

**MC5303: WEB PROGRAMMING ESSENTIALS**

<b>UNIT I</b>	<b>INTRODUCTION TO WWW</b>	<b>9</b>
Internet Standards – Introduction to WWW – WWW Architecture – SMTP – POP3 – File Transfer Protocol - Overview of HTTP, HTTP request – response — Generation of dynamic web pages.		
<b>UNIT II</b>	<b>UI DESIGN</b>	<b>9</b>
Markup Language (HTML5): Basics of Html -Syntax and tags of Html- Introduction to HTML5 - Semantic/Structural Elements -HTML5 style Guide and Coding Convention– Html Svg and Canvas – Html API's - Audio & Video - Drag/Drop - Local Storage - Web socket API– Debugging and validating Html. Cascading Style Sheet (CSS3): The need for CSS – Basic syntax and structure Inline Styles – Embedding Style Sheets - Linking External Style Sheets - Introduction to CSS3 – Backgrounds - Manipulating text - Margins and Padding - Positioning using CSS - Responsive Web Design - Introduction to LESS/SASS		
<b>UNIT III</b>	<b>OVERVIEW OF JAVASCRIPT</b>	<b>9</b>
Introduction - Core features - Data types and Variables - Operators, Expressions, and Statements Functions - Objects - Array, Date and Math Related Objects - Document Object Model - Event Handling - Controlling Windows & Frames and Documents - Form validations.		
<b>UNIT IV</b>	<b>ADVANCED FEATURES OF JAVASCRIPT</b>	<b>9</b>
Browser Management and Media Management – Classes – Constructors – Object-Oriented Techniques in JavaScript – Object constructor and Prototyping - Sub classes and Super classes – Introduction to JSON – JSON Structure –Introduction to jQuery –Introduction to AJAX-Bootstrap - Bootstrap components.		
<b>UNIT V</b>	<b>PHP</b>	<b>9</b>
Introduction - How web works - Setting up the environment (LAMP server) - Programming basics Print/echo - Variables and constants – Strings and Arrays – Operators, Control structures and looping structures – Functions – Reading Data in Web Pages - Embedding PHP within HTML - Establishing connectivity with MySQL database.		

**TOTAL : 45 PERIODS****REFERENCES:**

1. David Flanagan, “JavaScript: The Definitive Guide, Sixth Edition”, O'Reilly Media, 2011
2. Harvey & Paul Deitel& Associates, Harvey Deitel and Abbey Deitel, “Internet and World Wide Web - How To Program”, Fifth Edition, Pearson Education, 2011
3. James Lee, BrentWare , “Open Source Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP” AddisonWesley, Pearson 2009
4. Thomas A. Powell, “HTML & CSS: The Complete Reference”, Fifth Edition, 2010
5. Thomas A Powell, Fritz Schneider, “JavaScript: The Complete Reference”, Third Edition, Tata McGraw Hill, 2013
6. Thomas A Powell, “Ajax: The Complete Reference”, McGraw Hill, 2008

**MC5303: WEB PROGRAMMING ESSENTIALS****PART A - UNIT-I****1) Define Internet.**

Internet is the network of computer networks. It makes it possible for any computer connected to it to send and receive data from any other computer connected to it. The Internet is a global system of interconnected computer networks that use the standard Internet protocol suite (often called TCP/IP, although not all protocols use TCP) to serve billions of users worldwide.

**2) What is the difference between node and hosts?**

A node is any addressable device connected to a network whereas the host is a more specific descriptor that refers to a networked general-purpose computer rather than a single purpose device (such as a printer).

**3) What is IP address?**

The Internet address (or IP address) is 32 bits (for IPv4) that uniquely and universally defines a host or router on the Internet. The portion of the IP address that identifies the network is called the netid. The portion of the IP address that identifies the host or router on the network is called the hostid. An IP address defines a device's connection to a network.

**4) Explain various types of connecting devices?**

There are five types of connecting devices: repeaters, hubs, bridges, routers, and switches.

Repeaters regenerate a signal at the physical layer. A hub is a multiport repeater. Bridges have access to station addresses and can forward or filter a packet in a network. They operate at the physical and data link layers. Routers determine the path a packet should take. They operate at the physical, datalink, and network layers.

**5) What is a mail gateway?**

A machine that connects to two or more electronic mail systems and transfers mail messages among them. Mail gateways usually capture an entire mail message, reformat it according to the rules of the destination mail system, and then forward the message.

**6) Define protocol. Why are the protocols layered?**

A protocol is a formal set of rules that must be followed in order to communicate.

Layering protocols simplifies the task of communicating over the network and it allows for reuse of layers that are not specific to a particular application.

**7) Define DNS.**

DNS stands for Domain Name System. DNS provides the service of translating a host name to the corresponding IP address. Example WWW.Yahoo.com is called as domain name.

**8) Differentiate between Internet and WWW.**

Internet	World Wide Web
Internet is the physical aspects computers, networks servers.	WWW is an abstraction and set of services on the top of internet
Internet provides access to mail, network news and its with information resources	WWW makes it easier to use the internet by providing graphical interface wide and rapidly growing amount of information

**9) Define port. What do you mean by well-known ports?**

A port is a logical channel to an application running on a host. i.e., the applications running on the host machines are uniquely identified by port numbers. Port numbers can range from 1 to 65535; however ports 1 to 1023 are reserved. These reserved ports are referred to as well-known ports because the Internet Assigned Numbers authority publicly documents the applications that use them.

**10) What is FTP and the features provided by FTP?**

The TCP/IP standard, high level protocol for transferring files from one machine to another. It is a standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet. FTP is built on client-server architecture and utilizes separate control and data connections between the client and server.

**11) Give few commands for FTP.**

```
ftp> help ls      ls      list contents of remote directory
ftp> help cdup    cdup    change remote working to parent directory.
ftp> help glob    glob    toggle metacharacter expansion of local file names
ftp> help bell    bell    beep when command completed
```

To execute a command name: ftp> bell Bell mode on.

**12) What is meant by Name Resolution?**

Name Resolution is the process of mapping a hostname to its corresponding IP Address. One way to translate a hostname to an IP address is to look it up in a simple text file. The second way is the domain name service, which is a distributed database containing all registered hostnames on the Internet and their IP addresses.

**13) Define URI, URL, and URN.**

1. URI (Uniform Resource Identifier): It identifies an object on the Internet.
2. URL (Uniform Resource Locator): It is a specification for identifying an object such as a file, newsgroup, and CGI program or e-mail address by indicating the exact location on the internet.
3. URN (Uniform Resource Name): It is a method for referencing an object without declaring the full path to the object.

**14) What are the components of HTTP URL? Define URL encoding.**

The components are host, an optional port, path, filename, section and query string. URL encoding involves replacing all unsafe and nonprintable characters with a percent sign (%) followed by two hexadecimal digits corresponding to the character's ASCII value.

**15) What is HTTP?**

HTTP – HyperText Transfer Protocol used to transfer web documents from a server to a browser. **Hypertext Transfer Protocol Secure (HTTPS)** is a combination of the Hypertext Transfer Protocol (HTTP) with SSL/TLS protocol to provide encrypted communication and secure identification of a network web server. HTTPS connections are often used for payment transactions on the World Wide Web and for sensitive transactions in corporate information systems.

**16) What is CGI?**

CGI – Common Gateway Interface. A technology a server uses to create a web page dynamically when the request arrives. The **Common Gateway Interface (CGI)** is a standard method for web servers software to delegate the generation of web pages to executable files. Such files are known as CGI scripts; they are programs, often stand-alone applications

**17) What is SMTP and POP?**

Simple Mail Transfer Protocol specifies how two mail systems interact and the format of control messages they exchange to transfer mail. This protocol establishes the TCP connection to port 25, act as client, waits for the receiving machine operating as the server to talk first.

The protocol used to access and extract e-mail from a mail box. Email requires a special language for mail to be received or sent. POP is one of the technologies that allows email sent from anywhere in the world to arrive in your inbox. When a person sends an email to our address, it is transmitted over the Internet, and eventually lands on our email server.

**18) What is MIME?**

MIME – Multipurpose Internet Mail Extensions.

A standard used to encode data such as images as printable ASCII text for transmission through e-mail. MIME is the shortened form of the complete term i.e. Multipurpose Internet Mail Extensions. It is particular description used to format the non-ASCII messages in order to send them over the Internet. Nowadays a number of e-mail clients are supporting MIME that enabled them to received ad send the graphics, MIME also supports the messaging in the character set in addition to ASCII.

**19) What are the advantages and disadvantages of dynamic web pages?**

**Advantages:**

Enhances functionality of a website, Simple and Easy to update, New content is responsible for bringing traffic to the site that helps in the search engines optimization, Acts as a system to permit staff or users to collaborate

**Disadvantages:**

Expensive to develop, as web solutions developers use their expertise and knowledge to develop such websites, Hosting costs a little more, again the involvement of experts.

**20) What are the differences between SMTP and POP?**

SMTP is a relatively simple, text-based protocol, in which one or more recipients of a message are specified (and in most cases verified to exist) along with the message text and possibly other encoded objects. The message is then transferred to a remote server using a procedure of queries and responses between the client and server.

POP is a very simple protocol that only allows downloading of messages from your Inbox to your local computer. Generally, once transferred, the email is then on your local computer and is removed from FastMail.FM.

**21) State the functions of POP.**

In computing, the **Post Office Protocol (POP)** is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection. POP and IMAP (Internet Message Access Protocol) are the two most prevalent Internet standard protocols for e-mail retrieval

**22) What are the sequences of steps for each HTTP request from a client to the server?**

- Making the connection
- Making a request
- The response
- Closing the connection

**23) Explain about HTTP Connection.**

It is a communication channel between web browser and web server. It begins on the client's idea with the browser sending a request to the web server for a document.

Request Header Fields are

1. From ,2. Reference, 3. If\_modified\_since, 4. Pragma, 5. User Agent

**24) List any four common browsers.**

- Mosaic (1993) by NCSA (National Centre for supercomputer applications)
- Netscape Navigator
- Microsoft Internet Explorer

**25) Distinguish between Static and Static and Dynamic Web Pages. (MAY 2016)**

In simplest terms, static Web pages are those with content that cannot change without a developer editing its source code, while dynamic Web pages can display different content from the same source code. When it comes to using static or dynamic pages for parts of your company's website, having the most advanced code on each of your pages is not important. What is important is the purpose each page serves for your website.

**26) What is WWW?(MAY2015)**

WWW is an abstraction and set of services on the top of internet. WWW makes it easier to use the internet by providing graphical interface wide and rapidly growing amount of information.

**27) What is dynamic web page? (MAY2015)**

A dynamic web page is a kind of web page that has been prepared with fresh information (content and/or layout), for each individual viewing. It is not static because it changes with the time (ex. a news content), the user (ex. preferences in a login session), the user interaction (ex. web page game), the context (parametric customization), or any combination of the foregoing.

**28) What is the primary goal of ARPANET? (MAY 2016)**

The main purpose of ARPANET was to provide a communication network to the various branches of armed forces during and after a nuclear attack.

**UNIT – II**

**1) What is meant by HTML?**

HTML is the set of mark-up symbols or codes placed in a file intended for display on the Web browser page. These mark-up symbols and codes identify structural elements such as paragraphs, heading, and lists. HTML can be used to place media (such as graphics, video, and audio) on the Web page and describe fill-in-forms. A method is an implementation of an object's behavior.

**2) Define Tags.**

Tags signal the browser to inform about the formatting details, i.e. how the content should be displayed in the browser screen. Tags are enclosed between "<" and ">"

**3) What are the two different types of tags?**

Standalone tag only start tag is present and no end tag. Example <BR> and container tag have start and end tag will be present. Example <html>.... </html>

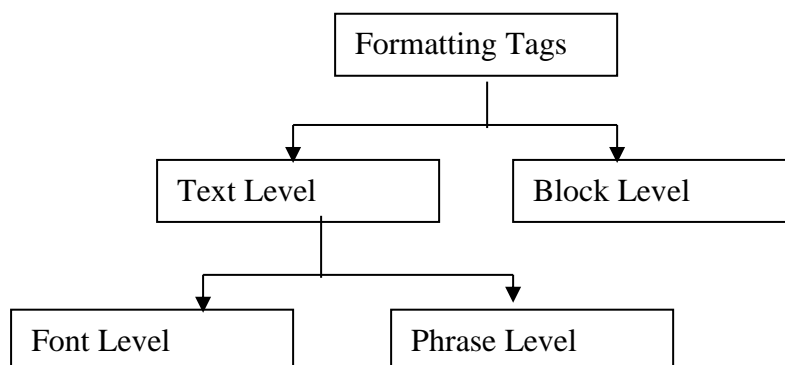
**4) What are the rules to define a tag?**

Attributes should be placed inside start tag, appears as Name-value pairs separated by blank spaces. Attributes should have only one value, values should be enclosed within either single (') or double (") quotes.

**5) Differentiate between standalone and container tag.**

S.no	Standalone	Container
1	Only start tag is present and no end tag.	Both start and end tag will be present
2	Can have only attributes and no parameters	Can have both attributes and parameters.
3	Example: 	Example:<html>.....</html>

6) List out the types of formatting tags.



**7) What is cellpadding and cell spacing attributes?**

The cellpadding allows to have some space between the contents of each cell and its borders. The distance between each cell is called cell spacing.

**8) What is the need of using form in HTML?**

Form is a typical layout on the web page by which user can interact with the web page. The components that can be placed on the form are text box, check box, radio buttons, and push

buttons and so on. Thus form is typically used to create an interactive Graphical User Interface.

**9) What is the purpose of using frames in HTML?**

The HTML frames allows the web designer to present the web document in multiple views. Using multiple views one can keep the formation visible and at the same time other views can be scrolled or replaced.

**10) What is the need for special character in HTML?**

There are some symbols that cannot be used directly in HTML document. For example <(less than) because this symbol is also used along with the tag. Hence this is called a special symbol and can be denoted with the help of entity reference.

**11) State how an unrecognized element or attribute treated by the HTML document?**

If any unrecognized element or attribute is written then the HTML document simply displays the contents. For example <title>testing</title> will display the string “testing” on the web page. It will not display it as a title of the web page.

**12) What is the use of hyperlink tag in HTML?**

The hyperlink tag is used to link logically with other page. Using this tag a web link can be specified. The <a> tag is used to specify the hyperlink in HTML.

To logically link one page with another, use of link to enhance readability of the web document, the navigation from one page to another is possible.

**13) What is BODY in HTML document?**

The effects which we want in the window are mentioned with the help of tags in the body. It is the place where the actual data is written in html. All the changes can be viewed by changing the tags content in the body whereas the head part is the introduction part and the body is the actual content part.<BODY>data content</BODY>

**14) What is an image map?**

An image map allows you to link to several web pages through one image. Simply define shapes within images and link these to the pages you want. Here's a video to help you learn more about images and links in HTML.

**15) What is CSS file?**

The cascading style is many times defined in a separate file. This file is called CSS file. In CSS file one or more style rules are given.

**16) What are style sheets?**

The style sheets are the collection of styles that can be either embedded within the HTML documents or can be externally applied. The Cascading style sheet is a markup language used to apply the styles to HTML elements.

**17) What is selector string? Specify any three forms of selectors.**

The rule set in CSS consists of selector string which is basically an HTML element. These selectors can be defined with the help of properties and values.

**18) Mention the need for cascading style sheet.**

The cascading style sheet allows separation between the information contained in a document and its presentation. Hence any change in presentation can be made without distributing the information of the document. The cascading style sheet allows the developer to give the consistent appearance to all elements of the web page.

**19) Enlist various categories of properties used in CSS.**

The various categories of properties in CSS are

1) Fonts 2) List 3) Alignment of Text 4) Color 5) Margins 6) Background 7) Borders.

**20) What is the use of Universal Selector?**

Using the universal selector the values can be defined for all the elements in the document. It is denoted by \*.

**21) What is generic class selector?**

The generic class applied to any tag in the HTML document. And thus the values defined within that generic selector can be applied to the corresponding tag. The class selector must be preceded by the dot operator.

**22) What are the advantages of External style sheet?**

When we use external style sheet then the style is defined in one file and actual contents of the web are defined in another file. Hence if we want to change the style of presentation of web page then we can simply modify the file in which the style is defined.

**23) What is the difference the external style sheet and embedded style sheet?**

The external style sheet is a kind of style sheet in which the styles are defined in a separate.css file and this file is mentioned at the beginning of the HTML document. When



we need to apply the particular style to more than one web documents then the external style sheet is used. The embedded style sheet is a method in which the style is specified within the HTML document itself. It is not defined in separate file. Due to embedded style sheet unique style can be applied to all the elements.

**24) What do you mean by the term inline element?**

The inline elements are those elements that do not form new blocks of content. The content is distributed in lines.

**25) What are the various style sheets?**

Inline, external, imported and embedded are the different types of style sheets.

**26) Explain inline, embedded and external style sheets.**

**Inline**

If only a small piece of code has to be styled then inline style sheets can be used.

**Embedded**

Embedded style sheets are put between the <head> </head> tags.

**External**

If you want to apply a style to all the pages within your website by changing just one style sheet, then external style sheets can be used.

**27) Define Hyperlink. (May 2014)**

Connection between a hypertext or graphical element (button, drawing, image), and one or more such items in the same or different electronic document. Upon clicking the mouse button on a hyper-linked item, the reader is automatically transferred to the other end of the hyperlink which could be another document or another website. Also called hotlink.

**28) How will you embed the external style sheet? (May 2014)**

In external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing just one file. Each page must include a link to the style sheet with the <link> tag. The <link> tag goes inside the head section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. An example of a style sheet file is shown below:

**"myStyle.css":**

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

**29) Differentiate between client-side scripting and server side scripting. (MAY 2015)**

Server side scripting is used to create dynamic pages based a number of conditions when the users browser makes a request to the server. Client side scripting is used when the users browser already has all the code and the page is altered on the basis of the users input. Client side scripting is that which runs on user browser. client side scripting is used for validation JavaScript, VB script are client side scripting. server side scripting is that which runs on server. Used to insert and fetch record from database PHP, ASP are server side scripting.

**30) What is the difference between hotspot and hyperlink? (MAY 2015)**

It is a tool that is used widely in Web development, Multimedia applications and gaming. On some web pages you may be presented with an image which fills the screen. On that picture there may hotspots which, when clicked, will take you to another page. A typical example of this would be a site map of an organization or a map of the world.

Hyperlink is a link by clicking on which you will get redirected to the different website or web page, following is the code to make any word, phrase or keywords hyperlinked.

**31) What is the differences between relative and absolute positioning? (MAY 2016)**

A relative positioned element is positioned relative to its normal position. To position an element relatively, the property position is set as relative. The difference between absolute and relative positioning is how the position is being calculated.

**32) Define Markup language.**

Markup languages are designed for the processing, definition and presentation of text.

The language specifies code for formatting, both the layout and style, within a text file. The code

used to specify the formatting are called tags. HTML is an example of a widely known and used markup language.

### UNIT- III

#### 1) What is JavaScript?

JavaScript is a platform-independent, event-driven, interpreted client-side scripting language developed by Netscape Communications Corp. and Sun Microsystems.

#### 2) What are the uses of JavaScript?

JavaScript can be used as an alternate to Java applets. JavaScript can be embedded in XHTML, JavaScript can be used to create cookies.

#### 3) What are the Basic Terminology of Programming Languages?

Expression, Statement, Keyword, Reserved Word.

#### 4) What is meant by statements in JavaScript?

Statements are the essence of a language like JavaScript. They are instructions to the interpreter to carry out specific actions. For example, one of the most common statements is an assignment. Assignment uses the `=` operator and places the value on the right-hand side into the variable on the left. For example, `x = y + 10;` adds 10 to y and places the value in x.

#### 5) What are the primitive data types in JavaScript?

JavaScript supports five primitive data types: number, string, Boolean, undefined, and null. These types are referred to as primitive types because they are the basic building blocks from which more complex types can be built. Of the five, only number, string, and Boolean are real data types in the sense of actually storing data. Undefined and null are types that arise under special circumstances.

#### 6) What are the Escape Codes Supported in JavaScript?

The escape codes supported in JavaScript are `\b` Backspace, `\t` Tab (horizontal), `\n` Linefeed (newline), `\v` Tab (vertical), `\f` Form feed, `\r` Carriage return, `\"` Double quote, `\'` Single quote, `\\` Backslash.

#### 7) Define Composite Types.

An object is a composite type that can contain primitive and composite types. An object is a collection that can contain primitive or composite data, including functions and other objects. The data members of an object are called properties, and member functions are known as methods.

#### 8) Define Array in JavaScript.

An array is an ordered list that can contain primitive and complex data types. Arrays are sometimes known as vectors or list in other programming languages and are actually Array objects in JavaScript. The members of an array are called elements. Array elements are numbered starting with zero. That is, each element is assigned an index, a non-negative integer indicating its position in the array.

#### 9) Define Function in JavaScript.

A function is another special type of JavaScript object, one that contains executable code. A function is called (or invoked) by following the function name with parentheses. Functions can take arguments (or parameters), pieces of data that are passed to the function when it is invoked. Arguments are given as a comma-separated list of values between the parentheses of the function call.

#### 10) What is meant by identifiers?

An identifier is a name by which a variable or function is known. In JavaScript, any combination of letters, digits, underscores, and dollar signs is allowed to make up an identifier.

#### 11) Define Operators in JavaScript.

JavaScript supports a variety of operators. Some of them, like those for arithmetic and comparison, are easy for even those new to programming to understand. Others, like the bitwise AND (&), increment (++), and some conditional (? :) operators.

#### 12) What is JavaScript name spacing? How and where is it used?

Using global variables in JavaScript is evil and a bad practice. That being said, namespacing is used to bundle up all your functionality using a unique name. In JavaScript, a namespace is really just an object that you've attached all further methods, properties and objects. It promotes modularity and code reuse in the application.

#### 13) How many looping structures can you find in JavaScript?

If you are a programmer, you know the use of loops. It is used to run a piece of code multiple times according to some particular condition. Javascript being a popular scripting language supports the following loops: for, while, do-while loop

**14) Mention The Various Java Script Object Models.**

Math Object, String Object, Date Object, Boolean and Number Object, Document Object window Object.

**15) How Scripting Language Is Differs From Html.**

HTML is used for simple web page design, HTML with FORM is used for both form design and Reading input values from user, Scripting Language is used for Validating the given input values whether it is correct or not, if the input value is incorrect, the user can pass an error message to the user, Using form concept various controls like Text box, Radio Button, Command Button, Text Area control and List box can be created.

**16) What are the different types of objects in JavaScript?**

Type	Example	Implementation Provided By	Governing Standard
User-defined	Programmer defined Customer or Circle	Programmer	None
Built-in	Array, Math	The browser via engine its JavaScript	ECMA-262
Browser	Window, Navigator	The browser	None (though some portions adhere to an adhoc standard)
Document	Image, HTMLInputElement	The browser via its DOM engine	W3C DOM

**17) Justify “JavaScript” is an event-driven programming”**

Javascript supports event driven programming. when user clicks the mouse or hit the keys on the keyboard or if user submits the form then these events and response to them can be handled using javascript. Hence javascript is mainly used in web programming for validating the data provided by the user.

**18) What is the use of pop up boxes in java script?**

There are three types of popup boxes used in javascript. Using these popup boxes the user can interact with the web application.

**19) What is DOM?**

Document Object Model (DOM) is a set of platform independent and language neutral application interface (API) which describes how to access and manipulate the information stored in XML, XHTML and javascript documents.

**20) Enlist any four mouse events.**

The MouseEvent are-mousedown, mouseup, mouseover, mousemove, mouseout.

**21) List ad various level of document object modeling.**

Various levels of DOM are DOM0, Dom1, Dom2, and Dom3

**22) Define Math Object.**

The Math object holds a set of constants and methods that enable more complex. The math object is static, so properties are accessed directly. For example, to compute the square root of 10, the sqrt() method is accessed through the Math object directly.

**23) What you meant BY Date Pickers?**

HTML5 introduces a number of new form fields for selecting date and time

**24) What are the three possible choices for form validation?**

They are three possible choice are before they happen, as they happen, after they happen

**25) What are they validation properties and methods?**

Validation properties and methods are checkvalidity (), validaionMessage, customerror, patternMismatch, rangeOverflow, rangeUnderflow, tooLong.

**26) Define event bubbling.**

Suppose, there is an element present inside another element. Then during the event handling, if the event which is present in the inner element is handled and then the event of the outer element is handled. This process of event handling is called event bubbling

**27) How to create arrays in Javascript?**

We can declare an array like this Var scripts = new Array();

We can add elements to this array like this



```

scripts[0] = "PHP";
scripts[1] = "ASP";
scripts[2] = "JavaScript";
scripts[3] = "HTML";

```

Now our array scripts has 4 elements inside it and we can print or access them by using their index number. Note that index number starts from 0. To get the third element of the array we have to use the index number 2. Here is the way to get the third element of an array. document.write(scripts[2]); We also can create an array like this var no\_array = new Array(21, 22, 23, 24, 25);

## 28) Write a simple program in JavaScript to validate the email-id. (May 2014)

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script>
```

```

function validateForm() {
var x = document.forms["myForm"]["email"].value;
var atpos = x.indexOf("@");
var dotpos = x.lastIndexOf(".");
if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length) {
    alert("Not a valid e-mail address");
    return false; } }

```

```
</script>
```

```
</head>
```

```
<body>
```

```

<form name="myForm" action="demo_form.asp" onsubmit="return validateForm();"
method="post">

```

```
Email: <input type="text" name="email">
```

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

```
</form>
```

```
</body>
```

```
</html>
```

## 29)How to write function using Java Script? Give Example. (May 2014)

A JavaScript function is defined with the **function** keyword, followed by a **name**, followed by parentheses ().

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

The parentheses may include parameter names separated by commas: (*parameter1*, *parameter2*, ...)

The code to be executed, by the function, is placed inside curly brackets: {}

```

functionName(parameter1, parameter2, parameter3) {
    code to be executed
}

```

Function **parameters** are the **names** listed in the function definition.

Function **arguments** are the real **values** received by the function when it is invoked.

Inside the function, the arguments are used as local variables

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>This example calls a function which performs a calculation, and returns the result:</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```

function myFunction(a, b) {
    return a * b; }

```

```
document.getElementById("demo").innerHTML = myFunction(4, 3);
```

```

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

```

This example calls a function which performs a calculation, and returns the result:

**30) Write JavaScript to display current time on the status bar.(MAY 2015)**

```

<html>
<head>
<title>date format</title>
<script language="javascript">
var t=new Date();
var m=t.getMonth()+1;
var d=t.getDate();
var y=t.getFullYear();
window.status= d + " - " + m + " - " + y
</script>
</head>
<BODY>
<P>CURRENT DATE DISPLAY ON STATUS BAR
</body>
</html>

```

**31) Name and mention the use of any four methods in JavaScript Math Object.(MAY 2015)**

The Math object holds a set of constants and methods that enable more complex. The math object is static, so properties are accessed directly. For example, to compute the square root of 10, the sqrt() method is accessed through the Math object directly., ceil ,round, Floor.

**32) What are the core features of JavaScript?(MAY 2016)**

- A lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-Platform.

**33) What is the need for event Handling?(MAY 2016)**

They are two ways of using Event Handlers are  
 Event Handlers as Attributes.  
 Event Handlers as Properties.

## UNIT-IV

**1) What are the different types of detection categories?**

Technical issues ex: javascript support, Visual issues, Delivery issues, user issues

**2) Define media queries.**

A media query takes the CSS media attribute and extends it with conditions that avoid using javascript for simple device capability detection. For example, web developers commonly are familiar with one style rule for print and one one for screen.

**3) What are the navigator properties for visual detection?**

The navigator properties for visual detection are availHeight, availleft, availTop, availWidth, colorDepth, Pixel dept

**4) What is a Cookie?**

A cookie is a variable that is stored on the visitor's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With JavaScript, you can both create and retrieve cookie values.

**5) What are the two different types of cookies?**

**Name cookie** - The first time a visitor arrives to your web page, he or she must fill in her/his name. The name is then stored in a cookie. Next time the visitor arrives at your page, he or she could get a welcome message like "Welcome John Doe!" The name is retrieved from the stored cookie

**Date cookie** - The first time a visitor arrives to your web page, the current date is stored in a cookie. Next time the visitor arrives at your page, he or she could get a message like "Your last visit was on Tuesday August 11, 2005!" The date is retrieved from the stored cookie.

**6) What is meant by event handlers?**

A function or method containing program statements that are executed in response to an event. An event handler typically is a software routine that processes actions such as keystrokes and mouse movements. With Web sites, event handlers make Web content dynamic. JavaScript is a common method of scripting event handlers for Web content.

**7) Define Protocol Handlers and content handlers..**

Protocol handlers are implemented as subclasses of the `URLConnectionHandler` class. The `URLConnectionHandler` class defines four access methods that can be overridden by its subclasses, but only the `openConnection()` method is required to be overridden.

Content handlers are given the ability to register a mime type to be handled by a specific URL. There are some Content Handler Generic, Word, Embed, and Sanitize.

**8) What are the advantages and disadvantages of cookie?**

Used to store user-specific information within a user system to reduce burden on server. i.e., user-specific information can be credentials (username and password), sessionid, security token.

Cookies can be disabled on user browsers, Cookies are transmitted for each HTTP request/response causing overhead on bandwidth, No security for sensitive data

**9) What does Rollover mean?**

Rollover is a JavaScript technique used by Web developers to produce an effect in which the appearance of a graphical image changes when the user rolls the mouse pointer over it. Rollover also refers to a button on a Web page that allows interactivity between the user and the Web page. It causes the button to react by either replacing the source image at the button with another image or redirecting it to a different Web page.

**10) What is meant by animating the canvas?**

Animation involves repeatedly clearing the canvas and drawing an image on it. To create smooth animations, you need to minimize the time between clearing the canvas and completing the new drawing, and you need to keep the changes relatively small from image to image.

For smooth animation, use the following sequence:

1. Model—calculate small changes in image substitution, position, rotation, color, or scale.
2. Clear—Clear part or all of the canvas.
3. Draw—Draw any images, using the pre-calculated values. Stroke or fill any paths, shapes, or strings., Repeat steps 1-3.

**11) Define Compositing**

In video terminology, compositing is the merging of two video tracks in order to produce a new single frame from the combined tracks. The term may also be used to describe the overlaying of text and titles on video clips.

**12) What is meant by Augmenting classes in JavaScript?**

JavaScript's prototype-based inheritance mechanism is dynamic: an object inherits properties from its prototype, even if the prototype changes after the object is created. This means that we can augment JavaScript classes simply by adding new methods to their prototype objects.

**13) What is the difference between SessionState and ViewState?**

ViewState is specific to a page in a session. Session state refers to user-specific data that can be accessed across all pages in the web application.

**14) How to create a function using function constructor?**

The following example illustrates this

It creates a function called square with argument x and returns x multiplied by itself.  
`var square = new Function ("x","return x*x");`

**15) Define Class in Javascript with examples.**

A new JavaScript class is defined by creating a simple function. When a function is called with the new operator, the function serves as the constructor for that class. Internally, JavaScript creates an Object, and then calls the constructor function. Inside the constructor, the variable `this` is initialized to point to the just created Object. This code snippet defines a new class, Foo, and then creates a single object of that class. `function Foo()`

```
{
  this.x = 1;
```

```
this.y = 2;}
obj = new Foo;
```

**16) What is a object constructor? (May 2014/2015)**

A constructor is any function which is used as a constructor. The language doesn't make a distinction. A function can be written to be used as a constructor or to be called as a normal function, or to be used either way. A constructor is used with the new keyword.

**17) Write short notes on object constructor and prototyping.**

An object constructor is merely a regular JavaScript function, so it's just as robust ie: define parameters, call other functions etc. The difference between the two is that a constructor function is called via the new operator (which you'll see below). By basing our object definition on the function syntax, we get its robustness as well. Lets use a real world item "cat" as an example. A property of a cat may be its color or name. A method may be to "meeyow". The important thing to realize, however is that every cat will have a different name or even meeyow noise. To create an object type that accommodates this need for flexibility, we'll use an object constructor:

**18) Write short notes on Chaining and constructor chaining.**

In object-oriented programming, a class B can extend or subclass another class A. We say that A is the super class and B is the subclass. Instances of B inherit the entire instance methods of A. The class B can define its own instance methods, some of which may override methods of the same name defined by class A. If a method of B overrides a method of A, the overriding method in B may sometimes want to invoke the overridden method in A: this is called method chaining. Similarly, the subclass constructor B() may sometimes need to invoke the superclass constructor A(). This is called constructor chaining.

