ("external_file.php");
?>
```

As there is no "external_file", the output displays a message, notifying it failed to include it. Still, it doesn't stop executing the script.

Result:
```
Warning error<br />
<b>Warning</b>:  include(external_file.php): failed to open stream: No such file or
directory in <b>[...][...]</b> on line <b>3</b><br />
<br />
<b>Warning</b>:  include(): Failed opening 'external_file.php' for inclusion
(include_path='.:/home/phpfunctions/phpVersions/7.3.5/lib/php') in <b>[...][...]</b> on
line <b>3</b><br />
```

Notice Error:
➔Notice errors are minor errors. They are similar to warning errors, as they also don't stop code execution.

➔Often, the system is uncertain whether it's an actual error or regular code. Notice errors usually occur if the script needs access to an undefined variable.

Example:

```
<?php
$a="Defined error";
echo "Notice error";
echo $b;
?>
```

In the script above, we defined a variable ($a), but called on an undefined variable ($b). PHP executes the script but with a notice error message telling you the variable is not defined.

Result:

```
Notice error<br />
<b>Notice</b>:  Undefined variable: b in <b>[...][...]</b> on line <b>4</b><br />
```

Parse Error (Syntax):

➔Parse errors are caused by misused or missing symbols in a syntax. The compiler catches the error and terminates the script.

Parse errors are caused by:

- Unclosed brackets or quotes
- Missing or extra semicolons or parentheses
- Misspellings

➔For example, the following script would stop execution and signal a parse error:

```
<?php
echo "Red";
echo "Blue";
echo "Green"
?>
```

It is unable to execute because of the missing semicolon in the third line.

Result:

```
<br />
<b>Parse error</b>:  syntax error, unexpected '';
' (T_ENCAPSED_AND_WHITESPACE), expecting ',' or ';' in <b>[...][...]</b> on line <b>2</b>
<br />
```

Fatal Error:

➔Fatal errors are ones that crash your program and are classified as critical errors. An undefined function or class in the script is the main reason for this type of error.

There are three (3) types of fatal errors:

- Startup fatal error (when thesystem can't run the code at installation)

munotes.in

- Compile time fatal error (when a programmer tries to use nonexistent data)
- Runtime fatal error (happens while the program is running, causing the code to stop working completely)

```
<?php
function sub()
{
$sub=6-1;
echo "The sub= ".$sub;
}
div();
?>
```

The output tells you why it is unable to compile, as in the image below

```
Result:
<br />
<b>Fatal error</b>:  Uncaught Error: Call to undefined function div() in [...][...]:7
Stack trace:
#0 {main}
  thrown in <b>[...][...]</b> on line <b>7</b><br />
```

12.6 SUMMARY

➔ In this chapter we studied PHP Regular Expression also known as regex are powerful pattern matching algorithm that can be performed in a single expression.

➔PHP has built in functions namely PHP preg_match(), PHP preg_split() and PHP preg_replace() that allow us to work with regular functions which we have learn in this PHP Regular Expressions.

➔Regular expressions are very useful when performing validation checks, creating HTML template.

➔ In this chapter we studied arrays types of arrays, number handling and basic php errors like warning error, notice error, parse error and fatal errors.

12.7 REFERENCES

https://www.guru99.com/
https://www.tutorialrepublic.com/
https://www.w3schools.com/
https://www.tutorialspoint.com/
https://phoenixnap.com/

munotes.in

12.8 UNIT END EXERCISE

1. Write a PHP script that checks if a string contains another string.
2. Write a PHP script that removes the last word from a string.

   Sample string: 'Welcome to University of Mumbai'

   Expected Output: Welcome to University of
3. Write a PHP script that removes the whitespaces from a string

   Sample string: 'The university " " of mumbai'

   Expected Output: Theuniversity""ofmumbai
4. Write a PHP script to remove nonnumeric characters except comma and dot.

   Sample string: '$123,34.00A'

   Expected Output: 12,334.00
5. Write a PHP script to remove all characters from a string except a-z A-Z 0-9 or " ".

   Sample string: abcde$ddfd @abcd)der]

   Expected Result : abcdeddfd abcd der

❖❖❖❖

munotes.in

13

INTRODUCTION TO ADVANCE PHP AND MYSQL

Unit Structure :

13.1 Introduction to Advanced PHP
13.2 Introduction to MYSQL
13.3 MySQL Functions
13.4 Integrating web forms and databases
13.5 Displaying queries in tables
13.6 Building Forms from queries
13.7 Summary
14.8 Exercise

13.1 INTRODUCTION TO ADVANCED PHP

PHP is stand for “Hypertext Preprocessor “. PHP is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications.

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. It is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

PHP is a Loosely Typed Language. It is the biggest blogging system on the web (WordPress).

PHP files can contain text, HTML, CSS, JavaScript, and PHP code. PHP code are executed on the server, and the result is returned to the browser as plain HTML

PHP files have extension ".php"

The advanced PHP is concept which is used to build elegant, scalable, reliable sites. The advanced PHP gives you the concepts, features, tools, and practical advice to use them together to build performant, secure, scalable, and reliable web applications. It provides latest Object-Oriented-Programming (OOP) and functional programming features.

In Advanced PHP we can implements OOPs concepts using classes and methods. This class enables developers to tackle the most advanced features of PHP. It is designed for experienced developers

13.2 INTRODUCTION TO MYSQL

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database.

It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database.

Advantages of MySQL

MySQL is an open-source database, so you don't have to pay a single penny to use it.

MySQL is a very powerful program that can handle a large set of functionality of the most expensive and powerful database packages.

MySQL is customizable because it is an open-source database, and the open-source GPL license facilitates programmers to modify the SQL software according to their own specific environment.

MySQL is quicker than other databases, so it can work well even with the large data set.

MySQL supports many operating systems with many languages like PHP, PERL, C, C++, JAVA, etc.

MySQL uses a standard form of the well-known SQL data language.

MySQL is very friendly with PHP, the most popular language for web development.

munotes.in

13.3 MYSQL FUNCTIONS

A function is a block of statements that can be used repeatedly in a program.

A function will not execute immediately when a page loads. A function will be executed by a call to the function.

- **Inbuilt Function**
- **User Defined Function**

Inbuilt Function:

There are several Inbuilt functions are available in php. Generally all its are single line functions.

User Defined Function :
A user defined function declaration starts with the word "function":
Syntax

