

ATME COLLEGE OF ENGINEERING

13th KM Stone, Bannur Road, Mysore - 560 028



A T M E
College of Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(ACADEMIC YEAR 2019-20)

LABORATORY MANUAL

**SUBJECT: WEB TECHNOLOGY LABORATORY WITH
MINI PROJECT**

SUB CODE: 15CSL77

SEMESTER: VII

**As per Choice Based Credit System (CBCS) scheme
(Effective from the academic year 2016 -2017)**

INSTITUTIONAL MISSION AND VISION

Objectives

- To provide quality education and groom top-notch professionals, entrepreneurs and leaders for different fields of engineering, technology and management.
- To open a Training-R & D-Design-Consultancy cell in each department, gradually introduce doctoral and postdoctoral programs, encourage basic & applied research in areas of social relevance, and develop the institute as a center of excellence.
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.
- To cultivate strong community relationships and involve the students and the staff in local community service.
- To constantly enhance the value of the educational inputs with the participation of students, faculty, parents and industry.

Vision

- Development of academically excellent, culturally vibrant, socially responsible and globally competent human resources.

Mission.

- To keep pace with advancements in knowledge and make the students competitive and capable at the global level.
- To create an environment for the students to acquire the right physical, intellectual, emotional and moral foundations and shine as torch bearers of tomorrow's society.
- To strive to attain ever-higher benchmarks of educational excellence.

Department of Computer Science & Engineering

Vision of the Department

- To develop highly talented individuals in Computer Science and Engineering to deal with real world challenges in industry, education, research and society.

Mission of the Department

- To inculcate professional behavior, strong ethical values, innovative research capabilities and leadership abilities in the young minds & to provide a teaching environment that emphasizes depth, originality and critical thinking.
- Motivate students to put their thoughts and ideas adoptable by industry or to pursue higher studies leading to research.

Program Outcomes (POs)

Engineering Graduates will be able to:

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives (PEO'S):

1. Empower students with a strong basis in the mathematical, scientific and engineering fundamentals to solve computational problems and to prepare them for employment, higher learning and R&D.
2. Gain technical knowledge, skills and awareness of current technologies of computer science engineering and to develop an ability to design and provide novel engineering solutions for software/hardware problems through entrepreneurial skills.
3. Exposure to emerging technologies and work in teams on interdisciplinary projects with effective communication skills and leadership qualities.

4. Ability to function ethically and responsibly in a rapidly changing environment by Applying innovative ideas in the latest technology, to become effective professionals in Computer Science to bear a life-long career in related areas.

Program Specific Outcomes (PSOs)

1. Demonstrate understanding of the principles and working of the hardware and software aspects of Embedded Systems.
2. Use professional Engineering practices, strategies and tactics for the development, implementation and maintenance of software.
3. Provide effective and efficient real time solutions using acquired knowledge in various domains.

WEB TECHNOLOGY LABORATORY

WITH MINI PROJECT

[As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017)

Subject Code	:	15CSL77	I.A. Marks	:	20
Hours/Week	:	01I + 02P	Exam Hours	:	03
Total Hours	:	40	Exam Marks	:	80

CREDITS – 02

Course objectives: This course will enable students to

1. Design and develop static and dynamic web pages.
2. Familiarize with Client-Side Programming, Server-Side Programming, Active server Pages.
3. Learn Database Connectivity to web applications.

Description (If any): NIL

Lab Experiments:

PART A

1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.
2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
3. Write a JavaScript code that displays text “TEXT-GROWING” with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays “TEXT-SHRINKING” in BLUE color. Then the font size decreases to 5pt.
4. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:
 - a) Parameter : A string
 - b) Output: The position in the string of the left-most vowel
 - c) Parameter : A number
 - d) Output: The number with its digits in the reverse order
5. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the

- College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
6. Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
 7. Write a PHP program to display a digital clock which displays the current time of the server.
 8. Write the PHP programs to do the following:
 - a) Implement simple calculator operations.
 - b) Find the transpose of a matrix
 - c) Multiplication of two matrices.
 - d) Addition of two matrices.
 9. Write a PHP program named states.py that declares variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following:
 - a) Search for a word in variable states that ends in xas. Store this word in element 0 of a list named states List. Find the transpose of a matrix
 - b) Search for a word in states that begins with k and ends in s. Perform a case-insensitive comparison. [Note: Passing re.I as a second parameter to method compile performs a case-insensitive comparison.] Store this word in element 1 of states List. Addition of two matrices.
 - c) Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.
 - d) Search for a word in states that ends in a. Store this word in element 3 of the list.
 10. Write a PHP program to sort the student records which are stored in the database using selection sort.

Study Experiment / Project:

Develop a web application project using the languages and concepts learnt in the theory and exercises listed in part A with a good look and feel effects. You can use any web technologies and frameworks and databases.

Note:

1. In the examination each student picks one question from part A.
2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.
3. The team must submit a brief project report (15-20 pages) that must include the following
 - a. Introduction
 - b. Requirement Analysis
 - c. Software Requirement Specification
 - d. Analysis and Design
 - e. Implementation
 - f. Testing

Course outcomes: The students should be able to:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.
- Learn how to link and publish web sites

Conduction of Practical Examination:

1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 30 Marks.
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
 - a) Part A: Procedure + Conduction + Viva: $10 + 35 + 5 = 50$ Marks
 - b) Part B: Demonstration + Report + Viva voce = $15 + 10 + 05 = 30$ Marks

Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

CONTENT LIST

SL.NO.	EXPERIMENT NAME	PAGE NO.
1.	Introduction	1
2.	Program 1 : JavaScript : Simple calculator	27
3.	Program 2 : JavaScript : Calculate squares and cubes of the numbers from 0 to 10	32
4.	Program 3 : JavaScript : TEXT-GROWING and TEXT-SHRINKING	35
5.	Program 4: HTML5 and JavaScript : a) position in the string of the left-most vowel b) number with its digits in the reverse order	39
6.	Program 5 : XML document to store information about a student	44
7.	Program 6: PHP: display the number of visitors visiting the web page.	48
8.	Program 7: PHP: display digital clock with current time of the server.	50
9.	Program 8: PHP: a) Implement simple calculator operations. b) Find the Transpose of a matrix, Multiplication of two matrices and addition of two matrices	52
10.	Program 9: PHP: program with variable states with value "Mississippi Alabama Texas Massachusetts Kansas"	60
11.	Program 10 : PHP : program to sort the student records using selection sort	64
12.	Viva Questions	71

CHAPTER 1: INTRODUCTION

1.1 Hyper Text Markup Language

- HTML stands for **H**yper**t**ext **M**arkup **L**anguage, and it is the most widely used language to write Web Page.
- **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus the link available on a webpage are called Hypertext.
- As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark up" a text document with tags that tell a Web browser how to structure it to display.
- Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.
- Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

Basic HTML Document

In its simplest form, following is an example of an HTML document:

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is document title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>Document content goes here.....</p>
  </body>
</html>
```

- Save it in an HTML file **test.htm** using your favorite text editor. Finally open it using a web browser like Internet Explorer or Google Chrome, or Firefox etc.
- It must show the following output:

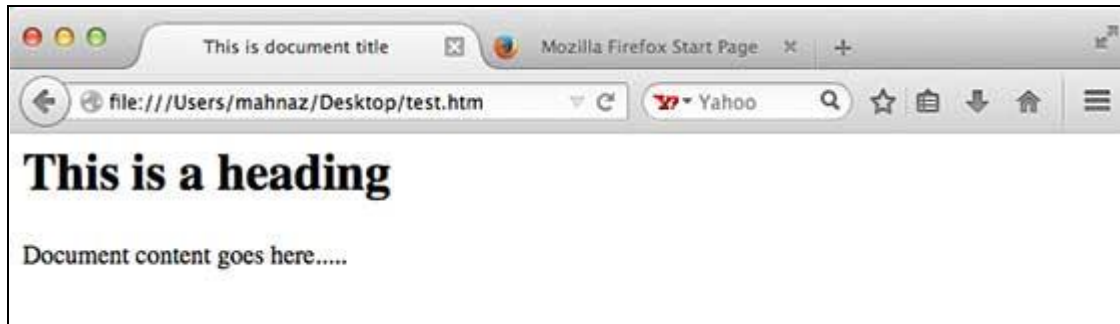


Fig 1: Sample Html Output

HTML Tags

- HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**.
- Except few tags, most of the tags have their corresponding closing tags.
- For example **<html>** has its closing tag **</html>** and **<body>** tag has its closing tag **</body>** tag etc.
- Above example of HTML document uses following tags:

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.
<h1>	This tag represents the heading.

<code><p></code>	This tag represents a paragraph.
------------------------	----------------------------------

World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML 4.

HTML Document Structure

A typical HTML document will have following structure:

Document declaration tag

```
<html>
```

```
  <head>
```

```
    Document header related tags
```

```
  </head>
```

```
  <body>
```

```
    Document body related tags
```

```
  </body>
```

```
</html>
```

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration:

```
<!DOCTYPE html>
```

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used.

1. 2 Cascading Style sheets

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Solved a Big Problem

- HTML was NEVER intended to contain tags for formatting a web page!
- HTML was created to **describe the content** of a web page, like:

`<h1>This is a heading</h1>`

`<p>This is a paragraph.</p>`

- When tags like ``, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.
- To solve this problem, the World Wide Web Consortium (W3C) created CSS.
- CSS removed the style formatting from the HTML page!

CSS Saves a Lot of Work!

- The style definitions are normally saved in external .css files.
- With an external stylesheet file, you can change the look of an entire website by changing just one file!

CSS Syntax

- A CSS rule-set consists of a selector and a declaration block:

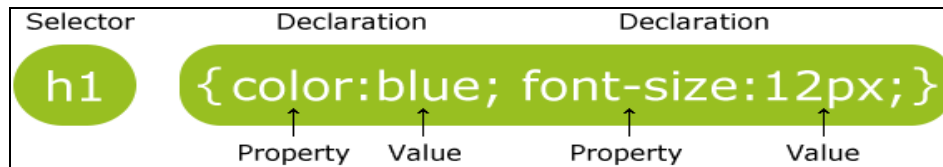


Fig 2: CSS rule-set

- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

In the following example all <p> elements will be center-aligned, with a red text color:Example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <style>
```

```
      p {
```

```
        color: red;
```

```
        text-align: center;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <p>Hello World!</p>
```

```
    <p>This paragraph is styled with CSS.</p>
```

```
  </body>
```

```
</html>
```

CSS Selectors

- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

The element Selector

- The element selector selects elements based on the element name.
- You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):
- Example

```
p {  
  
    text-align: center;  
    color: red;  
}
```

The id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element should be unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- The style rule below will be applied to the HTML element with id="para1":
- Example

```
#para1  
  
{  
    text-align: center;  
    color: red;  
}
```

- **Note:** An id name cannot start with a number!