**19) Define jQuery function.**

The jQuery function is the value of jQuery or of \$. This is the function that creates JQuery objects, registers handlers to be invoked when the DOM is ready, and that also serves as the jQuery namespace

**20) What is meant by JQueryobject?**

A jQuery object is an object returned by the jQuery function. A jQuery object represents a set of document elements and can also be called a "jQuery result," a "jQuery set," or a "wrapped set." "the selected elements" When you pass a CSS selector to the jQuery function, it returns a jQuery object that represents the set of document elements that match that selector.

**21) What is Ajax?(MAY 2016)**

Ajax is a set of client side technologies that provides asynchronous communication between user interfaces and web server. So the advantages of using Ajax are asynchronous communication, minimal data transfer and server is not overloaded with unnecessary load.

**22) What technologies are being used in AJAX?**

AJAX uses four technologies, which are as follows:

JavaScript, XMLHttpRequest, Document Object Model (DOM), Extensible HTML (XHTML) and Cascading Style Sheets (CSS)

**23) Describe AJAX Control Extender Toolkit.**

AJAX Control Toolkit is a set of extenders that are used to extend the functionalities of the ASP.NET controls. The extenders use a block of JavaScript code to add new and enhanced capabilities to the ASP.NET controls. AJAX Control Toolkit is a free download available on the Microsoft site. You need to install this toolkit on your system before using extenders.

**24) What is the syntax to create AJAX objects?**

AJAX uses the following syntax to create an object:

```
Var myobject = new AjaxObject("page path");
```

The page path is the URL of the Web page containing the object that you want to call.

The URL must be of the same domain as the Web page.

**25) How can you find out that an AJAX request has been completed?**

You can find out that an AJAX request has been completed by using the readyState property. If the value of this property equals to four, it means that the request has been completed and the data is available.

**26) What are the different ways to pass parameters to the server?**

We can pass parameters to the server using either the GET or POST method. The following code snippets show the example of both the methods:  
Get: XMLHttpRequest.Open("GET", "file1.txt", true);

Post: XMLHttpRequest.open("POST", "file2.txt", true);

**27) What are the extender controls?**

The extender controls uses a block of JavaScript code to add new and enhanced capabilities to ASP.NET. The developers can use a set of sample extender controls through a separate download - AJAX Control Toolkit (ACT).

**28) List out the advantages and limitations of AJAX. (May 2014)**

**Advantages:** Better interactivity, Easier navigation, Compact, Backed by reputed brands

**Disadvantages:** It is difficult to bookmark a particular state of the application, Function provided in the code-behind file do not work because the dynamic pages cannot register themselves on browsers history engine automatically.

**29) Define the super classes and subclasses with example. (MAY 2015)**

Every class can have a super class from which it inherits properties and methods. Any class can be extended, or sub-classed so the resulting subclass can inherit its parent's behavior. As we have seen, JavaScript supports prototype inheritance instead of class based. It's possible for inheritance to happen other ways, however.

```
<script language="javascript" type="text/javascript">
function superClass()
{
    this.supertest = superTest; //attach method superTest
}
function subClass()
{
    this.inheritFrom = superClass;
    this.inheritFrom();
    this.subtest = subTest; //attach method subTest
}
function superTest()
{
    return "superTest";
}
function subTest()
{
    return "subTest";
}
var newClass = new subClass();
alert(newClass.subtest()); // yields "subTest"
alert(newClass.supertest()); // yields "superTest"
</script>
```

**30) What is the use of Constructor? (MAY 2016)**

JavaScript constructors are special methods that create and initialize the properties for newly created objects. A constructor must be able to reference the object on which it is to operate.

## UNIT-V

**1) What is PHP?**

PHP is a powerful tool for making dynamic and interactive Web pages, PHP stands for Hypertext Preprocessor, PHP is a server-side scripting language, like ASP, PHP scripts are executed on the server, PHP supports many databases (MySQL, ORACLE), and PHP is open source software.

**2) How PHP works for the web?**

Understanding how the World Wide Web works is necessary to understand how PHP works for web. The Web is a network of computers that offer Web pages. Millions of Web sites are on the Web. To enable Web surfers to find the Web sites they want to visit, each Web page has an address, called a URL. This includes the Web site's domain name and the filename, such as www.mycompany.com/welcome.html. When Web surfers want to visit a Web page, they type the URL into their Web browsers. The following process is set in motion: The Web browser sends a message out onto the Web, requesting the Web page and the message is sent to the computer at the address specified in the URL.

**3) What is meant by LAMP server?**

LAMP stack is a group of open source software used to get web servers up and running. The acronym stands for Linux, Apache, MySQL, and PHP.



**4) What are the steps to set a LAMP server?****Step One—Install Apache**

Apache is free open source software which runs over 50% of the world's web servers. To install apache, open terminal and type in these commands:

**Step Two—Install MySQL**

MySQL is a widely-deployed database management system used for organizing and retrieving data. Step Three—Install PHP

PHP is an open source web scripting language that is widely used to build dynamic webpages.

*Step Four—RESULTS: See PHP on your Server*

*Although LAMP is installed, we can still take a look and see the components online by creating a quick php info page*

**5) What is a PHP File?**

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"

**6) What are the main functions for PHP?**

PHP can generate dynamic page content, PHP can create, open, read, write, and close files on the server, PHP can collect form data, PHP can send and receive cookies, PHP can add, delete, modify data in your database, PHP can restrict users to access some pages on your website, PHP can encrypt data.

**7) What is the basis of PHP?**

PHP is a server side scripting language used on the Internet to create dynamic web pages. It is often coupled with MySQL, a relational database server that can store the information and variables the PHP files may use. Together they can create everything from the simplest web site to a full blown business web site, an interactive web forum, or even an online role playing game.

**8) What is the difference between Print and Echo Statements?**

Differences between echo and print, is echo - can output one or more strings, print - can only output one string, and returns always 1

**9) Define Variables in PHP.**

PHP variables must begin with a "\$" sign, Case-sensitive (\$Foo != \$foo != \$fOo)

Global and locally-scoped variables, Global variables can be used anywhere, Local variables restricted to a function or class, Certain variable names reserved by PHP, they are two types one is Form variables (\$\_POST, \$\_GET), and second one is Server variables (\$\_SERVER)

**10) Write short notes on Constants in PHP.**

A constant is an identifier (name) for a simple value. A constant is case-sensitive by default. By convention, constant identifiers are always uppercase.

```
<?php
// Valid constant names
define("FOO", "something");
define("FOO2", "something else");
define("FOO_BAR", "something more");
// Invalid constant names (they shouldn't start
// with a number!)
define("2FOO", "something");
// This is valid, but should be avoided:
// PHP may one day provide a "magical" constant
// that will break your script
define("__FOO__", "something");
?>
```

**11) What are the Formatting and Printing Strings available in PHP?**

Function	Description
printf()	Displays a formatted string
sprintf()	Saves a formatted string in a variable
fprintf()	Prints a formatted string to a file
number_format()	Formats numbers as strings

**12) What are the Operators available in PHP?**

Arithmetic Operators: +, -, \*, /, %, ++, --, Assignment Operators: =, +=, -=, \*=, /=, %=

Comparison Operators: ==, !=, >, <, >=, <= ,Logical Operators: &&, ||, !, String Operators: . and .= (for string concatenation)

**13) What are the three different types of control structures in PHP?**

Control Structures are the structures within a language that allow us to control the flow of execution through a program or script, Grouped into conditional (branching) structures (e.g. if/else) and repetition structures (e.g. while loops) and For loops.

**14) Give one Example for If/Else statement.**

Example if/else if/else statement:

```
if ($foo == 0) {
    echo 'The variable foo is equal to 0';
}
else if (($foo > 0) && ($foo <= 5))
{
    echo 'The variable foo is between 1 and 5';
}
else {
    echo 'The variable foo is equal to ' . $foo;
}
```

**15) Write the syntax for Switch –Case Structures.**

```
Switch ($number)
{
    Case 1:
        echo "small";
        break;
    Case 2:
        echo "medium";
        break;
    Case 3:
        echo "large";
        break;
    Default:
        echo "That is not a valid code.";
}
```

**16) Write the syntax for while loop in PHP.**

PHP Syntax	Example
while (expression) { statements; }	\$timeleft = 10; while (\$timeleft > 0) { echo \$timeleft."seconds left "; \$timeleft = \$timeleft-1; }

**17) Write short notes on array.(MAY 2015)**

Arrays are complex variables that store a group of values under a single variable name.

An array is useful for storing a group of related values.

In PHP arrays are nothing but group of key/value associations under a single array name.

The keys can be either numeric or text. Depending on the type of key, the arrays are classified as Numeric Arrays (Keys are numbers), Associative Arrays (Keys are strings)

**19) What is meant by function in PHP?**

Callable code to which data can be passed and returned, Breaks up code so that the same piece of code can be executed from different points, without redundancy, Easy to comprehend

**20) How to create and calling Functions?**

Creating and Calling functions

Function definition syntax

```
function function_name($arg1, $arg2,.....)
{
```

```
statements;
return return_value
}
```

function and return are keywords, function\_name is the user given name to the function \$arg1, \$arg2..... are the parameters passed to the function. The arguments can be both variables and arrays return\_value is the value returned from the function Function is called by giving the name of function and giving the arguments within parentheses. If the function returns a value, it is assigned to a variable

## 20) Define References in PHP.

References are nothing but pointers. That is these are variables that hold the address of another variable. If one variable changes the value, then it will be reflected in the other variable also.

```
$val = 4;
$ref = &$val;
$ref++;
echo $val
```

Here the output will be 5 since both \$ref and \$val point to the same memory location

## 21) What are the two different forms tag attributes to reading data from a webpage?

A form is a construct in HTML which enables users to interact and enter data that can be further processed on the server, Form tag has two attributes

Method: specifies the way in which data is to be sent to the server (GET or POST)

Action: specifies the URL where the data is to be sent, If action attribute is not specified, then data is sent to the same URL.

## 22) How to handle text areas in PHP?

Handling text area

Text areas are used for multiline text input

Input tag:

```
<textarea name = "text" cols = 50 rows = 5>
</textarea>
```

In PHP

```
$data = $_REQUEST['text'];
```

## 23) How to handle checkboxes in PHP?

### Handling Check Boxes

These are square shaped controls that can be selected or deselected with a mouse. These are used typically when we want the user to give input in terms of Yes/No,

<input type = "checkbox" name = "check" value="Yes"> ,The string specified for the value attribute "Yes" is sent to the PHP script with the form ,data and the value can be accessed through the name attribute "check", • The value of the checkbox can be accessed only if the checkbox has been ticked. We can check whether the check box has been ticked using the isset() function. The isset function takes a variable name as parameter and checks whether the variable exists or not.

PHP script:

```
If(isset($_REQUEST['check']))
echo $_REQUEST['check'];
else
echo "Checkbox was not ticked";
```

## 24) How to handle radio buttons in PHP?

### Handling Radio Buttons

Check boxes are useful if we want the user to select multiple items from a number of Choices, Radio buttons are useful if we want user to select only one item from a number of choices

All the radio buttons that form a group should have the same name attribute,

```
<input type = "radio" name = "sex" value = "Male">
<input type = "radio" name = "sex" value = "Female">
if(isset($_REQUEST['sex']))
echo "You are a $_REQUEST['sex'];
```

## 25) Write short notes on embedding PHP within HTML.

PHP script may be embedded within HTML documents ie meaning PHP and HTML code can both happily co-exist in the same file. All embedded PHP code must be contained within<?php and?> tags so it can be readily recognized by the PHP engine for interpretation.

Typically the PHP code will write content into the body section of the html document. This is then sent to web browser.

## 26) Define MySQL.

**MySQL** is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. The official MySQL Workbench is a free integrated environment developed by MySQL AB that enables users to graphically administer MySQL databases and visually design database structure.

## 27) What are things to be considered for PHP with MySQL?

There are five things you want to be able to do from within PHP in relation to a MySQL database they are Connect to the MySQL database, Disconnect from the MySQL database, Execute MySQL queries, Work with the results from an SQL query, Check the status of your MySQL command.

## 28) write down the general structure for each looping structure in PHP. (May 2014)

**for** - loops through a block of code a number of times

**for/in** - loops through the properties of an object

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

**do/while** - also loops through a block of code while a specified condition is true

The for loop has the following syntax:

```
for (statement 1; statement 2; statement 3)
```

```
{
    code block to be executed
}
```

while loops are the simplest type of loop in PHP. They behave just like their C counterparts. The basic form of a while statement is:

```
While(expr)
```

```
statement
```

## 29) when print/echo statement used in PHP? (May 2014)

There are some differences between echo and print:

echo - can output one or more strings

print - can only output one string, and returns always 1

echo is marginally faster compared to print as echo does not return any value.

## 30) State how web works (MAY 2015).

The Web works on a client / server model in which client software (Browser) runs on a local computer. The Server software runs on a web host. To use the web you must make an Internet connection and then launch your web browser.

- Type the URL you want to visit
- Browser makes a connection to a server using HTTP protocol
- The Server then
  - Accepts the connection
  - Sends the file contents back to browser, and
  - Closes the connection

## 31) What are the features of PHP? (MAY 2016)

PHP is much easier to learn and use

PHP was written Specifically for Dynamic webpage Creation.

PHP is both free and cross-Platform

PHP is the most popular tool available for developing Dynamic Web sites.

## 32) What are the software components in LAMP? (MAY 2016)

The LAMP server (Linux, Apache, MySQL, PHP (or Perl)) is one of the most important servers one might ever set up.

## PART B

### UNIT I

#### 1) Explain in detail the basic internet protocols.

Protocols are set of rules that provides a communication service that higher level objects use to exchange messages.

A protocol defines two interfaces

Service Interface: Define the operation that local objects can perform on the protocol

Peer Interface: Defines the form and meaning of messages exchanges between protocol peers to implement the communication service.

For internet and World Wide Web the following protocols are most widely used.

1. HTTP 2. SMTP 3. POP3 4. File Transfer Protocol

**HTTP**

The Hyper Text Transfer Protocol (HTTP) is a request/response protocol. It is a communication protocol used to transfer the information on local network and World Wide Web

It is a standard protocol for communication between web browsers and web servers .It is a text oriented protocol.

**Steps:**

HTTP protocol defines four steps for each request from a client to the server. They are

**Making the Connection**

Making a request.

Operation	Description
Options	Request information about available options.
GET	Get the document identified in URL
HEAD	Get meta information about document identified in URL
POST	Gives information to the server.
PUT	Stores the document in specified URL
DELETE	Deletes the specified URL.
TRACE	It loopbacks request message
Connect	It is used by proxies

**Response.**

.closing the connection

**Making the Connection:**

The client establishes a TCP connection to the server. By default, the connection is made on port 80. Other ports may be specified in the URL.

**Making a Request:**

The client sends a message to the server requesting a page at the specified URL. There are 8 types of HTTP request. They are

Operation	Description
Options	Request information about available options.
GET	Get the document identified in URL
HEAD	Get meta information about document identified in URL
POST	Gives information to the server.
PUT	Stores the document in specified URL
DELETE	Deletes the specified URL.
TRACE	It loopbacks request message
Connect	It is used by proxies

**The Response**

The server sends a response to the client often the response begins with a response code, followed by MIME header information, blank line, then requested information or an error message

**Closing the Connection:**

The connection can be closed either by the client or by the server only both.

**Features of HTTP Protocol.**

1. It is communication protocol used between web browser and web server.
2. This protocol is based on request-response messaging. That means client makes the request of desired web page and then the server responds it by sending the requested resource.
3. It is stateless protocol. That means HTTP protocol cannot remember the previous user's information nor it remembers the number of times the user has visited particular websites.
4. The request-response message consists of plain text in fairly readable form.
5. The HTTP protocol has a cache control.



## SMTP

SMTP stands for Simple Mail Transfer Protocol. It is simple ASCII protocol. This protocol establishes the TCP connection to port 25, act as client, waits for the receiving machine operating as the server to talk first.

The server starts by sending a line of text giving its identify and telling whether it is prepared to revive mail. If it is not, the client releases the connection and tries again later.

If the server is willing to accept email, the client announces whom the email is coming from and whom the message is returned to.

If such a recipient exist the server gives the client the go ahead to send the message and the server acknowledge it. No checksum are needed because TCP provides a reliable byte stream.

When all the email has been exchanged in both directions the connections is released.

Example dialog for sending the message including the numerical codes used by SMTP.

S: Server

C:Client-

S:220xyz.com SMTP service ready

C:HELO abod.com

S:250 xyz.com says hello to abed.com

C:MAIL FROM:Rosaline@abcd.com

S:250 sender ok

C:RCPT TO:C\_Joseph@xyz.com

S:250 recipient ok

C:DATA

S:354 send mail; end with “,”On a line by itself

C: from:Rosaline@abcd.com

C:To C\_Joseph@xyz.com

C:MIME\_version:1.0

C: Messages\_id:0704760941.AA00747@abcd.com

C:Content\_Type:Multiport/alternative

C:Subject:Hi Son

C:Content\_Type:text/enriched

C:Happy birthday to you Joseph

C: Content\_type:audio/basic

C:Content\_transfer\_encoding:base64

C:

S:250 message accepted

C:QUIT

S:221 xyz.com closing connection

The commands used are

- 1.Hello- to Establish Connection.
- 2.RCPT- to send a single message to multiple receivers.
3. MAIL\_FROM- to introduce the sender to the receiver.
4. DATA- The message to be sent in e-mail which will end with.
5. HELP-to know what command the SMTP server will again.

Advantages:

Well Defined.

Simple to use

Disadvantages:

Messages should not exceed 64KB length.

Infinite mail can be triggered.

If the client and server have different timeout, one of them may give up while the other is till busy unexpectedly terminating the connection.

POP3:

POP3 stands for Post Office Protocol Version3.

POP3 allows user Transfer Agent to contract message transfer agent and allow e-mail to be copied from ISP to the user.

Two situations can be handled by POP3

If both the sender and reciver are online.

If sender is online but reciver is not.





### Data Transfer Connection

Steps:

1. POP3 begins when user starts mail reader.
2. Mail reader calls up the ISP and establishes a TCP Connection with MTA at port 110
3. POP3 goes through 3 states in sequence after connection establishment

Authorization:

It deals with having user login.

Transaction:

It deals with the user collecting the e-mails and marking them for deletion from the mailbox.

Update:

Actually causes email to be deleted.

Commands:

USER – Gets the user name

PASS- Gets the password of the user.

LIST- Causes the server to list the contents of mailbox, one message per line, giving the length of the message period (.) terminates the list.

RETR- Used to retrieve the messages

DELE- Used to mark the messages to be deleted.

QUIT- Used to terminate the transaction state and enter update state.

FTP:

The File transfer protocol sets the rules for transferring files between computers. When user wants to download a file from the server FTP is used. There are various issues that must be solved during such file transfer. These issues can be described as follows

Client must have an authority to download particular file.

The hardware and software on both the computers might be different.

The data interpretation and data formats on client and server might be different.

FTP provides the mechanism to handle these issues internally and user remains free different.

FTP uses two connections between client and server. One connection is used for actual data transfer and other is used for control information. This separation of data and commands makes the FTP more Efficient.

When client makes a request for particular file download then using the data transfer connection actual data gets transmitted from server to the client. At the same time server keeps track of how much data is sent so far and how much is remaining. This tracking can be done using the control transfer connection. Hence during the file downloading/uploading we can see a message about how many bytes are getting transferred and how much time is remaining. Various commands used in FTP are

S.No	Command	Meaning of the command
1	USER	It identifies the user
2	PASS	Password
3	QUIT	For Logging off
4	CWD	Change to another directory
6	DELE	Delete the file
7	LIST	Display the contents of the directory
8	RETR	Retrieves files from server to client
9	STOR	Stores file from client to server
10	APPE	Appending the existing file
11	HELP	Asking for the help

## TCP

It is a higher-level protocol extends IP to provide additional functionality

Reliable communication based on the concept of communication

TCP adds concept of a connection on top of IP

Provides guarantee that packets delivered

Provide two-way (full duplex) communication

A and B both send messages to one another at the same time.

Reliable data transmission by demanding an ACK for each packet it sends via IP

Splitting longer messages into shorter ones

Reassembling on receiver side.

TCP also adds concept of a port

The port concept allows TCP to communicate with many different applications on a machine.

TCP header contains port number representing an application program on the destination computer

Some port numbers have standard meanings

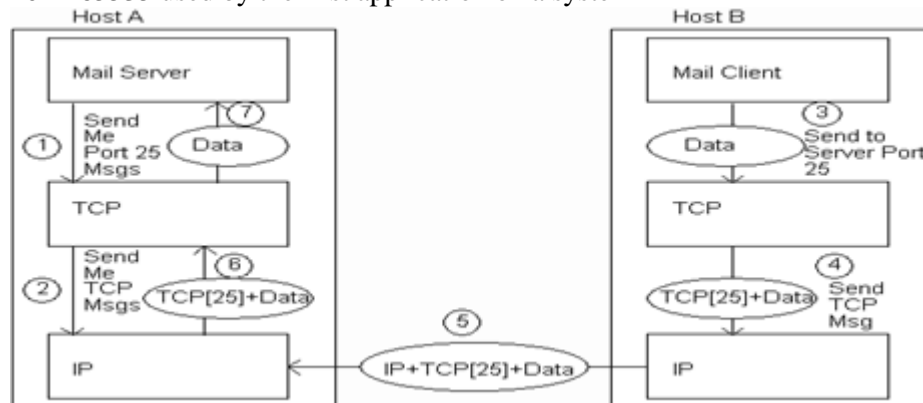
Example: port 25 is normally used for email transmitted using the Simple Mail Transfer Protocol (SMTP)

Other port numbers are available first-come-first served to any application

Assigned by IANA

0-1023 requested only by the applications that are run by the system at boot-up

1024-65535 used by the first application on a system



## 2) Discuss about the elements of WWW.

The Web is the collection of machines (Web servers) on the Internet that provide information, particularly HTML documents, via HTTP.

Machines that access information on the Web are known as Web clients. A Web browser is software used by an end user to access the Web.

Distinctive feature of Web: support for hypertext (text containing links)

It uses Hypertext Markup Language (HTML) for Document representation

Public Sharing of Information-Internet

Usenet newsgroup service- 1979

worldwide distributed Internet discussion system

Posting Information that could be read by users on other system

First Internet Chat software-Internet Relay Chat

Various technologies were developed for supporting information management and search on the internet.

Gopher-hierarchical view of documents

WAIS-(Wide area information system)-Used indexing

ARCHIE -Search online info archives via FTP

Unique feature of Web: support for hypertext (text containing links)

Communication via Hypertext Transport Protocol (HTTP)

Document representation using Hypertext Markup Language (HTML)

A Web browser is software used by an end user to access the Web.

The Hypertext Concept

The hypertext is a way of presenting information in such a way that the sequence of the information is left up to the reader.

This sequence is usually non-linear, however readers can follow a linear sequence.

The hypertext works by means of hyperlink, which are highlighted or underlined words (or graphics) that one can click to bring another document into view.

Web Browser

A web browser displays a web document and enables users to access web documents.

When the user clicks a hyperlink, the browser initiates a message to a web server.

This message requests the server to retrieve the requested information and send it back to the web browser through the telecommunications network.

#### Web Server

This is a program that waits patiently for the browser to request a web page.

The server looks for the requested information, retrieves it and sends it to the browser or sends an error message if the file is not found.

Uniform Resource Locators (URL) and Transfer protocols

These are the web addresses.

The resource locator is an addressing system that precisely states where a resource is located.

It is divided into four parts: protocol; server/ domain name; path and the resource name. eg: <http://www.cites.uiuc.edu/101/url101.html>

The first portion of the URL (http) designates the protocol that is used to locate the file or resource on the Web.

A protocol is a standardized means of communication among machines across a network. Protocols allow data to be taken apart for faster transmission, transmitted, and then reassembled at the destination in the correct order.

Here, http represents the HyperText Transfer Protocol, which is used to transfer Web pages across the Internet to Web browsers such as Netscape Navigator, Internet Explorer, or Lynx.

The portion of the URL following the protocol (www.cites.uiuc.edu) designates the host name of the computer you are accessing. The designator www, found in many URLs, is an acronym for World Wide Web. While using www is conventional, it is not necessary. Some Web servers omit the need to type www.

A URL may have any number of directories following it. In this example, 101 is a directory under the root directory of the host you are accessing. The file url101.html is located in the 101 directory.

#### Web Languages

##### HTML (HyperText Markup Language)

It is the universal language understood by all WWW (World Wide Web) clients.

An HTML document (program) is ASCII text with embedded instructions (markups) which affect the way the text is displayed.

The basic model for HTML execution is to fetch a document by its name (e.g. URL), interpret the HTML and display the document, possibly fetching additional HTML documents in the process.

It can accept user input and/or cause additional HTML documents to be fetched by URL.

Providing safety, platform independence, and the ability to interact with a variety of formats, protocols, tools, and languages makes it a universal language.

##### Java and Java Script

Java is a language for sending applets over the web, so that the computer can execute them.

JavaScript is a language that allows HTML to embed small programs called scripts in web pages.

The main purpose of applets and scripts is to speed up web page interactivity.

##### VBScript and ActiveX Controls

VBScript resembles Microsoft's Visual Basic and is used to add scripts to pages that are displayed by Internet explorer.

ActiveX controls, like Java, are used to embed executable programs into a web page.

#### XML

XML stands for EXtensible Markup Language

XML is a markup language much like HTML

XML was designed to carry data, not to display data

XML tags are not predefined. We must define our own tags

XML is designed to be self-descriptive

XML is a W3C Recommendation.

XML is not a replacement for HTML. XML and HTML were designed with different goals. XML was designed to transport and store data, with focus on what data is.

HTML was designed to display data, with focus on how data looks. HTML is about displaying information, while XML is about carrying information.

With XML we invent our Own Tags. The tags in the example below (like <to> and <from>) are not defined in any XML standard.

```
<note>
```

```
<to>Tove</to>
```

```
<from>Jani</from>
```

```
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
<note>
```

These tags are "invented" by the author of the XML document. That is because the XML language has no predefined tags.

The tags used in HTML (and the structure of HTML) are predefined. HTML documents can only use tags defined in the HTML standard (like <p>, <h1>, etc.). XML allows the author to define his own tags and his own document structure.

### 3) Explain the concept of web clients and web browsers.

#### Web Browser

A web browser displays a web document and enables users to access web documents.

When the user clicks a hyperlink, the browser initiates a message to a web server.

This message requests the server to retrieve the requested information and send it back to the web browser through the telecommunications network.

#### Web Server

This is a program that waits patiently for the browser to request a web page.

The server looks for the requested information, retrieves it and sends it to the browser or sends an error message if the file is not found.

#### Uniform Resource Locators (URL) and Transfer protocols

These are the web addresses.

The resource locator is an addressing system that precisely states where a resource is located.

It is divided into four parts: protocol; server/ domain name; path and the resource name. eg: <http://www.cites.uiuc.edu/101/url101.html>

The first portion of the URL (http) designates the protocol that is used to locate the file or resource on the Web.

A protocol is a standardized means of communication among machines across a network. Protocols allow data to be taken apart for faster transmission, transmitted, and then reassembled at the destination in the correct order.

Here, http represents the HyperText Transfer Protocol, which is used to transfer Web pages across the Internet to Web browsers such as Netscape Navigator, Internet Explorer, or Lynx.

The portion of the URL following the protocol (www.cites.uiuc.edu) designates the host name of the computer you are accessing. The designator www, found in many URLs, is an acronym for World Wide Web. While using www is conventional, it is not necessary. Some Web servers omit the need to type www.

A URL may have any number of directories following it. In this example, 101 is a directory under the root directory of the host you are accessing. The file url101.html is located in the 101 directory.

#### Web Languages

##### HTML (HyperText Markup Language)

It is the universal language understood by all WWW (World Wide Web) clients.

An HTML document (program) is ASCII text with embedded instructions (markups) which affect the way the text is displayed.

The basic model for HTML execution is to fetch a document by its name (e.g. URL), interpret the HTML and display the document, possibly fetching additional HTML documents in the process.

It can accept user input and/or cause additional HTML documents to be fetched by URL.

Providing safety, platform independence, and the ability to interact with a variety of formats, protocols, tools, and languages makes it a universal language.

##### Java and Java Script

Java is a language for sending applets over the web, so that the computer can execute them.

JavaScript is a language that allows HTML to embed small programs called scripts in web pages.

The main purpose of applets and scripts is to speed up web page interactivity.

##### VBScript and ActiveX Controls

VBScript resembles Microsoft's Visual Basic and is used to add scripts to pages that are displayed by Internet explorer.

ActiveX controls, like Java, are used to embed executable programs into a web page.

#### XML

XML stands for EXtensible Markup Language

XML is a markup language much like HTML

XML was designed to carry data, not to display data

XML tags are not predefined. We must define our own tags

XML is designed to be self-descriptive



XML is a W3C Recommendation.

XML is not a replacement for HTML. XML and HTML were designed with different goals. XML was designed to transport and store data, with focus on what data is.

HTML was designed to display data, with focus on how data looks. HTML is about displaying information, while XML is about carrying information.