```
function function_Name()
{
code to be executed;
}
```

Function names are NOT case-sensitive.

```
<?php  function fun() {
echo "Hello world!";
}
fun(); // call the function
?>
```

PHP MySQL Connection

PHP provides mysql_connect() function to open a database connection. This function takes three parameters and returns a MySQL link identifier on success or FALSE on failure.

There are some MySQL functions for database connectivity such as

- Mysql_connection()
- mysql_select_db()
- mysql_query()
- mysql_result()
- mysql_close()•

➢ **PHP mysql_connection() Function:**

PHP mysql_connect() function is used to connect with MySQL database. It returns resource if connection is established or null.

Syntax of mysql_connection
connection mysql_connect(server, user, password);
server:
Optional − The host name running database server. If not specified then default value is localhost:3306.

User:
Optional − The username accessing the database. If not specified then default is the name of the user that owns the server process.

password :
Optional − The password of the user accessing the database. If not specified then default is an empty password.

PHP mysql_select_db() Function:
The mysql_select_db() function sets the active MySQL database. This function returns TRUE on success, or FALSE on failure. mysql_create_db() function attempts to create a new database on the server associated with the specified link identifier.

The mysqli_select_db() function is used to change the default database for the connection.

Syntax of mysql_select_db() function
mysqli_select_db(connection, name)
connection : Required. Specifies the MySQL connection to use
name : Required. Specifies the database name

PHP mysql_query() Function:
mysql_query() Function performs a query against a database. This function selects a database and executes a query on it. If the optional link identifier isn't specified, the function will try to find an open link to the MySQL server and if no such link is found it'll try to create one as if mysql_connect() was called with no arguments.

The mysql_query() sends a query to the currently active database on the server that's associated with the specified link identifier. This function returns a resource identifier or FALSE if the query was not executed correctly.

Syntax of mysql_query() function
mysqli_query(database connection, Sql query);

Example:
```
$conn = mysqli_connect($host, $user, $pass,$dbname);
$sql = "CREATE DATABASE myDb";
mysql_query($conn, $sql);
```

PHP mysql_result() Function:
The mysql_result() returns the contents of one cell from a MySQL result set. When working on large result sets.Themysql_result() should not be mixed with calls to other functions that deal with the result set.

Syntax of mysql_result() function
mysql_result ($result , $row , $field);
Parameters of mysql_result

result: The result resource that is being evaluated. This result comes from a call to mysql_query().

row : The row number from the result that's being retrieved. Row numbers start at 0.

field: The name or offset of the field being retrieved.

Example:
- mysql_close()
- This function used to disconnect Mysql database

13.4 INTEGRATING WEB FORMS AND DATABASES

PHP has a pretty straight forward method to working with MySQL databases. To establish a connection to the MySQL database, but SQL which is the language used to put data in and get data out of a database. There are five steps to make PHP database interaction

- **Create a connection**
- **Select database**
- **Perform database query**
- **Use return data**
- **Close connection**

➢ **Create a connection**

It is very important when you want to start a dynamic website to create a connection with your SQL (we are talking about MySQL) by using mysql_connect() function. In mysql_connect() arguments should be a string and it's important to use die() function with mysql_error() function to check if there is a problem with the connection or not.

➢ **Select database**

Now we have to select the database which we are creating and saving our data by using mysql_select_db() function which the arguments are the name of the database and connection we made earlier.

➢ **Perform database query**

In this step, we have to select out data from the database and bring it into our web page. The best decision to use mysql_query() function which it will Send a MySQL query with 2 arguments as displayed in the above code.

➢ **Use return data**

To display the result on your web page, you have to use while() loop function in addition to mysql_fetch_array() function which fetches a result row as an associative array, a numeric array, or both.

➢ **Close connection**

To close connection, it is better to use mysql_close() function to close MySQL connection. This is a very important function as it closes the connection to the database server. Your script will still run if you do not include this function. And too many open MySQL connections can cause problems for your account. This is a good practice to close the MySQL connection once all the queries are executed.

munotes.in