The class Selector

- The class selector selects elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the name of the class.

In the example below, all HTML elements with class="center" will be red

- and center-aligned:
- Example

```
.center  
  
{  
    text-align: center;  
    color: red;  
}
```

- You can also specify that only specific HTML elements should be affected by a class.
- In the example below, only <p> elements with class="center" will be center-aligned:
- Example

```
p.center {  
    text-align: center;  
    color: red;  
}
```

- HTML elements can also refer to more than one class.
- In the example below, the <p> element will be styled according to class="center" and to class="large":
- Example

```
<p class="center large">This paragraph refers to two classes.</p>
```

- **Note:** A class name cannot start with a number!

Grouping Selectors

- If you have elements with the same style definitions, like this:

```
h1 {  
    text-align: center;  
    color: red;  
}
```

```
h2 {  
    text-align: center;  
    color: red;  
}
```

```
p {  
    text-align: center;  
    color: red;  
}
```

- It will be better to group the selectors, to minimize the code.
- To group selectors, separate each selector with a comma.
- In the example below we have grouped the selectors from the code above:
- Example

```
h1, h2, p { text-align: center;  
            color: red;  
}
```

CSS Comments

- Comments are used to explain the code, and may help when you edit the source code at a later date.
- Comments are ignored by browsers.
- A CSS comment starts with `/*` and ends with `*/`. Comments can also span multiple lines:
- Example

```
p {  
    color: red;  
    /* This is a single-line comment */  
    text-align: center;  
}  
  
/* This is  
a multi-line  
comment */
```

1.3 JavaScript

- JavaScript is a lightweight, interpreted programming language.
- It is designed for creating network-centric applications. It is complimentary to and integrated with Java.
- JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

What is JavaScript

- JavaScript is a dynamic computer programming language.
- It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages.
- It is an interpreted programming language with object-oriented capabilities.

Client-side JavaScript

- Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.
- It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over

- traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.
- The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.
- JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript

- **Less server interaction** – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** – They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multithreading or multiprocessor capabilities.
- JavaScript is lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

To execute JavaScript

Try following example –

```
<html>
<body>

    <script language="javascript" type="text/javascript">
        <!--
            document.write("Hello World!")
        //-->
    </script>

</body>
```

</html>

output:

- Hello World

1.4 Extended Markup Language

- XML stands for EXtensible Markup Language
- XML is a markup language much like HTML
- XML was designed to store and transport data
- XML was designed to be self-descriptive
- XML is a W3C Recommendation

XML Does Not DO Anything

- Maybe it is a little hard to understand, but XML does not DO anything.
- This note is a note to Tove, from Jani, stored as XML:

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

- The note is quite self-descriptive. It has sender and receiver information. It also has a heading and a message body.

But still, this XML document does not DO anything. XML is just information wrapped in tags. Someone must write a piece of software to send, receive,

- store, or display it:

Difference between XML and HTML

XML and HTML were designed with different goals:

- XML was designed to carry data - with focus on what data is
- HTML was designed to display data - with focus on how data looks
- XML tags are not predefined like HTML tags are

XML Does Not Use Predefined Tags

- The XML language has no predefined tags.
- The tags in the example above (like <to> and <from>) are not defined in any XML standard. These tags are "invented" by the author of the XML document.
- HTML works with predefined tags like <p>, <h1>, <table>, etc.
- With XML, the author must define both the tags and the document structure.

XML is Extensible

- Most XML applications will work as expected even if new data is added (or removed).
- Imagine an application designed to display the original version of note.xml (<to> <from> <heading> <data>).
- Then imagine a newer version of note.xml with added <date> and <hour> elements, and a removed <heading>.
- The way XML is constructed, :

```
<note>
  <date>2015-09-01</date>
  <hour>08:30</hour>
  <to>Tove</to>
  <from>Jani</from>
  <body>Don't forget me this weekend!</body>
</note>
```

Note

To: Tove

From: Jani

Head: Reminder

Don't forget me this weekend!

Note

To: Tove

From: Jani

1.5 HTML5

What is New in HTML5?

The DOCTYPE declaration for HTML5 is very simple:

```
<!DOCTYPE html>
```

The character encoding (charset) declaration is also very simple:

```
<meta charset="UTF-8">
```

HTML5 Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Title of the document</title>
```

```
</head>
```

```
<body>
```

Content of the document.....

```
</body>
```

```
</html>
```

New HTML5 Elements

The most interesting new HTML5 elements are:

New semantic elements like <header>, <footer>, <article>, and <section>.

New attributes of form elements like number, date, time, calendar, and range.

New graphic elements: <svg> and <canvas>.

New multimedia elements: <audio> and <video>.

New HTML5 API's (Application Programming Interfaces)

The most interesting new API's in HTML5 are:

HTML Geolocation

HTML Drag and Drop

HTML Local Storage

HTML Application Cache

HTML Web Workers

HTML SSE

Removed Elements in HTML5

The following HTML4 elements have been removed in HTML5:

Removed Element	Use Instead
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	
	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS, <s>, or
<tt>	CSS

Define Semantic Elements as Block Elements

HTML5 defines eight new semantic elements. All these are block-level elements.

To secure correct behavior in older browsers, you can set the CSS display property for these HTML elements to **block**:

```
header, section, footer, aside, nav, main, article, figure {  
    display: block;  
}
```

Add New Elements to HTML

You can also add new elements to an HTML page with a browser trick.

This example adds a new element called <myHero> to an HTML page, and defines a style for it:

Example :

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<script>document.createElement("myHero")</script>  
  
<style>  
myHero {  
    display: block;  
    background-color: #dddddd;  
    padding: 50px;  
    font-size: 30px;  
}  
</style>  
</head>  
  
<body>
```

```
<h1>A Heading</h1>
<myHero>My Hero Element</myHero>
</body>
</html>
```

New Semantic/Structural Elements

HTML5 offers new elements for better document structure:

Tag	Description
<article>	Defines an article in a document
<aside>	Defines content aside from the page content
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<main>	Defines the main content of a document
<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<progress>	Represents the progress of a task
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)

<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time
<wbr>	Defines a possible line-break

New Form Elements

Tag	Description
<datalist>	Specifies a list of pre-defined options for input controls
<output>	Defines the result of a calculation

New Input Types

New Input Types	New Input Attributes
Color	autocomplete
Date	autofocus
Datetime	form
datetime-local	formaction
email	formenctype
month	formmethod
number	formnovalidate
range	formtarget
search	height and width
tel	list
time	min and max
url	multiple
week	pattern (regexp)

HTML5 - New Attribute Syntax

HTML5 allows four different syntaxes for attributes.

This example demonstrates the different syntaxes used in an `<input>` tag:

Type	Example
Empty	<code><input type="text" value="John" disabled></code>
Unquoted	<code><input type="text" value=John></code>
Double-quoted	<code><input type="text" value="John Doe"></code>
Single-quoted	<code><input type="text" value='John Doe'></code>

In HTML5, all four syntaxes may be used, depending on what is needed for the attribute.

1.6 PHP

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Common uses of PHP

PHP performs system functions, i.e. from files on a system it can create,

- open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user.
- You add, delete, modify elements within your database thru PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible –

- Simplicity
- Efficiency
- Security

- Flexibility
- Familiarity

"Hello World" Script in PHP

- To create simple PHP scripts. "Hello, World!" script.

Since PHP is embedded in HTML. That means that we will normally

- have PHP statements like this –

```
<html>

  <head>
    <title>Hello World</title>
  </head>

  <body>
    <?php echo "Hello, World!";?>
  </body>

</html>
```

It will produce following result –

Hello, World!

- If you examine the HTML output of the above example, you'll notice that the PHP code is not present in the file sent from the server to your Web browser.
- All of the PHP present in the Web page is processed and stripped from the page; the only thing returned to the client from the Web server is pure HTML output.
- All PHP code must be included inside one of the three special markup tags that are recognised by the PHP Parser.

```
<?php PHP code goes here ?>
<?  PHP code goes here ?>
<script language="php"> PHP code goes here </script>
```

1.7 MYSQL

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications.

MySQL Database:

MySQL is a fast, easy-to-use RDBMS being used for many small and

- big businesses.

MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.

MySQL is a very powerful program in its own right. It handles a large

- subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

Execute simple SQL commands using MySQL Client:

You can connect to your MySQL server by using MySQL client using

- **mysql** command.
- At this moment, you do not need to give any password as by default it will be set to blank.