With XML we invent our Own Tags. The tags in the example below (like <to> and <from>) are not defined in any XML standard.

```
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

These tags are "invented" by the author of the XML document. That is because the XML language has no predefined tags.

The tags used in HTML (and the structure of HTML) are predefined. HTML documents can only use tags defined in the HTML standard (like <p>, <h1>, etc.). XML allows the author to define his own tags and his own document structure.

#### 4) Describe the services available in the internet in detail?

Participate in off-line (indirectly, not in real time) discussions via e-mail with people with similar interests through 'mailing lists' and 'News Groups'.

Participate live (in real time) in a conversation with another person via the 'Internet video phone' (like Skype), or audio-video conferencing between multiple people using a specially designed computer programs and equipment.

Participate in on-line (directly, in real time) written discussion with a larger group of people who use the 'Internet Relay Chat' (IRC) service - chat rooms.

To work on a remote computer using the 'Telnet' service or some per function quite the same.

Take files of any type (download) from remote computers and deliver files (upload) them with an FTP (File Transfer Protocol) service.

Reading complex documents using 'hypertext'. Clicking on a keyword or image on the screen the user is automatically goes to other facilities within the same or other domains.

Read multimedia documents found on WWW (World Wide Web) that contain text, graphics, sound, and video using intelligent browser web presentation, as 'Google Chrome', 'Firefox' or 'Internet Explorer' program support.

Learning and practicing for the exam and achieve appropriate certification.

Search the Web, documents, various WWW sites or via dedicated international service, as 'Google' and 'Yahoo!', over subscribed keywords to find the desired documents.

Publicly disclose their images or pictures of his grandchildren, and who wants to can publish a picture of mother in law.

Advertise your business in a variety of ways, from setting up video clips to the creation of their own website.

Paying bills through the 'Internet Banking'.

Buy and spend money :-), advertise and offer for sale.

Read web editions of newspapers or say 'IT Alphabet'.

Play simple games

E-mail

Exchange electronic mail (e-mail) to any Internet user in any location on the planet.

The concept of sending electronic text messages between parties in a way analogous to mailing letters or memos predates the creation of the Internet. Even today it can be important to distinguish between Internet and internal e-mail systems. Internet e-mail may travel and be stored unencrypted on many other networks and machines out of both the sender's and the recipient's control. During this time it is quite possible for the content to be read and even tampered with by third parties, if anyone considers it important enough. Purely internal or intranet mail systems, where the information never leaves the corporate or organization's network, are much more secure, although in any organization there will be IT and other personnel whose job may involve monitoring, and occasionally accessing, the e-mail of other employees not addressed to them.

5) Write short notes on Domain Name System.

Domain Name System

The name of each host computer consists of a series of words separated by dots. The last part of the domain name is called the top-level domain (TLD).

The last two parts of a host computer name constitute the domain.

The second to last part of the name (second-level-domain) is chosen by the organization that owns the computer. Eg: yahoo.com

Because most organizations own more than one computer on the internet, most host computer names have at least one more part, preceding the domain name called third-level-domain. Eg: www.yahoo.com, mail.yahoo.com.

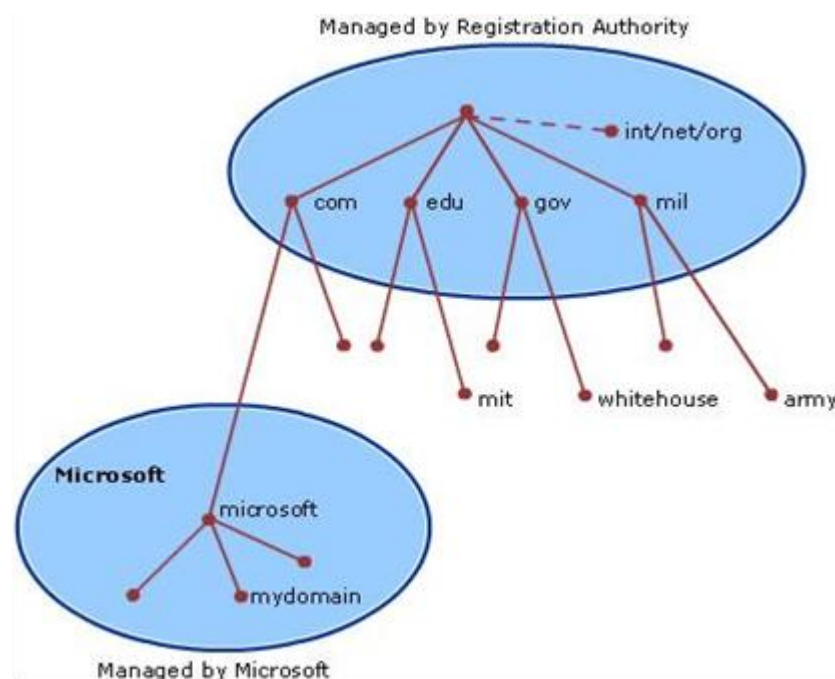
A domain name system (DNS) server translates between the numeric IP addresses that identify each host computer on the internet and the corresponding domain names.

How the Domain name system works

The DNS domain namespace, as shown in the following figure, is based on the concept of a tree of named domains. Each level of the tree can represent either a branch or a leaf of the tree. A branch is a level where more than one name is used to identify a collection of named resources. A leaf represents a single name used once at that level to indicate a specific resource.

DNS Domain Name Hierarchy

Types of DNS Domain Names



Types of DNS Domain Names

Name Type	Description	Example
Root Domain	This is the top of the tree, representing an unnamed level; it is sometimes shown as two empty quotation marks (""), indicating a null value.	A single period (.) or a period used at the end of a name, such as "example.microsoft.com."

Top level domain	A name used to indicate a country/region or the type of organization using a name.	“.com”, which indicates a name registered to a business for commercial use on the Internet.
Second level domain	Variable-length names registered to an individual or organization for use on the Internet.  These names are always based upon an appropriate top-level domain, depending on the type of organization or geographic location where a name is used.	“microsoft.com”, which is the second-level domain name registered to Microsoft by the Internet DNS domain name registrar.
Subdomain	Additional names that an organization can create that are derived from the registered second-level domain name. These include names added to grow the DNS tree of names in an organization and divide it into departments or geographic locations.	“example.microsoft.com”, which is a fictitious subdomain assigned by Microsoft for use in documentation example names.
Host or Resource name	Names that represent a leaf in the DNS tree of names and identify a specific resource. Typically, the leftmost label of a DNS domain name identifies a specific computer on the network. For example, if a name at this level is used in a host (A) RR, it is used to look up the IP address of computer based on its host name.	“host-a.example.microsoft.com”, where the first label (“host-a”) is the DNS host name for a specific computer on the network.

### DNS and Internet Domains

The Internet Domain Name System is managed by a Name Registration Authority on the Internet, responsible for maintaining top-level domains that are assigned by organization and by country/region. These domain names follow the International Standard 3166. Some of the many existing abbreviations, reserved for use by organizations, as well as two-letter and three-letter abbreviations used for countries/regions are shown in the following table:

#### Some DNS Top-level Domain Names (TLDs)

##### DNS Domain Name Type of Organization

Com	Commercial organizations
Edu	Educational institutions
Org	Non-profit organizations
Net	Networks (the backbone of the Internet)
Gov	Non-military government organizations
Mil	Military government organizations
Arpa	Reverse DNS

“xx” Two-letter country code (i.e. us, au, ca, fr)

- 5) Write short notes on Domain Name System.

Refer previous answer.

- 6) i) Discuss any four HTTP request methods.

HTTP defines eight methods (sometimes referred to as "verbs") indicating the desired action to be performed on the identified resource.

S.N.	Method and Description
1	<b>GET</b> The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
2	<b>HEAD</b> Same as GET, but only transfer the status line and header section.
3	<b>POST</b> A POST request is used to send data to the server, for example customer information, file upload etc using HTML forms.
4	<b>PUT</b> Replace all current representations of the target resource with the uploaded content.
5	<b>DELETE</b> Remove all current representations of the target resource given by URI.
6	<b>CONNECT</b> Establish a tunnel to the server identified by a given URI.
7	<b>OPTIONS</b> Describe the communication options for the target resource.
8	<b>TRACE</b> Perform a message loop-back test along the path to the target resource.

- ii) Explain in detail the functions of a web server.

Various functions of web browsers are

- Reformat the URL and send a valid HTTP request.
- When user gives the address of particular web site it is in the form of domain name. the web browser converts the DNS to corresponding IP address.
- The web browser establishes a TCP connection with the web browser while processing the user's request.
- The web browsers send the HTTP request to the web server.
- The web server processes the HTTP request sent by the web browser and returns the desired web page to the client machine. The web browser on the client's machine displays this web page in appropriate format.

- 7) Explain the generation of web pages with merits and demerits.

Dynamic Web Sites

A dynamic web page is a kind of web page that has been prepared with fresh information (content and/or layout), for each individual viewing. It is not static because it changes with the time (ex. anews content), the user (ex. preferences in a login session), the user interaction (ex. web page game), the context (parametric customization), or any combination of the foregoing.

Two types of dynamic web sites

Client-side scripting and content creation

Using client-side scripting to change interface behaviors within a specific web page, in response to mouse or keyboard actions or at specified timing events. In this case the dynamic behavior occurs within the presentation. Such web pages use presentation technology called rich interfaced pages.

Client-side scripting languages like JavaScript or ActionScript, used for Dynamic HTML (DHTML) and Flash technologies respectively, are frequently used to orchestrate media types (sound, animations, changing text, etc.) of the presentation. The scripting also allows use of remote scripting, a technique by which the DHTML page requests additional information from a server, using a hidden Frame, XMLHttpRequests, or a Web service. The Client-side content is generated on the user's computer.

The web browser retrieves a page from the server, then processes the code embedded in the page (often written in JavaScript) and displays the retrieved page's content to the user. The innerHTML property (or write command) can illustrate the client-side dynamic page generation: two distinct pages, A and B, can be regenerated as document.innerHTML = A and document.innerHTML = B; or "on load dynamic" by document.write(A) and document.write(B).

**Server-side scripting and content creation**

Using server-side scripting to change the supplied page source between pages, adjusting the sequence or reload of the web pages or web content supplied to the browser. Server responses may be determined by such conditions as data in a posted HTML form, parameters in the URL, the type of browser being used, the passage of time, or a database or server state. Such web pages are often created with the help of serverside languages such as PHP, Perl, ASP, ASP.NET, JSP, ColdFusion and other languages. These server-side languages typically use the Common Gateway Interface (CGI) to produce dynamic web pages. These kinds of pages can also use, on the client-side, the first kind (DHTML, etc.).

Server-side dynamic content is more complicated:

- (1) The client sends the server the request.
- (2) The server receives the request and processes the server-side script such as [PHP] based on the query string, HTTP POST data, cookies, etc.

The dynamic page generation was made possible by the Common Gateway Interface, stable in 1993. Then Server Side Includes pointed a more direct way to deal with server-side scripts, at the web servers.

**Combining client and server side**

Ajax is a web development technique for dynamically interchanging content with the server-side, without reloading the web page. Google Maps is an example of a web application that uses Ajax techniques and database.

**Application areas of Dynamic Website**

Dynamic web page is required when following necessities arise:

- Need to change main pages more frequently to encourage clients to return to site.
- Long list of products / services offered that are also subject to up gradation
- Introducing sales promotion schemes from time to time
- Need for more sophisticated ordering system with a wide variety of functions
- Tracking and offering personalized services to clients.
- Facility to connect Web site to the existing back-end system

The fundamental difference between a static Website and a dynamic Website is a static website is no more than an information sheet spelling out the products and services while a dynamic website has wider functions like engaging and gradually leading the client to online ordering.

But both static web site design and dynamic websites design can be designed for search engine optimization. If the purpose is only to furnish information, then a static website should suffice.

Dynamic website is absolutely necessary for e-commerce and online ordering

**Advantages:**

Enhances functionality of a website, Simple and Easy to update, New content is responsible for bringing traffic to the site that helps in the search engines optimization, Acts as a system to permit staff or users to collaborate

**Disadvantages:**

Expensive to develop, as web solutions developers use their expertise and knowledge to develop such websites, Hosting costs a little more, again the involvement of experts

## 8) Explain FTP protocol in detail (May 2016)

**FTP:**

The File transfer protocol sets the rules for transferring files between computers. When user wants to download a file from the server FTP is used. There are various issues that must be solved during such file transfer. These issues can be described as follows

Client must have an authority to download particular file.



The hardware and software on both the computers might be different.  
 The data interpretation and data formats on client and server might be different.  
 FTP provides the mechanism to handle these issues internally and user remains free different.  
 FTP uses two connections between client and server. One connection is used for actual data transfer and other is used for control information. This separation of data and commands makes the FTP more Efficient.

When client makes a request for particular file download then using the data transfer connection actual data gets transmitted from server to the client. At the same time server keeps track of how much data is sent so far and how much is remaining. This tracking can be done using the control transfer connection. Hence during the file downloading/uploading we can see a message about how many bytes are getting transferred and how much time is remaining. Various commands used in FTP are

S.No	Command	Meaning of the command
1	USER	It identifies the user
2	PASS	Password
3	QUIT	For Logging off
4	CWD	Change to another directory
6	DELE	Delete the file
7	LIST	Display the contents of the directory
8	RETR	Retrieves files from server to client
9	STOR	Stores file from client to server
10	APPE	Appending the existing file
11	HELP	Asking for the help

9) i) What is SMTP? Explain the working of SMTP in detail. (May 2015/2016)

#### SMTP

SMTP stands for Simple Mail Transfer Protocol. It is simple ASCII protocol. This protocol establishes the TCP connection to port 25, act as client, waits for the receiving machine operating as the server to talk first.

The server starts by sending a line of text giving its identify and telling whether it is prepared to revive mail. If it is not, the client releases the connection and tries again later.

If the server is willing to accept email, the client announces whom the email is coming from and whom the message is returned to.

If such a recipient exist the server gives the client the go ahead to send the message and the server acknowledge it. No checksum are needed because TCP provides a reliable byte stream.

When all the email has been exchanged in both directions the connections is released.

Example dialog for sending the message including the numerical codes used by SMTP.

S: Server

C:Client-

S:220xyz.com SMTP service ready

C:HELO abod.com

S:250 xyz.com says hello to abed.com

C:MAIL FROM:Rosaline@abcd.com

S:250 sender ok

C:RCPT TO:C\_Joseph@xyz.com

S:250 recipient ok

C:DATA

S:354 send mail; end with “,”On a line by itself

C: from:Rosaline@abcd.com

C:To [C\\_Joseph@xyz.com](mailto:C_Joseph@xyz.com)

C:MIME\_version:1.0

C: Messages\_id:0704760941.AA00747@abcd.com

C:Content\_Type:Multiport/alternative

C:Subject:Hi Son

C:Content\_Type:text/enriched

C:Happy birthday to you Joseph

C: Content\_type:audio/basic

C:Content\_transfer\_encoding:base64

C:

S:250 message accepted

C:QUIT

S:221 xyz.com closing connection

The commands used are

- 1.Hello- to Establish Connection.
- 2.RCPT- to send a single message to multiple receivers.
3. MAIL\_FROM- to introduce the sender to the receiver.
4. DATA- The message to be sent in e-mail which will end with.
5. HELP-to know what command the SMTP server will again.

**Advantages:**

Well Defined.

Simple to use

**Disadvantages:**

Messages should not exceed 64KB length.

Infinite mail can be triggered.

If the client and server have different timeout, one of them may give up while the other is till busy unexpectedly terminating the connection.

ii) Explain the architecture of WWW in detail. (May 2016)

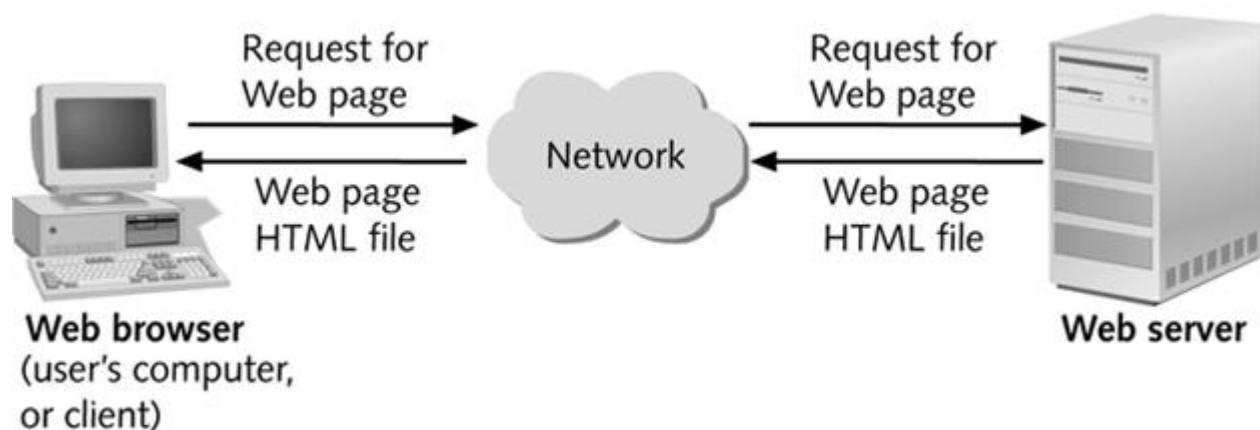
**Basic Web Architecture**

Client Architectures, Server Architectures, Trade-offs, AJAX and LAMP

The web is a two-tiered architecture.

A web browser displays information content,  
and a web server that transfers information to the client.

A web browser runs on the client, Internet Explorer, Netscape, A web server runs on the server, IIS, Apache, The client and server communicate using the http protocol.



**Web Browser**

The primary purpose is to bring information resources to the user.

An application for retrieving, presenting, and traversing information resources.

**Web Server**

The term web server or webservice can mean one of two things:

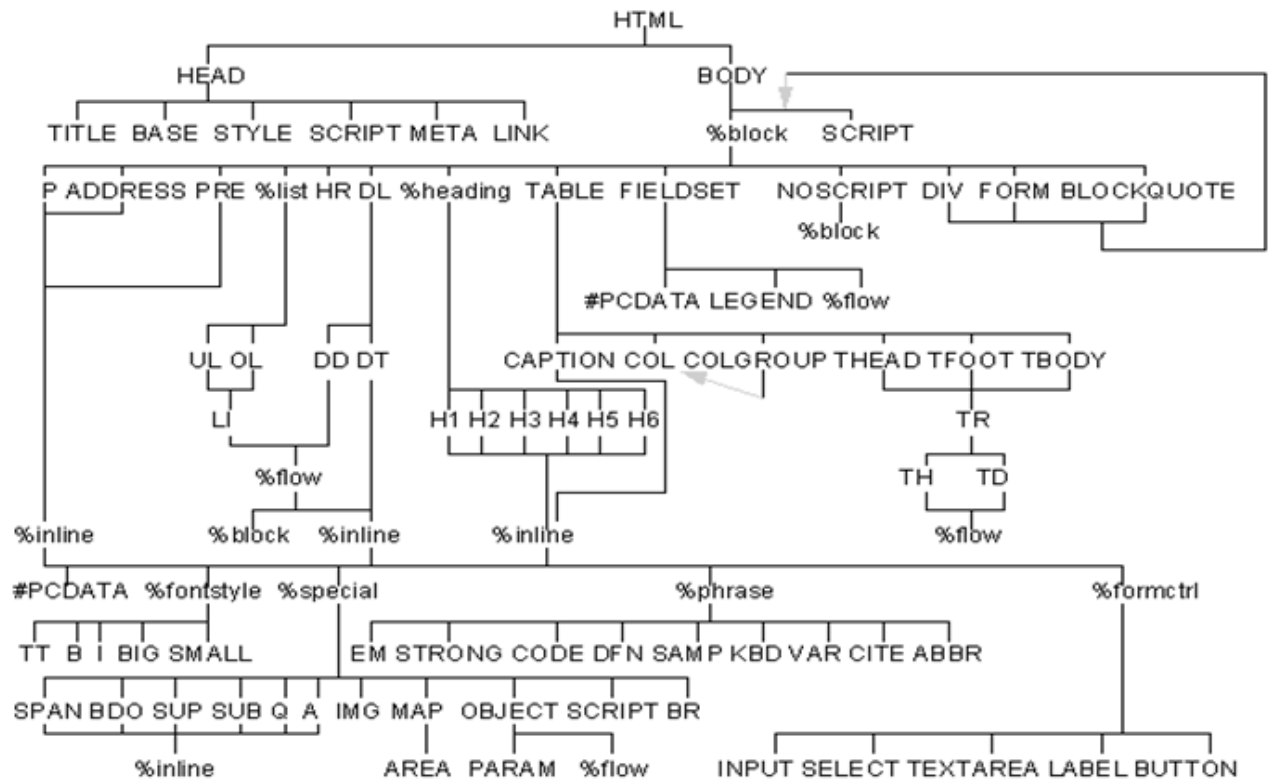
A computer program that accepts HTTP requests and return HTTP responses with optional data content.

A computer that runs a computer program as described above.

**HyperText Markup Language**

Document layout language (not a programming language)

Defines structure and appearance of Web pages

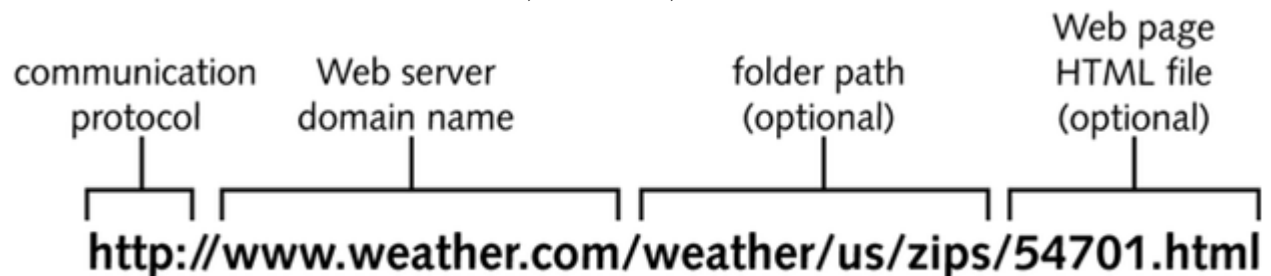


### Universal Resource Identifier

## URLs are location dependent

It contains four distinct parts: the protocol type, the machine name, the directory path and the file name.

There are several kinds of URLs: file URLs, FTP URLs, and HTTP URLs.



10) i) What is HTTP and how it works? **(May 2015)**

## HTTP

The Hyper Text Transfer Protocol (HTTP) is a request/response protocol. It is a communication protocol used to transfer the information on local network and World Wide Web

It is a standard protocol for communication between web browsers and web servers .It is a text oriented protocol.

Steps:

HTTP protocol defines four steps for each request from a client to the server. They are

## Making the Connection

### Making a request.

Response.

.closing the connection

### Making the Connection:

The client establishes a TCP connection to the server. By default, the connection is made on port 80. Other ports may be specified in the URL.

### Making a Request:

The client sends a message to the server requesting a page at the specified URL. There are 8 types of HTTP request. They are

Operation	Description
-----------	-------------

Options	Request information about available options.
GET	Get the document identified in URL
HEAD	Get meta information about document identified in URL
POST	Gives information to the server.
PUT	Stores the document in specified URL
DELETE	Deletes the specified URL.
TRACE	It loopbacks request message
Connect	It is used by proxies

#### The Response

The server sends a response to the client often the response begins with a response code, followed by MIME header information, blank line, then requested information or an error message

#### Closing the Connection:

The connection can be closed either by the client or by the server only both.

#### Features of HTTP Protocol.

1. It is communication protocol used between web browser and web server.
2. This protocol is based on request-response messaging. That means client makes the request of desired web page and then the server responds it by sending the requested resource.
3. It is stateless protocol. That means HTTP protocol cannot remember the previous user's information nor it remembers the number of times the user has visited particular websites.
4. The request-response message consists of plain text in fairly readable form.
5. The HTTP protocol has a cache control.

ii) What is POP3 and What are the advantages and limitations of POP3?

#### POP3:

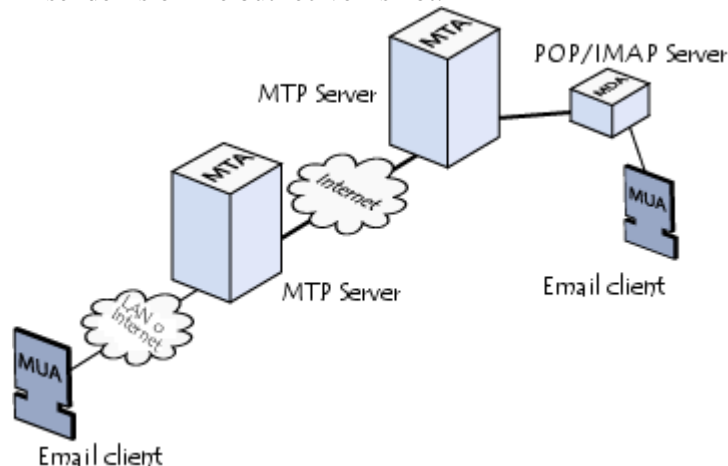
POP3 stands for Post Office Protocol Version3.

POP3 allows user Transfer Agent to contract message transfer agent and allow e-mail to be copied from ISP to the user.

Two situations can be handled by POP3

If both the sender and receiver are online.

If sender is online but receiver is not.



#### Steps:

1. POP3 begins when user starts mail reader.
2. Mail reader calls up the ISP and establishes a TCP Connection with MTA at port 110
3. POP3 goes through 3 states in sequence after connection establishment

#### Authorization:

It deals with having user login.

#### Transaction:

It deals with the user collecting the e-mails and marking them for deletion from the mailbox.

#### Update:

Actually causes email to be deleted.

#### Commands:

USER – Gets the user name

PASS- Gets the password of the user.

LIST- Causes the server to list the contents of mailbox, one message per line, giving the length of the message period (.) terminates the list.

RETR- Used to retrieve the messages

DELE- Used to mark the messages to be deleted.

QUIT- Used to terminate the transaction state and enter update state.

### Advantages and Disadvantages

POP3 has certain distinct advantages over and above the pOP2 systems. POP3 accounts allow users to have mailboxes on a server with their domain name. Post Office Protocol Version 3 provides a simple, standardized way for users to access mailboxes and download messages to their computers. The main advantage of this is that once the mail messages are delivered to the client PC, they can be read with or without connecting to the internet.

POP3 is the most popular protocol and is being supported by almost all email clients. Hence, it has become a standardized way for users to access and download messages to their computers.

The main disadvantage of the POP3 protocol is that it is inadequate for the mobile user. Since messages are downloaded to their mailboxes, the user will be able to access the messages from a PC only.

11) What are the various internet standards used in web? Explain (May 2016)

a web service also include components such as WSDL, UDDI, and SOAP that contribute to make it active. The next step is to learn WSDL, UDDI, and SOAP.

### WSDL

WSDL is an XML-based language for describing web services and how to access them.

WSDL describes a web service, along with the message format and protocol details for the web service.

To learn more about WSDL, visit our [WSDL Tutorial](#).

### UDDI

UDDI is an XML-based standard for describing, publishing, and finding web services.

To learn more about UDDI, visit our [UDDI Tutorial](#).

### SOAP

SOAP is a simple XML-based protocol that allows applications to exchange information over HTTP.

## UNIT II

1) Discuss the various features available in HTML to format the text with example.

Html formatting, Fonts

HTML Text Formatting Tags

Tag	Description
<a href="#"><u>&lt;b&gt;</u></a>	Defines bold text
<a href="#"><u>&lt;em&gt;</u></a>	Defines emphasized text
<a href="#"><u>&lt;small&gt;</u></a>	Defines smaller text
<a href="#"><u>&lt;strong&gt;</u></a>	Defines important text
<a href="#"><u>&lt;sub&gt;</u></a>	Defines subscripted text
<a href="#"><u>&lt;sup&gt;</u></a>	Defines superscripted text

```
<b>
<!DOCTYPE html>
<html>
<body>
<p>This is normal text - <b>and this is bold text</b>.</p>
</body>
</html>
```

This is normal text - and this is bold text

```
<em>
<!DOCTYPE html>
<html>
<body>
<em>Emphasized text</em><br>
<strong>Strong text</strong><br>
<dfn>Definition term</dfn><br>
<code>A piece of computer code</code><br>
<samp>Sample output from a computer program</samp><br>
<kbd>Keyboard input</kbd><br>
```



```
<var>Variable</var>
```

```
</body>
```

```
</html>
```

*Emphasized text*

**Strong text**

*Definition term*

A piece of computer code

Sample output from a computer program

Keyboard input

*Variable*

*Subscript and superscript*

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>This text contains <sub>subscript</sub> text.</p>
```

```
<p>This text contains <sup>superscript</sup> text.</p>
```

```
</body>
```

```
</html>
```

This text contains subscript text.

This text contains superscript text.

Fonts

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p><font size="3" color="red">This is some text!</font></p>
```

```
<p><font size="2" color="blue">This is some text!</font></p>
```

```
<p><font face="verdana" color="green">This is some text!</font></p>
```

```
<p><strong>Note:</strong> The font element was deprecated in HTML 4.01, and is not supported in HTML5.</p>
```

```
</body>
```

```
</html>
```

**This is some text!**

**This is some text!**

**This is some text!**

2) List and explain any four html elements in detail.

i) Html formatting, Fonts

HTML Text Formatting Tags

Tag	Description
<a href="#"><u>&lt;b&gt;</u></a>	Defines bold text
<a href="#"><u>&lt;em&gt;</u></a>	Defines emphasized text
<a href="#"><u>&lt;small&gt;</u></a>	Defines smaller text
<a href="#"><u>&lt;strong&gt;</u></a>	Defines important text
<a href="#"><u>&lt;sub&gt;</u></a>	Defines subscripted text
<a href="#"><u>&lt;sup&gt;</u></a>	Defines superscripted text

```
<b>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>This is normal text - <b>and this is bold text</b>.</p>
```

```
</body>
```

```
</html>
```

This is normal text - and this is bold text

```
<em>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<em>Emphasized text</em><br>
```

```

<strong>Strong text</strong><br>
<dfn>Definition term</dfn><br>
<code>A piece of computer code</code><br>
<samp>Sample output from a computer program</samp><br>
<kbd>Keyboard input</kbd><br>
<var>Variable</var>
</body>
</html>

```

*Emphasized text*

**Strong text**

*Definition term*

A piece of computer code

Sample output from a computer program

Keyboard input

*Variable*

*Subscript and superscript*

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>This text contains <sub>subscript</sub> text.</p>
```

```
<p>This text contains <sup>superscript</sup> text.</p>
```

```
</body>
```

```
</html>
```

This text contains <sub>subscript</sub> text.

This text contains <sup>superscript</sup> text.

Fonts

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p><font size="3" color="red">This is some text!</font></p>
```

```
<p><font size="2" color="blue">This is some text!</font></p>
```

```
<p><font face="verdana" color="green">This is some text!</font></p>
```

```
<p><strong>Note:</strong> The font element was deprecated in HTML 4.01, and is not supported in HTML5.</p>
```

```
</body>
```

```
</html>
```

**This is some text!**

**This is some text!**

**This is some text!**

ii)commenting code, Linking using Anchors,

HTML Comments-> Comments can be inserted into the HTML code to make it more readable and understandable. Comments are ignored by the browser and are not displayed.

Comments are written like this: Example <!-- This is a comment -->

HTML Hyperlinks (Links)

A hyperlink (or link) is a word, group of words, or image that you can click on to jump to a new document or a new section within the current document. When you move the cursor over a link in a Web page, the arrow will turn into a little hand. Links are specified in HTML using the <a> tag. The <a> tag can be used in two ways:

1. To create a link to another document, by using the href attribute
2. To create a bookmark inside a document, by using the name attribute

Syntax of linking <a href="url">Link text</a>

The target Attribute: The target attribute specifies where to open the linked document.

Example

```
<a href="http://www.gurukpo.com/" target="_blank">Visit gurukpo</a>
```

Anchor Tag

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>
```

```
<a href="default.asp">HTML Tutorial</a> This is a link to a page on this website.
```

```
</p>
```

```

<p>
<a href="http://www.w3.org/">W3C</a> This is a link to a website on the World Wide Web.
</p>
</body>
</html>

```

iii) Background color and images

```

<!DOCTYPE html>
<html>
<body bgcolor="#E6E6FA">
<h1>Hello world!</h1>
<p><a href="http://www.w3schools.com">Visit W3Schools.com!</a></p>
<p><b>Note:</b> If using color names, different browsers may render different colors, and Firefox have
problems if the attribute value is in RGB code (will not display correct color).</p>
<p><b>Tip:</b> To produce equal results in all browsers, always use hex code to specify colors.</p>
<p>However, the bgcolor attribute was deprecated in HTML 4, and is not supported in HTML 4.01 Strict
DTD or in XHTML 1.0 Strict DTD. Use CSS instead.</p>
</body>
</html>

```

Hello world!

Note: If using color names, different browsers may render different colors, and Firefox have problems if the attribute value is in RGB code (will not display correct color).

Tip: To produce equal results in all browsers, always use hex code to specify colors.

However, the bgcolor attribute was deprecated in HTML 4, and is not supported in HTML 4.01 Strict DTD or in XHTML 1.0 Strict DTD. Use CSS instead.

## IMAGES

### HTML Images - The <img> Tag and the Src Attribute

In HTML, images are defined with the <img> tag. The <img> tag is empty, which means that it contains attributes only, and has no closing tag. To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display. Syntax for defining an image:  The URL points to the location where the image is stored.

#### The Alt Attribute:

The required alt attribute specifies an alternate text for an image, if the image cannot be displayed. The value of the alt attribute is an author-defined text:  The alt attribute provides alternative information for an image if a user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

### HTML Images - Set Height and Width of an Image

The height and width attributes are used to specify the height and width of an image. The attribute values are specified in pixels by default: 

#### iv) List

**HTML Unordered Lists** An unordered list starts with the <ul> tag. Each list item starts with the <li> tag. The list items are marked with bullets (typically small black circles).

<ul> <li>Coffee</li> <li>Milk</li> </ul> How the HTML code above looks in a browser:

Coffee

Milk

### HTML Ordered Lists

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag. The list items are marked with numbers. <ol> <li>Coffee</li> <li>Milk</li> </ol> How the HTML code above looks in a browser:

1. Coffee

2. Milk

### HTML Definition Lists

A definition list is a list of items, with a description of each item.

The <dl> tag defines a definition list.

The <dl> tag is used in conjunction with <dt> (defines the item in the list) and

<dd> (describes the item in the list):

<dl> <dt>Coffee</dt> <dd>- black hot drink</dd> <dt>Milk</dt> <dd>- white cold drink</dd> </dl>

How the HTML code above looks in a browser: Coffee - black hot drink Milk - white cold drink

## 3) Explain commenting code, Linking using Anchors in detail

**HTML Comments->** Comments can be inserted into the HTML code to make it more readable and

understandable. Comments are ignored by the browser and are not displayed.

Comments are written like this: Example `<!-- This is a comment -->`

### HTML Hyperlinks (Links)

A hyperlink (or link) is a word, group of words, or image that you can click on to jump to a new document or a new section within the current document. When you move the cursor over a link in a Web page, the arrow will turn into a little hand. Links are specified in HTML using the `<a>` tag. The `<a>` tag can be used in two ways:

1. To create a link to another document, by using the href attribute
2. To create a bookmark inside a document, by using the name attribute

Syntax of linking `<a href="url">Link text</a>`

The target Attribute: The target attribute specifies where to open the linked document.