```
<?php
// Five steps to PHP database connections:
//  Create a database connection
// $connection allows us to keep referring to this connection after it is established
$connection = mysql_connect("localhost","root","myPassword");
if (!$connection) {
die("Database connection failed: " . mysql_error());
}
//  Select a database to use
$db_select = mysql_select_db("widget_corp",$connection);
if (!$db_select) {
die("Database selection failed: " . mysql_error());
}
//  Perform database query
$result = mysql_query("SELECT * FROM subjects", $connection);
if (!$result) {
die("Database query failed: " . mysql_error());
}
// Use returned data
while ($row = mysql_fetch_array($result)) {
echo $row["menu_name"]." ".$row["position"]."<br />";
}
//  Close connection
mysql_close($connection);
```

13.5 DISPLAYING QUERIES IN TABLES

Select Data From a MySQL Database

The SELECT statement is used to select data from one or more tables. SELECT column_name(s) FROM table_name

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
```

```
die("Connection failed: " . $conn->connect_error); }

$sql = "SELECT id, firstname, lastname,Address FROM student_info";
$result = $conn->query($sql);
if ($result->num_rows> 0) {
   // output data of each row
while($row = $result->fetch_assoc()) {
      echo "<br> id: ". $row["id"]. " - Name: ". $row["firstname"]. " " .
$row["lastname"] . " ” . $row["Address"] .  "<br>";
   }
} else {
   echo "0 results";}
$conn->close();
?>
</body>
</html><br>";
   }
} else {
   echo "0 results";
}
$conn->close();
?>
```

Output:
Id: 1 Name: Sachin Sharma Address: Mumbai
Id: 2 Name: Kapil Varma Address: Delhi
Id: 3 Name: Rajesh kumarAddress:Surati

Insert Data into MySQL Database:

My SQL INSERT statement is used to store or add data in MySQL table within the database. We can perform insertion of records in two ways using a single query in MySQL:

- Insert record in a single row
- Insert record in multiple rows

Syntax:

The below is generic syntax of SQL INSERT INTO command to insert a single record in MySQL table:

```
INSERT INTO table_name( field1, field2,...fieldN )
VALUES   ( value1, value2,...valueN );
<?php
if( $_POST['name'] || $_POST["address"] )
        {
```

```
        if (preg_match("/[^A-Za-z'-]/",$_POST['name'] ))
         {
          die ("invalid name and name should be alpha");
         }
         echo "Welcome ". $_POST['name']. "<br />";
          // echo "Your Message is : ". $_POST['message']. "<br />";
        //exit();

          //connecting to the database
define('DB_HOST', 'localhost');
define('DB_NAME', 'mydb');
          define('DB_USER','root');
      define('DB_PASSWORD','');
     $con=mysql_connect(DB_HOST,DB_USER,DB_PASSWORD)or
die("Failed to connect to MySQL: " . mysql_error());
    $db=mysql_select_db(DB_NAME,$con) or
die("Failed to connect to MySQL: " . mysql_error());
          //inserting Record to the database
          $name = $_POST['name'];
          $address = $_POST['address'];
          $gender = $_POST['gender'];
          echo $name;
          echo "<br/>";
          echo $address;
          echo "<br/>";
          echo $gender;
          echo "<br/>";
          $sql="INSERT INTO student_info(Name,Address,Gender)
        VALUES('$name','$address','$gender')";
         // $result = mysql_query($sql)or die(mysql_error());
//$query =”insert into contact (contactName,
contact,Email,message)VALUES('$name','$email','$message')”;
mysql_query($sql)or die(mysql_error());
        echo "Successfully updated database";
/*
          if($result)
             {
                echo "Successfully updated database";
             }
             else
             {
```

```
die('Error: '.mysql_error($con));
            }
            */
mysql_close($con);
   }
   ?><html>
<body>
<form action ="student_info.html" method ="POST">
<input type ="submit" value="Go Back"  />
</form>
</body>
</html>
```