So just use following command

```
[root@host]# mysql
```

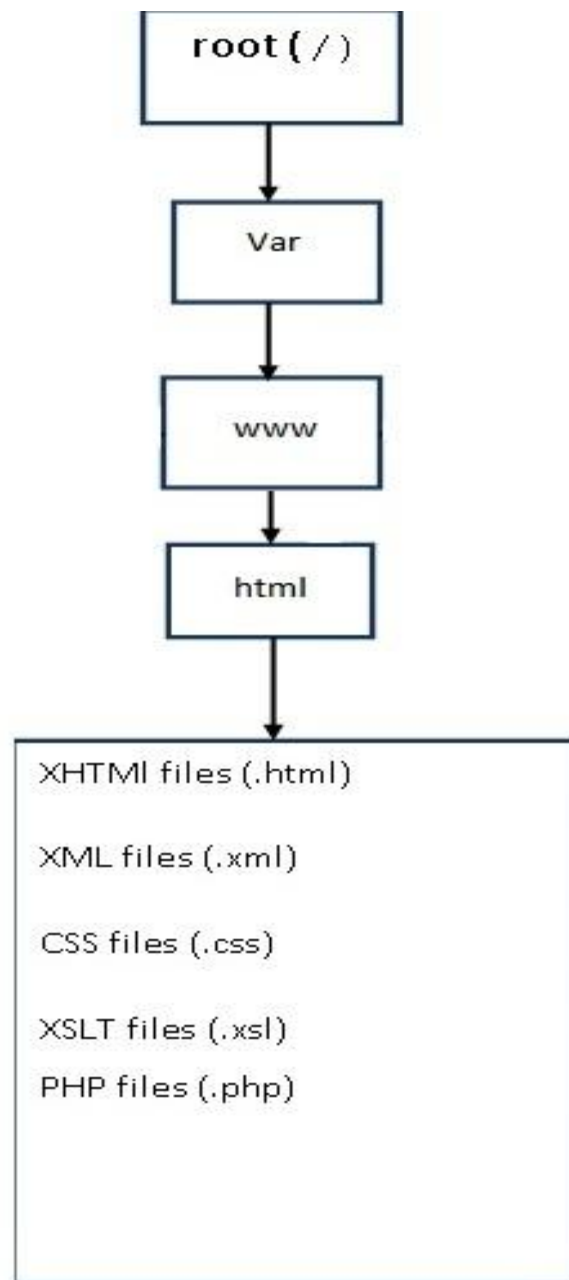
It should be rewarded with a mysql> prompt. Now, you are connected to the MySQL server and you can execute all the SQL command at mysql> prompt as follows:

```
mysql> SHOW DATABASES;
```

```
+-----+  
| Database |  
+-----+  
| mysql   |  
| test    |  
+-----+
```

```
2 rows in set (0.13 sec)
```

Location to store files used in Web Programming Laboratory :



Steps of executing program:

- i. Html, xml, css & php files should save in /var/www/html/ path directory.
- ii. Perl & cgi files should save in /var/www/cgi-bin/ path directory.
- iii. Run all files through browser using `http://localhost/filename.extension`
- iv. In perl program, you have to give file permission before execute perl program through browser.

For eg:- `root@localhost />chmod 777 filename.ext.`

Program 1:

1) Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.

Program Objective:

- To Design and develop static and dynamic web pages.

1.html

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

```
<head>
```

```
<style>
```

```
table, td, th
```

```
{
```

```
border: 1px solid black;
```

```
width: 33%;
```

```
text-align: center;
```

```
background-color: DarkGray;
```

```
border-collapse: collapse;
```

```
}
```

```
table { margin: auto; }
```

```
input { text-align: right; }
```

```
</style>

<script type="text/javascript">

function calc(clicked_id)

{

Var val1 = parseFloat(document.getElementById("value1").value);
var val2 = parseFloat(document.getElementById("value2").value);
if(isNaN(val1)||isNaN(val2))

alert("ENTER VALID NUMBER");

else if(clicked_id=="add")
document.getElementById("answer").value=val1+val2;
else if(clicked_id=="sub")
document.getElementById("answer").value=val1-val2;
else if(clicked_id=="mul")
document.getElementById("answer").value=val1*val2;
else if(clicked_id=="div")
document.getElementById("answer").value=val1/val2;

}
function cls()
{
value1.value="0";

value2.value="0";

answer.value="";
}
</script>
</head>
```

```
<body>
<table>
<tr><th colspan="4"> SIMPLE CALCULATOR </th></tr>
<tr><td>value1</td><td><input type="text" id="value1" value="0"/></td>
<td>value2</td><td><input type="text" id="value2" value="0"/> </td></tr>
<tr><td><input type="button" value="Addition" id = "add"
onclick="calc(this.id)"/></td>

<td><input type="button" value="Subtraction" id = "sub"

onclick="calc(this.id)"/></td>

<td><input type="button" value="Multiplication" id = "mul"

onclick="calc(this.id)"/></td>

<td><input type="button" value="Division" id = "div"

onclick="calc(this.id)"/></td></tr>

<tr><td>Answer:</td><td> <input type="text" id="answer" value=""

disabled/></td>

<td colspan="2"><input type="button" value="CLEAR ALL"

onclick="cls()"/></td> </tr>

</table>

</body>

</html>
```

Output:

SIMPLE CALCULATOR			
value1	<input type="text"/>	value2	<input type="text"/>
Addition	Subtraction	Multiplication	Division
Answer:	<input type="text"/>	CLEAR ALL	

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Test Cases :

Test No.	Input Parameters	Expected Output	Obtained Output	Remarks
1	value1=50.56 value2=24.39	Addition =74.95 Subtraction =26.17 Multiplication=1233.1584 Division=2.0729807298	Addition =74.95 Subtraction =26.17 Multiplication=1233.1584 Division=2.0729807298	PASS
2	value1= 0 value2= 45	Addition =45 Subtraction =-45 Multiplication=0 Division=0	Addition =45 Subtraction =-45 Multiplication=0 Division=0	PASS

3	value1= 45 value2= 0	Addition =45 Subtraction =45 Multiplication=0 Division=Infinity	Addition =45 Subtraction =45 Multiplication=0 Division=Infinity	PASS
4	value1 = abc value2 = 23	ENTER VALID NUMBER	ENTER VALID NUMBER	PASS
5	value1 = 50 value2 =xyz	ENTER VALID NUMBER	ENTER VALID NUMBER	PASS

Viva Questions:

1. What is HTML?
2. What is Tag?
3. How can I include comments in HTML?
4. How do I align a table to the right (or left)?
5. What is JavaScript?
6. How is JavaScript different from Java?

Program 2

2) Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.

Program Objective:

- To Design and develop static and dynamic web pages.

2.html

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

```
<head>
```

```
<style> table,tr, td
```

```
{
```

```
border: solid black;
```

```
width: 33%;
```

```
text-align: center;
```

```
border-collapse: collapse; background-color:lightblue;
```

```
}
```

```
table { margin: auto; }
```

```
</style>
```

```
<script>
```

```
document.write( "<table><tr><thcolspan='3'> NUMBERS FROM 0 TO 10 WITH  
THEIR SQUARES AND CUBES </th></tr>" );
```

```
document.write( "<tr><td>Number</td><td>Square</td><td>Cube</td></tr>" );
```

```
for(var n=0; n<=10; n++)  
  
{  
  
document.write( "<tr><td>" + n + "</td><td>" + n*n + "</td><td>" + n*n*n +  
"</td></tr>" );  
  
}  
  
document.write( "</table>" );  
  
</script>  
  
</head>  
  
</html>
```

Output:

NUMBERS FROM 0 TO 10 WITH THEIR SQUARES AND CUBES		
Number	Square	Cube
0	0	0
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Viva Questions:

1. How to create a button which acts like a link?
2. What is difference between HTML and XHTML?
3. What are logical and physical tags in HTML?
4. What is a Hypertext link?
5. What is a DOCTYPE? Which one do I use?
6. What is the official JavaScript website?

Program 3:

3) Write a JavaScript code that displays text “TEXT-GROWING” with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays “TEXT-SHRINKING” in BLUE color. Then the font size decreases to 5pt.

Program Objectives:

- To Design and develop static and dynamic web pages.

3a.html

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {
```

```
position: absolute;
```

```
top: 50%;
```

```
left: 50%;
```

```
transform: translate(-50%, -50%);
```

```
}
```

```
</style>
```

```
</head>
<body>

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

<script>

var var1 = setInterval(inTimer, 1000);

var fs = 5;

var ids = document.getElementById("demo");

function inTimer() {

ids.innerHTML = 'TEXT GROWING'; ids.setAttribute('style', "font-size: " + fs +
"px; color: red"); fs += 5;

if(fs >= 50 ){

clearInterval(var1);

var2 = setInterval(deTimer, 1000);

}

}

function deTimer() {

fs -= 5;

ids.innerHTML = 'TEXT SHRINKING'; ids.setAttribute('style', "font-size: " + fs +
"px; color: blue");
```

```
if(fs === 5 ){
```

```
clearInterval(var2);
```

```
}
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Output:

TEXT-GROWING

TEXT SHRINKING

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Viva Questions:

1. What are JavaScript types?
2. How do you submit a form using Javascript?
3. What's a way to append a value to an array?
4. What does isNaN function do?
5. How to read and write a file using javascript?

Program 4:

4) Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:

- a) Parameter: A string**
- b) Output: The position in the string of the left-most vowel**
- c) Parameter: A number**
- d) Output: The number with its digits in the reverse order**

Program Objectives:

- To Design and develop static and dynamic web pages.