Example

```
<a href="http://www.gurukpo.com/" target="_blank">Visit gurukpo</a>
```

Anchor Tag

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>
```

```
<a href="default.asp">HTML Tutorial</a> This is a link to a page on this website.
```

```
</p>
```

```
<p>
```

```
<a href="http://www.w3.org/">W3C</a> This is a link to a website on the World Wide Web.
```

```
</p>
```

```
</body>
```

```
</html>
```

4) Explain how tables can be inserted into HTML document with example.

### HTML Tables

Tables are defined with the `<table>` tag. A table is divided into rows (with the `<tr>` tag), and each row is divided into data cells (with the `<td>` tag). `td` stands for "table data," and holds the content of a data cell. A `<td>` tag can contain text, links, images, lists, forms, other tables, etc.

Table Example

```
<table border="1">
```

```
<tr> <td>row 1, cell 1</td>
```

```
<td>row 1, cell 2</td>
```

```
</tr>
```

```
<tr> <td>row 2, cell 1</td>
```

```
<td>row 2, cell 2</td>
```

```
</tr>
```

```
</table>
```

Row1 cell1	Row1 cell2
Row 2 cell2	Row2 cell 2

HTML Tables and the Border Attribute

If you do not specify a border attribute, the table will be displayed without borders. Sometimes this can be useful, but most of the time, we want the borders to show. To display a table with borders, specify the border attribute:

```
<table border="1">
```

```
<tr> <td>Row 1, cell 1</td>
```

```
<td>Row 1, cell 2</td>
```

```
</tr>
```

```
</table>
```

### HTML Table Headers

Header information in a table are defined with the `<th>` tag. All major browsers will display the text in the `<th>` element as bold and centered.

```
<table border="1">
```

```
<tr> <th>Header 1</th>
```

```
<th>Header 2</th>
```

```
</tr>
```

```
<tr>
```

```
<td>row 1, cell 1</td>
```

```

<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>

```

Header 1	Header 2
Row1 cell1	Row 1 cell2
Row 2 cell 1	Row2 cell2

5) What is the significance of using forms on the web page? Enlist various components used on form. **(May 2015)**

Form is a typical layout on the web page by which a user can interact with can interact with the web page. Typical component of forms are

HTML Forms

HTML forms are used to pass data to a server.

An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

The <form> tag is used to create an HTML form:

```
<form>
```

```
.
```

```
input elements
```

```
.
```

```
</form>
```

HTML Forms - The Input Element

The most important form element is the <input> element.

The <input> element is used to select user information.

An <input> element can vary in many ways, depending on the type attribute. An <input> element can be of type text field, checkbox, password, radio button, submit button, and more.

The most common input types are described below.

---

Text Fields

<input type="text"> defines a one-line input field that a user can enter text into:

```
<form>
```

```
First name: <input type="text" name="firstname"><br>
```

```
Last name: <input type="text" name="lastname">
```

```
</form>
```

How the HTML code above looks in a browser:

Top of Form

First name:

Last name:

Bottom of Form

Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

Password Field

<input type="password"> defines a password field:

```
<form>
```

```
Password: <input type="password" name="pwd">
```

```
</form>
```

How the HTML code above looks in a browser:

Top of Form

Password:

Bottom of Form

Note: The characters in a password field are masked (shown as asterisks or circles).

Radio Buttons

<input type="radio"> defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices:

```
<form>
```



```
<input type="radio" name="sex" value="male">Male<br>
<input type="radio" name="sex" value="female">Female
</form>
```

How the HTML code above looks in a browser:

Top of Form

☐ Male  
☐ Female

Bottom of Form

Checkboxes

`<input type="checkbox">` defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices.

```
<form>
<input type="checkbox" name="vehicle" value="Bike">I have a bike<br>
<input type="checkbox" name="vehicle" value="Car">I have a car
</form>
```

How the HTML code above looks in a browser:

Top of Form

☐ I have a bike  
☐ I have a car

Bottom of Form

Submit Button

`<input type="submit">` defines a submit button.

A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute. The file defined in the action attribute usually does something with the received input:

```
<form name="input" action="html_form_action.asp" method="get">
```

```
Username: <input type="text" name="user">
```

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

```
</form>
```

How the HTML code above looks in a browser:

Top of Form

Username:

Bottom of Form

If you type some characters in the text field above, and click the "Submit" button, the browser will send your input to a page called "html\_form\_action.asp". The page will show you the received input.

6) Explain in detail about Cascading Style Sheets with suitable examples. **May 2014/May 2016)**

CSS are powerful mechanism for adding styles (e.g. Fonts, Colors, Spacing) to web documents.

They enforce standards and uniformity throughout a web site and provide numerous attributes to create dynamic effects.

The advantage of a style sheet includes the ability to make global changes to all documents from a single location. Style sheets are said to cascade when they combine to specify the appearance of a page.

To create an inline style

Add the style attribute to the HTML tag.

The style declaration must be enclosed within double quotation marks.

To create an embedded style

Insert a `<style>` tag within the head section of HTML file.

Within the `<style>` tag, enclose the style declarations need to the entire Web page.

The style sheet language identifies the type of style used in the document.

The default and the most common language is "text/css" for use with CSS.

To create an External styles

Create a text file containing style declarations

Create a link to that file in each page of the Web site using a `<link>` tag.

Specify the link attributes, such as href, rel, and type.

Link a style sheet, the value of the href attribute should be the "URL" of the linked document, the value of the rel attribute should be "stylesheet" and the value of the type attribute should be "text/css".

EXTERNAL.CSS:

```
body{ background-color: gray;}
```

```
p { color: blue; }
h3{ color: white; }
```

EXTERNAL.HTML:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="EXTERNAL.css" /><!--Link tag for External CSS-->
</head>
<body>
<h3> A White Header </h3>
<p> This paragraph has a blue font.
The background color of this page is gray because
we changed it with CSS! </p>
</body>
</html>
```

INTERNAL.HTML:

```
<html>
<head>
<style> <!--Style tag for Internal CSS-->
body { background-color: blue; }
p { color: white; }
</style>
</head>
<body>
<h2>Internal CSS</h2>
<p>This page uses internal CSS. Using the style tag we are able to modify
the appearance of HTML elements.</p>
</body>
</html>
```

INLINE.HTML:

```
<html>
<head>
</head>
<body>
<h2>InLINE CSS</h2>
<p style="color:sienna;margin-left:20px"><!--Style Attribute(INLINE)-->
This page uses INLINE CSS. Using the style ATTRIBUTE we are able to modify
the appearance of HTML elements.
</p>
</body>
</html>
```

7) i) Explain the features of cascading style sheet.

CSS stands for Cascading style sheets, which is a type of web language, called style sheet language which standardizes the layout throughout a website. Therefore used for describing the look and formatting of a document, from document presentation, including elements such as the layout, colours, and fonts. It's commonly used to style web pages, written in HTML and XHTML.

Using CSS mainly enables the separation of document content, focusing on the appearance and formatting. The separation can improve content accessibility, provide more flexibility and can control the presentation characteristics, to suit specific requirements. CSS also functions the layout of a website, it can reduce the size, clutter and load time of web pages. Additionally it enables multiple pages to share formatting and also reduces complexity and repetition in the structural content. Also, without CSS, a webpage can become teeming with tables nested inside other tables, which causes the speed of a website to be slow, therefore using CSS can also improve the speed of a webpage [1]. CSS additionally prevents clutter, it organizes the web code, it makes it simple and straightforward, this also creates fast loading. As well, CSS permits web pages to be displayed and understood, even if the CSS is not working, or removed. It is also supported well by most modern web browsers.

ii) Explain any eight CSS text properties.

### Color

You can set the color of text with the following:

color: value;

Possible values are

color name – example:(red, black...)

hexadecimal number – example:(#ff0000, #000000)

RGB color code – example:(rgb(255, 0, 0), rgb(0, 0, 0))

### Letter Spacing

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

letter-spacing: value;

Possible values are

normal

length

Example:

T h e s e l e t t e r s a r e s p a c e d a t 5 p x .

### Text Align

You can align text with the following:

text-align: value;

Possible values are

left

right

center

justify

Examples:

This text is aligned left.

This text is aligned in the center.

This text is aligned right.

This text is justified.

### Text Decoration

You can decorate text with the following:

text-decoration: value;

Possible values are

none

underline

overline

line through

blink

Examples:

This text is underlined.

This text is overlined.

~~This text has a line through it.~~

This text is blinking (not in internet explorer).

### Text Indent

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

text-indent: value;

Possible values are

length

percentage

Examples:

This text is indented 10px pixels.

### Text Transform

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

text-transform: value;

Possible values are

none

capitalize

lowercase

uppercase

Examples:

This First Letter In Each Word Is Capitalized, Though It Is Not In My File.

THIS TEXT IS ALL UPPERCASE, THOUGH IT IS ALL LOWERCASE IN MY FILE.

this text is all lowercase. though it is all uppercase in my file.

### White Space

You can control the whitespace in an (X)HTML element with the following:

```
white-space: value;
```

Possible values are

normal

pre

nowrap

### Word Spacing

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

```
word-spacing: value;
```

Possible values are

normal

length

Example:

These words are spaced at 5px.

### 8. Explain the terms margins and padding in detail.(May 2016)

Margins means the space between the content and its neighboring content. There can be top margin, bottom margin, left margin and right margin. The values of these properties can be given in px and in.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p
```

```
{
```

```
background-color:yellow;
```

```
}
```

```
p.margin
```

```
{
```

```
margin-top:100px;
```

```
margin-bottom:100px;
```

```
margin-right:50px;
```

```
margin-left:50px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is a paragraph with no specified margins.</p>
```

```
<p class="margin">This is a paragraph with specified margins.</p>
```

```
</body>
```

```
</html>
```

This is a paragraph with no specified margins.

This is a paragraph with specified margins.

Padding means the space between the contents and its border. Various properties of padding are padding-left, padding-right, padding-top and padding-bottom. These values can also be given in ps or in in.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```

p
{
background-color:yellow;
}
p.padding
{
padding-top:25px;
padding-bottom:25px;
padding-right:50px;
padding-left:50px;
}
</style>
</head>

<body>
<p>This is a paragraph with no specified padding.</p>
<p class="padding">This is a paragraph with specified paddings.</p>
</body>

</html>

```

This is a paragraph with no specified padding.

This is a paragraph with specified padding's.

9) Explain is CSS positioning? Explain it with illustrative examples.

### **Positioning**

The CSS positioning properties allow you to position an element. It can also place an element behind another, and specify what should happen when an element's content is too big.

Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.

There are four different positioning methods.

#### **Static Positioning**

HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page.

Static positioned elements are not affected by the top, bottom, left, and right properties.

#### **Fixed Positioning**

An element with fixed position is positioned relative to the browser window.

It will not move even if the window is scrolled:

#### **Example**

```

p.pos_fixed
{
position:fixed;
top:30px;
right:5px;
}

```

Note: IE7 and IE8 support the fixed value only if a !DOCTYPE is specified.

Fixed positioned elements are removed from the normal flow. The document and other elements behave like the fixed positioned element does not exist.

Fixed positioned elements can overlap other elements.

#### **Relative Positioning**

A relative positioned element is positioned relative to its normal position.

#### **Example**

```

h2.pos_left
{
position:relative;
left:-20px;
}
h2.pos_right
{

```



```
position:relative;
left:20px;
}
```

The content of relatively positioned elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow.

#### Example

```
h2.pos_top
{
position:relative;
top:-50px;
}
```

Relatively positioned elements are often used as container blocks for absolutely positioned elements.

#### Absolute Positioning

An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>:

#### Example

```
h2
{
position:absolute;
left:100px;
top:150px;
}
```

Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist.

Absolutely positioned elements can overlap other elements.

#### Overlapping Elements

When elements are positioned outside the normal flow, they can overlap other elements.

The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

An element can have a positive or negative stack order:

#### Example

```
img
{
position:absolute;
left:0px;
top:0px;
z-index:-1;
}
```

An element with greater stack order is always in front of an element with a lower stack order.

**10) i) Explain how to use CSS to control the style and layout of multiple web pages.(May 2015)**

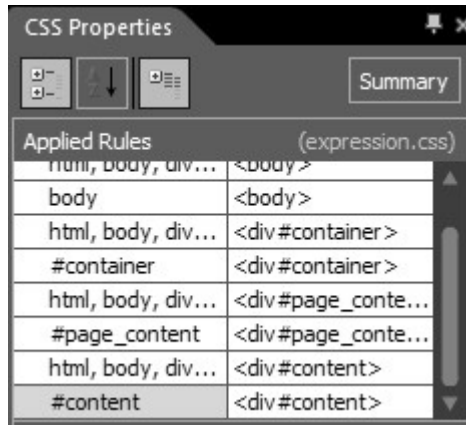
#### CSS Properties Task Panel

The 'CSS Properties' panel shows you the styles that are used by the current selection in your web page and also shows you the styles' order of precedence. In addition, this panel provides you with a comprehensive list of all CSS properties, which enables you to add properties to an existing style, modify properties you've already set, or create new inline styles.

In 'Design view' or 'Code view', in your web page, select the content you want to add an 'inline style' to. In the 'CSS Properties' panel, under 'Applied Rules', either right-click the 'selector' you want to apply the 'inline style' to or if "(no rules applied)" appears then right-click anywhere under 'Applied Rules', and in the shortcut menu, click 'New Inline Style'. Under 'CSS Properties', set the 'properties' you want.

Open your .css file

Click the link in brackets above the 'Applied Rules' pane to quickly open your .css file. (See Fig 3 below)



In the 'CSS Properties' Task Panel which can be opened under the 'Panels' Menu you can start typing the selector - body - and the IntelliSense you set up in the configuration section above now comes into play. The 'IntelliSense' dialog box will appear and allow you to select body, if you then insert { the end tag will also be inserted } and IntelliSense will suggest properties and then values for you to insert.

Once you have just one style inserted CSS Properties will be populated and you can select and make any of the styles using the CSS Properties Panel.

### Modify a Style

When you want to Modify the properties of a style you can do so from a variety of locations.

CSS Properties Task Pane - Enables you to edit only styles that the current web page uses. (This means styles you can select in Code view or any style in an open .CSS file.

Apply Styles Task Pane - Enables you to edit any style contained in the Style Sheets of the Current web page. (Also styles that web page does not use)

Manage Styles Task Pane - Enables you to edit any style similar to Apply Styles Task Pane

Modify Style dialog box enables you to design a new or existing style and preview the style's appearance as you design it.

Please visit Modify the properties of a style to find out how to do so. You might also want to Rename a Style.

### Apply a Style

The 'Apply Styles' panel enables you to create, modify, apply, remove, or delete styles, and attach or remove an external CSS. The panel identifies style types, and shows you if the style is used in the current web page and by the current selection.

When you want to Apply a class or id selector from your style sheet you can use one of the following:

Apply Styles Panel

Manage Styles Panel

Style Toolbar

The difference between them is that 'Apply Styles' Panel allows multiple styles to a selection whilst the 'Manage Styles' Panel allow only one, with the 'Style Toolbar' it enables you to name and apply new undefined styles within your page in a timely manner.

To apply a style

In Design view or Code view, in your web page, select the item you want to apply a style to.

Do one of the following:

To apply multiple existing styles, in the 'Apply Styles' panel, press **CTRL** while you click each style that you want.

To apply a single existing style in the 'Apply Styles' panel, click the **class or ID style** you want.

To apply a **single existing style** in the 'Style toolbar', in either the **Class or ID drop-down box**, click the style you want.

To apply a single existing style in the 'Manage Styles' panel, right-click the class or ID style you want, and in the shortcut menu, click 'Apply Style'.

To create and apply a new style without defining any properties, in the **Style toolbar**, in either the Class or ID box, type a unique name for the new style.

Delete a style from a cascading style sheet

When you no longer need a particular style, you can delete the style from your cascading style sheet (CSS) by using the Apply Styles panel or Manage Styles panel, depending upon the type of style you want to delete.

**Attach or detach an external cascading style sheet**

**Caution!** If you select '**All Pages**' in the '**Link Style Sheet**' dialog box, Expression Web will erase all style sheets that are on your pages and replace them with the new style sheet choices that you have selected. This can be a quick way to apply new styles to your entire Web site but can cause problems if you didn't intend to remove existing style sheets.

When you want to attach the external CSS to multiple pages use the 'Attach Style Sheet' dialog box instead. When you no longer want a web page to use the styles of an attached external CSS, you can detach the CSS from that page in one step by using the Apply Styles or Manage Styles panel. When you want to detach an external CSS from multiple web pages or a whole site, you can use the Manage Style Sheet Links dialog box instead.

Drag the .css file from either the 'Folder List' panel or the 'Site View' onto the page.

To attach an external CSS to one or more web pages or to a .css file

To select the files that you want to attach the CSS to, do one of the following:

To attach the CSS to a **single web page**, open the web page.

To attach the CSS to **all web pages in a site**, open the site.

To attach the CSS to **multiple pages**, in the '**Folder List**' panel or '**Site View**', select the web pages that you want to use the CSS.

To attach the CSS to all the pages in a folder, in the '**Folder List**' panel or the '**Folders view**' of the '**Site View**' window, select the folder.

ii) What is frame? What are the benefits of frame? Explain creation of different kinds of frames in detail. (May 2016)

HTML Frames with frames, we can display more than one HTML document in the same browser window. Each HTML document is called a frame, and each frame is independent of the others.

The disadvantages of using frames are:

Frames are not expected to be supported in future versions of HTML

Frames are difficult to use. (Printing the entire page is difficult). The web developer must keep track of more HTML documents

The HTML frameset Element

The frameset element holds one or more frame elements. Each frame element can hold a separate document.

The frameset element states HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.

The HTML frame Element

The <frame> tag defines one particular window (frame) within a frameset. In the example below we have a frameset with two columns.

The first column is set to 25% of the width of the browser window. The second column is set to 75% of the width of the browser window. The document "frame\_a.htm" is put into the first column, and the document "frame\_b.htm" is put into the second column:

```
<frameset cols="25%,75%"> <frame src="frame_a.htm" /> <frame src="frame_b.htm" />
```

```
</frameset>
```

Note: The frameset column size can also be set in pixels (cols="200,500"), and one of the columns can be set to use the remaining space, with an asterisk (cols="25%,\*").

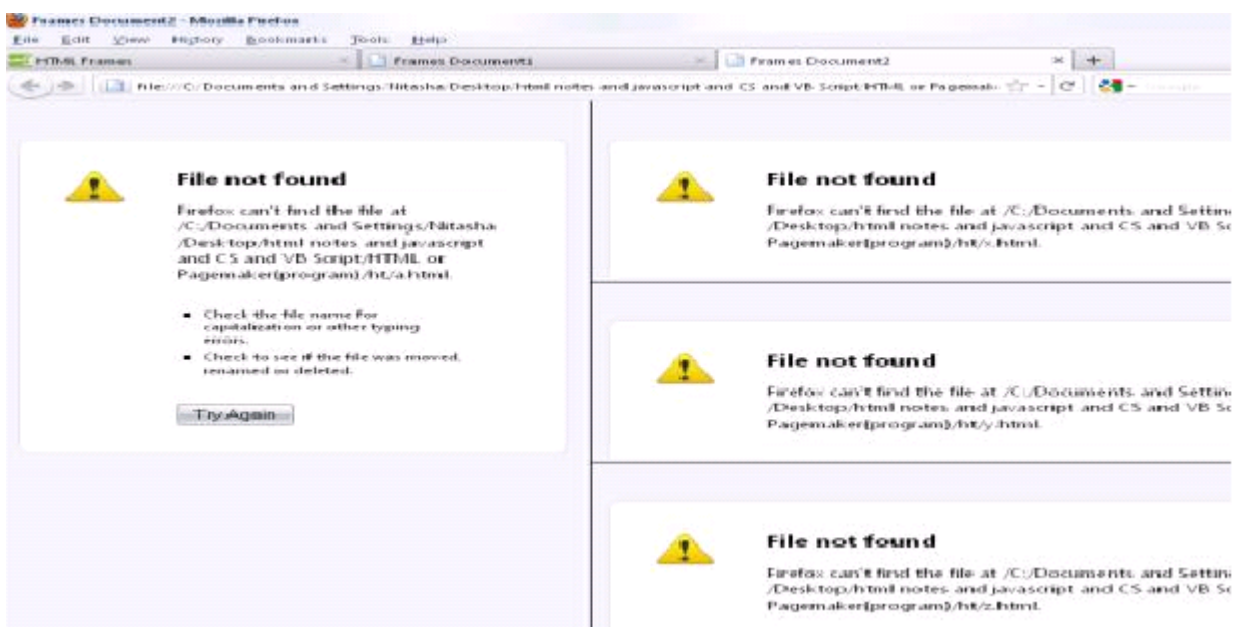
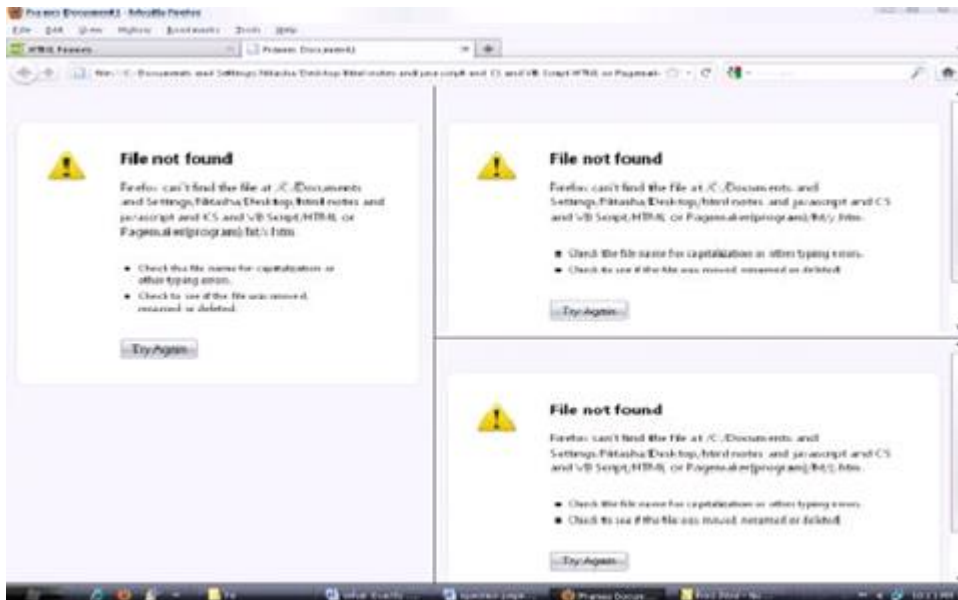
Basic Notes - Useful Tips

Tip: If a frame has visible borders, the user can resize it by dragging the border. To prevent a user from doing this, you can add noresize="noresize" to the <frame> tag.

Note: Add the <noframes> tag for browsers that do not support frames.

Important: You cannot use the <body></body> tags together with the <frameset></frameset> tags! However, if you add a <noframes> tag containing some text for browsers that do not support frames, you will have to enclose the text in <body></body> tags! See how it is done in the first example below.

Create a Frame



<html>  
<head>

<title>Frames Document1</title> </head>

<frameset cols="40%,50%"> <frame src="x.htm"> <frameset rows="25%,25%"> <frame src="y.htm">  
<frame src="z.htm"> </frameset>

</html>

11) Describe the tags in HTML for embedding multimedia content in a web page.

Attributes

Like all other HTML elements, this element supports the global attributes.

autoplay

A Boolean attribute; if specified, the video automatically begins to play back as soon as it can do so without stopping to finish loading the data.

autobuffer

A Boolean attribute; if specified, the video will automatically begin buffering even if it's not set to play automatically. Use this attribute only when it is very likely that the user will watch the video. The video is buffered until the media cache is full.

buffered

An attribute you can read to determine the time ranges of the buffered media. This attribute contains a TimeRanges object.

controls

If this attribute is present, the browser will offer controls to allow the user to control video playback, including volume, seeking, and pause/resume playback.

crossorigin

This enumerated attribute indicates whether to use CORS to fetch the related image. CORS-enabled resources can be reused in the <canvas> element without being *tainted*. The allowed values are:

anonymous

Sends a cross-origin request without a credential. In other words, it sends the Origin: HTTP header without a cookie, X.509 certificate, or performing HTTP Basic authentication. If the server does not give credentials to the origin site (by not setting the Access-Control-Allow-Origin: HTTP header), the image will be *tainted*, and its usage restricted.

use-credentials

Sends a cross-origin request with a credential. In other words, it sends the Origin: HTTP header with a cookie, a certificate, or performing HTTP Basic authentication. If the server does not give credentials to the origin site (through Access-Control-Allow-Credentials: HTTP header), the image will be *tainted* and its usage restricted.

When not present, the resource is fetched without a CORS request (i.e. without sending the Origin: HTTP header), preventing its non-tainted use in <canvas> elements. If invalid, it is handled as if the enumerated keyword **anonymous** was used. See CORS settings attributes for additional information.

height

The height of the video's display area, in CSS pixels.

loop

A Boolean attribute; if specified, we will, upon reaching the end of the video, automatically seek back to the start.

muted

A Boolean attribute which indicates the default setting of the audio contained in the video. If set, the audio will be initially silenced. Its default value is false, meaning that the audio will be played when the video is played.

played

A TimeRanges object indicating all the ranges of the video that have been played.

preload

This enumerated attribute is intended to provide a hint to the browser about what the author thinks will lead to the best user experience. It may have one of the following values:

none: indicates that the video should not be preloaded.

metadata: indicates that only video metadata (e.g. length) is fetched.

auto: indicates that the whole video file could be downloaded, even if the user is not expected to use it. the *empty string*: synonym of the auto value.

If not set, its default value is browser-defined (i.e. each browser may have its default value). The spec advises it to be set to metadata.

#### Usage notes:

The autoplay attribute has precedence over preload. If autoplay is specified, the browser would obviously need to start downloading the video for playback.

The specification does not force the browser to follow the value of this attribute; it is a mere hint.

poster

A URL indicating a poster frame to show until the user plays or seeks. If this attribute isn't specified, nothing is displayed until the first frame is available; then the first frame is shown as the poster frame.

src

The URL of the video to embed. This is optional; you may instead use the <source> element within the video block to specify the video to embed.

width

The width of the video's display area, in CSS pixels.

Events

The <video> element can fire many different events.

Examples

```
<!-- Simple video example -->
```

```
<video src="videofile.ogg" autoplay poster="posterimage.jpg">
```

```
  Sorry, your browser doesn't support embedded videos,
```

```
  but don't worry, you can <a href="videofile.ogg">download it</a>
```

```
  and watch it with your favorite video player!
```

```
</video>
```

```
<!-- Video with subtitles -->
```

```
<video src="foo.ogg">
```

```
  <track kind="subtitles" src="foo.en.vtt" srclang="en" label="English">
```



```
<track kind="subtitles" src="foo.sv.vtt" srclang="sv" label="Svenska">
</video>
```

## 12) Compare: HTML, XML and XHTML (May 2016)

What are HTML, XML and XHTML?

The original language of the World Wide Web is HTML (HyperText Markup Language), often referred to by its current version, HTML 4.01 or just HTML4 for short. HTML was originally an application of SGML (Standard Generalized Markup Language), a sort of meta-language for making markup languages. SGML is quite complicated, and in practice most browsers do not actually follow all of its oddities. HTML as actually used on the web is best described as a custom language influenced by SGML.

Another important thing to note about HTML is that all HTML user agents (this is a catchall term for programs that read HTML, including web browsers, search engine web crawlers, and so forth) have extremely lenient error handling. Many technically illegal constructs, like misnested tags or bad attribute names, are allowed or safely ignored. This error-handling is relatively consistent between browsers. But there are lots of differences in edge cases, because this error handling behavior is not documented or part of any standard. This is why it is a good idea to validate your documents.

XML and XHTML are quite different. XML (eXtensible Markup Language) grew out of a desire to be able to use more than just the fixed vocabulary of HTML on the web. It is a meta-markup language, like SGML, but one that simplifies many aspects to make it easier to make a generic parser. XHTML (eXtensible HyperText Markup Language) is a reformulation of HTML in XML syntax. While very similar in many respects, it has a few key differences.

First, XML always needs close tags, and has a special syntax for tags that don't need a close tag. In HTML, some tags, such as `img` are always assumed to be empty and close themselves. Others, like `p` may close implicitly based on other content. And others, like `div` always need to have a close tag. In XML (including XHTML), any tag can be made self-closing by putting a slash before the code angle bracket, for example ``. In HTML that would just be ``

Second, XML has draconian error-handling rules. In contrast to the leniency of HTML parsers, XML parsers are required to fail catastrophically if they encounter even the simplest syntax error in an XML document. This gives you better odds of generating valid XML, but it also makes it very easy for a trivial error to completely break your document.

### HTML-compatible XHTML

When XML and XHTML were first standardized, no browser supported them natively. To enable at least partial use of XHTML, the W3C came up with something called "HTML-compatible XHTML". This is a set of guidelines for making valid XHTML documents that can still more or less be processed as HTML. The basic idea is to use self-closing syntax for tags where HTML doesn't want a close tag, like `img`, `br` or `link`, with an extra space before the slash. So our ever-popular image example would look like this: ``. The details are described in the Appendix C of the XHTML 1.0 standard.

It's important to note that this is kind of a hack, and depends on the de facto error handling behavior of HTML parsers. They don't really understand the XML self-closing syntax, but writing things this way makes them treat `/` as an attribute, and then discard it because it's not a legal attribute name. And if you tried to do something like `<div />`, they wouldn't understand that the `div` is supposed to be empty.

There are also many other subtle differences between HTML and XHTML that aren't covered by this simple syntax hack. In XHTML, tag names are case sensitive, scripts behave in subtly different ways, and missing implicit elements like `<tbody>` aren't generated automatically by the parser.

So if you take an XHTML document written in this style and process it as HTML, you aren't really getting XHTML at all – and trying to treat it as XHTML later may result in all sorts of breakage.

**UNIT III**

- 1) Explain the core features of javascript in detail.

**JavaScript Core Features****Basic Features****JavaScript is case sensitive**

- Whitespace

– Whitespace is generally ignored in JavaScript statements and between JavaScript statements but not always consider

- $x = x + 1$  same as  $x=x + 1$

$s = \text{typeof } x$ ; is same as  $s=\text{typeof } x$  but it not the same as  $s=\text{typeof } x$ ; or  $s= \text{type of } x$ ;

**Statements**

– A script is made up of individual statements

– JavaScript statements are terminated by returns or semi-colons (;)

– So  $x = x+1$ ; same as  $x = x+1$

$\text{alert}(x)$ ;

$\text{alert}(x)$ ;

**Blocks**

To group together statements we can create a block using curly braces.

Blocks are used with functions as well as larger decision structures like if statements.

**Function add(x,y)**

{

var result = x + y;

Return result;

}

If(x>10)

{

X=0;

Y=10;

}

**Variables**

Variables store data in a program

The name of a variable should be unique well formed identifier starting with a letter and followed by letters or digits

Variable names should not contain special characters or white space

Variable names should be well considered

– X versus sum

Define a variable using the var statement

– var x;

If undefined a variable will be defined on its first use

Variables can be assigned at declaration time

– var x = 5;

Commas can be used to define many variables at once

– `var x, y = 5, z;`

## Basic Data Types

Every variable has a data type that indicates what kind of data the variable holds

Basic data types in JavaScript

– Strings (“thomas”, ‘x’, “Who are you?”)

Strings may include special escaped characters

– ‘This isn’t hard’

Strings may contain some formatting characters

– “Here are some newlines \n\n\n and tabs \t\t\t yes!”

– Numbers (5, -345, 56.7, -456.45677)

Numbers in JavaScript tend not to be complex (e.g. higher math)

– Booleans (true, false)

Also consider the values null and undefined as types

Weak Typing

JavaScript is a weakly type language meaning that the contents of a variable can change from one type to another.

– Some languages are more strongly type in that you must declare the type of a variable and stick with it. Example of dynamic & weak typing a variable initially holding a string can later hold a number

`= "hello"; x = 5; x = false;`

While weak typing seems beneficial to a programmer it can lead to problems

## Type Conversion

- Consider the following example of weak typing in action

`document.write(4*3);`

`document.write("<br>"); document.write("5" + 5); document.write("<br>"); document.write("5" - 3);`

`document.write("<br>"); document.write(5 * "5");`

You may run into significant problems with type conversion between numbers and strings use functions like `parseFloat()` to deal with these problems

– Prompt demo

Composite Types

JavaScript supports more advanced types made up of a collection of basic types.