Delete Data from MySQL Database:

PHP mysql_query() function is used to delete record in a table.

```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
<body>
<?php
define('DB_HOST', 'localhost');
define('DB_NAME', 'mydb');
   define('DB_USER','root'); define('DB_PASSWORD','');
   $test = $_POST['id'];
  $con=mysql_connect(DB_HOST,DB_USER,DB_PASSWORD)or
die("Failed to connect to MySQL: " . mysql_error());
  $db=mysql_select_db(DB_NAME,$con) or
die("Failed to connect to MySQL: " . mysql_error());
$sql = "DELETE from student_info WHERE id=$test";
if (mysql_query($sql))
{
  echo "Record Deleted successfully";
}
else
 {
    echo "Error deleting record:  " .mysql_error($con);
} ?>
</body></html>
```

munotes.in

13.6 BUILDING FORMS FROM QUERIES

We can create and use forms in PHP. To get form data, the form request may be get or post. To retrieve data from get request, we need to use $_GET, for post request $_POST.

To show the values in the input fields after the user hits the submit button, we add a little PHP script inside the value attribute of the following input fields: name, email, and website.

```
<html>
<head>
<title>Sign-In</title>
<link rel="stylesheet" type="text/css" href="CSS/style.css">
</head>
<body id="body-color">
<div id="Sign-In">
<fieldset style="width:30%"><legend>LOG-IN HERE</legend>
<form method="POST" action="connectivity.php">
User <br><input type="text" name="user" size="40"><br>
Password <br><input type="password" name="pass" size="40"><br>
<input id="button" type="submit" name="submit" value="Log-In">
</form>
</fieldset>
</div></body></html>
```

student_Login.html

```
<?php
if( $_POST['name'] || $_POST["address"] )
{
if (preg_match("/[^A-Za-z'-]/",$_POST['name'] ))
{
die ("invalid name and name should be alpha");
}

echo "Welcome ". $_POST['name']. "<br />";
// echo "Your Message is : ". $_POST['message']. "<br />";
//exit();
//connecting to the database
define('DB_HOST', 'localhost');
define('DB_NAME', 'mydb');
define('DB_USER','root');
```

```
define('DB_PASSWORD','');
$con=mysql_connect(DB_HOST,DB_USER,DB_PASSWORD) or
die("Failed to connect to MySQL: " . mysql_error());
$db=mysql_select_db(DB_NAME,$con) or
die("Failed to connect to MySQL: " . mysql_error());
//inserting Record to the database
$name = $_POST['name'];
$address = $_POST['address'];
$gender = $_POST['gender'];
echo $name;
echo "<br/>";
echo $address;
echo "<br/>";
echo $gender;
echo "<br/>";
 $sql="INSERT INTO student_info(Name,Address,Gender)
VALUES('$name','$address','$gender')";
//$query = "INSERT INTO
contact(contactName,contactEmail,message)VALUES('$name','$e
mail','$message')";
// $result = mysql_query($sql)or die(mysql_error());
mysql_query($sql)or die(mysql_error());
echo "Successfully updated database";
/*
if($result)
  {
    echo "Successfully updated database";
  }
  else
  {
die('Error: '.mysql_error($con));
  }
  */
mysql_close($con);  }    ?>
<html>
<body>
<form action ="student_info.html" method ="POST">
<input type ="submit" value="Go Back"  />
</form>
</body>
</html>
```

student_insert.php

13.6 SUMMARY

➔ In this chapter we are more focused on advantages of MySQL and its inbuilt and user defined functions.

➔The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP.

➔ MySQL is a RDBMS (Relational database management system). We use such a system in order to store, retrieve and manage data related to our applications.