4a.html

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var str = prompt("Enter the Input","");
```

```
if(!(isNaN(str)))
```

```
{
```

```
var num,rev=0,remainder;
```

```
num = parseInt(str);
```

```
while(num!=0) {
```

```
remainder = num%10;
```

```
num = parseInt(num/10);
```



```
rev = rev * 10 + remainder;

}

alert("Reverse of "+str+" is "+rev);

}

else

{

str = str.toUpperCase();

for(var i = 0; i < str.length; i++) {

var chr = str.charAt(i);

if(chr == 'A' || chr == 'E' || chr == 'I' || chr == 'O' || chr == 'U')break;

}

if( i < str.length )

alert("The position of the left most vowel is "+(i+1));

else

alert("No vowel found in the entered string");

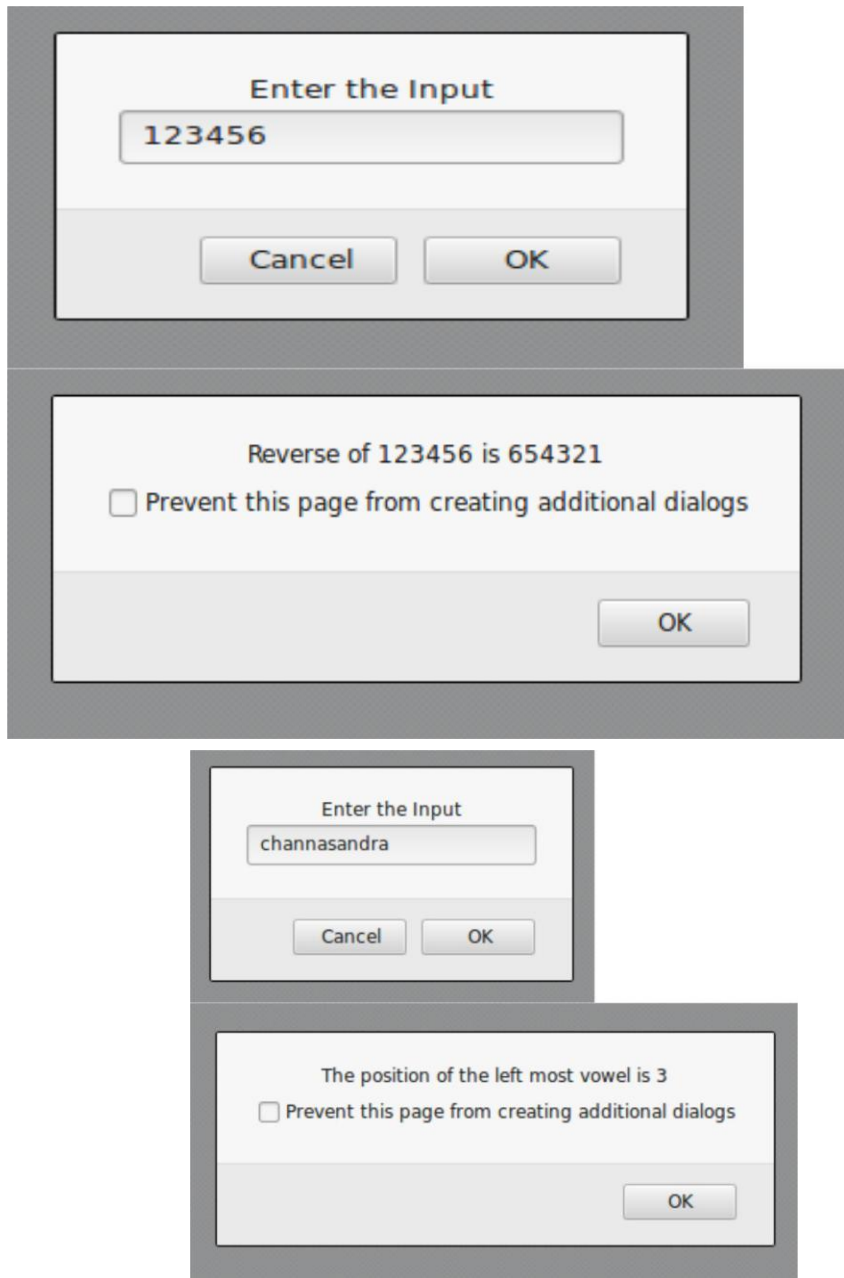
}

</script>

</body>

</html>
```

Output:



Test Cases :

Test No.	Input Parameters	Expected Output	Obtained Output	Remarks
1	123	Reverse of 123 is 321	Addition =74.95 Subtraction =26.17 Multiplication=1233.1584 Division=2.072980729807298	PASS
2	CHANNASA NDRA	The position of the leftmost vowel is 3	The position of the leftmost vowel is 3	PASS
3	SKY	No vowel found in the entered string	No vowel found in the entered string	PASS
4	MNKTO	The position of the leftmost vowel is 5	The position of the leftmost vowel is 5	PASS

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Viva Questions:

1. What is HTML5?
2. Name some of the new features of HTML5.
3. Which browsers support HTML5?
4. Are HTML tags case sensitive?
5. What is the purpose of 'section' tag in HTML5?
6. What are custom attributes in HTML5?
7. What are the drawbacks of cookies?
8. What is the purpose of 'video' tag in HTML5?

Program 5:

5) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.

Program Objectives:

- To Design and develop static and dynamic web pages.

5. xml

```
<?xml-stylesheet type="text/css" href="5.css" ?>

<!DOCTYPE HTML>

<html>

<head>

<h1> STUDENTS DESCRIPTION </h1>

</head>

<students>

<student>

<USN>USN: 4AD15CS001</USN>

<name>NAME : SANTHOSH</name>

<college>COLLEGE : ATME</college>

<branch>BRANCH : Computer Science and Engineering</branch>

<year>YEAR : 2015</year>

<e-mail>E-Mail : santosh@gmail.com</e-mail>
```

</student>

<student>

<USN>USN: 4AD15CS002</USN>

<name>NAME : MANORANJAN</name>

<college>COLLEGE : ATME</college>

<branch>BRANCH : Computer Science and Engineering</branch>

<year>YEAR : 2015</year>

<e-mail>E-Mail : manaranjan@gmail.com</e-mail>

</student>

<student>

<USN>USN: 4AD15CS003</USN>

<name>NAME : CHETHAN</name>

<college>COLLEGE : ATME </college>

<branch>BRANCH : Computer Science and Engineering</branch>

<year>YEAR : 2015</year>

<e-mail>E-Mail : chethan@gmail.com</e-mail>

</student>

</students>

</html>

5. css

```
student{  
display:block; margin-top:10px; color:Navy;  
}
```

```
USN{  
display:block; margin-left:10px;font-size:14pt; color:Red;  
}
```

```
name{  
display:block; margin-left:20px;font-size:14pt; color:Blue;  
}
```

```
college{  
display:block; margin-left:20px;font-size:12pt; color:Maroon;  
}
```

```
branch{  
display:block; margin-left:20px;font-size:12pt; color:Purple;  
}
```

```
year{  
display:block; margin-left:20px;font-size:14pt; color:Green;  
}
```

```
e-mail{  
display:block; margin-left:20px;font-size:12pt; color:Blue;  
}
```

Output:

STUDENTS DESCRIPTION

USN : 4AD15CS001

NAME : SANTHOSH

COLLEGE : ATME

BRANCH : Computer Science and Engineering

YEAR : 2015

E-Mail : santosh@gmail.com

USN : 4AD15CS002

NAME : MANORANJAN

COLLEGE : ATME

BRANCH : Computer Science and Engineering

YEAR : 2015

E-Mail : manaranjan@gmail.com

USN : 4AD15CS003

NAME : CHETHAN

COLLEGE : ATME

BRANCH : Computer Science and Engineering

YEAR : 2015

E-Mail : chethan@gmail.com

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Viva Questions:

1. Explain inline, embedded and external style sheets.
2. What are style sheet properties?
3. What is CSS?
4. What is XML?
5. How does the XML structure is defined?
6. How do you display XML with XSLT?
7. What is a CDATA section in XML?
8. List the rules to be followed by an XML document.

Program 6:

6) Write a PHP program to keep track of the number of visitors visiting the webpage and to display this count of visitors, with proper headings.

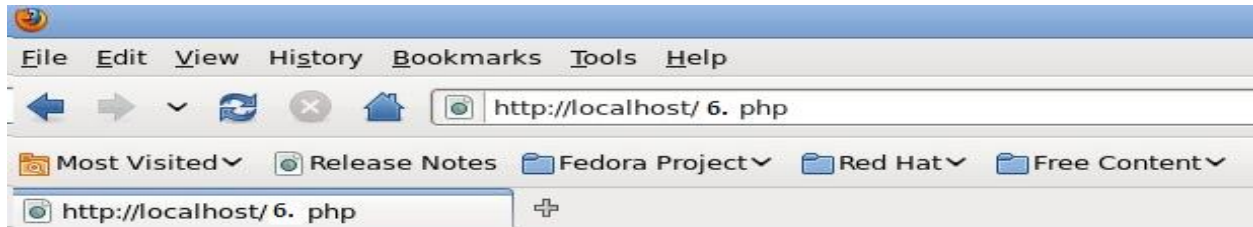
Program Objectives:

- Design and develop static and dynamic web pages.
- Familiarize with Client-Side Programming, Server-Side Programming, Active server Pages.

6.php

```
<?php  
  
print "<h3> REFRESH PAGE </h3>";  
  
$name="counter.txt";  
  
$file = fopen($name,"r");  
  
$hits= fscanf($file,"%d");  
  
fclose($file);  
  
$hits[0]++;  
  
$file = fopen($name,"w");  
  
fprintf($file,"%d", $hits[0]);  
  
fclose($file);  
  
print "Total number of views: ".$hits[0];  
  
?>
```

Output:



REFRESH PAGE

Total number of views: 10

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Viva Questions:

1. What is PHP?
2. What are the common usage of PHP?
3. In how many ways you can embed PHP code in an HTML page?
4. Is PHP a case sensitive language?

Program 7:

7) Write a PHP program to display a digital clock which displays the current time of the server.

Program Objectives:

- To Design and develop static and dynamic web pages.
- Familiarize with Client-Side Programming, Server-Side Programming, and Active server Pages.

7.php

```
<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="refresh" content="1"/>

<style>

p {

color:white;

font-size:90px;