## Arrays

– An ordered set of values grouped together with a single identifier

Defining arrays

– `var myArray = [1,5,1968,3];`

– `var myArray2 = ["Thomas", true, 3, - 47];`

– `var myArray3 = new Array();`

```
– var myArray4 = new Array(10)
```

Arrays

Access arrays by index value

```
– var myArray = new Array(4)
```

```
– myArray[3] = "Hello";
```

Arrays in JavaScript are 0 based given

```
– var myArray2 = ["Thomas", true, 3, -47];
```

```
– myArray2[0] is "Thomas", myArray[1] is true and so on
```

– Given new Array(4) you have an array with an index running from 0 – 3

– To access an array length you can use *arrayName.length*

```
alert(myArray2.length);
```

Objects

Underneath everything in JavaScript are objects.

An object is a collection of data types as well as functions in one package

The various data types called properties and functions called methods are accessed using a dot notation.

*objectname.propertyname*

We have actually been using these ideas already, for example `document.write( hello )` says using the document object invoke the write() method and give it the string "hello" this results in output to the string

## Working with Objects

There are many types of objects in JavaScript

– Built-in objects (primarily type related)

– Browser objects (navigator, window, etc.)

– Document objects (forms, images, etc.)

– User defined objects

Given the need to use objects so often shortcuts are employed such as the with statement

```
with (document)
```

```
{
```

```
write("This is easier"); write("This is even easier");
```

```
}
```

We also see the use of the short cut identifier `this` when objects reference themselves

## Expressions and Operators

Make expressions using operators in JavaScript

Basic Arithmetic

– + (addition), - (subtraction/unary negation), /

(division), \* (multiplication), % (modulus)

**Increment decrement**

++ (add one) -- (subtract one)

**Comparison**

– &gt;, &lt;, &gt;=, &lt;=, != (inequality), == (equality), === (type equality)

**Logical**

– &amp;&amp; (and) || (or) ! (not)

**More Operators****Bitwise operators (&, |, ^)**

– Not commonly used in JavaScript except maybe cookies?

– Shift operators (&gt;&gt; right shift, &lt;&lt; left shift)

**String Operator**

– + serves both as addition and string concatenation

– document.write("JavaScript" + " is " + " great! ");

– You should get familiar with this use of +

**Be aware of operator precedence**

– Use parenthesis liberally to force evaluations

– var x = 4 + 5 \* 8 versus x = (4+5) \* 8

**Flow Control**

- Basic program execution control handled in

JavaScript using the if statement

• if ( <i>expression</i> )	or	if ( <i>expression</i> )
----------------------------	----	--------------------------

<i>true-case</i>		<i>true-case;</i>
------------------	--	-------------------

Else
<i>false-case;</i>

if (x &gt; 10)

alert("x bigger than 10"); else

alert("x smaller than 10");

**More on If Statements**

You can use { } with if statements to execute program blocks rather than single statements

if (x &gt; 10)

{

alert("X is bigger than 10");

```
alert("Yes it really is bigger");
```

```
}
```

- Be careful with ;s and if statements

```
if (x > 10);
```

```
alert("I am always run!?!");
```

### Switch Statements

If statements can get messy so you might consider using a switch statement instead

```
switch (condition)
```

```
{
```

```
case (value) : statement(s) break;
```

```
...
```

```
default: statement(s);
```

```
}
```

The switch statement is not supported by very old JavaScript aware browsers (pre-JavaScript 1.2), but today this is not such an important issue

### Switch Example

```
var x=3; switch (x)
```

```
{
```

```
case 1: alert('x is 1'); break;
```

```
case 2: alert('x is 2'); break;
```

```
case 3: alert('x is 3'); break;
```

```
case 4: alert('x is 4'); break;
```

```
default: alert('x is not 1, 2, 3 or 4');
```

```
}
```

### Loops

JavaScript supports three types of loops: while, do/while, and for

Syntax of while:

```
while(condition) statement(s)
```

Example:

```
var x=0;
```

```
while (x < 10)
```

```
{
```

```
document.write(x);
```

```
document.write("<br />"); x = x + 1;
```

```
}
```

```
document.write("Done");
```

### Do Loop



- The difference between loops is often when the loop condition check is made, for example

```
var x=0; do
```

```
{
```

```
document.write(x); x = x + 1;
```

```
} while (x < 10);
```

In the case of do loops the loop always executes at least once since the check happens at the end of the loop

### For Loop

The most compact loop format is the for loop which initializes, checks, and increments/decrements all in a single statement

```
for (x=0; x < 10; x++)
```

```
{
```

```
document.write(x);
```

```
}
```

With all loops we need to exercise some care to avoid infinite loops. See example

### For/In Loop

One special form of the for loop is useful with looking at the properties of an object. This is the for/in loop.

```
for (var aProp in window)
```

```
{
```

```
document.write(aProp) document.write("<br />");
```

```
}
```

We will find this construct useful later on when looking at what we can do with a particular object we are using

## Loop Control

We can control the execution of loops with two statements: break and continue

break jumps out of a loop (one level of braces)

continue returns to the loop increment

```
var x=0;
```

```
while (x < 10)
```

```
{
```

```
x = x + 1; if (x == 3) continue;
```

```
document.write("x = "+x); if (x == 5)
```

```
break;
```

```
}
```

## Functions

Functions are useful to segment code and create a set of statements that will be used over and over again  
The basic syntax is

```
function name(parameter list)  
{  
  function statement(s) return;  
}
```

For example

```
function add(x, y)  
{  
  var sum = x + y; return sum;  
}
```

We can then invoke a function using the function name with ( )'s

```
var result = add(2, 3);
```

- We can also pass variable values as well as literals

```
var a = 3, b=5; var result;  
result = add(a,b);
```

Variables are passed to function by value so you must use return to send things back.

You can return a value or not from a function and you can have as many return statements as you like

## Input/Output in JavaScript

### Special dialog forms

#### – Alert

```
alert("Hey there JavaScript coder! ");
```

#### – Confirm

```
if (confirm('Do you like cheese?')) alert("Cheese lover");  
else
```

```
  alert("Cheese hater");
```

#### – Prompts

```
var theirname = prompt("What's your name? ", "");
```

### Writing to the HTML document

#### – document.write()

#### – document.writeln()

Writing should be done before or as the document loads.

In traditional JavaScript the document is static after that,

though with the DOM everything is rewritable

Since we are writing to an (X)HTML document you may write out tags and you will have to consider the white space handling rules of (X)HTML

## Comments and Formatting

When writing JavaScript commenting is useful

Two methods – C and C++ style

– `/* This is a multiple line style comment */`

– `// This is a single line comment`

2). Explain the primitive data types used in javascript? (May 2015)

### Basic Data Types

Every variable has a data type that indicates what kind of data the variable holds

Basic data types in JavaScript

– Strings (“thomas”, ‘x’, “Who are you?”)

Strings may include special escaped characters

– ‘This isn’t hard’

Strings may contain some formatting characters

– “Here are some newlines \n\n\n and tabs \t\t\t yes!”

– Numbers (5, -345, 56.7, -456.45677)

Numbers in JavaScript tend not to be complex (e.g. higher math)

– Booleans (true, false)

Also consider the values null and undefined as types

### Weak Typing

JavaScript is a weakly type language meaning that the contents of a variable can change from one type to another.

– Some languages are more strongly type in that you must declare the type of a variable and stick with it. Example of dynamic & weak typing a variable initially holding a string can later hold a number

= `"hello"; x = 5; x = false;`

While weak typing seems beneficial to a programmer it can lead to problems

### Type Conversion

- Consider the following example of weak typing in action  
`document.write(4*3);`

```
document.write("<br>"); document.write("5" + 5); document.write("<br>"); document.write("5" - 3);
document.write("<br>"); document.write(5 * "5");
```

You may run into significant problems with type conversion between numbers and strings use functions like `parseFloat()` to deal with these problems

– Prompt demo

Dealing with Type

You can also use the `typeof` operator to figure out type

```
var x = "5"; alert (typeof x);
```

- Be aware that using operators like equality or even `+` may not produce expected results  
`x=5;`

Produces a rather interesting result. We see the inclusion of a type equality operator (`===`) to deal with this Composite Types

JavaScript supports more advanced types made up of a collection of basic types.

Arrays

- An ordered set of values grouped together with a single identifier
- 2) Explain the operators with examples.

### Operators

An operator generally performs some kind of calculation (operation) or comparison with two values (the value on each side of an operator is called operand) to reach a third value.

Type of operator: Arithmetic Operator, Comparison Operator, Assignment Operators, Bitwise Operator  
Increment and Decrement Operator, Logical Operator

JavaScript was designed to add interactivity to HTML pages

JavaScript is a scripting language - a scripting language is a lightweight programming language

A JavaScript is lines of executable computer code

A JavaScript is usually embedded directly in HTML pages

JavaScript is an interpreted language (means that scripts execute without preliminary compilation)

Everyone can use JavaScript without purchasing a license

JavaScript is supported by all major browsers, like Netscape and Internet Explorer

Arithmetic operators

Operators are also tokens. JavaScript operators shown in following table include arithmetic operators for addition, subtraction, multiplication, division, and the modulus operator for returning the remainder from division. These are all binary operators, which means that they require two operands, one to the left of the operator and one to the right. The addition and subtraction operators can also function as unary operators with a single operand to the right of the operator, eg `-x`.

Operator	Symbol	Example
Addition	+	3+4
Subtraction	-	z-10
Multiplication	*	A*b
Division	/	a/3
Modulus	%	17%3 =2

Basic operators

Operator	Description	Example	Result
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	X=4

## Assignment Operators

The JavaScript assignment operator is the symbol `=`. Thus the JavaScript statement `x = a+b`

Comparison Operators in Javascript.

Decision-making has its own operators, which allow anyone to test conditions. There are many comparison operators of which most useful are summarized in the following table

Operator	Example	Is The Same As
<code>=</code>	<code>x=y</code>	<code>x=y</code>
<code>+=</code>	<code>x+=y</code>	<code>x=x+y</code>
<code>-=</code>	<code>x-=y</code>	<code>x=x-y</code>
<code>*=</code>	<code>x*=y</code>	<code>x=x*y</code>
<code>/=</code>	<code>x/=y</code>	<code>x=x/y</code>
<code>%=</code>	<code>x%=y</code>	<code>x=x%y</code>

## Logical operators

Javascript allows us to use multiple conditions; to do the logical operators AND, OR, and NOT are in use. The symbols for these are listed below

Operator	Description	Example
<code>&amp;&amp;</code>	and	<code>x=6</code> <code>y=3</code> <code>(x &lt; 10 &amp;&amp; y &gt; 1)</code> returns true
<code>  </code>	or	<code>x=6</code> <code>y=3</code> <code>(x==5    y==5)</code> returns false
<code>!</code>	not	<code>x=6</code> <code>y=3</code> <code>!(x==y)</code> returns true

## Increment operator and decrement operator

These operators are represented by `++` and `--` respectively. They increase or decrease value by 1.

## String Operator in Javascript

A string is most often text, for example "Hello World!". To stick two or more string variables together, use the `+` operator.

```
txt1="What a very"
```

```
txt2="nice day!"
```

```
txt3=txt1+txt2
```

The variable `txt3` now contains "What a verynice day!".

To add a space between two string variables, insert a space into the expression, OR in one of the strings.

```
txt1="What a very"
```

```
txt2="nice day!"
```

```
txt3=txt1+" "+txt2
```

or

```
txt1="What a very "
```

```
txt2="nice day!"
```

```
txt3=txt1+txt2
```

The variable `txt3` now contains "What a very nice day!".

4) Explain how functions can be written in Java script with an example.

A function contains some code that will be executed by an event or a call to that function. A function is a set of statements. You can reuse functions within the same script, or in other documents. You define functions at the beginning of a file (in the head section), and call them later in the document. It is now time to take a lesson about the alert-box:

This is JavaScript's method to alert the user.

```
Alert("This is a message")
```

## Define a Function

To create a function you define its name, any values ("arguments"), and some statements:

```
function myfunction(argument1,argument2,etc)
{
```

*some statements*

}

A function with no arguments must include the parentheses:

```
function myfunction()
```

```
{
```

*some statements*

```
}
```

Arguments are variables used in the function. The variable values are values passed on by the function call. By placing functions in the head section of the document, you make sure that all the code in the function has been loaded before the function is called.

Some functions return a value to the calling expression

```
function result(a,b)
```

```
{
```

```
c=a+b
```

```
return c
```

```
}
```

Call a Function

A function is not executed before it is called.

You can call a function containing arguments:

```
myfunction(argument1,argument2,etc)
```

or without arguments:

```
myfunction()
```

The return Statement

Functions that will return a result must use the "return" statement. This statement specifies the value which will be returned to where the function was called from. Say you have a function that returns the sum of two numbers:

```
function total(a,b)
```

```
{
```

```
result=a+b
```

```
return result
```

```
}
```

When you call this function you must send two arguments with it:

```
sum=total(2,3)
```

The returned value from the function (5) will be stored in the variable called sum.

## 5) Discuss javascript Array object in detail

### Arrays

#### Declaring and Allocating Arrays

- Arrays occupy space in memory. An array in JavaScript is an Array object. The programmer uses operator new to allocate dynamically the number of elements required by each array.
- The process of creating new objects is also known as creating an instance, or instantiating an object, and operator new is known as the dynamic memory allocation operator. Array objects are allocated with new because arrays are considered to be objects, and all objects must be created with new. To allocate 12 elements for integer array c, use the statement

```
var c = new Array( 12 );
```

#### Passing Arrays to Functions

- To pass an array argument to a function, specify the name of the array (a reference to the array) without brackets. For example, if array hourlyTemperatures has been declared as

```
var hourlyTemperatures = new Array( 24 );
```

then the function call

```
modifyArray( hourlyTemperatures );
```

passes array hourlyTemperatures to function modifyArray.

- In JavaScript, every array object —knows its own size (via the length attribute). Thus, when we pass an array object into a function, we do not pass the size of the array separately as an argument.
- Although entire arrays are passed by using call-by-reference, individual numeric and boolean array elements are passed by call-by-value exactly as simple numeric and Boolean variables are passed (the



objects referred to by individual elements of an Array of objects are still passed by call-by-reference). Such simple single pieces of data are called scalars, or scalar quantities. To pass an array element to a function, use the subscripted name of the element as an argument in the function call.

### Multiple-Subscripted Arrays

- Multiple-subscripted arrays with two subscripts often are used to represent tables of values consisting of information arranged in rows and columns.
- To identify a particular table element, we must specify the two subscripts; by convention, the first identifies the element's row, and the second identifies the element's column. Arrays that require two subscripts to identify a particular element are called double-subscripted arrays (also called two-dimensional arrays).
- Multiple-subscripted arrays can have more than two subscripts. JavaScript does not support multiple-subscripted arrays directly, but does allow the programmer to specify single-subscripted arrays whose elements are also single-subscripted arrays, thus achieving the same effect.
- Every element in array *a* is identified by an element name of the form *a*[*i*][*j*]; *a* is the name of the array, and *i* and *j* are the subscripts that uniquely identify the row and column, respectively, of each element in *a*.

The names of the elements in the first row all have a first subscript of 0; the names of the elements in the fourth column all have a second subscript of 3. Multiple-subscripted arrays can be initialized in declarations like a single-subscripted array. Array *b* with two rows and two columns could be declared and initialized with the statement

```
var b = [ [ 1, 2 ], [ 3, 4 ] ];
```

- The values are grouped by row in square brackets. So, 1 and 2 initialize *b*[0][0] and *b*[0][1], and 3 and 4 initialize *b*[1][0] and *b*[1][1]. The interpreter determines the number of rows by counting the number of sub-initializer lists (represented by sets of square brackets) in the main initializer list. The interpreter determines the number of columns in each row by counting the number of initializer values in the sub-initializer list for that row. Multiple-subscripted arrays are maintained as arrays of arrays. The declaration 

```
var b = [ [ 1, 2 ], [ 3, 4, 5 ] ];
```

 creates array *b* with row 0 containing two elements (1 and 2) and row 1 containing three elements (3, 4 and 5).

6) i) Write short notes on document object model. (May 2014/May2016)

### Document object model

Every web page resides a browser window which can be considered as an object.

A document object represents the HTML documents that is displayed in that window. The document object has various properties that refer to other objects which allow access to and modification of document content.

The way that document content is accessed and modified is called the Document Object Model. The Objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a web document.

Window Object: Top of the hierarchy. It is the outmost element of the object hierarchy.

Document Object: Each HTML document that gets loaded into a window becomes a document object. The document contains the content of the page.

Form Object: Everything enclosed in the <form>... </form> tags sets the form object.

Form Control Elements: The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

Here is a simple hierarchy of few important objects

Window			
Frame	Parent	Self	top

### Levels of Document Object Model

The Document Object Model has been under development by W3C and it is developed in various levels. These levels are

Level	Description
DOM 0	THIS MODEL IS SUPPORTED BY THE EARLY BROWSERS. THIS LEVEL COULD SUPPORT Javascript. This version was implemented in Netscape 3.0 and Internet Explorer 3.0 browsers.
DOM 1	This version was issued in 1998 which was focused on XHTML and XML
DOM 2	This version was issued in 2000 that could specify the style sheet. It also supports the event model and traversal within the documents.
DOM 3	This is the current release of DOM specification published in 2004. This version could deal with XML

with DTD and schema, document validations, document views and formatting.
---

Basically DOM is an Application Programming Interface(API) that defines the interface between XHTML document and application program. That means, suppose application program is written in java and this java program wants to access the elements of XHTML web document then it is possible by using a set of Application Programming Interfaces(API) which belongs to the DOM.

ii) Explain dom event handling in detail.

Event is an activity that represents a change in environment. For example mouse clicks, pressing a particular key of keyboard represent the events. Such events are called intrinsic events.

Event handler is a script that gets executed in response to these events. Thus event handler enables the web document to respond the user activities through the browser window.

### Ways of Event Handling

A number of ways exist to connect your code to an event handlers; we're going to look at two of the more useful ways viz.

1)Event Handlers as attributes.

2)Event Handlers as Properties.

Event Handlers as attributes

The first most common method is to add the event handlers name and the code we want to execute to the HTML tag's attributes.

For example

```
<html>
<head>
<script language = "Javascript">
<!--Beginning of java script-->
function MsgBox(textstring){
alert(textstring)}
//--End of Javascript-->
</script>
</head>
<body>
<form>
<input name = "text1" type = Text>
<input name = "submit" type = button value = "show Me"
onclick = "MsgBox(form.text1.value)">
</form>
</body>
</html>
```

Output will be type the text and click the show me button the text will be shown in the dialog box

Event Handlers as Properties

In this way of connecting to events, we first to define the function that will be executed when the event occurs. Then we need to set that objects event handler property to the function we defined. This is illustrated in the following object

For example

```
<script language = "JavaScript"
Function linkSomePage_Onclick()
{
Alert('this is going nowhere');
Return false;
}
</script>
<Ahref = "somepage.htm" name="linkSomepage">
Click Me
</A>
<Script language = "Javascript" type = "text/JavaScript">
Window.document.links[0].Onclick = "linkSomepage_Onclick;
</script>
```

Explanation

We define the function linkSomepage\_Onclick(), much as we did before. As before we can return a value indicating whether we want the normal action of that object to happen. Next we have the </A>tag, whose object's event we are connecting to. You will notice there is no mention of the event handler or the

function within the attributes of the tag.

7) i) What is meant by event propagation?

The concept of *event bubbling* was introduced to deal with situations where a single event, such as a mouse click, may be handled by two or more event handlers defined at different levels of the *Document Object Model* (DOM) hierarchy. If this is the case, the event bubbling process starts by executing the event handler defined for individual elements at the lowest level (e.g. individual hyperlinks, buttons, table cells etc.). From there, the event *bubbles up* to the containing elements (e.g. a table or a form with its own event handler), then up to even higher-level elements (e.g. the BODY element of the page). Finally, the event ends up being handled at the highest level in the DOM hierarchy, the document element itself (provided that your document has its own event handler).

The term *event propagation* is often used as a synonym of *event bubbling*. However, strictly speaking, event propagation is a wider term: it includes not only event bubbling but also event capturing. *Event capturing* is the opposite of bubbling (events are handled at higher levels first, then *sink* down to individual elements at lower levels). Event capturing is supported in fewer browsers and rarely used; notably, Internet Explorer prior to version 9.0 does not support event capturing.

ii) Explain the terms capturing and event bubbling.

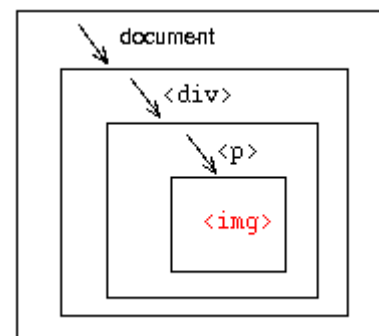
Event Capture

Let's say that your document contains a <div> which contains a <p> which contains an <img>. Further, let's say you've added an event listener to all of them. When a user clicks on the image, a mouseclick event occurs.

Even though the user clicked the image, the image doesn't get the event first. Instead, the event listener attached to the document grabs the event first and processes it. (That is, it captures the event before it gets to its intended target.) The event is then passed down to the <div>'s event listener. The event then goes to the <p>, and finally to the <img>. That is, all of the clicked-on object's "ancestors" higher up in the document capture the event for processing before sending it down the chain to its intended target.

You can [try event capture in a pop-up window](#). (This only works in Mozilla.)

The term *event propagation* is often used as a synonym of *event bubbling*. However, strictly speaking, event propagation is a wider term: it includes not only event bubbling but also event capturing. *Event capturing* is the opposite of bubbling (events are handled at higher levels first, then *sink* down to individual elements at lower levels). Event capturing is supported in fewer browsers and rarely used; notably, Internet Explorer prior to version 9.0 does not support event capturing.



8) Explain any three objects that can be defined in JavaScript with the help of examples. (May 2016)

Javascript objects has collection of related properties and methods that can be used to manipulate a certain kind of data.

They are three types of objects

1) Math object

The math object provides a number of useful mathematical functions and number manipulation methods. There is no need to declare a variable as Math object or define a new math object before being able to use it, making it a little bit easier to use.

Some math objects are as following

1) abs() Method: The abs() method returns the absolute value of the number passes as its parameter.

Essentially, this means that it returns the positive value of the number. So -1 is returned as 1, -4 as 4 and so on. However, 1 would be returned as 1 because it's already positive.

2) ceil() Method: The ceil() method always round a number up to the next largest whole number or integer. So 10.01 becomes 11 and 9.99 becomes -9(because -9 is greater than -10). Ceil() method has just one parameter, namely the number you want rounded up.

3) floor() Method: Like the ceil() method, the floor method removes any numbers after the decimal point, and returns a whole number or integer. The difference is that floor() always rounds the number down, so if we pass 10.01 we would be returned 10, and if we pass 9.99, we will see -10 returned.

4) round Method: The round method is very similar to ceil() and floor(), except that instead of always rounding up or rounding down, it rounds up only if the decimal part is 0.5 or greater, and rounds down otherwise.

5) random() Method: The random method returns a random floating point number in the range between 0 and 1, where 0 is included and 1 is not. This can be very useful for displaying random number banner images for writing a javascript game.

#### Date Object

The date object handles everything to do with date and time in javascript. Using it, we can find out the date time now, store our own dates and time, do calculations with these dates, and converts the dates into strings.

#### Creating Date Object

The Date() objects have the following steps:

1) Declaration and Initialization of Date object: We can declare and initialize a Date object in many ways. In the method of creation we just declare a new Date object on the PC on which the script is run.

```
Var theDate1 = new Date();
```

```
Var theDate3 = new Date("31 January 2000");
```

2) Getting Date Values: it is all very nice having stored a date, but how do we get the information out again? Well, we just use the get methods. These are summarized in the following table

Method	Returns
getDate()	The day of the month
getDay()	The day of the week as an integer, with Sunday as 0, Monday as 1 and so on.
getMonth()	The month as an integer, with January as 0, February as 1 and so on.
getFullYear()	The year as a four-digit number
toString()	Returns the full date based on the current time zone as a human-readable string.

3) Setting Date values: To change part of a date into date object, we have a group of set functions, which pretty much replicate the get functions described earlier, except that we are setting not getting values.

Method	Description
setDate()	The date of the month is passed in as parameter to set the date
setMonth()	The month of year is passed in as an integer parameter, whose 0 is January, 1 as February and so on.
setFullYear()	This sets the year to the four digit integer number passed in as a parameter.

#### Document object model

Every web page resides in a browser window which can be considered as an object.

A document object represents the HTML documents that are displayed in that window. The document object has various properties that refer to other objects which allow access to and modification of document content.

The way that document content is accessed and modified is called the Document Object Model. The Objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a web document.

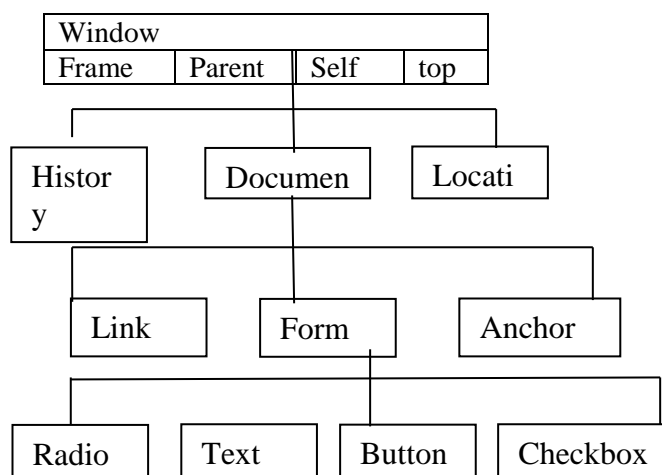
Window Object: Top of the hierarchy. It is the outmost element of the object hierarchy.

Document Object: Each HTML document that gets loaded into a window becomes a document object. The document contains the content of the page.

Form Object: Everything enclosed in the <form>... </form> tags sets the form object.

Form Control Elements: The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

Here is a simple hierarchy of few important objects:



9) i) Write short notes on Inline Frames and Applied Frames.(or) Explain any four JavaScript window methods to control windows. **(May 2015)**

Javascript includes powerful features for working with browser windows.

Creating a new window

One of the most convenient uses for the window object is to create a new window. You can do this to display a document , eg: the instructions for a game- without clearing the current window.

You can create a new browser window with the window.open() method. A typical statement to open a new window looks like this:

```
WinObj = window.open("URL","windowName","Feature List");
```

The following are the components of window.open() statement

1) The WinObj variable is used to store the new window object. You can access methods and properties of the new object by using the name.

2) The first parameter of the window.open() method is a URL, which will be loaded into the new window. If it is left blank, no web page will be loaded.

3) The second parameter specifies a window name (here window name). this is assigned to the window object's name property and is used to refer the window.

4) the third parameter is a list of optional features , separated by commas.

Opening and closing window

You can close windows as well. The window.close() method closes a window.

Frames

A frame is a special type of window that can display multiple independently scrollable frames on a single screen, each with its own distinct URL. Frames can point to different URL's and be targeted by other URL's , all within the same window. A series of frames makes up a page.

The following diagram shows a window containing three frames:

Music Club Artists Jazz -T.Alan -j.jack Soul -b.britto	Alen sing a song
Alphabetical by category Musician Description	

This frame is named navigateFrame

Creating a frame

You can create a frame by using the <FRAMESET> tag in an HTML document.

Example

```
<FRAMESET ROWS = "90%,10%">
```

```
<FRAMESET COLS = "30%,70%">
```

```
<FRAME SRC = category.html NAME = "list frame">
```

```
<FRAME SRC = titles.html NAME = "contentFrame">
```

```
</FRAMESET>
```

```
<FRAME SRC = navigate.html NAME = "navigateFrame">
```

```
</FRAMESET>
```

DOCUMENTS

Document object is an object that is created by the browser for each new HTML page(document) that is viewed. By doing this, javascript gives you access to a number of properties and methods that can affect the document in various ways.

Document object has a number of properties associated with it, which are following:

Property	Description
formName	Not a property itself, but creates a new property with each named form placed in the document
Forms	An array of all form tags in a document
Frames	An array of all frames used in the document.
Height	Returns the height, in pixels, of the body element of the document.
Images	An array of all the images(img) tags in the document.
Layers	An array of all the layer tags in the document.
Links	An array of all the link<a> tags in the document.

Scripts	An array of all the script tags used in the document.
URL	Returns the URL of the current document.

ii) Explain the form handling and form validation. **(May 2015/2016)**

Javascript can be used to validate data in HTML forms before sending off the content to a server. Form data that typically are checked by a javascript could be :

- 1) Has the user left required fields empty?
- 2) Has the user entered a valid e-mail address?
- 3) Has the user entered a valid date?
- 4) Has the user entered text in a numeric field?

Field Validation

The function checks if a field has been left empty. If field is blank , an alert box alerts a message , the function return false , and the form will not be submitted.

```
<html>
<head>
<script>
function validateForm()
{
var x=document.forms["myForm"]["fname"].value
if(x==null||x== "")
{
alert("First name must be filled out ");
return false;
}
}
</script>
</body>
<form name = "myForm" action = "demo_form.asp" onsubmit= "return validateForm()" method = "post">
First name:<input type = text" name = "fname">
<input type = "submit" value = "Submit">
</form>
</body>
</html>
```

10) Designing Quiz Application Personal Information System/ Using JavaScript

```
<form name = "myform">
Which is the largest animal on land?<br>
Aardvark <input type = "radio" name = "rad1" onclick = "getAnswer('1a', '1')">
Whale <input type = "radio" name = "rad1" onclick = "getAnswer('1b', '2')">
Elephant <input type = "radio" name = "rad1" onclick = "getAnswer('1c', '3')">
Rhinosaurus <input type = "radio" name = "rad1" onclick = "getAnswer('1d','4')">
<br><br>
<div id = "para"><div>
</form>
```

```
<script type = "text/javascript">
var reltext = [];
reltext[1] = "The aardvark is a medium-sized, burrowing, nocturnal mammal native to Africa."
reltext[2] = "The whale is the common name for various marine mammals of the order Cetacea."
reltext[3] = "The African elephant is the largest living land mammal. "
reltext[4] = "The rhinoceros is a large, primitive-looking mammal that dates from millions of years ago."
```

```
function getAnswer(ans, rel) {
document.getElementById("para").innerHTML = reltext[rel];
if (ans == "1c") {alert ("That is the right answer!")
}
else {alert ("No - try again")
}
}
```



</script>

Output:



11) Write JavaScript code to check whether the user has entered a valid e-mail address and a valid date.

**VALIDATION.HTML:**

```
<html>
<head>
<script type="text/javascript">
var fl=1;
function check()
{
var n=document.forms[0].elements[0].value;
var re=new RegExp("[0-9]");

xyz.innerHTML="";
if(n.length<1)
{
xyz.innerHTML="invalid user name";
fl=0;
}
if(re.test(n))
{
xyz.innerHTML="User name should not start with a digit";
fl=0;
}
}
function check1()
{
xyz1.innerHTML="";
```

```
var name1=document.getElementById("pwd").value;
if(name1.length<1)
{
    xyz1.innerHTML="Field should be filled";
    fl=0;
}
}
function check2()
{
    xyz2.innerHTML="";
    var name2=document.getElementById("pwd1").value;
    var name1=document.getElementById("pwd").value;
    if(name2.length<1)
    {
        xyz2.innerHTML="Field should be filled";
        fl=0;
    }
    if(name1!=name2)
    {
        xyz2.innerHTML="passwords mismatched";
        fl=0;
    }
}
function email()
{
    p1.innerHTML="";
    var m1=document.getElementById("mail").value;
    var re1=new RegExp("^[0-9]");

    if(m1=="")
    {
        p1.innerHTML="please enter email id";
        fl=0;
    }

    if(re1.test(m1))
    {
        p1.innerHTML="mail id should not start with a digit";
        fl=0;
    }
}
function addr()
{
    p2.innerHTML="";
    var adr=document.getElementById("ad").value;
    if(adr=="")
    {
        p2.innerHTML="field should be filled";
        fl=0;
    }
}
function showit()
{
    if(fl==0)
        alert("Your registration was not successful");
    else
        alert("your registration was successful");
}
</script>
```

69

**OUTPUT:**  
**Login: Karups**

## Email Registration

**User Name** :

**Password** :

**Confirm password:**

**Email Address** :  @  .   please enter email id

**Address** :

**Line1** :

**Line2** :

**Pin** :

invalid user name

Field should be filled

Field should be filled

field should be filled

**Message from webpage** ✖

Your registration was not successful

- 13) Explain the different types of looping statements in JavaScript. (May 2016)  
Refer the first question answer.