➔In this chapter we have learn how to design dynamic webpages We can create and use forms in PHP. To get form data, the form request may be get or post. To retrieve data from get request, we need to use $_GET, for post request $_POST

13.7 EXERCISE

1. Create a simple HTML form and accept the user name and display the name through PHP echo statement.
2. How can we connect to a MySQL database from a PHP script?
3. How can we access the data sent through the URL with the GET method?
4. Design PHP Registration form using get and post method.
5. Create a database called registration. In the registration database, add a table called users. The users table will take the following four fields like id, username varchar (100), email varchar(100) and password varchar(100)

❖❖❖❖

14

ADVANCED PHP

Unit Structure:

14.1 PHP String
14.2 String Function
14.3 Regular Expressions
14.4 PHP Sessions
14.5 PHP Cookies
14.6 HTTP
14.7 E-Mail
14.8 Summary
14.9 References
14.10 Exercise

14.1 PHP STRING

PHP string is a sequence of characters. It is used to store and manipulate text.

There are some string literal in PHP such as single quoted, double quoted, heredoc syntax.

Single Quoted: We can create a string in PHP by enclosing the text in a single-quote. It is the easiest way to specify string in PHP.

Example: $str='Hello World , This text within single quote';

Double Quoted: we can specify string through enclosing text within double quote.

Example: $str="Hello World , This text within double quote";

Heredoc: Heredoc syntax (<<<) is the third way to delimit strings. In Heredoc syntax, an identifier is provided after this heredoc<<< operator, and immediately a new line is started to write any text.

```
<?php
/* $code = 'PHP'
   $str = <<<Display

Hello! Welcome in Web Technology, and This is demo of $code.
Display;
echo $str;
?>
```

Output: Hello! Welcome in Web Technology, and This is demo of PHP.

14.2 STRING FUNCTION

strlen() Function:
The strlen() function is used to find the length of a string. The strlen() function is predefined function of PHP. It is used to find the length of string or returns the length of a string including all the whitespaces and special character.

Syntax of str_word_count() function
strlen("string")
Example of strlen() function
```
<?php
echostrlen("Web Technology!"); // outputs 15
?>
```

Output: 15
str_word_count() Function:
The PHP str_word_count() function counts the number of words in a string.

This is in-built function of PHP. It is used to return information about words used in a string or counts the number of words in a string.

Syntax of str_word_count() function
str_word_count(string,return,char)
Example of str_word_count function
```
<?php
echostr_word_count("Web Technology!"); // outputs 2
?>
```

Output: 2
stringstrcmp() Function:
String comparison is one of the most common tasks in programming and development. strcmp() is a string comparison function in PHP. It is a built-in function of PHP, which is case sensitive.

This function compares two strings and tells whether a string is greater, less, or equal to another string. strcmp() function accepts two strings parameter, which is mandatory to pass in the function body. Both parameters are mandatory to pass in strcmp() function().

Syntax of strcmp() function
strcmp($str1,$str2);
Return 0 - It returns 0 if both strings are equal, i.e., $str1 = $str2
Return < 0 - It returns negative value if string1 is less than string2, i.e., $str1 < $str2
Return >0 - It returns positive value if string1 is greater than string 2, i.e., $str1 > $str2
Example of str_word_count function
```
<html>
```

```
<body>
<?php
echostrcmp("Hello", "Hello") ;  //Output 0
echo "<br>";
echostrcmp("hello", "Hello") ; //Output 1
?>
</body></html>
```

Output: 0
1

stringstrrev() Function:
The strrev() function is predefined function of PHP. It is used to reverse a string. It is one of the most basic string operations which are used by programmers and developers.

Syntax of strrev() function
stringstrrev (string $string);
Example of strrev function

```
<?php
echostrrev("Hello world!"); // outputs !dlrowolleH
echo "<br>";
echostrrev("PHP"); //PHP?>
```

Output:!dlrowolleH
PHP

stringsubstr_count() Function:
The substr_count() is a built-in function of PHP. It counts the number of times a substring occurs in the given string. The substr_count() is a case-sensitive function.

Syntax of substr_count() function
substr_count($string, $substring, $start, $length)
$string (required): The $string parameter is the main string parameter in which occurrence's of substring is counted. It is a mandatory parameter.

$substring (required): The value passed in this parameter is searched in $string parameter and returns the counted occurrence of substring. It is also a mandatory parameter.

$start (optional): This parameter consists of an integer value. It specifies that from where to start the counting.

$length (optional): This parameter is an optional parameter and depends on $start parameter.
Example of substr_count() function

```
<?php
 $str1 = "God is one, God is every where";
   $str2 = "God";
echosubstr_count($str1, $str2); //Output 2
?>
```

Output: 2

stringstrpos()Function:

The strops() is in-built function of PHP. It is used to find the position of the first occurrence of a string inside another string or substring in a string.

The PHP strpos() function searches for a specific text within a string. If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.