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);
```

```
}  
  
body{background-color:black;}  
  
</style>  
  
<p> <?php echo date(" h: i : s A");?> </p>  
  
</head>
```

Output:



10: 44 : 08 AM

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Viva Questions:

1. What are the characteristics of PHP variables?
2. What are the different types of PHP variables?
3. What are rules for naming a PHP variable?
4. What is NULL?

Program 8:

8) Write the PHP programs to do the following:

- a) Implement simple calculator operations.**
- b) Find the transpose of a matrix.**
- c) Multiplication of two matrices.**
- d) Addition of two matrices.**

Program Objectives:

- To Design and develop static and dynamic web pages.
- Familiarize with Client-Side Programming, Server-Side Programming, Active server Pages.

8a.php

<html>

<head>

<style>

table, td, th

{

border: 1px solid black;

width: 35%;

text-align: center;

```
background-color: DarkGray;
```

```
}
```

```
table { margin: auto; }
```

```
input,p { text-align:right; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form method="post">
```

```
<table>
```

```
<caption><h2> SIMPLE CALCULATOR </h2></caption>> <tr><td>First  
Number:</td><td><input type="text" name="num1" /></td>
```

```
<td rowspan="2"><input type="submit" name="submit"  
value="calculate"></td></tr>
```

```
<tr><td>Second Number:</td><td><input type="text"  
name="num2"/></td></tr>
```

```
</form>
```

```
<?php
```

```
if(isset($_POST['submit'])) // it checks if the input submit is filled  
{
```

```
$num1 = $_POST['num1'];

$num2 = $_POST['num2'];

if(is_numeric($num1) and is_numeric($num2) )

{

echo "<tr><td> Addition :</td><td><p>".($num1+$num2)."</p></td>";

echo "<tr><td> Subtraction :</td><td><p>".($num1-$num2)."</p></td>";

echo "<tr><td> Multiplication :</td><td><p>".($num1*$num2)."</p></td>";

echo "<tr><td>Division :</td><td><p>".($num1/$num2)."</p></td>";

echo "</table>";

}

else

{

echo "<script type='text/javascript' > alert(' ENTER VALID  
NUMBER');</script>";

}

}

?>

</body>

</html>
```

Output:**SIMPLE CALCULATOR**

First Number:	50	calculate
Second Number:	25	
Addition :	75	
Subtraction :	25	
Multiplication :	1250	
Division :	2	

Test Cases :

Test No.	Input Parameters	Expected Output	Obtained Output	Remarks
1	value1=50.56 value2=24.39	Addition =74.95 Subtraction =26.17 Multiplication=1233.1584 Division=2.072980729807298	Addition =74.95 Subtraction =26.17 Multiplication=1233.1584 Division=2.072980729807298	PASS
2	value1= 0 value2= 45	Addition =45 Subtraction =-45 Multiplication=0 Division=0	Addition =45 Subtraction =-45 Multiplication=0 Division=0	PASS
3	value1= 45 value2= 0	Addition =45 Subtraction =45 Multiplication=0 Division=Infinity	Addition =45 Subtraction =45 Multiplication=0 Division=Infinity	PASS
4	value1 = abc value2 = 23	ENTER VALID NUMBER	ENTER VALID NUMBER	PASS
5	value1 = 50 value2 =xyz	ENTER VALID NUMBER	ENTER VALID NUMBER	PASS

8b.php

```
<?php

$a = array(array(1,2,3),array(4,5,6),array(7,8,9));

$b = array(array(7,8,9),array(4,5,6),array(1,2,3));

$m=count($a);

$n=count($a[2]);

$p=count($b);

$q=count($b[2]);

echo "the first matrix      :". "<br/>";

for ($row = 0; $row < $m; $row++) {

for ($col = 0; $col < $n; $col++)

echo " ".$a[$row][$col];

echo "<br/>";

}

echo "the second matrix  :". "<br/>";

for ($row = 0; $row < $p; $row++) {

for ($col = 0; $col < $q; $col++)

echo " ".$b[$row][$col];

echo "<br/>";

}
```

```
echo "the transpose for the first matrix is:."<br/>"; for ($row = 0; $row < $m;
$row++) {

for ($col = 0; $col < $n; $col++)

echo " ".$a[$col][$row];

echo "<br/>";

}

if(($m=== $p) and ($n=== $q)) {

echo "the addition of matrices is:."<br/>";

for ($row = 0; $row < 3; $row++) {

for ($col = 0; $col < 3; $col++)

echo " ".$a[$row][$col]+$b[$row][$col]." "; echo "<br/>";

}

}

if($n=== $p){

echo " The multiplication of matrices: <br/>";

$result=array();

for ($i=0; $i < $m; $i++) {

for($j=0; $j < $q; $j++){

$result[$i][$j] = 0;

for($k=0; $k < $n; $k++)
```

```
$result[$i][$j] += $a[$i][$k] * $b[$k][$j];  
  
}  
  
}  
  
for ($row = 0; $row < $m; $row++) {  
  
for ($col = 0; $col < $q; $col++)  
  
echo " ".$result[$row][$col];  
  
echo "<br/>";  
  
}  
  
}  
  
?>
```

Output:

the first matrix:

1 2 3

4 5 6

7 8 9

the second matrix:

7 8 9

4 5 6

1 2 3

the transpose of the first matrix:

1 4 7

2 5 8

3 6 9

the addition of matrices is:

8 10 12

8 10 12

8 10 12

the multiplication of matrices:

18 24 30

54 69 84

90 114 138

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Viva Questions:

1. How will you define a constant in PHP?
2. What is the purpose of constant() function?
3. What are the differences between PHP constants and variables?
4. What are PHP magic constants?

Program 9:

9) Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following:

- a) a)Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList.**
- b) b)Search for a word in states that begins with k and ends in s. Perform a case-insensitive comparison. [Note: Passing re.I as a second parameter to method compile performs a case-insensitive comparison.] Store this word in element1 of statesList.**
- c) c)Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.**
- d) d)Search for a word in states that ends in a. Store this word in element 3 of the list.**

Program Objectives:

- To Design and develop static and dynamic web pages.
- Familiarize with Client-Side Programming, Server-Side Programming, Active server Pages.

9.php

```
<?php
```

```
$states = "Mississippi Alabama Texas Massachusetts Kansas"; $statesArray = [];
```

```
$states1 = explode(' ', $states);
```

```
echo "Original Array :<br>";
```

```
foreach ( $states1 as $i => $value )
```

```
print("STATES[$i]=$value<br>");
```

```
foreach($states1 as $state) {
```

```
if(preg_match( '/xas$/', ($state)))
```

```
$statesArray[0] = ($state);
```

```
}
```

```
foreach($states1 as $state) {
```

```
if(preg_match('/^k.*s$/i', ($state)))
```

```
$statesArray[1] = ($state);
```

```
}
```

```
foreach($states1 as $state) {
```

```
if(preg_match('/^M.*s$/', ($state)))
```

```
$statesArray[2] = ($state);
```

```
}
```

```
foreach($states1 as $state){
```

```
if(preg_match('/a$/', ($state)))
```

```
$statesArray[3] = ($state);
```

```
}
```

```
echo "<br><br>Resultant Array :<br>";

foreach ( $statesArray as $array => $value )

print("STATES[$array]=$value<br>");

?>
```

Output:

```
Original Array :
STATES[0]=Mississippi
STATES[1]=Alabama
STATES[2]=Texas
STATES[3]=Massachusetts
STATES[4]=Kansas
```

```
Resultant Array :
STATES[0]=Texas
STATES[1]=Kansas
STATES[2]=Massachusetts
STATES[3]=Alabama
```

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Viva Questions:

1. What is the use of "echo" in php?
2. Differences between GET and POST methods?
3. How to set cookies in PHP?
4. How to create a mysql connection?
5. How to select a database?

Program 10:

10) Write a PHP program to sort the student records which are stored in the database using selection sort.

Program Objectives:

- To Design and develop static and dynamic web pages.
- To Familiarize with Client-Side Programming, Server-Side Programming, Active server Pages.
- To Learn Database Connectivity to web applications.

10.php

Goto Mysql and then type :

```
create database weblab;
```

```
use weblab;
```

```
create table student(usnvarchar(10),name varchar(20),address varchar(20));
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<style>
```

```
table, td, th
```

```
{
```

```
border: 1px solid black;
```

```
width: 33%;
```

```
text-align: center;

border-collapse: collapse;

background-color: lightblue;

}

table { margin: auto; }

</style>

<?php

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "weblab";

$a=[];

// Create connection

// Opens a new connection to the MySQL server

$conn = mysqli_connect($servername, $username, $password, $dbname);

//      Check connection and return an error description from the last connection
error, if any

if ($conn->connect_error)

die("Connection failed: " . $conn->connect_error);

$sql = "SELECT * FROM student";
```

```
//      performs a query against the database $result = $conn->query($sql);

echo "<br>";

echo "<center> BEFORE SORTING </center>"; echo "<table border='2'>";

echo "<tr>";

echo "<th>USN</th><th>NAME</th><th>Address</th></tr>"; if ($result-
>num_rows> 0)

{

//      output data of each row and fetches a result row as an associative array

while($row = $result->fetch_assoc()){

echo "<tr>";

echo "<td>". $row["usn"]."</td>";

echo "<td>". $row["name"]."</td>";

echo "<td>". $row["addr"]."</td></tr>";

array_push($a,$row["usn"]);

}

}

else

echo "Table is Empty";

echo "</table>";

$n=count($a);
```

```
$b=$a;

for ( $i = 0 ; $i< ($n - 1) ; $i++ )

{

$pos= $i;

for ( $j = $i + 1 ; $j < $n ; $j++ ) {

if ( $a[$pos] > $a[$j] )

$pos= $j;

}

if ( $pos!= $i ) {

$temp=$a[$i];

$a[$i] = $a[$pos];

$a[$pos] = $temp;

}

}

$c=[];

$d=[];

$result = $conn->query($sql);

if ($result->num_rows> 0)// output data of each row

{
```

```
while($row = $result->fetch_assoc()) {  
  
    for($i=0;$i<$n;$i++) {  
  
        if($row["usn"]== $a[$i]) {  
  
            $c[$i]=$row["name"];  
  