### UNIT IV

1) Explain Browser Management in detail. (or) What methods are used for browser Management in JavaScript? Explain (MAY 2015)

Javascript can be used to manage the browser and user experience

All browser offers ways to manage private data, which can range from a record of sites one has viewed to copies of web pages stored for the purposes of faster load times on subsequent visits. Each browser provides a way to peruse this data, as well as various methods to remove it from computer or portable device. These browsers also give the ability to surf the web without saving any of these sensitive data components.

Although each browser may have its own unique name for certain private data components, one will find that terms are interchangeable and that all browsers store the same items. There are some exceptions, that each browser represents items in different ways.

#### Navigator Object

The navigator objects of javascript shows how to use it to perform browser detection, whether the subject is Firefox, Internet Explorer, Opera, etc.

Javascript's navigator objects provides properties that indicate the type of browser and other useful information about the user accessing the page.

The navigator object of javascript contains the following core properties.

Properties	Description
appCodeName	The code name of the browser
appName	The name of the browser
appVersion	Version information for the browser
cookieEnabled	Boolean that indicates whether the browser has cookies enabled.
Language	Returns the default language of the browser version.
online	Determines whether the browser is online
Platform	Returns for which platform the browser is complied.
Product	Returns the engine name of the browser.
userAgent	Returns the user-agent header sent by the

browser to the server.
------------------------

```
<!DOCTYPE html>
<html>
<body>
<div id="example"></div>
<script>
txt="<p>Browser CodeName:"+navigator.appCodeName+"</p>";
txt+="<p>Browser Name:"+navigator.appName+"</p>";
txt+="<p>Browser Version:"+navigator.appVersion+"</p>";
txt+="<p>Cookies Enabled:"+navigator.cookieEnabled+"</p>";
txt+="<p>Browser Language:"+navigator.language+"</p>";
txt+="<p>BrowserOnline:"+navigator.OnLine+"</p>";
txt+="<p>Platform:"+navigator.platform+"</p>";
txt+="<p>User-agent header:"+navigator.userAgent+"</p>";
txt+="<p>User-agentlanguage:"+navigator.systemLanguage+"</p>";
document.getElementById("example").innerHTML= txt;
</script>
</body>
</html>
```



## BROWSER DETECTION

Anyone who has built more than a few web pages has surely come across some of the numerous differences between browser types and versions. A page that looks perfect on one computer screen does not look quite the same on other computer screen. It looks vastly different that is small shift of content or container size.

Hence user simplifies their site technology to the so called lowest common denominator. In order to meet of older browsers, often lacking support for even CSS or javascript and viewing pages in a low resolution environment.

Falling somewhere in between these extremes is the more adaptive type of site that modifies itself to suit the browser's capabilities or indicates to users their inability to use the site. This "sense and adapt" concept is often termed browser detection or browser sniffing. It is integral part of javascript tasks of any complexity.

Browser sniffing is a technique used in website and web applications in order to determine the web browser a visitor is using, and to serve browser-appropriate content to the visitor.

They are different types of detection

- Information detection

- Technology detection

- Javascript detection

- Plug-in Detection

- Visual detection

Information detection

When using javascript to detect the capabilities of web site or application's visitors, one can break up the useful detectable information in four categories.



Technical issues(for example, javascript, color depth and screen size)

Visual issues(for example, color depth and screen size)

Delivery issues(for example, connection speed or type)

User issues(for example, language setting and previous visitor)

### Technology detection

Browser support the following things.

Markup versions and specific elements.

Style sheets and specific properties.

Scripting language version and specific objects , methods and properties.

Java

Object technology(plug-ins and active controls)

### Javascript detection

Javascript support is probably the easiest technology to detect. If a script does not run, this condition implicitly shows that the browser does not support javascript or that it is turned off. It have two types javascript version and object detection.

### Visual detection

The screen object encapsulates information about the display on which the window is being rendered, including the size of the screen and the number of colors that it can display. The following are the commonly used screen properties.

Property	Description
avaailHeight	Returns the height of the screen
availWidth	Returns the width of the screen
colorDepth	Returns the bit depth of the color palette for displaying images
Height	Returns the total height of the screen
pixelDepth	Returns the color resolution
Width	Returns the total width of the screen.

### Browser control

Using the window object it is possible to change window appearance. For example one might scroll or resize the window using window.scrollTo() or window.resizeTo() or the set the browser status message using window.status or window.defaultStatus.

For more control , one might consider opening a new window (window.open) and removing the browser ‘s or even going full screen.

### State Management

Cookies are data, stored in small text files on computer. When a web server has sent a web page to a browser , the connection is shut down and the server forgets everything about the user.

### Working with cookies

Server sends data to the vistor’s browser in the form of cookie. Cookies are a plain text data record of five variable-length fields.

Expires: the date the cookies will expire. If this is blank, the cookie will expire when the vistor quits the browser.

Domain: the domain name of web site.

Path: The path to the directory or web page that set the cookie. This may be blank if anyone wants to retrieve the cookie from any directory or page.

Secure: If this field contains the word “secure” then the cookie may only be retrived with a secure server. If this field is blank, no such restrictions exists.

Name=value: cookies are set and retrieved in the form of key and value points.

## 2)Explain the media management in detail.(May 2015)

### Media management

Web pages are not composed solely of text, but generally include a variety of depedent media items such as images, videos , audioclips, and multimedia elements.

### Image handling

One can use javascript to get at and manipulate images. This may involve creating new images, setting up an array of images, changing the source attribute in the <body> tag or just doing some simple animation with them.

Any images user places between the two <body> tags of HTML pages can be seen by javascript. This is because whenever one adds a picture with the IMG tag it is placed in a collection, which is a sort of array.

The first picture that the browser can see, the one closest to the opening BODY tag, is placed at position 0 in the collection. The second picture will be placed at position 1, the third at position 2 etc. for web page, move the two <script> tags inside the <body> of the html:

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>IMAGE COLLECTION</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE = "JAVASCRIPT">
</SCRIPT>
</BODY>
</HTML>
<BODY>
<IMG SRC = "C:\Documents and Settings\BOSS\Desktop\DSC06396.JPEG">
<P>
<IMG SRC = "C:\Documents and Settings\BOSS\Desktop\DSC06397.JPEG">
<P>
<INPUT TYPE = "BUTTON" VALUE = "GO" ONCLICK = "IMAGE_TEST()">
<P>
<LABEL ID = "LABEL_1">IMAGE NAMES</LABEL>
<SCRIPT LANGUAGE = "JAVASCRIPT">
</SCRIPT>
</BODY>
```

### ANIMATION

One can use javascript to create a complex animation which includes but not limited to:

Fireworks.

Fade effect.

Roll-in or Roll-out.

Page-in or Page-out.

Object movements.

Javascript provides following two functions to be frequently used in animation programs

Set(Timeout(function, duration): this function calls function after duration milliseconds from now.

setInterval(function, duration): this function calls function after every duration milliseconds.

clearTimeout(setTimeout\_variable); this function calls clear any timer set by the setTimeout() functions.

Drawing lines

```
<html>
<head>
<style>
body
{
margin:0px;
padding:0px;
}
</style>
</head>
<body>
<canvas id="myCanvas"width="578" height="200"></canvas>
<script>
```

```

var canvas = document.getElementById("my Canvas");
var context = canvas.getContext('2d');
context.beginPath();
context.moveTo(100,150);
context.lineTo(450,50);
context.lineWidth = 20;
context.strokeStyle = '#0000ff';
context.lineCap = 'round';
context.stroke();
</script>
</html>

```

3)What is constructor? Explain how constructors can be used in javascript.

Constructor are the methods in which the properties of newly created objects get initialized. While initializing the properties of an object, the constructor needs to refer the corresponding object. We can refer to this object using a predefined variable called this.

Every new expression must include a call to a constructor, whose name is the same as the object being created.

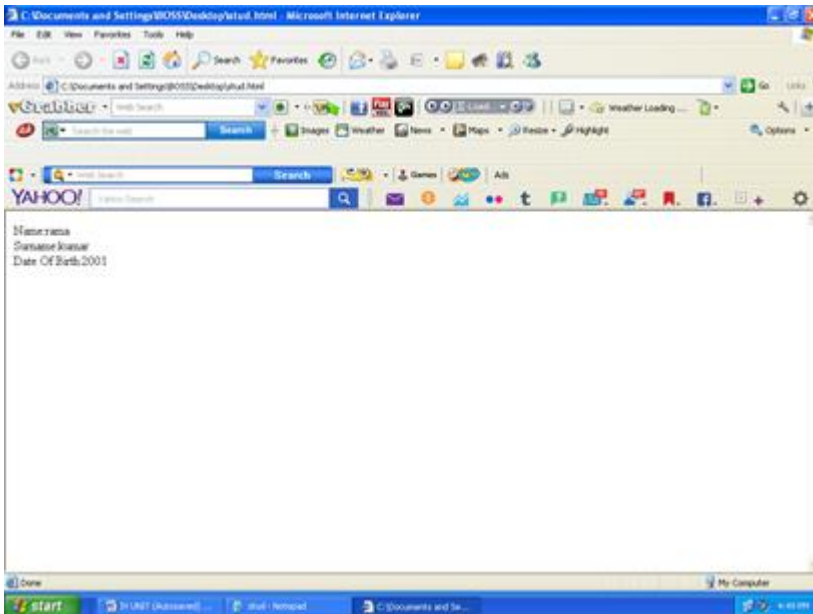
```

<html>
<head>
<script type = "text/javascript">
function Student(Stud_name,Stude_surname,Stud_DOB)
{
this.name=Stud_name;//Initializing the data properties
this.surname=Stude_surname;
this.DOB=Stud_DOB;
this.display=StudInfo;//Initializing the method
}
function StudInfo()
{
document.write("Name:"+this.name+"<br/>");
document.write("Surname:"+this.surname+"<br/>");
document.write("Date Of Birth:"+this.DOB+"<br/>");
}
</script>
</head>
<body>
<script type = "text/javascript">
var Std_III=new Student("rama","kumar",2001);//creating constructor
Std_III.display();//invoking the method of an object
document.write("<br/>");
</script>
</body>
</html>

```

### Script Explanation

In above written script we have created an object Student using new operator. While creating the object we have written a constructor to which the initializing values are passed as parameter. Then using the variable this we can access the data properties of the object Student. Similarly the method display is initialized in this constructor. This method is then invoked in the body section using the dot operator as-



Std\_III.display(). Hence as an output we can see the values that gets assigned to the data properties.

Thus we create a collection of objects using the same constructor then it resembles the concept of class in the object oriented programming. But there is such class concept in javascript , although it supports the concept of object. Hence javascript is called as object based programming.

#### 4) Explain on object constructor and prototyping.

Javascript provides a special predefined constructor function called Object(), used with the new keyword, to build a generic object. The return value of the Object() constructor is assigned to a variable. The variable contains a reference to an instance of a new object. Program below create a user-defined function that will act as a constructor.

Syntax:

```
Var myobj = new Object();
```

For example

```
<html>
<head><title>The object() Constructor</title>
<script type = "text/javascript">
Var cat = new Object();
Alert(cat);
</script>
<body>
</body>
</html>
```

Explanation

The Object() constructor creates and returns a reference to a cat object.

The returned value from the Object() constructor is a reference to an object.[object Object] means that the cat is a descendant of the Object object from which all objects are derived.

#### Prototyping

In a language such as java or c++ , to create a class as an extension of another, one define the class in such a way that it inherits the functionality of the higher-level object and overrides whatever functionality ha/she does not want.

Javascript , on the other hand , provides for a constructor, via object, that allows one to create new objects. It is the object constructor that allocates the memory for the object , including all of its properties.

The Object also provides a prototype property that enables to extend any object, including the built-in ones such as String and Number. One uses this prototype to derive new object methods and properties, rather than class inheritance:

```
Number.prototype.add = function(val1,val2) {return val1+val2;}
```

```
Var num = new Number();
```

```
Num.prototype.add = function(val1,val2)
```

```
Var sum =num.add(8,3);//sum is 11
```

This demonstrates of the dynamic nature of Javascript, and how one can add new methods and properties to objects at runtime. There is a difference between adding a new property or method to an instance of an

object and extending the object through prototype,

The following code is not the same as the preceding code snippet because the add method now applies only to the object instance, and not to all instances of an object:

```
Var num = new Number();
Num.add=function(val1,val2){return val1+val2};
Var sum = num.add(8,3);
```

When anyone extent an object through prototype, the method or property is available to all instances of the object. When he/she extend an object instance with a new property or method, it is available only to the instance.

The concept of extending objects via prototyping is best explained through a demonstration.

The following example demonstrates how to extend the employee object by defining a new method, trim, using the underlying object prototype property. Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<script>
```

```
function employee(name,jobtitle,born)
```

```
{
```

```
  this.name=name;
```

```
  this.jobtitle=jobtitle;
```

```
  this.born=born;
```

```
}
```

```
var fred=new employee("Fred Flintstone","Caveman",1970);
```

```
employee.prototype.salary=null;
```

```
fred.salary=20000;
```

```
document.write(fred.salary);
```

```
</script>
```

```
</body>
```

```
</html>
```

```
20000
```

### 5) Explain JQuery Getters and Setters in detail. (May 2014)

Some of the simplest, and most common, operations on jQuery objects are those that get or set the value of HTML attributes, CSS styles, element content, or element geometry. This section describes those methods.

First, however, it is worth making some generalizations about getter and setter methods in jQuery:

Rather than defining a pair of methods, jQuery uses a single method as both getter and setter. If you pass a new value to the method, it sets that value; if you don't specify a value, it returns the current value.

When used as setters, these methods set values on every element in the jQuery object, and then return the jQuery object to allow method chaining.

When used as getters, these methods query only the first element of the set of elements and return a single value. (Use map() if you want to query all elements.) Since getters do not return the jQuery object they are invoked on, they can only appear at the end of a method chain.

Three simple, but useful, jQuery methods for DOM manipulation are:

text() - Sets or returns the text content of selected elements

html() - Sets or returns the content of selected elements (including HTML markup)

val() - Sets or returns the value of form fields

Set Content - text(), html(), and val()

We will use the same three methods from the previous page to set content:

text() - Sets or returns the text content of selected elements

html() - Sets or returns the content of selected elements (including HTML markup)

val() - Sets or returns the value of form fields

### 6) Explain technologies are being used in AJAX?

AJAX (asynchronous Javascript and XML) is a group of interrelated web development techniques used on the client-side to create interactive web applications.

AJAX is the art of exchanging data with a server, and updating parts of web page-without reloading the whole page.

AJAX is a new technique for creating better, faster and more interactive web applications with the help of XML,HTML,CSS and javascript..

AJAX uses XHTML for content and CSS for presentation, as well as the Document Object Model and Javascript for dynamic content display.

With AJAX when submit is pressed, javascript will make a request to the server, interpret the results and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

AJAX is a web browser technology independent of web server software. A user can continue to use the application while the client program requests information from the server in the background.

AJAX is very simple technology based on the following open standards:

i)javascript.

ii)XML.

iii)HTML.

iv)CSS

working of AJAX

AJAX involves full-page updates. If request is made by the client(browser), the server processes the request and sends the output(usually HTML).

The client renders the HTML page. This way of posting a page will do well with processing a non-interactive web Form, but what about instances where is expected to be interactive. A key requirement of such a type of transaction is to have the ability to fill up a partial page, consisting of a field or a set of fields in a Form, rather than the entire page.

#### CREATING XMLHTTPREQUEST OBJECT

```
<!DOCTYPE html>
<html>
<head>
<script>
function loadXMLDoc()
{
var xmlhttp;
if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
    }
else
    { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
xmlhttp.onreadystatechange=function()
    {
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
        document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
        }
    }
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
}
</script>
</head>
<body>

<div id="myDiv"><h2>Let AJAX change this text</h2></div>
<button type="button" onclick="loadXMLDoc()">Change Content</button>

</body>
</html>
```





AJAX is not a new programming language.  
 AJAX is a technique for creating fast and dynamic web pages.  
 Change Content

### Advantages of AJAX

**Better Interactivity:** AJAX allows easier and quicker interaction between user and website as pages are not reloaded for

content to be display.

**2)Easier Navigation:** AJAX application on web-sites can be built to allow easier navigation to users in comparison to using the traditional back and forward button on a web browser.

**Compact:** with AJAX, several multi-purpose applications and features can be handled using a single web page, avoiding the need for clutter with several web pages.

**Backed by Reputed Brands:** another assuring reason to use AJAX on website is the fact that several complex web applications are handles using AJAX , google Maps is the most impressive and obvious example, other powerful , popular scripts such as the vBulltien forum software has also incorporated AJAX into their latest version.

### Disadvantages of AJAX

**Back and refresh Button are Rendered Useless:** with AJAX as , all functions are loaded on a dynamic page without the page beging reloaded or more importantly a URL being changed, clicking the back or refresh button would take you to an entirely different web page or to the beginning of what your dynamic web page was processing. this is the main drawback behind AJAX but fortunately with good programming skills this issue can be fixed.

**Security constraints:** due to security constraints, one can only use to access information from the host that server the initial page. If anyone needs to display information from server, it is not possible within the AJAX.

7)Explain the concept of JSON concept with example..  
 JAVASCRIPT OBJECT NOTATION (JSON)

### INTRODUCTION

JSON is a syntax for storing and exchanging text information. JSON is smaller than XML and faster and easier to parse.

JSON is a light weight data interchange format. Rather than attempt to chain references as comma delimited strings, or have to deal with the complexity (and over head) of XML, JSON provides a format that converts a server side structure into a JavaScript object that one can use practically right out of the box.

For Example,

```
{
  "employees":
```

```
[
  {"firstName": "Arjun", "lastName": "Singh"},
  {"firstName": "Sanjay", "lastName": "Verma"},
  {"firstName": "Peter", "lastName": "Sharma"},
]
```

The employees object is an array of 3 employee records (object)

Array

Array elements are generally of a basic type (number, string, boolean, or null), however can also be a nested array or object. Elements are comma-delimited and contained within brackets:

```
myArray = ["John Doe"29,true,null]
```

```
myArray = [], {}]
```

```
myArray[2] true
```

Array with Objects

This array contains comma-delimited objects which each contain multiple comma-delimited objects which each contain multiple comma delimited key/value pairs. Objects within an array can be accessed using the array name and index:

```
myArray =
```

```
[
  {"name": "John Doe", "age": 29},
  {"name": "Anna Smith", "age": 24},
  {"name": "peter Jones", "age": 39},
]
```

```
myArray[0].name John Doe
```

```
//method of the object to be initialised
```

```
//This is called constructor chaining.
```

```
Rectangle.call(this, w, h);
```

```
//Now store the position of the upper-left corner of the rectangle
```

```
this.x = x;
```

```
this.y = y;
```

```
}
```

```
//Create a prototype for the subclass that inherits from the prototype
```

```
//of the superclass. We do this with the heir() function.
```

```
function heir(p) {
  function f() {}
  f.prototype = p;
  return new f();
}
```

```
PositionedRectangle.prototype =heir(Rectangle.prototype);
```

```
//Since the subclass prototype object was created with the heir()
function,
```

```
//it does not have a meaningful constructor property. So set that now.
```

```
PositionedRectangle.prototype.constructor = PositionedRectangle;
```

```
//Now that we've configured the prototype object for our subclass,
```

```
//we can add instance methods to it.
```

```
PositionedRectangle.prototype.contains = function (x,y) {
  return (x > this.x &&
    x < this.x + this.width &&
    y > this.y &&
    y < this.y + this.height);
};
```

As one can see from above code, creating a subclass in JavaScript is not as simple as creating a class that inherits directly from Object. First, there is the issue of invoking the superclass construction from the subclass constructor. Take care when you do this that the superclass constructor is invoked as a method of the newly created object.

JSON syntax is a subset of the JavaScript object notation syntax:

Data is in name/value pairs

Data is separated by commas

Curly braces hold objects

Square brackets hold arrays

```
<!DOCTYPE html>
<html>
<body>
<h2>Create Object from JSON String</h2>
<p>
First Name: <span id="fname"></span><br>
Last Name: <span id="lname"></span><br>
</p>
<script>
var txt = '{"employees":[" +
'{"firstName":"John","lastName":"Doe" },' +
'{"firstName":"Anna","lastName":"Smith" },' +
'{"firstName":"Peter","lastName":"Jones" }]}';

var obj = eval("(" + txt + ")");
document.getElementById("fname").innerHTML=obj.employees[1].firstName;
document.getElementById("lname").innerHTML=obj.employees[1].lastName;
</script>
</body>
</html>
```



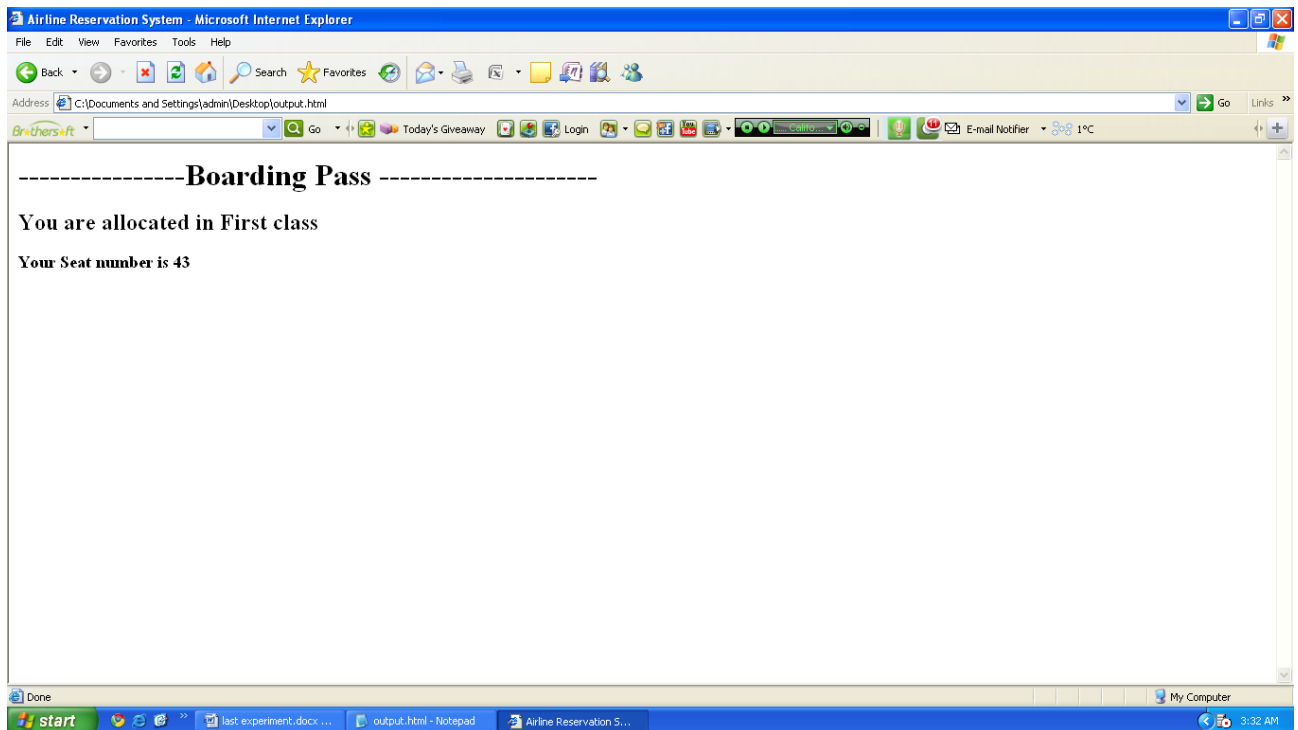
```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!-- Solution11.16 -->
<!-- Airline Reservation System-->
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<title>Airline Reservation System</title>
<script type = "text/javascript">
var input;
var secondInput;
var element;
var secondElement;
var firstCount = 0;
var economyCount = 0;
var seats = [ ,0,0,0,0,0,0,0,0,0]; //allocate 10-element Array
function startArray()
{
for(var i=0; i<11; i++)
{
input = window.prompt("Please type 1 for First Class and Please type 2 for Economy.", "0");
if (input == 1 || input == 2)
{
element = linearSearch(seats);
```

```
if (element==1 && input == 1)
{
document.writeln("The First Class is already fully booked<br/>");
secondQuestion(seats);
}
else if (element ==-1 && input == 2)
{
document.writeln("The Economy Class is already fully booked<br/>");
secondQuestion(seats);
}
else
boardingPass(input);
}
//to terminate the program
else
{
window.status = "Bye-bye!";
System.exit(0);
}
}
}
function linearSearch(theArray)
{
if (input == 1)
{
for (var n=0; n<6 ; n++)
if (theArray [n] == 0)
return n;
}
else if (input == 2)
{
for (var n=6; n<11 ; n++)
if (theArray [n] == 0)
return n;
}
return -1;
}
function boardingPass(theInput)
{
if (input ==1)
{
document.writeln("-----BOARDING PASS-----<br/>");
document.writeln("You are allocated in the First Class<br/>");
document.writeln("Your seat number is "+ element+"<br/>");
document.writeln("-----<br/>");
seats[element]= 1;
firstCount++;
}
else if (input ==2)
{
document.writeln("-----BOARDING PASS-----<br/>");
document.writeln("You are allocated in the Economy Class<br/>");
document.writeln("Your seat number is "+ element + "<br/>");
document.writeln("-----<br/>");
seats[element]= 1;
economyCount++;
}
}
function secondQuestion(theArray)
{
if (input == 1)
```

```
{
for (var n=6; n<11 ;n++)
{
if (theArray [n] == 0)
{
secondInput = window.prompt("Do you want to move to Economy Class? (If YES, please press 1. If NO,
please press 2)","0");
if ( secondInput == 1)
{
input = 2;
element=linearSearch(seats);
document.writeln("You have been allocated to Economy Class<br/>");
boardingPass(input);
break;
}
else if (secondInput == 2)
{
document.writeln("Next flight leaves in 3 hours<br/>");
break;
}
}
}
else if (input == 2)
{
for (var n=0; n<6 ;n++)
{
if (theArray [n] == 0)
{
secondInput = window.prompt("Do you want to move to First Class? (If YES, please press 1. If NO, please
press 2)","0");
if ( secondInput == 1)
{
input = 1;
element=linearSearch(seats);
document.writeln("You have been allocated to First Class<br/>");
boardingPass(input);
break;
}
else if (secondInput == 2)
{
document.writeln("Next flight leaves in 3 hours<br/>");
break;
}
}
}
}
}
</script>
</head>
<body onload = "startArray()"></body>
</html>
```

OUTPUT:

Login: karups



9) Explain object oriented techniques in detail. (May 2014)

**Constructor**

Constructors are the methods in which the properties of newly created objects get initialized. While initializing the properties of an object, the constructor needs to refer the corresponding object. We can refer to this object using a predefined variable called `this`.

Every new expression must include a call to a constructor, whose name is the same as the object being created.

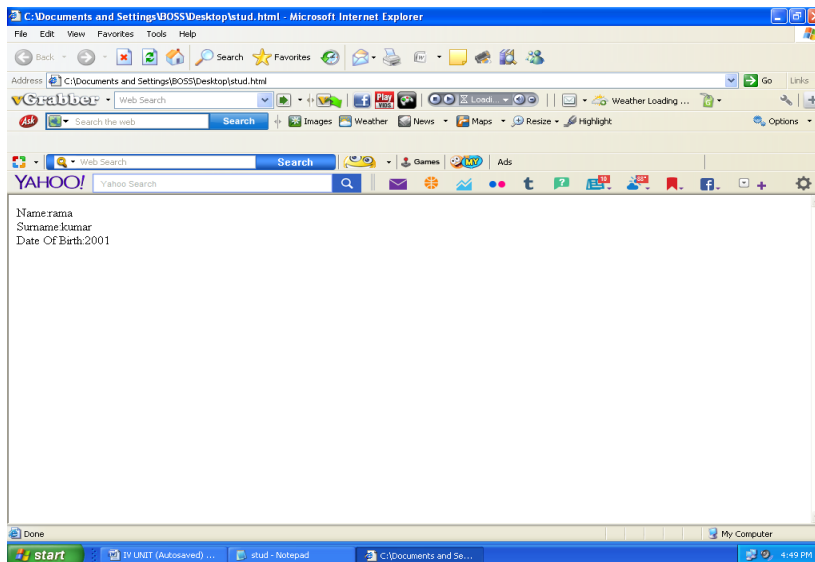
```
<html>
<head>
<script type = "text/javascript">
function Student(Stud_name,Stude_surname,Stud_DOB)
{
this.name=Stud_name;//Initializing the data properties
this.surname=Stude_surname;
this.DOB=Stud_DOB;
this.display=StudInfo;//Initializing the method
}
function StudInfo()
{
document.write("Name:"+this.name+"<br/>");
document.write("Surname:"+this.surname+"<br/>");
document.write("Date Of Birth:"+this.DOB+"<br/>");
}
</script>
</head>
<body>
<script type = "text/javascript">
var Std_III=new Student("rama","kumar",2001);//creating constructor
Std_III.display();//invoking the method of an object
document.write("<br/>");
</script>
</body>
</html>
```

**Script Explanation**

In above written script we have created an object `Student` using `new` operator. While creating the object we have written a constructor to which the initializing values are passed as parameter. Then using the variable `this` we can access the data properties of the object `Student`. Similarly the method `display` is initialized in



this constructor. This method is then invoked in the body section using the dot operator as-



Std\_III.display(). Hence as an output we can see the values that gets assigned to the data properties. Thus we create a collection of objects using the same constructor then it resembles the concept of class in the object oriented programming. But there is such class concept in javascript , although it supports the concept of object. Hence javascript is called as object based programming.

### Object constructor and prototyping.

Javascript provides a special predefined constructor function called Object(), used with the new keyword, to build a generic object. The return value of the Object() constructor is assigned to a variable. The variable contains a reference to an instance of a new object. Program below create a user-defined function that will act as a constructor.

Syntax:

```
Var myobj = new Object();
```

For example

```
<html>
<head><title>The object() Constructor</title>
<script type = "text/javascript">
Var cat = new Object();
Alert(cat);
</script>
<body>
</body>
</html>
```

Explanation

The Object() constructor creates and returns a reference to a cat object.

The returned value from the Object() constructor is a reference to an object.[object Object] means that the cat is a descendant of the Object object from which all objects are derived.

### Prototyping

In a language such as java or c++ , to create a class as an extension of another, one define the class in such a way that it inherits the functionality of the higher-level object and overrides whatever functionality ha/she does not want.

Javascript , on the other hand , provides for a constructor, via object, that allows one to create new objects. It is the object constructor that allocates the memory for the object , including all of its properties.

The Object also provides a prototype property that enables to extend any object, including the built-in-ones such as String and Number. One uses this prototype to derive new object methods and properties, rather than class inheritance:

```
Number.prototype.add = function(val1,val2) {return val1+val2;}
```

```
Var num = new Number();
```

```
Num.prototype.add = function(val1,val2)
Var sum =num.add(8,3);//sum is 11
```

This demonstrates of the dynamic nature of Javascript, and how one can add new methods and properties to objects at runtime. There is a difference between adding a new property or method to an instance of an object and extending the object through prototype,

The following code is not the same as the preceding code snippet because the add method now applies only to the object instance, and not to all instances of an object:

```
Var num = new Number();
Num.add=function(val1,val2){return val1+val2};
Var sum = num.add(8,3);
```

When anyone extent an object through prototype, the method or property is available to all instances of the object. When he/she extend an object instance with a new property or method, it is available only to the instance.

The concept of extending objects via prototyping is best explained through a demonstration.