Syntax of strpos() function

stringstrrev ($string, $Search_string);

Example of strpos function

```
<?php
<?php
echostrpos("Hello World, God is Great", "God"); //Output 13
?>?>
```

Output: 13

14.3 PHP REGULAR EXPRESSIONS

A regular expression is a sequence of characters that forms a search pattern. When you search for data in a text. Regular expressions are nothing more than a sequence or pattern of characters it. They provide the foundation for pattern-matching functionality.

These are nothing more than a pattern or a sequence of characters, which describe a special search pattern as text string. Regular expression allows you to search a specific string inside another string. Even we can replace one string by another string and also split a string into multiple chunks.

Advantages and uses of Regular expressions:

- Regular expressions help in validation of text strings which are of programmer's interest.
- It offers a powerful tool for analyzing, searching a pattern and modifying the text data.
- It helps in searching specific string pattern and extracting matching results in a flexible manner.
- It helps in parsing text files looking for a defined sequence of characters for further analysis or data manipulation.
- With the help of in-built regexes functions, easy and simple solutions are provided for identifying patterns.
- It effectively saves a lot of development time, which are in search of specific string pattern.

Regular Expression Modifiers
Modifiers can change how a search is performed.

Modifier	Description
i	Performs a case-insensitive search
m	Performs a multiline search (patterns that search for the beginning or end of a string will match the beginning or end of each line)
u	Enables correct matching of UTF-8 encoded patterns.

Regular Expression Meta characters:
Meta characters are characters with a special meaning:

Metacharacter	Description
|	Find a match for any one of the patterns separated by | as in: ball|bat|net
.	Find just one instance of any character
^	Finds a match as the beginning of a string as in: ^Hello
$	Finds a match at the end of the string as in: World$
\d	Find a digit
\s	Find a whitespace character

munotes.in

Regular Expression Functions
PHP provides a variety of functions that allow you to use regular expressions.

Function	Description
preg_match()	Returns 1 if the pattern was found in the string and 0 if not
preg_match_all()	Returns the number of times the pattern was found in the string, which may also be 0
preg_replace()	Returns a new string where matched patterns have been replaced with another string

PHP preg_match() function:

The preg_match() function will tell you whether a string contains matches of a pattern. This function searches the pattern inside the string and returns true if the pattern exists otherwise returns false.

Syntax of preg_match() function
preg_match($pattern_string , $string);

Example of preg_match() function

```
<html>
<body>
<?php
$str = "Welcome in Web Technology";
$pattern = "/Technology /i";
echopreg_match($pattern, $str);
?>
</body>
</html>
```

Output: The preg_match() function returns 1

PHP preg_match_all() function:

The preg_match_all() function will tell you how many matches were found for a pattern in a string. This function matches all the occurrences of pattern in the string.

Syntax of preg_match_all() function
preg_match_all ($pattern_string , $string);

Example of preg_match() function

```
<?php
$str = "The India my country all Indians are brother & sister";
$pattern = "/India/i";
echopreg_match_all($pattern, $str); // Outputs 2
?>
```

Output: The preg_match_all() function returns 2

PHP preg_replace() function:

The preg_replace() function will replace all of the matches of the pattern in a string with another string. Thepreg_replace() function is similar to the ereg_replace() function, except that the regular expressions can be used in search and replace.

Syntax of preg_replace() function
preg_replace($pattern_string , Sreplace_String, $Original_string);

Example of preg_match() function

```
<html>
<body>
<?php
$str = "Hello World, Welcome into World of Web technology!";
$pattern = "/Technology/i";
```

```
echopreg_replace($pattern, "Technology", $str); // Outputs "Hello World, Welcome into World of Web Technology!"
?>
</body>
</html>
```

Output: The preg_match_all() function returns “Hello World, Welcome into World of Web Technology!"

PHP preg_split() function:
The preg_split() function is an inbuilt function in PHP which is used to convert the given string into an array. The function splits the string into smaller strings or sub-strings of length which is specified by the user. If the limit is specified then small string or sub-strings up to limit return through an array.

Syntax of preg_split() function
preg_split($pattern, $subject, $limit, $flag)
Example of preg_split() function

```
<html>
<body>
<?php
$result = preg_split("/[\s,]+/", "Web Technology");
  // Display result
print_r($result);
?>
</body>
</html>
```

Output: The preg_split() function returns Array ([0] => Web [1] => Technology)

14.4 PHP SESSIONS

PHP session is object which is used to store and pass information from one page to another temporarily. A session is a way to store information to be used across multiple pages.

Sessions are a simple way to store data for individual users against a unique session ID.

This can be used to persist state information between page requests. An alternative way to make data accessible across the various pages of an entire website is to use a PHP Session.