            $d[$i]=$row["addr"];  
  
        }  
  
    }  
  
}  
  
echo "<br>";  
  
echo "<center> AFTER SORTING <center>";  
  
echo "<table border='2'>";  
  
echo "<tr>";  
  
echo "<th>USN</th><th>NAME</th><th>Address</th></tr>";  
for($i=0;$i<$n;$i++) {  
  
    echo "<tr>";  
  
    echo "<td>". $a[$i]."</td>";  
  
    echo "<td>". $c[$i]."</td>";  
    echo "<td>". $d[$i]."</td></tr>";  
  
}  
  
echo "</table>";
```

```
$conn->close();
```

```
?>
```

```
</body>
```

```
</html>
```

Output

BEFORE SORTING

USN	NAME	ADDRESS
4AD14	chandan	bengaluru
4AD07	arun	mysore
4AD01	abhi	tumkur
4AD38	manoranjan	mandya

AFTER SORTING

USN	NAME	ADDRESS
4AD01	abhi	tumkur
4AD07	arun	mysore
4AD14	chandan	bengaluru
4AD38	manoranjan	mandya

Program Outcome:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Have a good understanding of Web Application Terminologies, Internet Tools other web services.

Viva Questions:

1. What is the difference between Session and Cookie?
2. How to create a session?
3. How to remove data from a session?
4. How to find the length of a string?
5. What is the use of rand() in php?

Viva Questions

1) What is HTML?

HTML, or HyperText Markup Language, is a Universal language which allows an individual using special code to create web pages to be viewed on the Internet.

2) What is a tag?

In HTML, a tag tells the browser what to do. When we write an HTML page, we enter tags for many reasons -- to change the appearance of text, to show a graphic, or to make a link to another page.

3) What is the simplest HTML page?

HTML Code:

```
<HTML>
<HEAD>
<TITLE>This is my page title! </TITLE>
</HEAD>
<BODY>
    This is my message to the world!
</BODY>
</HTML>
```

4) How do I create frames? What is a frameset?

Frames allow an author to divide a browser window into multiple (rectangular) regions. Multiple documents can be displayed in a single window, each within its own frame. Graphical browsers allow these frames to be scrolled independently of each other, and links can update the document displayed in one frame without affecting the others.

You can't just "add frames" to an existing document. Rather, you must create a frameset document that defines a particular combination of frames, and then display your content documents inside those frames. The frameset document should also include alternative non-framed content in a NOFRAMES element.

The HTML 4 frames model has significant design flaws that cause usability problems for web users. Frames should be used only with great care.

5) How can I include comments in HTML?

Technically, since HTML is an SGML application, HTML uses SGML comment syntax. However, the full syntax is complex, and browsers don't support it in its entirety anyway. Therefore, use the following simplified rule to create HTML comments that both have valid syntax and work in browsers:

An HTML comment begins with "<!--", ends with "-->", and does not contain "--" or ">" anywhere in the comment.

The following are examples of HTML comments:

* <!-- This is a comment. -->

* <!-- This is another comment,
and it continues onto a second line. -->

* <!-->

6) What is a Hypertext link?

A hypertext link is a special tag that links one page to another page or resource. If you click the link, the browser jumps to the link's destination.

7) What is a DOCTYPE? Which one do I use?

According to HTML standards, each HTML document begins with a DOCTYPE declaration that specifies which version of HTML the document uses. Originally, the DOCTYPE declaration was used only by SGML-based tools like HTML validators, which needed to determine which version of HTML a document used (or claimed to use).

Today, many browsers use the document's DOCTYPE declaration to determine whether to use a stricter, more standards-oriented layout mode, or to use a "quirks" layout mode that attempts to emulate older, buggy browsers.

8) How do I align a table to the right (or left)?

You can use <TABLE ALIGN="right"> to float a table to the right. (Use ALIGN="left" to float it to the left.) Any content that follows the closing </TABLE> tag will flow around the table. Use <BR CLEAR="right"> or <BR CLEAR="all"> to mark the end of the text that is to flow around the table, as shown in this example:

The table in this example will float to the right.

```
<table align="right">...</table>
```

This text will wrap to fill the available space to the left of (and if the text is long enough, below) the table.

```
<br clear="right">
```

This text will appear below the table, even if there is additional room to its left.

9) Explain Cell Padding and Cell Spacing ?

Cell Padding: it refers to the gap or space between the cell content and cell border or cell wall.

Cell Spacing: It refers to the gap between the two cells of same tables.

In HTML cell spacing and padding both are used with Table Border layout.

Example:

```
<table border cellpadding=2>
```

```
<table border cellspacing=2>
<table border cellpadding=2 cellspacing=2>
```

10) How to create a button which acts like a link?

To create buttons which act as a hyperlink, there are two ways:

```
<FORM ACTION="[url]" METHOD=get>
<INPUT TYPE=submit VALUE="Text on button">
</FORM>

<INPUT TYPE="submit" VALUE="Go to my link location"
ONCLICK=" http://www.careerride.com/;" />
```

11) What is difference between HTML and XHTML?

The differences between HTML and XHTML are:

HTML is application of Standard Generalized Markup Language(SGML) whereas XML is application of Extensible Markup Language(XML).

HTML is a static Web Page whereas XHTML is dynamic Web Page.

HTML allows programmer to perform changes in the tags and use attribute minimization whereas XHTML when user need a new markup tag then user can define it in this.

HTML is about displaying information whereas XHTML is about describing the information

12) How many types CSS can be include in HTML?

There are three ways to include the CSS with HTML:

Inline CSS: it is used when only small context is to be styled.

o To use inline styles add the style attribute in the relevant tag.

External Style Sheet: is used when the style is applied to many pages.

o Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

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

Internal Style Sheet: is used when a single document has a unique style.

o Internal styles sheet needs to put in the head section of an HTML page, by using the <style> tag, like this:

```
<head>
<style type="text/css">
hr {color:sienna}
p {margin-left:20px}
body {background-image:url("images/back40.gif")}
</style>
</head>
```

13) What are logical and physical tags in HTML?

Logical tags are used to tell the meaning of the enclosed text. The example of the logical tag is `` `` tag. When we enclosed text in strong tag then it tell the browser that enclosed text is more important than other text.

Physical text are used to tell the browser that how to display the text enclosed in the physical tag. Some example of the physical tags are: `` , `<big>` , `<i>`

14) What are new Media Elements in HTML5?

Following are the New Media Elements are present in HTML5:

`<audio>` tag: for playing audio.

`<video>` tag: for playing video.

`<source>` tag: For media resources for media elements.

`<embed>` tag: For embedded content,

`<track>` tag: For text tracks used in media players

15) Explain various HTML list tags.

In HTML we can list the element in two ways:

Ordered list: in this list item are marked with numbers.

Syntax: ``

`` first item ``

``second item `` ``

Display as: 1. First item

2. Second item.

Unordered Lists: in this item are marked with bullets.

Syntax: ``

`` first item ``

``second item `` ``

Display as:

- First item
- Second item.

16) Explain HTML background.

There are two types of background in HTML:

Colored Background: in this the background of the html is colored.

The Syntax is: <body bgcolor = “red”>

The value of the bgcolor can be set in three ways by hexadecimal number, an RGB value and Color name.

Example: <body bgcolor = “black”>

<body bgcolor = “rgb(0,0,0)”>

<body bgcolor = “#000000”>

Image Background: in this we set the background of the website by an image. Syntax used for this is : <body background=”study.gif”>

17) What is CSS?

CSS stands for Cascading Style Sheets. By using CSS with HTML we can change the look of the web page by changing the font size and color of the font. CSS plays an important role in building the website. Well written CSS file can be used to change the presentation of each web page. By including only one CSS file. It gives web site developer and user more control over the web pages.

18) How to insert Javascript in HTML?

We can insert JavaScript in HTML using <Script tag>. JavaScript can be enclosed in <script type = text/javascript> and ending with </script>.

Example:

```
<html>
  <body>
    <script type="text/javascript">
      ...JavaScript....
    </script>
  </body>
</html>
```

19) What is the Use of SPAN in HTML and give one example?

SPAN: Used for the following things:

Highlight the any color text

For adding colored text

For adding background image to text.

Example:

```
<p>  
<span style="color:#000000;">  
In this page we use span.  
</span>  
</p>
```

20)What are style sheet properties?

CSS Background

CSS Text

CSS Font

CSS Border

CSS Outline

CSS Margin

CSS Padding

CSS List

CSS Table

List various font attributes used in style sheet.

font-style

font-variant

font-weight

font-size/line-height

font-family

caption

icon

menu

message-box

small-caption

status-bar

21) 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.

22) How do I create a link that opens a new window?

 opens a new, unnamed window.

 opens a new window named "example", provided that a window or frame by that name does not already exist.

Note that the TARGET attribute is not part of HTML 4 Strict. In HTML 4 Strict, new windows can be created only with JavaScript. links that open new windows can be annoying to your readers if there is not a good reason for them.

What is the difference between the HTML form methods GET and POST?

The method parameter specifies which method the client is using to send information to the WEB server. The method determines which parameter you will find the CGI request data in:

* POST - post_args

* GET – http args

23)What is the DOM?

DOM is a platform independent, World Wide Web Consortium (W3C) standard form of representation of structured documents as an object-oriented model. It is an application programming interface so as to access HTML and XML documents.

Document Object Model (DOM) is used to query, traverse and manipulate documents like XML or HTML documents. DOM is best suited where the document must be accessed repeatedly or out of sequence order. DOM allows accessing the contents of a web page. It also allows dealing with events that allows capturing and responding to user's actions. There are different levels of DOM standards depending on the compatibility of the browsers.

24) What is the HTML DOM?

The HTML DOM API specializes and adds the functionality to relate to HTML documents and elements. It addresses the issues of backwards compatibility with the Level 0 of DOM and provides mechanisms for common and frequent operations on HTML documents

25) What is JavaScript?

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

JavaScript is a general-purpose programming language designed to let programmers of all skill levels control the behavior of software objects. The language is used most widely today in Web browsers whose software objects tend to represent a variety of HTML elements in a document and the document itself.

But the language is used with other kinds of objects in other environments. For example, Adobe Acrobat Forms uses JavaScript as its underlying scripting language to glue together objects that are unique to the forms generated by Adobe Acrobat.

Therefore, it is important to distinguish JavaScript, the language, from the objects it can communicate with in any particular environment.

When used for Web documents, the scripts go directly inside the HTML documents and are downloaded to the browser with the rest of the HTML tags and content.

26) How is JavaScript different from Java?

Don't be fooled by the term Java in both. Both are quite different technologies.

JavaScript was developed by Brendan Eich of Netscape; Java was developed at Sun Microsystems. While the two languages share some common syntax, they were developed independently of each other and for different audiences. Java is a full-fledged programming language tailored for network computing; it includes hundreds of its own objects, including objects for creating user interfaces that appear in Java applets (in Web browsers) or standalone Java applications. In contrast, JavaScript relies on whatever environment it's operating in for the user interface, such as a Web document's form elements.

JavaScript was initially called LiveScript at Netscape while it was under development. A licensing deal between Netscape and Sun at the last minute let Netscape plug the "Java" name into the name of its scripting language. Programmers use entirely different tools for Java and JavaScript. It is also not uncommon for a programmer of one language to be ignorant of the other. The two languages don't rely on each other and are intended for different purposes. In some ways, the "Java" name on JavaScript has confused the world's understanding of the

differences between the two. On the other hand, JavaScript is much easier to learn than Java and can offer a gentle introduction for newcomers who want to graduate to Java and the kinds of applications you can develop with it.

27) What is the official JavaScript website?

This is a trick question used by interviewers to evaluate the candidate's knowledge of JavaScript. Most people will simply say javascript.com is the official website.

The truth is- there is no official website for Javascript you can refer to. It was developed by Brendan Eich for Netscape. It was based on the ECMAScript language standard; ECMA-262 being the official JavaScript standard.

28) What are JavaScript types?

Number, String, Boolean, Function, Object, Null, Undefined.
How do you convert numbers between different bases in JavaScript?
Use the parseInt() function, that takes a string as the first parameter, and the base as a second parameter. So to convert hexadecimal 3F to decimal, use parseInt ("3F", 16);

29) How do you submit a form using Javascript?

Use document.forms[0].submit();

30) How do you assign object properties?

obj["age"] = 22 or obj.age = 22.

31) What's a way to append a value to an array?

arr[arr.length] = value;

32) What does isNaN function do?

Return true if the argument is not a number.

33 What's relationship between JavaScript and ECMAScript?

ECMAScript is yet another name for JavaScript (other names include LiveScript). The current JavaScript that you see supported in browsers is ECMAScript revision 3.

34) How to read and write a file using javascript?

I/O operations like reading or writing a file is not possible with client-side javascript.

35) How do you convert numbers between different bases in JavaScript?

Use the `parseInt()` function, that takes a string as the first parameter, and the base as a second parameter. So to convert hexadecimal FF to decimal, use `parseInt ("FF", 16);`

36) What boolean operators does JavaScript support?

&&, and !

37) How to get the contents of an input box using Javascript?

Use the "value" property.
`var myValue = window.document.getElementById("textboxID").value;`

38) How to determine the state of a checkbox using Javascript?

`var checkedP = window.document.getElementById("CheckBoxID").checked;`

39) How to set the focus in an element using Javascript?

```
<script> function setFocus() { if(focusElement != null) {  
document.forms[0].elements["myelementname"].focus(); } } </script>
```

40) How to access an external javascript file that is stored externally and not embedded?

This can be achieved by using the following tag between head tags or between body tags.
`<script src="raj.js"></script>`How to access an external javascript file that is stored externally and not embedded? where abc.js is the external javascript file to be accessed.

41) What is the difference between an alert box and a confirmation box?

An alert box displays only one button which is the OK button whereas the Confirm box displays two buttons namely OK and cancel.

42)What is a prompt box?

A prompt box allows the user to enter input by providing a text box.

43) Can javascript code be broken in different lines?

Breaking is possible within a string statement by using a backslash \ at the end but not within any other javascript statement.
that is ,
`document.write("Hello \ world");`
is possible but not `document.write \`
`("hello world");`

44) What looping structures are there in JavaScript?

for, while, do-while loops, but no foreach.

45) How do you create a new object in JavaScript?

`var obj = new Object();` or `var obj = { };`

46) What is this keyword?

It refers to the current object.

47) 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.

48) What looping structures are there in JavaScript?

for, while, do-while loops, but no foreach.

49) To put a "close window" link on a page ?

```
<a href='javascript:window.close()' class='mainnav'> Close </a>
```

50) How to hide javascript code from old browsers that dont run it?

Use the below specified style of comments `<script language=javascript> <!-- javascript code goes here // -->` or Use the `<NOSCRIPT>`some html code `</NOSCRIPT>` tags and code the display html statements between these and this will appear on the page if the browser does not support javascript

51) How to comment javascript code?

Use `//` for line comments and
`/*`
`*/` for block comments
Name the numeric constants representing max,min values
`Number.MAX_VALUE`
`Number.MIN_VALUE`

52) What does javascript null mean?

The null value is a unique value representing no value or no object.
It implies no object,or null string,no valid boolean value,no number and no array object.

53) How do you create a new object in JavaScript?

`var obj = new Object();` or `var obj = {};`
How do you assign object properties?
`obj["age"] = 23` or `obj.age = 23.`

54) What's a way to append a value to an array?

`arr[arr.length] = value;`

55) To set all checkboxes to true using JavaScript?

```
//select all input tags
function SelectAll() {
var checkboxes = document.getElementsByTagName("input");
for(i=0;i<checkboxes.length;i++) {
if(checkboxes.item(i).attributes["type"].value == "checkbox") {
checkboxes.item(i).checked = true;
}}}
```

56) What does undefined value mean in javascript?

Undefined value means the variable used in the code doesn't exist or is not assigned any value or the property doesn't exist.

57) What is the difference between undefined value and null value?

(i)Undefined value cannot be explicitly stated that is there is no keyword called undefined whereas null value has keyword called null
(ii)typeof undefined variable or property returns undefined whereas typeof null value returns object

58) What are undefined and undeclared variables?

Undeclared variables are those that are not declared in the program (do not exist at all),trying to read their values gives runtime error.But if undeclared variables are assigned then

implicit declaration is done .Undefined variables are those that are not assigned any value but are declared in the program.Trying to read such variables gives special value called undefined value.

59) How to disable an HTML object ?

```
document.getElementById("myObject").disabled = true;
```

60) **How to create a popup warning box?**

```
alert('Warning: Please enter an integer between 0 and 1000.');
```

61) **How to create a confirmation box?**

```
confirm("Do you really want to launch the missile?");
```

62) **How to create an input box?**

```
prompt("What is your temperature?");
```

63) **How to force a page to go to another page using JavaScript ?**

```
<script language="JavaScript" type="text/javascript" ><!--  
location.href="http://rajeshstutorials.blogspot.com"; //--></script>
```

64) **What's Math Constants and Functions using JavaScript?**

The Math object contains useful constants such as Math.PI, Math.E
Math.abs(value); //absolute value
Math.max(value1, value2); //find the largest
Math.random() //generate a decimal number between 0 and 1
Math.floor(Math.random()*101) //generate a decimal number between 0 and 100

65) **How to get value from a textbox?**

```
alert(document.getElementById('txtbox1').value);
```

66) **How to get value from dropdown (select) control?**