The following example demonstrates how to extend the employee object by defining a new method, trim, using the underlying object prototype property. Example

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

<script>
function employee(name,jobtitle,born)
{
this.name=name;
this.jobtitle=jobtitle;
this.born=born;
}

var fred=new employee("Fred Flintstone","Caveman",1970);
employee.prototype.salary=null;
fred.salary=20000;

document.write(fred.salary);
</script>

</body>
</html>
20000
```

### Sub classes and Super classes in JavaScript.

Java, c++ and other class-based object-oriented languages have an explicit concept of the class hierarchy. Every class can have a superclass from which it inherits properties and methods. Any class can be extended, or subclasses, so that the resulting sub class inherits its behavior. Javascript supports inheritance instead of class-bases inheritance. Still javascript analogies to the class hierarchy can be drawn.

In javascript , the Object class is the most generic, and all other classes are specialized versions, or subclasses, of it. Another way to say this is that object is the superclass of all built-in classes, and all classes inherit in a few basic methods from object.

The classes are all direct subclasses of object. When necessary however, it is possible to subclass any other class. For example

```
<script language="javascript" type="text/javascript">
<!--
```

```
// thanks to webreference
```

```
function superClass() {
  this.supertest = superTest; //attach method superTest
}
```

```

function subClass() {
    this.inheritFrom = superClass;
    this.inheritFrom();
    this.subtest = subTest; //attach method subTest
}

function superTest() {
    return "superTest";
}

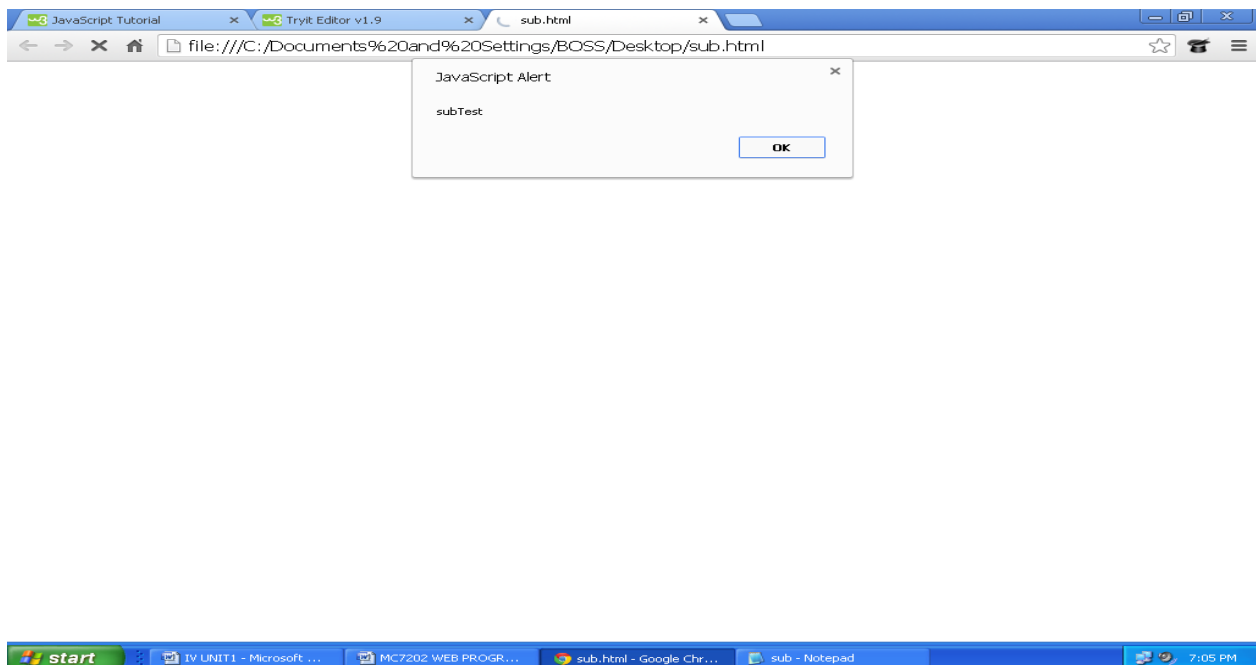
function subTest() {
    return "subTest";
}

var newClass = new subClass();

alert(newClass.subtest()); // yields "subTest"
alert(newClass.supertest()); // yields "superTest"

//-->
</script>

```



#### 10) Explain sub classes and super classes in javascript. (May 2015)

Java, c++ and other class-based object-oriented languages have an explicit concept of the class hierarchy. Every class can have a superclass from which it inherits properties and methods. Any class can be extended, or subclasses, so that the resulting sub class inherits its behavior.

Javascript supports inheritance instead of class-based inheritance. Still javascript analogies to the class hierarchy can be drawn.

In javascript, the Object class is the most generic, and all other classes are specialized versions, or subclasses, of it. Another way to say this is that object is the superclass of all built-in classes, and all classes inherit in a few basic methods from object.

The classes are all direct subclasses of object. When necessary however, it is possible to subclass any other class. For example

```

<script language="javascript" type="text/javascript">
<!--

```

// thanks to webreference

```
function superClass() {
    this.supertest = superTest; //attach method superTest
}
```

```
function subClass() {
    this.inheritFrom = superClass;
    this.inheritFrom();
    this.subtest = subTest; //attach method subTest
}
```

```
function superTest() {
    return "superTest";
}
```

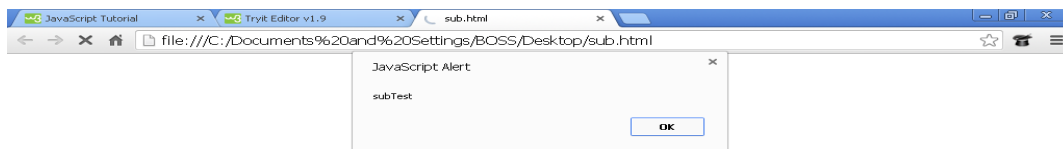
```
function subTest() {
    return "subTest";
}
```

```
var newClass = new subClass();
```

```
alert(newClass.subtest()); // yields "subTest"
alert(newClass.supertest()); // yields "superTest"
```

```
//-->
```

```
</script>
```



11) i) Explain any four jQuery AJAX methods in detail. (May 2015)

jQuery provides **load()** method to do the job –

Syntax

Here is the simple syntax for **load()** method –

```
[selector].load( URL, [data], [callback] );
```

Here is the description of all the parameters –

- **URL** – The URL of the server-side resource to which the request is sent. It could be a CGI, ASP, JSP, or PHP script which generates data dynamically or out of a database.

- **data** – This optional parameter represents an object whose properties are serialized into properly encoded parameters to be passed to the request. If specified, the request is made using the **POST** method. If omitted, the **GET** method is used.
- **callback** – A callback function invoked after the response data has been loaded into the elements of the matched set. The first parameter passed to this function is the response text received from the server and second parameter is the status code.

Consider the following HTML file with a small JQuery coding –

```
<html>

<head>
  <title>The jQuery Example</title>
  <script type = "text/javascript"
    src = "http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

  <script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
      $("#driver").click(function(event){
        $('#stage').load('/jquery/result.html');
      });
    });
  </script>
</head>

<body>

  <p>Click on the button to load /jquery/result.html file –</p>

  <div id = "stage" style = "background-color:cc0;">
    STAGE
  </div>

  <input type = "button" id = "driver" value = "Load Data" />

</body>

</html>
```

Here **load()** initiates an Ajax request to the specified URL **/jquery/result.html** file. After loading this file, all the content would be populated inside <div> tagged with ID *stage*. Assuming, our /jquery/result.html file has just one HTML line –

S.N.	Methods & Description
1	<u><b>jQuery.ajax( options )</b></u> Load a remote page using an HTTP request.
2	<u><b>jQuery.ajaxSetup( options )</b></u> Setup global settings for AJAX requests.
3	<u><b>jQuery.get( url, [data], [callback], [type] )</b></u> Load a remote page using an HTTP GET request.
4	<u><b>jQuerygetJSON( url, [data], [callback] )</b></u> Load JSON data using an HTTP GET request.
5	<u><b>jQuery.getScript( url, [callback] )</b></u> Loads and executes a JavaScript file using an HTTP GET request.
6	<u><b>jQuery.post( url, [data], [callback], [type] )</b></u> Load a remote page using an HTTP POST request.
7	<u><b>load( url, [data], [callback] )</b></u> Load HTML from a remote file and inject it into the DOM.
8	<u><b>serialize( )</b></u> Serializes a set of input elements into a string of data.
9	<u><b>serializeArray( )</b></u> Serializes all forms and form elements like the .serialize() method but returns a JSON data structure for you to work with.

ii) With an example, illustrate how to convert a JSON Text to a JavaScript object.

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

<h2>Create Object from JSON String</h2>

<p id="demo"></p>

<script>
var text = '{ "employees":[' +
'{"firstName":"John","lastName":"Doe" },' +
'{"firstName":"Anna","lastName":"Smith" },' +
'{"firstName":"Peter","lastName":"Jones" } ]}';
```



```
obj = JSON.parse(text);
document.getElementById("demo").innerHTML =
obj.employees[1].firstName + " " + obj.employees[1].lastName;
</script>
```

```
</body>
```

```
</html>
```

Create Object from JSON String

Anna Smith

## 12) Compare JSON and jQuery. (May 2016)

```
var origArrayGroups = {c: 0, d: 100, a: "AT", b: "1002"};
var userArrayGroups = {a: "AT", b: "1000"};

var diff = {};
for (var prop in userArrayGroups) {
  console.log("Comparing element: " + prop);
  if(userArrayGroups[prop] !== origArrayGroups[prop])
  {
    diff[prop] = origArrayGroups[prop];
    console.log("Result: " + userArrayGroups[prop] + "!=" + origArrayGroups[prop]);
  }
  else
  {console.log("Result: " + userArrayGroups[prop] + "==" + origArrayGroups[prop]);}
}

console.log(diff);
Actual Return:
```

```
Object {b: "1002"}
```

## 14) How to implement inheritance in JavaScript? Discuss with example. (May 2016)

A definition

If there was one thing common to all definitions of inheritance, it would probably be the identification of a hierarchy between types (or objects in the case of JavaScript) that permits some kind of delegation between members of said hierarchy.

Given this, how about:

Inheritance in JavaScript is the ability to have an object delegate some or all of its implementation to another, by way of a hierarchical link.

JavaScript can be said to provide such a link natively via the prototype-chain (i.e. the chain of [[Prototype]] references).

Inheritance over time

Prior to ES5, inheritance was achieved thus:

```
function Parent() {
  this.foo = function() {
    console.log('foo');
  };
}
Parent.prototype.bar = function() {
  console.log('bar');
}
function Child() {
}
Child.prototype = p = new Parent();
Child.prototype.constructor = Child;
var c = new Child();
c instanceof Parent; // true
c instanceof Child; // true
```

```

c.__proto__ === p; // true
c.__proto__.__proto__ === Parent.prototype; // true
c.__proto__.__proto__.__proto__ === Object.prototype; // true
c.__proto__.__proto__.__proto__.__proto__ === null; // true
c.foo(); // foo
c.bar(); // bar

```

This configured the inheritance relationship thus:

```

c => "actual instance of Parent" => Parent.prototype (instance of Object) => Object.prototype
(instance of Object) => null

```

Which is usually exactly what we wanted. The important things work: `instanceof` behaves as expected, functions on the prototype of the base object `Parent` are available to us, and `Parent.prototype` is not the direct prototype of `Child` (thereby avoiding the possibility of accidental interference with other objects sharing that prototype object).

#### 15) How to create prototypes in JavaScript? Explain with example.

When you create an object using the `new` keyword, it creates a new object, passes it in as `this` to the constructor function (letting that function do to it whatever it wants) and then, and this is the important part, it takes the object instance that is pointed to by that function's **prototype** property and assigns it as.

If you inspect the prototype of the cats in Examples A and B you get the following result:

```

catA.__proto__;
Object { }

```

But, if you inspect the cat in Example C you get this result:

```

catC.__proto__;
Cat {age: 0}

```

even though they were not created using a constructor function. Since we didn't specify a prototype for these, they have a prototype of `Object`. It is possible to create objects without a prototype using this syntax: `var myObject = Object.create(null);`, but that is very rare and I've never seen a use for it. So barring that example, I think it is safe to say that all objects eventually inherit from `Object`.

This is even true of `catC`; look what happens when we inspect `catC` further:

```

catC.__proto__;
Cat {age: 0}

catC.__proto__.__proto__;
Object { }

catC.__proto__.__proto__.__proto__;
null

```

Notice that `catC` has a prototype (`__proto__`) of `Cat`. Actually, to say it that way is not really accurate. `Cat` is a function, so it cannot be a prototype; remember in the definitions above that a prototype is not a function it is an *object instance*. This is important to remember when thinking about prototypes -- an object's prototype is not the function that created it but an instance of an object. Since this is important, let's explore it a bit further:

```

Cat;
function Cat(name, color) {
  this.name = name;
  this.color = color;
}

Cat.prototype;
Cat {age: 0}

catC;
Cat {name: "Fluffy", color: "White", age: 0}

```

Look at the difference between `Cat`, `Cat.prototype` and `catC`. What this is showing is that `Cat` is a

function but `Cat.prototype` and `catC` are objects. It further shows that `Cat.prototype` has an `age` property (set to 0) and `catC` has three properties -- including `age...sort of (stay tuned)`. When you define a function, it creates more than just the function, it also creates a new *object* with that function as its type and assigns that new *object* to the function's prototype property. When we first created the `Cat` function, before we executed the line `Cat.prototype.age = 0`; if we would have inspected `Cat`'s prototype it would have looked like this: `Cat { }`, an object with no properties, but of type `Cat`. It was only after we called `Cat.prototype.age = 0`; that it looked like this: `Cat {age: 0}`.

## UNIT-V

- 1) Explain the setting up the environment for LAMP server.

PHP is a simple yet powerful language designed for creating HTML content. It describes the nature and history of PHP, which platform it runs on; and how to download, install and configure it.

### Setting up the LAMP SERVER

The LAMP server (Linux, Apache, MYSQL, PHP (or perl)) is one of the most important servers one might ever set up.

Since the LAMP server's underlying foundation is Linux, it has rock-solid reliability, security and can be installed on all kinds of hardware.

Here we installing the LAMP server on an Ubuntu 9.10 machine. This machine can be either a standard installation or an Ubuntu Server installation. Either way, it is all command lines from here. Because the OS is already installed, all we have to install is Apache, MYSQL and PHP. So there are only three major steps to take care of in order to get the LAMP server up and running.

### Apache

Apache is a web server. From within the terminal window type the command:

`Sudo apt-get install apache2`

If, by chance, you are using a distribution that does not use Sudo, you will need `su` to the root user and issue the above command without the `sudo` command.

Depending on your OS installation, the above command might need to pick up some dependencies. If so, those dependencies. At the end of the installation, apache should automatically start. If it does not, issue the following command

`Sudo/etc/init.d/apache2 start`

You can now open up a browser and point it to the IP address of the server to get the famous "It works" page. You are ready to move on to PHP.

### PHP

To begin the process of installing PHP, issue the following command

`Sudo apt-get install php5 libapache2-mod-php5`

When the installation is complete, restart Apache with the command:

`Sudo/etc/init.d/apache2 restart`

Now, let us give PHP a little test to make sure it has installed. In the terminal window, create a new file called `test.php`

Save that file place it in `/var/www/`. Now open up the browser to the address

`http://ADDRESS_OF_SERVER/test.php` and where `ADDRESS_OF_SERVER` is the actual address of the server. You should see "test PHP page" in the browser.

### MYSQL

Mysql is the database piece of the puzzle. This installation requires a few more steps than what users have just experienced. The first step is to install the server itself with the command

`Sudo apt-get install mysql-server`

Again, depending upon the OS installation, there might be some dependencies to be installed

After the installation is complete users need to log into the Mysql prompt and give the administrative user a password. Do this by following steps:

Log into MYSQL with the command `mysql-u root-p`.

As no password has been configured, you will only need to hit enter when prompted for the password. Enter the command `SET PASSWORD FOR 'root'@'localhost' = PASSWORD('YOUR PASSWORD');` Where `YOURPASSWORD` is the password you want to use for the administrative user.

Now quit the MYSQL prompt by issuing the command `quit` and hitting enter.

Start the MYSQL server with the command `sudo/etc/init.d/mysql start`.

Now the LAMP SERVER IS NOW UP AND RUNNING. FROM THE TERMINAL WINDOW, ISSUE THE COMMAND

`Sudo taskset`

This command will open a curses-based tool which allows one to select numerous software options for installation.

One of those selections is a LAMP server. All you need to do is mark LAMP server for installation.

Once you have selected LAMP server, hit the Tab key on the “button” and hit the Enter key. You will have to answer a single question when you get to the MYSQL portion of the install.

## 2) Explain variables and constants with examples.

Variables:

Variables are used for storing values, like text strings, numbers or arrays.

When a variable is declared, it can be used over and over again in your script.

All variables in PHP start with a \$ sign symbol.

A variable should start with a letter or by the symbol(\_)underscore.

Creation of a Variable:

The correct way of declaring a variable in PHP:

```
$var_name = value;
```

\*Datatype is not specified during declaration. According to the assigned value the variable will act as an integer, float or string.

Example:

```
<?php
```

```
$txt="Hello World!"; //Here the variable txt is of string datatype.
```

```
$x=16; //Here the variable x is of type integer.
```

```
$x1=16.2345; //Here the variable x1 is of float datatype.
```

```
?>
```

The following simple data types are supported by PHP

Integer: A whole number (no fractions), such as -43, 0, 1, 27, or 5438. The range of integers that is allowed varies, depending on the operating system

Floating point number: A number (usually not a whole number) that includes decimal places, such as 5.24 or 23.456789. This is often called a *real number* or a *float*.

Character string: A series of single characters, such as hello. There is no practical limit on the length of a string.

Boolean: A TRUE or FALSE value. The string FALSE (can be upper- or lowercase), the integer 0, the float 0.0, an empty string and the constant NULL are considered FALSE. When Boolean values are displayed FALSE displays as a blank string and TRUE echoes a 1.

Conversion of Data Types of variables

If there are two variables of different data types in an expression, PHP automatically converts and promotes the data type of one variable to the data type of the other variable with higher precision.

Example:

```
$firstNumber = 1; // PHP stores it as an integer
```

```
$secondNumber = 1.1; // PHP stores it as a float
```

```
$sum = $firstNumber + $secondNumber;
```

In the above piece of code, the data type of \$firstnumber is automatically converted to float by PHP. If a number is enclosed within a string, in arithmetic expressions, the string containing numbers is automatically converted to a number.

Example:

```
$var1 = "1";
```

```
$var2 = 2;
```

```
$total = $var1 + $var2;
```

Here, the value in \$total is 3 because PHP automatically converts “1” to 1.

If a String is used in an arithmetic expression and it does not contain any number, then PHP treats the string as number 0 in evaluating the expression.

The programmer also can change the way a variable is stored by using specific type cast.

The value in the variable on the right side of the equal sign is stored in the variable on the left side as the specified type.

Example:

```
$intvar = 10;
```

```
$floatvar = 100.10;
```

```
$intvar = (int)$floatvar;
```

In the above example, the value in \$floatvar is converted to integer and stored in \$intvar.

After the type cast, \$intvar contains the value 100.

Displaying the variable:

The value in a variable can be displayed using the echo statement.

The data type of a variable can be displayed using the var\_dump function. To display the data type of a variable \$var we can use var\_dump(\$var).

Removing Variables

The information in a variable can be removed setting the variable to an empty string. For example setting \$var = "" removes the information stored in the variable. If the variable \$var is displayed, nothing is echoed.

A variable can also be uncreated using the unset function. Passing a variable or variables as parameter to the unset function uncreates the variable. Any further use of the variable will give an undefined variable error.

Constants

Constants are similar to variables, that is they act as a placeholder for storing values. But the value stored in a constant cannot change during the execution of the script.

Constant creation

Constants are created using the define statement. The format for creating a constant is define("constantname", "constantvalue");

Points to remember when using constants

The name of a constant does not begin with a \$

We can store either a string or a number in a constant

By convention constant names are given in Capital letters but PHP accepts lowercase letters also.

When using a constant in an expression, the double quotes must be removed.

Using a constant ensures that the value won't be changed accidentally somewhere in the script, leading to the wrong value being used in statements later in the script

Example

```
define("INTEREST", 10.2);
```

The following statement defines a constant named INTEREST and assigns to it the value 10.2. This constant can be used as \$a=INTEREST \*\$b;

### 3) Explain String and array concepts with examples. (May 2016)

STRINGS:

Strings

A *character string* is a series of characters and *characters* are letters, numbers, and punctuation. Character strings are stored in a variable by using double quotes("") or single quotes (') which mark the beginning and end of the string.

Special characters like \n (for newline) \t(for tab) can be used only within strings created using double quotes. In single quoted strings special characters have no meaning and they are output as any other character.

The difference between using single quotes and double quotes are as follows

Single-quoted strings are stored literally, with the exception of \', which is stored as an apostrophe. \n and \t are echoed literally. A variable \$var inside a single quoted string is displayed as \$var itself and not by its value.

In double-quoted strings, variables and special characters are evaluated before the string is stored. For example \n is treated as a newline and display moves to the next line when echo is used \$var inside a double quoted string is evaluated for its value.

Escaping characters within strings

Characters that tell PHP to remove the special meaning of certain characters in a string and treat them as literals are called escaping characters. The special meaning of a character can be removed by preceding it with a \.

Examples:

Preceding \$ with \ removes its special meaning and the characters following \$ are not treated as variables. echo "\\$var"; will output \$var and not the value in \$var.

Similarly \ and " can also be escaped by using \ before them.

String Concatenation:

The process of joining strings is called as concatenation. Strings can be joined in PHP using the .(dot) operator.

Example:

```
$a = "Hello";
```

```
$b = "World";
```

```
$c = $a.$b; // $c = HelloWorld
```

Inbuilt functions for String manipulations

1. trim(\$str), ltrim(\$str) and rtrim(\$str)

The trim functions are used for removing leading and/or trailing spaces in a string. trim() removes both leading and trailing spaces, ltrim removes leading spaces and rtrim removes trailing spaces.

2. str\_word\_count(\$str)

This function returns the number of words in the string \$str.

This function can also be invoked with one more parameter. If this function is called as str\_word\_count(\$str,1), then the words in the string are returned as an array with the first word at index 0.

3. str\_repeat(\$str,\$n)

This function repeats the string \$str \$n times and returns it as a string

4. str\_replace(\$a,\$b,\$str)

This function replaces all occurrences of \$a with \$b in the string \$str and returns the new string.

5. strlen(\$str)

This function returns the length of the string \$str.

6. strpos(\$str, \$substr)

This function returns the position of first \$substr beginning in \$str.

7. strrev(\$str)

This function reverses the string and returns it.

8. strtolower(\$str)

This function returns the lowercase version of the string \$str.

9. strtoupper(\$str)

This function returns the uppercase version of the string \$str.

10. substr(\$str,\$n1,\$n2)

This function returns the substring from position \$n1 for upto \$n2 characters in \$str and returns it.

11. strcmp(\$str1,\$str2) and strcasecmp(\$str1,\$str2)

Compares two strings on alphabetical and numerical order. Returns -1 if \$str1 is less, 0 if two strings are equal, or +1 if \$str1 is greater. strcmp is casesensitive. strcasecmp is not.

## Arrays

Arrays are complex variables that store a group of values under a single variable name.

An array is useful for storing a group of related values.

In PHP arrays are nothing but group of key/value associations under a single array name.

The keys can be either numeric or text. Depending on the type of key, the arrays are classified as

Numeric Arrays (Keys are numbers)

Associative Arrays (Keys are strings)

### Creation of Arrays

Arrays can be created in the following ways

11.1.1. By specifying the key/value association for every element:

Example for creating numeric array:

```
$studname[1] = "Anil";
```

```
$studname[2] = "Bheema";
```

This creates an array called \$studname with two elements and they can be accessed as \$studname[1] and \$studname[2]. The key/value associations are 1/Anil and 2/Bheema.

Example for creating Associative array:

```
$state['TN'] = "Tamil Nadu";
```

```
$state['KA'] = "Karnataka";
```

```
$state['AP'] = "Andhra Pradesh";
```

This creates an associative array with 3 elements.

By not giving any key for the array

An array can be created by not giving any specific key to the values. This automatically creates an array that starts at key number 0.

Example:

```
$streets[] = "MG St.";
```

```
$streets[] = "Renga St.";
```

```
$streets[] = "Andar St.";
```



This creates an array called \$street with 3 elements and the first element can be located at \$streets[0].

By using the array function

Example for creating numeric array:

```
$vowels = array('a','e','i','o','u');
```

This creates an array with 5 elements and the first element starts at index 0.

Example for creating associative array

```
$counCap = array("India" => "NewDelhi","China"=>"beijing");
```

This creates an associative array with 2 elements. The following key/value associations are made

```
$counCap["India"] = "NewDelhi";
```

```
$counCap["China"] = "Beijing";
```

By specifying a range

An array with a range of values can be created by using the following statement:

```
$years = range(2001, 2010);
```

The resulting array looks like the following:

```
$years[0] = 2001
```

```
$years[1] = 2002
```

```
...
```

```
$years[8] = 2009
```

```
$years[9] = 2010
```

Appending an existing array

An existing array can be appended by not specifying the key for the new value to be added. PHP automatically gives the next number as the key for both numeric and associative arrays.

Example:

In the above \$counCap array, the following entry can be made

```
$counCap[] = "Colombo";
```

Here, "Colombo" gets added to the array and is automatically associated with the key 0.

Similarly, in the above \$years array, the statement \$years[] = 2011; associates 2011 to key 10.

Viewing Arrays

The structure and values of any array can be viewed by using one of two statements — var\_dump or print\_r.

Example:

```
$customers= array("Ram","Laksman","Sita");
```

print\_r(\$customers) will give the following output

```
(
```

```
[1] => Ram
```

```
[2] => Lakshman
```

```
[3] => Sita
```

```
)
```

Var\_dump(\$customers) gives the following output

```
array(3) {
```

```
[1]=>
```

```
string(9) "Ram"
```

```
[2]=>
```

```
string(9) "Laksman"
```

```
[3]=>
```

```
string(10) "Sita"
```

```
}
```

Removing values from Arrays

A value can be removed from an array using the unset(\$arrname[key]) statement. An array also can be completely removed using the unset(\$arrname) statement.

Using Arrays in echo

To display an array element \$arrname[2], it can be used in echo as echo \$arrname[2];

To display an array element value in an echo statement containing double quotes, then the array element reference must be enclosed within curly braces as follows:

```
echo "The value of array element 2 is {$arrname[2]}";
```

Handling arrays with loops:

Arrays and loops are natural combination –arrays are indexed using an array index, and

loops use a loop counter. By using the loop counter as the array index we can increment through an entire array.

EX:

```
<?php
$sactors[0]="sathyam";
$sactors[1]="sivam";
$sactors[2]="sundaram";
For($loop_index=0;$loop_index<count($sactors);$loop_index++)
{
Echo " $sactors($loop_index)=",$sactors($loop_index),"<br>";
}
```

Here the count(\$sactors) gives the no. of values stored in the array actor.i.e.,3.

From index 0 to 2 the values are extracted from the array .the output of the pgm is

\$sactors[0]=sathyam

\$sactors[1]=sivam

\$sactors[2]=sundaram

4) Explain the different types of operator in php with example program.

Arithmetic Operators

Comparison Operators

Logical (or Relational) Operators

Assignment Operators

Conditional (or ternary) Operators

Arithmetic Operators:

There are following arithmetic operators supported by PHP language:

Assume variable A holds 10 and variable B holds 20 then:

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiply both operands	A * B will give 200
/	Divide numerator by denominator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0
++	Increment operator, increases integer value by one	A++ will give 11
--	Decrement operator, decreases integer value by one	A-- will give 9

```
<html>
```

```
<head><title>Arithmetical Operators</title></head>
```

```
<body>
```

```
<?php
```

```
$a = 42;
```

```
$b = 20;
```

```
$c = $a + $b;
```

```
echo "Addition Operation Result: $c <br/>";
```

```
$c = $a - $b;
```

```
echo "Substraction Operation Result: $c <br/>";
```

```
$c = $a * $b;
```

```
echo "Multiplication Operation Result: $c <br/>";
```

```
$c = $a / $b;
```

```
echo "Division Operation Result: $c <br/>";
```

```
$c = $a % $b;
```

```
echo "Modulus Operation Result: $c <br/>";
```

```
$c = $a++;
```

```

    echo "Increment Operation Result: $c <br/>";
    $c = $a--;
    echo "Decrement Operation Result: $c <br/>";
?>
</body>
</html>

```

This will produce following result

```

Addition Operation Result: 62
Substraction Operation Result: 22
Multiplication Operation Result: 840
Division Operation Result: 2.1
Modulus Operation Result: 2
Increment Operation Result: 42
Decrement Operation Result: 43

```

#### Comparison Operators:

There are following comparison operators supported by PHP language

Assume variable A holds 10 and variable B holds 20 then:

Operator	Description	Example
==	Checks if the value of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!=	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

```

<html>
<head><title>Comparision Operators</title></head>
<body>
<?php
    $a = 42;
    $b = 20;

    if( $a == $b ){
        echo "TEST1 : a is equal to b<br/>";
    }else{
        echo "TEST1 : a is not equal to b<br/>";
    }

    if( $a > $b ){
        echo "TEST2 : a is greater than b<br/>";
    }else{
        echo "TEST2 : a is not greater than b<br/>";
    }

```

```

if( $a < $b ){
    echo "TEST3 : a is less than b<br/>";
}else{
    echo "TEST3 : a is not less than b<br/>";
}
if( $a != $b ){
    echo "TEST4 : a is not equal to b<br/>";
}else{
    echo "TEST4 : a is equal to b<br/>";
}
if( $a >= $b ){
    echo "TEST5 : a is either grater than or equal to b<br/>";
}else{
    echo "TEST5 : a is nieghter greater than nor equal to b<br/>";
}
if( $a <= $b ){
    echo "TEST6 : a is either less than or equal to b<br/>";
}else{
    echo "TEST6 : a is nieghter less than nor equal to b<br/>";
}
?>
</body>
</html>

```

This will produce following result

```

TEST1 : a is not equal to b
TEST2 : a is greater than b
TEST3 : a is not less than b
TEST4 : a is not equal to b
TEST5 : a is either grater than or equal to b
TEST6 : a is nieghter less than nor equal to b

```

#### Logical Operators:

There are following logical operators supported by PHP language

Assume variable A holds 10 and variable B holds 20 then:

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true then then condition becomes true.	(A and B) is true.
or	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	(A or B) is true.
&&	Called Logical AND operator. If both the operands are non zero then then condition becomes true.	(A && B) is true.
	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	(A    B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is false.

```

<html>
<head><title>Logical Operators</title></head>
<body>
<?php

```

```

$a = 42;
$b = 0;

if( $a && $b ){
    echo "TEST1 : Both a and b are true<br/>";
}else{
    echo "TEST1 : Either a or b is false<br/>";
}
if( $a and $b ){
    echo "TEST2 : Both a and b are true<br/>";
}else{
    echo "TEST2 : Either a or b is false<br/>";
}
if( $a || $b ){
    echo "TEST3 : Either a or b is true<br/>";
}else{
    echo "TEST3 : Both a and b are false<br/>";
}
if( $a or $b ){
    echo "TEST4 : Either a or b is true<br/>";
}else{
    echo "TEST4 : Both a and b are false<br/>";
}
$a = 10;
$b = 20;
if( $a ){
    echo "TEST5 : a is true <br/>";
}else{
    echo "TEST5 : a is false<br/>";
}
if( $b ){
    echo "TEST6 : b is true <br/>";
}else{
    echo "TEST6 : b is false<br/>";
}
if( !$a ){
    echo "TEST7 : a is true <br/>";
}else{
    echo "TEST7 : a is false<br/>";
}
if( !$b ){
    echo "TEST8 : b is true <br/>";
}else{
    echo "TEST8 : b is false<br/>";
}
?>
</body>
</html>

```

This will produce following result

```

TEST1 : Either a or b is false
TEST2 : Either a or b is false
TEST3 : Either a or b is true
TEST4 : Either a or b is true
TEST5 : a is true
TEST6 : b is true
TEST7 : a is false
TEST8 : b is false

```

Assignment Operators:

There are following assignment operators supported by PHP language:

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

```
<html>
<head><title>Assignment Operators</title></head>
<body>
<?php
    $a = 42;
    $b = 20;

    $c = $a + $b; /* Assignment operator */
    echo "Addtion Operation Result: $c <br/>";
    $c += $a; /* c value was 42 + 20 = 62 */
    echo "Add AND Assigment Operation Result: $c <br/>";
    $c -= $a; /* c value was 42 + 20 + 42 = 104 */
    echo "Subtract AND Assignment Operation Result: $c <br/>";
    $c *= $a; /* c value was 104 - 42 = 62 */
    echo "Multiply AND Assignment Operation Result: $c <br/>";
    $c /= $a; /* c value was 62 * 42 = 2604 */
    echo "Division AND Assignment Operation Result: $c <br/>";
    $c %= $a; /* c value was 2604/42 = 62*/
    echo "Modulus AND Assignment Operation Result: $c <br/>";
?>
</body>
</html>
```

This will produce following result

```
Addtion Operation Result: 62
Add AND Assigment Operation Result: 104
Subtract AND Assignment Operation Result: 62
Multiply AND Assignment Operation Result: 2604
Division AND Assignment Operation Result: 62
Modulus AND Assignment Operation Result: 20
```

### Conditional Operator



There is one more operator called conditional operator. This first evaluates an expression for a true or false value and then execute one of the two given statements depending upon the result of the evaluation. The conditional operator has this syntax:

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

```
<html>
<head><title>Arithmetical Operators</title></head>
<body>
<?php
    $a = 10;
    $b = 20;

    /* If condition is true then assign a to result otheriwse b */
    $result = ($a > $b ) ? $a :$b;
    echo "TEST1 : Value of result is $result<br/>";
    /* If condition is true then assign a to result otheriwse b */
    $result = ($a < $b ) ? $a :$b;
    echo "TEST2 : Value of result is $result<br/>";
?>
</body>
</html>
```

This will produce following result

```
TEST1 : Value of result is 20
TEST2 : Value of result is 10
```

#### Operators Categories:

All the operators we have discussed above can be categorised into following categories:

Unary prefix operators, which precede a single operand.