Session IDs are normally sent to the browser via session, cookies and the ID is used to retrieve existing session data. Session variables solve this problem by storing user information to be used across multiple pages. By default, session variables last until the user closes the browser.

munotes.in

Starting a PHP Session

A PHP session is easily started by making a call to the session_start() function. This function first checks if a session is already started and if none is started then it starts one. It is recommended to put the call to session_start() at the beginning of the page.

Session variables are set with the PHP global variable: $_SESSION. Session variables are stored in associative array called $_SESSION[]. These variables can be accessed during lifetime of a session.

The following example starts a session then registers a variable called counter that is incremented each time the page is visited during the session.

Make use of isset() function to check if session variable is already set or not.

Syntax of Session

$_SESSION[“session_object"] = “Value";

Example:

```
<?php
// Set session variables
$_SESSION[“session_object"] = “Value";
$_SESSION["student"] = "Sachin";
?>
PHP Session demonstration
To Create Session variable
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// Set session variables
$_SESSION["SportName”] = "Cricket";
$_SESSION["Country"] = "India";
echo "Session variables are set.";
?></body></html>
```

Save this file session_create.php

Transfer the information (data) which is store in session variables to next page. After retrieving into second page display it using $_SESSION [] array variables.

```
<?php
session_start();
?>
<!DOCTYPE html>
<html><body>
```

munotes.in

```
<?php
// Echo session variables that were set on previous page
echo "Favorite Sport is " . $_SESSION["SportName "] . ".<br>";
echo "Favorite Country is " . $_SESSION["Country"] . ".";
?>
```

Retrieve session information from session_create.php

Send the data from second page to third page and destroy the session.

```
<?php
session_start();
?>
<!DOCTYPE html>
<html><body>
<?php
// remove all session variables
session_unset();
// destroy the session
session_destroy();
echo "All session variables are now removed, and the session is destroyed."
?></body></html>
```

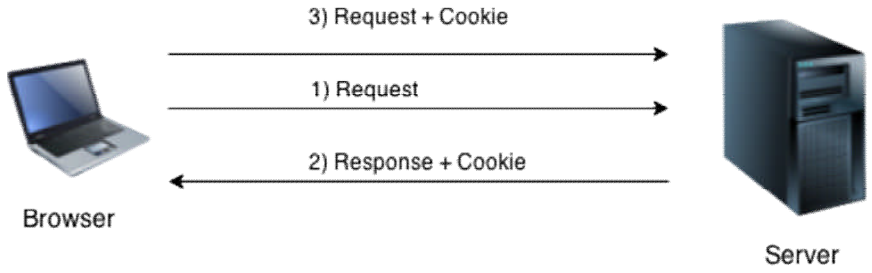
14.5 PHP COOKIES

A cookie is a small file with the maximum size of 4KB that the web server stores on the client computer.

PHP cookie is a small piece of information which is stored at client browser. It is used to recognize the user.

Cookie is created at server side and saved to client browser. Each time when client sends request to the server, cookie is embedded with request. Such way, cookie can be received at the server side.

A cookie created by a user can only be visible to them. Other users cannot see its value.



Setting a Cookie in PHP

The setcookie() function is used to set a cookie in PHP. Make sure you call the setcookie() function before any output generated by your script otherwise cookie will not set. The basic syntax of this function can be given with:

PHP setcookie() function

PHP setcookie() function is used to set cookie with HTTP response. Once cookie is set, you can access it by $_COOKIE superglobal variable.

setcookie(name, value, expire, path, domain, secure);

Parameter	Description
name	The name of the cookie.
value	The value of the cookie. Do not store sensitive information since this value is stored on the user's computer.
expires	The expiry date in UNIX timestamp format. After this time cookie will become inaccessible. The default value is 0.
path	Specify the path on the server for which the cookie will be available. If set to /, the cookie will be available within the entire domain.
domain	Specify the domain for which the cookie is available to e.g www.example.com.
secure	This field, if present, indicates that the cookie should be sent only if a secure HTTPS connection exists.

```
<?php
// Setting a cookie
setcookie("username", "KapilDev", time()+1*60*60); // using expiry in 1 hour(1*60*60 seconds or 3600 seconds)
?>
```

Retrieving the Cookie value

The PHP $_COOKIE super global variable is used to retrieve a cookie value. It typically an associative array that contains a list of all the cookies values sent by the browser in the current request, keyed by cookie name. Create another file named “cookies_read.php” with the following code.