Binary operators, which take two operands and perform a variety of arithmetic and logical operations.

The conditional operator (a ternary operator), which takes three operands and evaluates either the second or third expression, depending on the evaluation of the first expression.

Assignment operators, which assign a value to a variable.

#### Precedence of PHP Operators:

Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated. Certain operators have higher precedence than others; for example, the multiplication operator has higher precedence than the addition operator:

For example  $x = 7 + 3 * 2$ ; Here x is assigned 13, not 20 because operator \* has higher precedence than + so it first get multiplied with  $3*2$  and then adds into 7.

Here operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedence operators will be evaluated first.

Category	Operator	Associativity
Unary	! ++ --	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right
Logical AND	&&	Left to right
Logical OR		Left to right
Conditional	?:	Right to left

Assignment	= += -= *= /= %=	Right to left
------------	------------------	---------------

### 5) Explain Looping structures in detail. (May 2014/May 2015)

#### Conditional Statements

Very often when you write code, you want to perform different actions for different decisions.

You can use conditional statements in your code to do this.

In PHP we have the following conditional statements:

if statement - use this statement to execute some code only if a specified condition is true

if...else statement - use this statement to execute some code if a condition is true and another code if the condition is false

if...elseif....else statement - use this statement to select one of several blocks of code to be executed

switch statement - use this statement to select one of many blocks of code to be executed

#### The if Statement

Use the if statement to execute some code only if a specified condition is true.

##### Syntax

*if (condition) code to be executed if condition is true;*

The following example will output "Have a nice weekend!" if the current day is Friday:

```
<html> <body> <?php $d=date("D"); if ($d=="Fri") echo "Have a nice weekend!"; ?> </body> </html>
```

Notice that there is no ..else.. in this syntax. The code is executed only if the specified condition is true.

#### The if...else Statement

Use the if....else statement to execute some code if a condition is true and another code if a condition is false.

##### Syntax

*if (condition) code to be executed if condition is true; else code to be executed if condition is false;*

##### Example

The following example will output "Have a nice weekend!" if the current day is Friday, otherwise it will output "Have a nice day!":

```
<html> <body>
<?php $d=date("D"); if ($d=="Fri") echo "Have a nice weekend!"; else echo "Have a nice day!"; ?>
</body> </html>
```

If more than one line should be executed if a condition is true/false, the lines should be enclosed within curly braces:

```
<html> <body> <?php $d=date("D"); if ($d=="Fri") { echo "Hello!<br />"; echo "Have a nice weekend!";
echo "See you on Monday!"; } ?> </body> </html>
```

#### The if...elseif....else Statement

Use the if....else if...else statement to select one of several blocks of code to be executed.

##### Syntax

*if (condition) code to be executed if condition is true; elseif (condition) code to be executed if condition is true; else code to be executed if condition is false;*

##### Example

The following example will output "Have a nice weekend!" if the current day is Friday, and "Have a nice Sunday!" if the current day is Sunday. Otherwise it will output "Have a nice day!":

```
<html> <body> <?php $d=date("D"); if ($d=="Fri") echo "Have a nice weekend!"; elseif ($d=="Sun")
echo "Have a nice Sunday!"; else echo "Have a nice day!"; ?> </body> </html>
```

#### PHP Switch Statement

Conditional statements are used to perform different actions based on different conditions.

#### The PHP Switch Statement

Use the switch statement to select one of many blocks of code to be executed.

##### Syntax

```
switch (n) { case label1: code to be executed if n=label1; break; case label2: code to be executed if
n=label2; break; default: code to be executed if n is different from both label1 and label2; }
```

This is how it works: First we have a single expression *n* (most often a variable), that is evaluated once.

The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use break to prevent the code from running into the next case automatically. The default statement is used if no match is found.

##### Example

```
<html> <body> <?php $x=1; switch ($x) { case 1: echo "Number 1"; break; case 2: echo "Number 2";
break; case 3: echo "Number 3"; break; default: echo "No number between 1 and 3"; } ?> </body> </html>
```

## 6) Explain the embedding PHP with HTML.(May 2015/2016)

### PHP Functions

The real power of PHP comes from its functions.

In PHP, there are more than 700 built-in functions.

### PHP Functions

To keep the script from being executed when the page loads, you can put it into a function.

A function will be executed by a call to the function.

You may call a function from anywhere within a page.

### Create a PHP Function

A function will be executed by a call to the function.

### Syntax

```
function functionName() { code to be executed; }
```

### PHP function guidelines:

Give the function a name that reflects what the function does

The function name can start with a letter or underscore (not a number)

### Example

A simple function that writes my name when it is called:

```
<html> <body> <?php function writeName() { echo "Kai Jim Refsnes"; } echo "My name is ";
writeName(); ?> </body> </html>
```

### Output:

My name is Kai Jim Refsnes

### PHP Functions - Adding parameters

To add more functionality to a function, we can add parameters. A parameter is just like a variable.

Parameters are specified after the function name, inside the parentheses.

### Example 1

The following example will write different first names, but equal last name:

```
<html> <body> <?php function writeName($fname) { echo $fname . " Refsnes.<br />"; } echo "My name
is "; writeName("Kai Jim"); echo "My sister's name is "; writeName("Hege"); echo "My brother's name is
"; writeName("Stale"); ?> </body> </html>
```

### Output:

My name is Kai Jim Refsnes. My sister's name is Hege Refsnes. My brother's name is Stale Refsnes.

### Example 2

The following function has two parameters:

```
<html> <body> <?php function writeName($fname,$punctuation) { echo $fname . " Refsnes" .
$punctuation . "<br />"; } echo "My name is "; writeName("Kai Jim","."); echo "My sister's name is ";
writeName("Hege","!"); echo "My brother's name is "; writeName("Ståle","?"); ?> </body> </html>
```

### Output:

My name is Kai Jim Refsnes. My sister's name is Hege Refsnes! My brother's name is Ståle Refsnes?

### PHP Functions - Return values

To let a function return a value, use the return statement.

### Example

```
<html> <body> <?php function add($x,$y) { $total=$x+$y; return $total; } echo "1 + 16 = " . add(1,16);
?> </body> </html>
```

### Output:

1 + 16 = 17

## 7) Explain Establish connectivity with MYSQL database. (May 2014/May 2015/2016)

### Embedding PHP in regular HTML

If you have been paying attention to our earlier articles, you hopefully have picked up on how PHP can be embedded into a regular HTML document. For example, we should already know that the following is an example of how PHP is embedded:

```
<html>
<head>
<title>My first PHP Page</title>
</head>
```

```
<body>
This is normal HTML code
<?php

        // php code goes here

?>
```

Back into normal HTML

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

Further, we also learned how we could quickly output variables from PHP code without all of the hassle of an echo statement by doing the following:

```
<?=$variable?>
```

Today, we will extend our knowledge of embedded PHP by discussing how PHP can be used to control the flow of a web page through conditionals or repetition-control structures.

How it works

As mentioned in earlier articles, PHP will only process things that are enclosed within one of its valid code blocks (such as `<?php` and `?>`). Because of this, PHP effectively ignores everything that it was not specifically told to process and can be used to our advantage. For example, what will the output from the following be?

```
<?php

    $var = 5;
?>

    $var = 10;<br />
```

The variable \$var has a value of:

```
<?=$var?><br />
```

Is this a valid script? Yes, the output would be the following:

```
$var = 10;
```

The variable \$var has a value of: 5

Notice that with the second assignment of \$var, when we attempt to change the value from 5 to 10, it has no effect because it is not enclosed within valid PHP code-block syntax. So, instead of being processed, it is simply displayed to the web browser.

Embedded conditionals

What if we wanted to display HTML only under specific conditions? For example, we only want PHP to display a navigation menu if the user is validated -- how could we do this? Using what we already know, we could simply put all the code for the HTML menu within echo statements (paying careful attention to add a backslash any time we need to output a quote for our HTML to the web browser) but this method can quickly become cumbersome and difficult to follow. Instead, to deal with situations like this, we will introduce a new flavor of our code-block syntax:

```
<?php if(conditions) { ?>
... HTML CODE ...
<?php } ?>
```

Although this may be confusing, remember how PHP will process this code. To start, it will evaluate the first line of a normal if statement and then begin a code block. Then, we turn off PHP parser and jump into normal HTML code (all of which PHP will simply output to the browser and ignore) until, finally, we return to PHP code and close our if statement. The result of this technique is a way for us to control regular and standard HTML with nearly no intrusion by PHP into the syntax. Although the above example works, a special syntax is provided for instances where PHP is being used simply to control the output of standard HTML code:

```
<?php if(conditions): ?>
... HTML CODE ...
<?php endif; ?>
```

This syntax is identical in function to the original example provided.

8) Using PHP and MySQL, develop a program to accept book information viz.

HANDLING DATABASE CONNECTIONS

. Definitions

Database: A *database* is an electronic file cabinet that stores information in an organized manner so that we can find the necessary information when we need it.

Database Management System: A set of programs that is used to access the information stored in the database. The most popular kind of DBMS is the Relational DBMS (RDBMS).

Database structure:

A database consists of two parts: one the structure of the database and the other the data itself.

In a relational DBMS, the structure consists of collection of tables and the collection as a whole form the database.

Each table consists of a set of rows and columns. Each row has complete information about an entity and different rows correspond to different entities. (Entities are objects that have certain well defined characteristic in the real world)

The columns contain the characteristics/information of the entity. Each column corresponds to different characteristic.

Using PHP with database

Most database software understands SQL (Structured Query Language), a computer language is use to communicate with a database. An SQL statement, called a query, is sent to the RDBMS that tells it what to do. SQL queries can instruct the RDBMS to create a database, create tables in a database, store data, retrieve data, delete data, and perform many other actions.

PHP communicates with databases by using functions designed specifically to interact with databases. PHP includes a set of functions for each database it supports. PHP constructs SQL queries as strings and sends the queries to the database with the use of specific functions provided for that purpose.

The descriptions given below assume the use of MYSQL database server.

Steps for interacting with a database

Connecting to a database:

The PHP script first should establish a connection with the database server before it can do any further interaction. The `mysql_connect()` function is used for that purpose.

`$connect=mysql_connect($hostname, $user, $password)` function is used to connect to the database.

`$hostname`-This variable gives the location of the database which is the name of the computer on which the database is located. This name can be a domain name line `mycompany.com` or an IP address. If the database is on the same computer as the PHP, then `$hostname="localhost"`

`$user`-This is the user name that the database administrator sets up for using the database.

`$password`-This is the password for the corresponding `$user`

`$connect` is a variable that holds the handle to the database. This handle is used in other functions while communicating with the database.

Selecting a database:

A particular database must be selected before giving queries to it. The name of the database to be used and a connection handle are needed to select the database.

`mysql_select_db("Database Name", $connect);` selects a database

"Database Name" – Name of the database

`$connect` – Connection handle returned from the `mysql_connect(...)` function

If a new database has to be created SQL query is used to create the database.

After creating the database, it should be selected.

Querying the database:

After PHP has established a connection to the database, we can perform whatever action we desire, such as get data, change data, or insert new data. The SQL query tells the database what action we want to perform.

The function `$res=mysql_query($qry)` is used to query the database. Here `$qry` is the query string containing the SQL query.

`$res` is the data returned by the query. If the query does not return data, then on successful execution of the query `TRUE` is returned. If the query does not execute successfully, then `FALSE` is returned.

Database queries are executed on tables in the database. If there are no tables and the database is a new database, then tables must be created, populated with values before they can be queried.

Processing the data returned by the query

The returned data by a SQL query will be stored as a table of rows and columns and the `$res` returned by `mysql_query()` is the pointer to that table.

The returned data may need some processing to display as a HTML table. The

mysql\_fetch\_assoc() function is used to process the data row by row. When there are no more rows in the table, this function returns FALSE.

A while loop can be used with this function to process all the rows.

```
while($row = mysql_fetch_assoc($res))
{
    Process the row
}
```

Closing a database connection:

The connection to the database can be terminated using the mysql\_close(\$connect) function where \$connect is the connection variable obtained while connecting to the database using mysql\_connect();

Handling Errors

All the mysql related functions return TRUE on the successful execution of the function or return FALSE if the function could not be executed successfully.

After a function returns FALSE, it is not desirable to continue the execution of the PHP script.

A function die() is provided for this purpose. This function is used in the following manner:

f() or die();

If the function f() returns TRUE, execution of the script continues. If f() returns false, die() is executed and the script stops executing. die() is used in conjunction with mysql\_error(). mysql\_error() gives the error message returned by MYSQL database server which is the cause for the unsuccessful execution of the mysql related functions.

Examples:

1. Connecting to a database:

```
<?php
$host="localhost";
$user="root";
$password="";
$con = mysql_connect($host,$user,$password);
if($con)
    echo "Connected successfully to MYSQL\n";
else
    echo "Unable to connect";
?>
```

2. Creating a database

```
<?php
$createdb = mysql_query("CREATE DATABASE IF NOT EXISTS studentdb")
or die(mysql_error());
?>
```

Here, studentdb is the name of the database to be created and the string within mysql\_query() is the SQL query string.

3. Selecting a database

```
<?php
$con = mysql_connect($host,$user,$password);
if($con)
    echo "Connected successfully to MYSQL\n";
else
    echo "Unable to connect";
mysql_select_db("studentdb",$con)
or die("Cant select student database");
?>
```

4. Creating a table

```
<?php
$studtable = "CREATE TABLE student (
    stud_id int NOT NULL auto_increment,
    stud_name varchar(200) NOT NULL,
    stud_dob date NOT NULL,
    stud_course varchar(50) NOT NULL,
    stud_addr text NOT NULL,
    PRIMARY KEY (stud_id)
```



```

)";
$res = mysql_query($studtable)
or die(mysql_error());
?>
5. Populating a database with values
$stud_data = "INSERT INTO student (stud_name, stud_dob, stud_course, stud_addr)".
"VALUES ('Vidya','1980-01-11','MTEch','Trichy'),".
"('Rani','1978-02-24','MSIT','Chennai'),".
"('Ramesh','1982-05-22','MCA','Mumbai'),".
"('Shankar','1983-07-12','BCA','Madurai)";
$res = mysql_query($stud_data)
or die(mysql_error());
6. Processing data from a table
$stud_data = "SELECT * FROM student";
$res = mysql_query($stud_data)
or die(mysql_error());
echo "<table border = '1'>";
echo "<th> Student ID </th> <th> Name </th> <th> Date of Birth </th>";
echo "<th> Course </th> <th> Address </th>";
while($row = mysql_fetch_assoc($res))
{
echo "<tr>";
foreach($row as $value)
{
echo "<td> $value </td>";
}
echo "</tr>";
}
echo "</table>";

```

#### 8) Develop a web application for Airline Reservation System using PHP

```

<?php
$myacno= $_POST['acc_no'];
$mytitle = $_POST['tit'];
$myauth = $_POST['author'];
$myedition = $_POST['edition'];
if (empty($myacno) || empty($mytitle) || empty($myauth) || empty($myedition))

{
echo "<script>alert('Empty fields not allowed')</script>";
echo "<script>location.href='admin.php'</script>";
die();
}

$con=mysql_connect('localhost','root','');
mysql_select_db("csv_db",$con);
$result= mysql_query("select * from table1 where (edition='$myedition' && tit ='$mytitle' &&
author='$myauth') || (accno='$myacno')");
$rows=mysql_num_rows($result);
if ($rows>=1)
{
echo "Book Id or Bood is already exist";
header("Location:admin.php");
die();
}
mysql_query("insert into table1
(accno,tit,author,edition)VALUES('$myacno','$mytitle','$myauth','$myedition')");
echo "<script>alert('Record Added')</script>";
header("Location:admin.php");
?>

```



SEARCH CODE

```

<html><body>
<center>
<a href="admin.php"></a>
<a href="view.php"></a>
<form method = "post" action = "bs.php">
  <div style = 'top: 200; left: 560; position: absolute; z-index:1;'>
    <input type="text" name = "wala" value="name of book..." onblur="if (this.value == '') { this.value=
'name of book...'; }" onfocus="if (this.value == 'name of book...') { this.value= ''; }"/>
    <button name = "search" style="background-color:transparent;"> Search</button>

  </div>
</form>
</BODY>
</HTML>

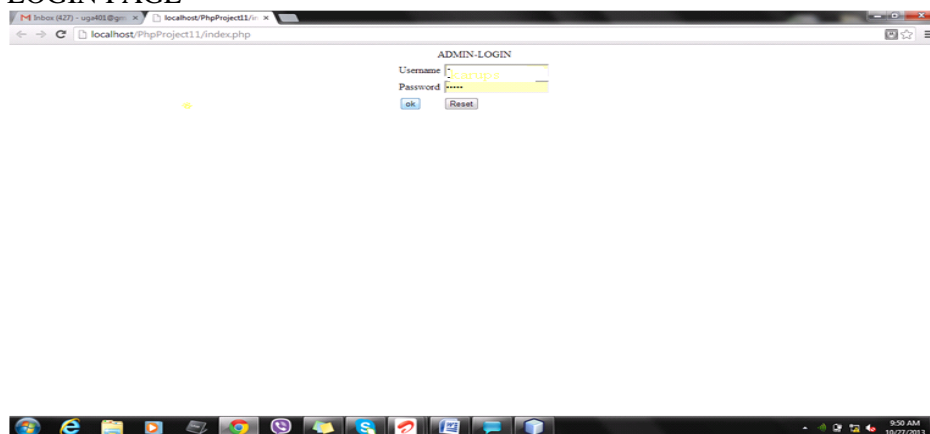
```

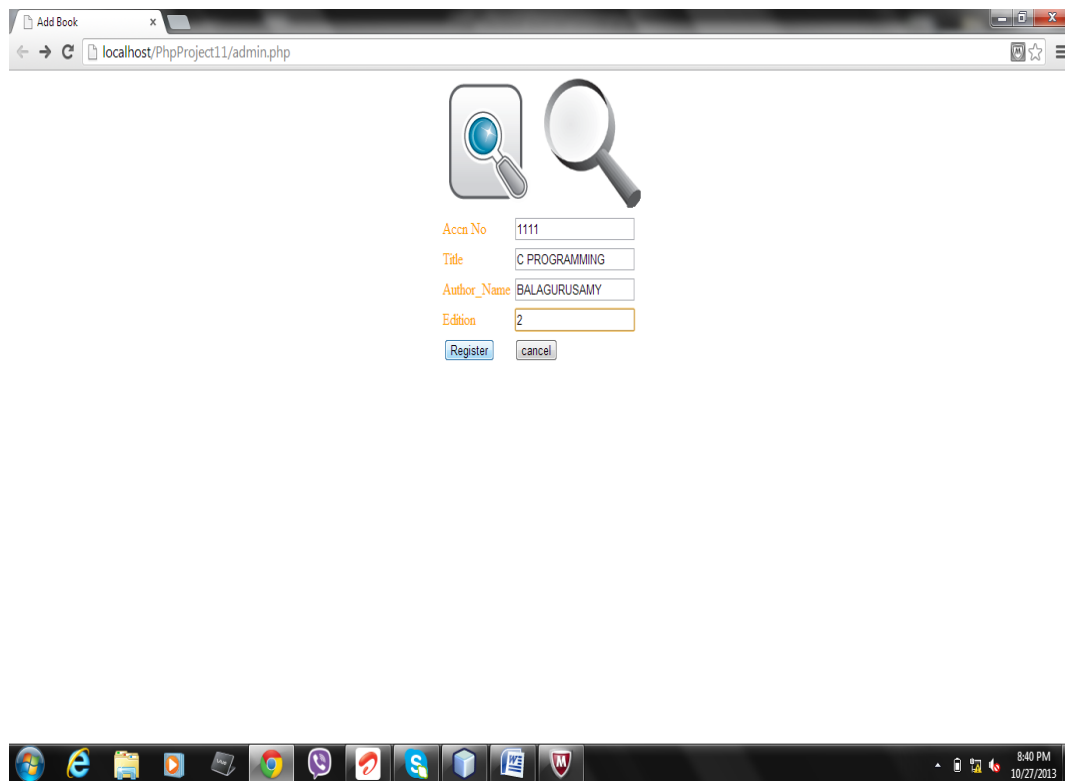
EDIT CODE

```

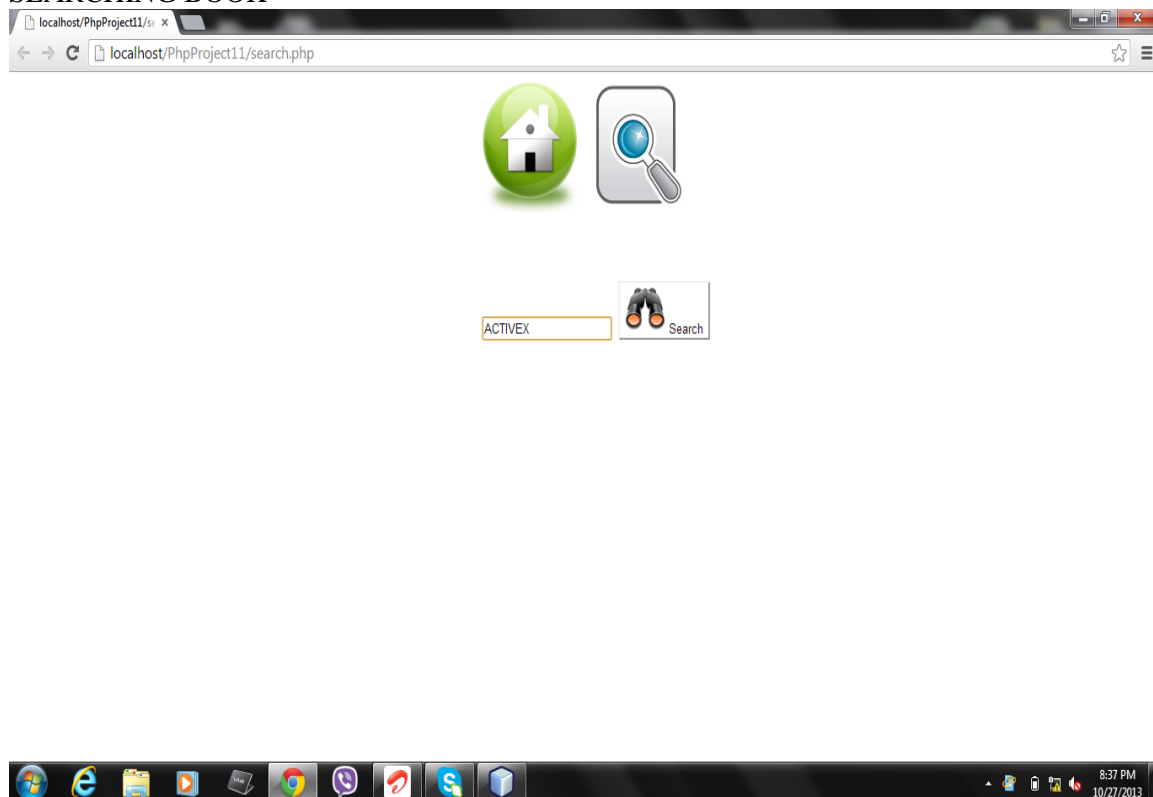
<?php
$mykey1=$_REQUEST['key1'];
$mykey2=$_REQUEST['key2'];
$mykey3=$_REQUEST['key3'];
$mykey4=$_REQUEST['key4'];
echo "<form action='update.php?key1=" . $mykey1 . "' method='POST'>";
ECHO "<CENTER>EDIT-INFORMATION</CENTER>";
echo "<center><table>";
echo "<TR><TD>Accn No</TD><TD><INPUT TYPE=text name='accno'
value=$mykey1></TD></TR>";
echo "<TR><TD>Title</TD><TD><INPUT TYPE=text name='tit' value=$mykey2 ></TD></TR>";
echo "<TR><TD>Author_Name</TD><TD><INPUT TYPE=text name='author' value=$mykey3
></TD></TR>";
echo "<TR><TD>Edition</TD><TD><INPUT TYPE=text name='edition'
value=$mykey4></TD></TR>";
echo "<TR><TD><INPUT TYPE=submit value=UPDATE></TD><TD><INPUT TYPE=reset
value='cancel'></TD></TR></table>";
?>

```

**Output:****LOGIN PAGE**



## SEARCHING BOOK

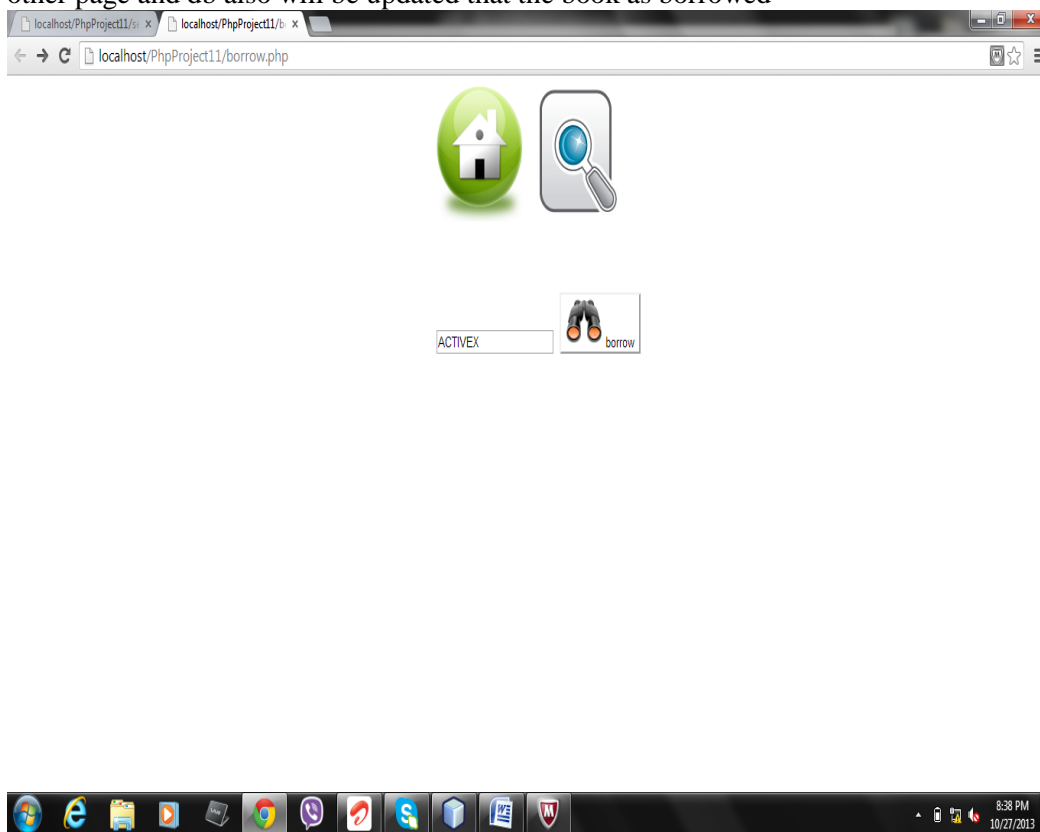


**VIEW-** by using view u can view the entire db

Here u have two symbols at the side of each record for deleting and editing purpose .By click the icon u can perform the operation.



BOOK BORROW-in this form by clicking borrow u get a confirmation msg that u borrowed by linking to other page and db also will be updated that the book as borrowed



#### 9) Write PHP code to find factorial of a number.(MAY 2015)

```

html>
<body>
    <title>Factorial Program in PHP</title>
    <!-- form post method -->
    <form action="" method="post">
        <label>Enter Number to calculate Factorial</label> <input type="text" name="number"
size="2" />
    </form>
</body>
</html>

<?php
    //check for post value if there then allow inside
    if($_POST){

```

```

$fact = 1;

//get the value from input text box 'number'
$number = $_POST['number'];

echo "Factorial of $number:<br><br>";

//loop till the iterator $i equals to $number
for ($i = 1; $i <= $number; $i++){

    //formula to calculate factorial is to
    //multiply the iterator $i value with $fact value.
    $fact = $fact * $i;

    //print $fact to show the factorial pyramid
    //or put this line out of this 'for loop' to show only the total value
    print $fact . "<br>";

}

?>

```

**10)** How to read from a web page? Explain with example.(May 2016)

Refer previous question.

**11) i)** Write a PHP recursive function to generate a Fibonacci series of n terms(May 2016)

Fibonacci Series

Fibonacci series is the one in which you will get your next term by adding previous two numbers.

For example,

1. 0 1 1 2 3 5 8 13 21 34
2. Here,  $0 + 1 = 1$
3.        $1 + 1 = 2$
4.        $3 + 2 = 5$

and so on.

Logic:

- Initializing first and second number as 0 and 1.
- Print first and second number.
- From next number, start your loop. So third number will be the sum of the first two numbers.

Fibonacci series using Recursive function

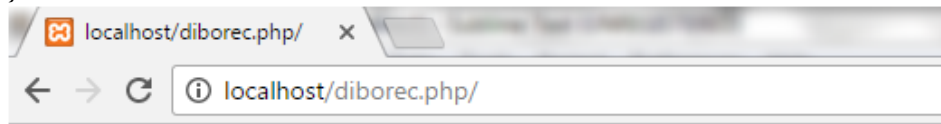
Recursion is a phenomenon in which the recursion function calls itself until the base condition is reached.

```

<?php
/* Print fibonacci series upto 12 elements. */
$num = 12;
echo "<h3>Fibonacci series using recursive function:</h3>";
echo "\n";
/* Recursive function for fibonacci series. */
function series($num){
    if($num == 0){
        return 0;
    }else if( $num == 1){
        return 1;
    } else {
        return (series($num-1) + series($num-2));
    }
}
/* Call Function. */
for ($i = 0; $i < $num; $i++){

```

```
echo series($i);
echo "\n";
}
```



## Fibonacci series using recursive function:

0 1 1 2 3 5 8 13 21 34 55 89

ii) Create a form for student information. Write JavaScript code to find Total, Average, Result & Grade. Refer the question no.9.

- 12) (i) Write a JavaScript to check whether a given number is prime or not. (May 2016)  
Prime number in PHP

Example:

Here is the Program to list the first 15 prime numbers.

```
<?php
$count = 0;
$num = 2;
while ($count < 15 )
{
    $div_count=0;
    for ( $i=1; $i<=$num; $i++)
    {
        if (($num%$i)==0)
        {
            $div_count++;
        }
    }
    if ($div_count<3)
    {
        echo $num." , ";
        $count=$count+1;
    }
    $num=$num+1;
}
?>
```

Output:

Prime Number using Form in PHP

Example:

We'll show a form, which will check whether a number is prime or not.

```
<form method="post">
Enter a Number: <input type="text" name="input"><br><br>
<input type="submit" name="submit" value="Submit">
</form>
<?php
if($_POST)
{
    $input=$_POST['input'];
    for ($i = 2; $i <= $input-1; $i++) {
```

```
    if ($input % $i == 0) {  
        $value= True;  
    }  
}  
if (isset($value) && $value) {  
    echo 'The Number '. $input . ' is not prime';  
} else {  
    echo 'The Number '. $input . ' is prime';  
}  
}  
?>
```

- (ii) Write a HTML code to develop a website for University. Provide the links to department, Examinations and Results with the use of suitable tags.  
Refer previous answers