PHP $_COOKIE

PHP $_COOKIE superglobal variable is used to get cookie.

```
$value=$_COOKIE["CookieName"];//returns cookie value
<?php
print_r($_COOKIE);    //output the contents of the cookie array variable
?>
```

Delete Cookies:

If you want to destroy a cookie before its expiry time, then you set the expiry time to a time that has already passed. You can delete a cookie by calling the same setcookie() function with the cookie name and any value however this time you need the set the expiration date in the past, as shown in the example below:

```
<?php
setcookie ("CookieName", "", time() - 3600);// set the expiration date to one hour ago
?>
```

Example of deleting Cookies

```
<?php
setcookie(" username ", time() - 360,'/');
?>
```

14.6 PHP HTTP

The PHP HTTP(Hyper Text Transfer Protocol) functions allow us to handle the information which is sent to the browser by the web server. The purpose of the HTTP extension is to provide comfort and robust set of functionality for major applications.

The HTTP extensions ease handling of HTTP URLs, dates, redirects, headers and message in an HTTP context.

PHP HTTP header() Function

The PHP HTTP header() functions sends a raw HTTP header. This function returns no value.

```
<?php
header("Location: https://www.google.com/");
?>
```

PHP HTTP headers_list() Function:

The PHP HTTP **headers_list( )** function returns a list of response headers sent to the client.

Syntax of headers_list()

arrayheaders_list (void);

PHP HTTP headers_sent() Function:

The PHP HTTP **headers_sent( )** function checks if or where headers have seen sent. The headers_sent( ) function will return false if no HTTP headers have already been sent or true otherwise.

```
<?php
setcookie('Sachin', 'Kapil’);
header("MCNSolution: pqr");
header('Content-type: text/plain');
var_dump(headers_list());
?>
```

14.7 E-Mail

PHP mail () function is used to send email in PHP. You can send text message, html message and attachment with message using PHP mail() function. Sending email messages are very common for a web application.

The mail() function allows you to send emails directly from a script. For the mail functions to be available, PHP requires an installed and working email system. The program to be used is defined by the configuration settings in the php.ini file.

PHP makes use of mail() function to send an email. This function requires three mandatory arguments that specify the recipient's email address, the subject of the message and the actual message additionally there are other two optional parameters.

The basic syntax of this function can be given with:

mail(to, subject, message, headers, parameters)

Parameter	Description
to	The recipient's email address.
subject	Required. Specifies the subject of the email. This parameter cannot contain any newline characters
message	Required. Defines the message to be sent. Lines should not exceed 70 characters
Optional — The following parameters are optional	
headers	Optional. Specifies additional headers, like From, Cc, and Bcc.
parameters	Optional. Specifies an additional parameter to the send mail program

PHP Mail: Send HTML Message

To send HTML message, you need to mention Content-type text/html in the message header.

```
<?php
  $to = "sample@maildemo.com";//change receiver address
  $subject = "This is Email  subject";
  $message = "<h1>This is HTML heading</h1>";
    $header = "From:xyz@ maildemo.com \r\n";
  $header .= "MIME-Version: 1.0 \r\n";
  $header .= "Content-type: text/html;charset=UTF-8 \r\n";
  $result = mail ($to,$subject,$message,$header);
if( $result == true ){
echo "Message sent successfully...";
}else{
echo "Sorry, unable to send mail...";     }  ?>
```

munotes.in

14.8 SUMMARY

This Chapter provides a hands-on learning experience complete with exercise learning. Advanced PHP is focused on 3 major areas: Object-Oriented programming, XML and MySQL support. This Chapter covers the following topics:

- How PHP works with your web browser and web server
- PHP language basics, including data, variables, logic and looping
- Working with arrays and functions
- Making web forms
- Working with databases like MySQL
- Remembering users with sessions & Cookies.

14.9 REFERENCES

- "PHP: A Beginner's Guide" by VikramVaswani
- "PHP Object – Oriented Solutions" by David Powers
- "Build Your Own Database Driven Web Site Using PHP & MySQL" by Kevin Yank.
- "Programming PHP: Creating Dynamic Web Pages" by Kevin Tatroe and Peter Mactyre.

14.10 QUESTIONS:

1) How a variable is declared in PHP?
2) What is the difference between "echo" and "print" in PHP?
3) What is the array in PHP?
4) Explain some of the PHP array functions?
5) Explain setcookie() function in PHP?
6) What is a session?
7) What is the method to register a variable into a session?
8) What is $_SESSION in PHP?
9) What is the difference between session and cookies in PHP?
10) How to create and destroy cookies in PHP?
11) How do you connect MySQL database with PHP?
12) How to create database connection and query in PHP?
13) What is Cookie? How to create cookies in PHP?
14) How to send email using php script?
15) Explain the difference between mysql_fetch_array(), mysql_fetch_object()?

❖❖❖❖

munotes.in