```
alert(document.getElementById('dropdown1').value);
```

67) What is event bubbling?

Event bubbling describes the behavior of events in child and parent nodes in the Document Object Model (DOM); that is, all child node events are automatically passed to its parent nodes. The benefit of this method is speed, because the code only needs to traverse the DOM tree once. This is useful when you want to place more than one event listener on a DOM element since you can put just one listener on all of the elements, thus code simplicity and reduction. One application of this is the creation of one event listener on a page's body element to respond to any click event that occurs within the page's body.

66) What is XML?

XML is the Extensible Markup Language. It improves the functionality of the Web by letting you identify your information in a more accurate, flexible, and adaptable way. It is extensible because it is not a fixed format like HTML (which is a single, predefined markup language). Instead, XML is actually a metalanguage—a language for describing other languages—which lets you design your own markup languages for limitless different types of documents. XML can do this because it's written in SGML, the international standard metalanguage for text document markup (ISO 8879).

67) What is a markup language?

A markup language is a set of words and symbols for describing the identity of pieces of a document (for example 'this is a paragraph', 'this is a heading', 'this is a list', 'this is the caption of this figure', etc). Programs can use this with a style sheet to create output for screen, print, audio, video, Braille, etc.

Some markup languages (e.g. those used in word processors) only describe appearances ('this is italics', 'this is bold'), but this method can only be used for display, and is not normally re-usable for anything else.

68) What is the difference between XML and HTML?

XML is no way clashes with HTML, since they are for two different purposes.

HTML	XML
HTML is for displaying purpose.	Whereas XML is for data representation.

HTML is used to mark up text so it can be displayed to users.	XML is used to mark up data so it can be processed by computers.
HTML describes both structure (e.g. <p>, <h2>,) and appearance (e.g. , , <i>)	XML describes only content, or “meaning”
HTML uses a fixed, unchangeable set of tags	In XML, you make up your own tags

69) What are the benefits of XML?

There are many benefits of using XML on the Web :

Simplicity- Information coded in XML is easy to read and understand, plus it can be processed easily by computers.

Openness- XML is a W3C standard, endorsed by software industry market leaders.

Extensibility - There is no fixed set of tags. New tags can be created as they are needed.

Self-description- In traditional databases, data records require schemas set up by the database administrator. XML documents can be stored without such definitions, because they contain meta data in the form of tags and attributes.

Contains machine-readable context information- Tags, attributes and element structure provide context information that can be used to interpret the meaning of content, opening up new possibilities for highly efficient search engines, intelligent data mining, agents, etc.

Separates content from presentation- XML tags describe meaning not presentation. The motto of HTML is: "I know how it looks", whereas the motto of XML is: "I know what it means, and you tell me how it should look." The look and feel of an XML document can be controlled by XSL style sheets, allowing the look of a document to be changed without touching the content of the document. Multiple views or presentations of the same content are easily rendered.

Supports multilingual documents and Unicode-This is important for the internationalization of applications.

Facilitates the comparison and aggregation of data - The tree structure of XML documents allows documents to be compared and aggregated efficiently element by element.

Can embed multiple data types - XML documents can contain any possible data type - from multimedia data (image, sound, video) to active components (Java applets, ActiveX).

Can embed existing data - Mapping existing data structures like file systems or relational databases to XML is simple. XML supports multiple data formats and can cover all existing data structures and .

Provides a 'one-server view' for distributed data - XML documents can consist of nested elements that are distributed over multiple remote servers. XML is currently the most sophisticated format for distributed data - the World Wide Web can be seen as one huge XML database.

70) **What is XSL-FO?.**

XSL-FO stands for Extensible Stylesheet Language Formatting Objects. It is used to format XML data and is also a W3C recommendation.

71) **What is a well-formed XML document?**

If a document is syntactically correct it can be called as well-formed XML documents. A well-formed document conforms to XML's basic rules of syntax:

Every open tag must be closed.

The open tag must exactly match the closing tag: XML is case-sensitive.

All elements must be embedded within a single root element.

Child tags must be closed before parent tags.

A well-formed document has correct XML tag syntax, but the elements might be invalid for the specified document type.

72) **What is a valid XML document?**

If a document is structurally correct then it can be called as valid XML documents. A valid document conforms to the predefined rules of a specific type of document:

These rules can be written by the author of the XML document or by someone else.

The rules determine the type of data that each part of a document can contain.

Note: Valid XML document is implicitly well-formed, but well-formed may not be valid

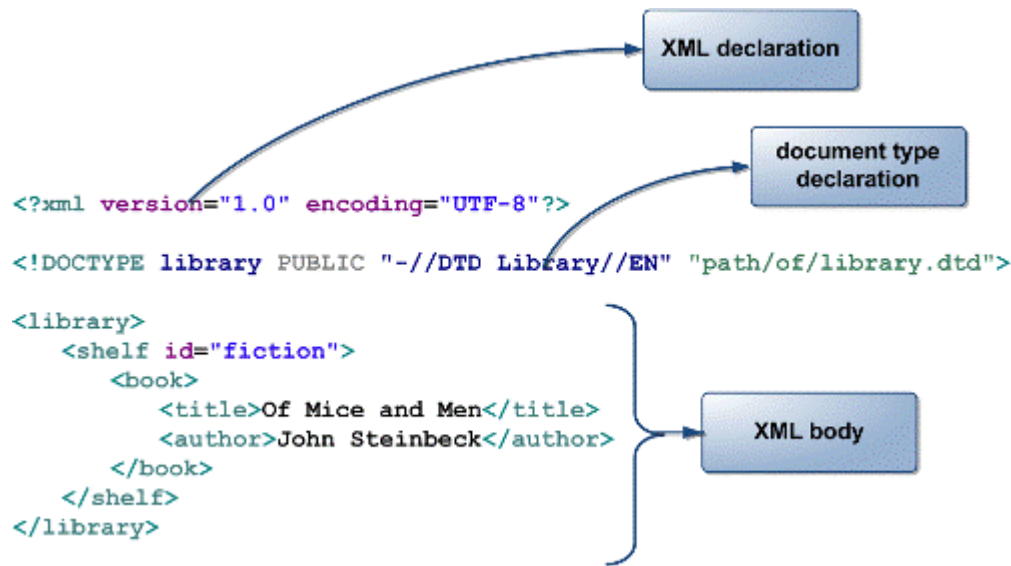
73) What is the structure of XML document ?

Figure 1: XML Structure

74) What is a Processing Instruction in XML?

A ProcessingInstruction is the information which we would like to give to application. Through a ProcessingInstruction an application would get idea about how to process the document. A ProcessingInstruction can appear anywhere and any no. of times in a document.

75) How does the XML structure is defined?

XML document will have a structure which has to be defined before we can create the documents and work with them. The structural rules can be defined using many available technologies, but the following are popular way of doing so-

Document Type Definition (DTD)

Schema

76) What is DTD?

A Document Type Definition (DTD) defines the legal building blocks of an XML document. It defines rules for a specific type of document, including:

Names of elements, and how and where they can be used

The order of elements

Proper nesting and containment of elements

Element attributes

To apply a DTD to an XML document, you can:

Include the DTD's element definitions within the XML document itself.

Provide the DTD as a separate file, whose name you reference in the XML document.

77) What is XML Schema?

An XML Schema describes the structure of an XML instance document by defining what each element must or may contain. XML Schema is expressed in the form of a separate XML file.

XML Schema provides much more control on element and attribute datatypes.

Some datatypes are predefined and new ones can be created.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
  <xsd:element name="test">  
    <xsd:complexType>
```

Schema Elements

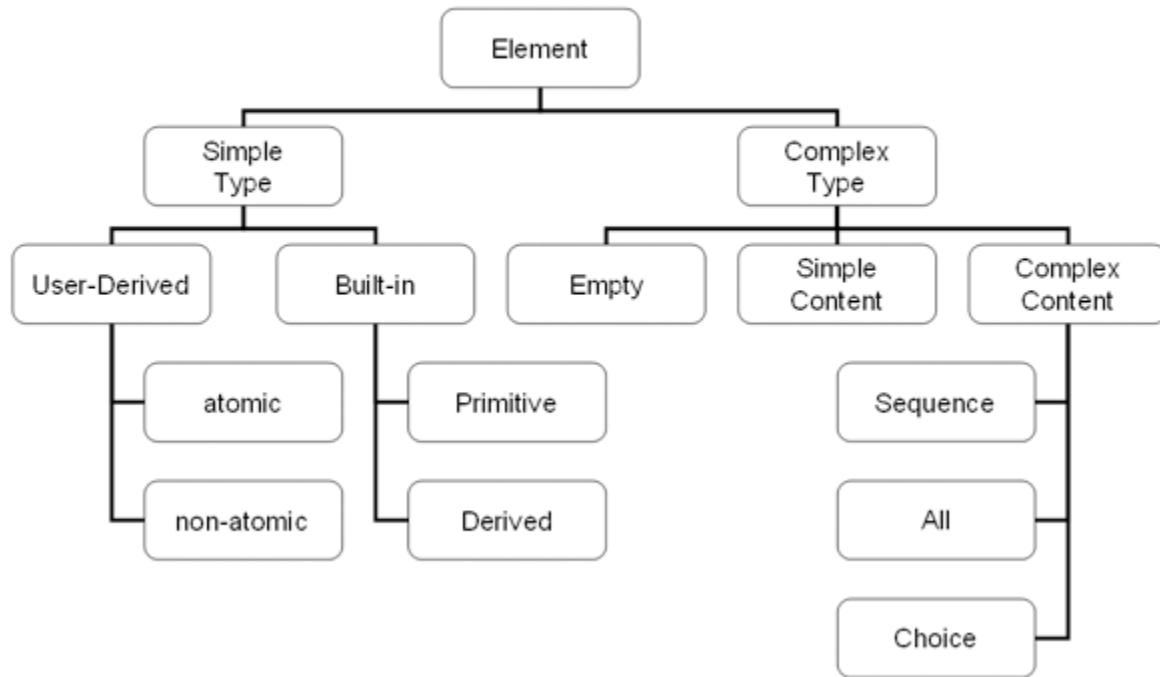


Figure 2: XML Schema

78) What are differences between DTDs and Schema?

Schema	DTD
Schema document is an XML document i.e., the structure of an XML document is specified by another XML document.	DTDs follow SGML syntax.
Schema supports variety of dataTypes similar to programming language.	In DTD everything is treated as text.
In Schema, It is possible to inherit and create relationship among elements.	This is not possible in DTD without invalidating existing documents.

In Schema, It is possible to group elements and attributes so that they can be treated as single logical unit.	Grouping of elements and attributes is not possible in DTD.
In Schemas, it is possible to specify an upper limit for the number of occurrences of an element	It is not possible to specify an upper limit of an element in DTDs

79) What is the version information in XML?

"Version" tag shows which version of XML is used.

80) If XML does not have closing tag will it work?

No, every tag in XML, which is opened, should have a closing tag.

81) Is XML case sensitive?

Yes, XML is case sensitive.

82) What is XML DOM?

The DOM stands for Document Object Model, which describes the logical formation of documents and provides the way to access and manipulate a document. It supplies an Application Programming Interface (API) to XML documents. It is built around the object-oriented design; therefore, it is known as DOM. The DOM model considers an XML document as a composition of objects and every object consists of properties and behaviors that can be manipulated by the DOM methods. The DOM allows creating and building XML documents, navigating the structure of documents, and managing the elements and their data. You can use the DOM methods and objects with any language, such as C#, VB, JavaScript, and VBScript.

83) What is XSL?

XSL is a language for expressing style sheets. An XSL style sheet is a file that describes the way to display an XML document.

Using XSL stylesheets, we can separate the XML document content and its styling.

An XSL style sheet begins with the XML declaration:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

<xsl:stylesheet> defines that the document is an XSLT style sheet document.
The <xsl:template> element defines a template.

84) Define CSS and XSL.

XSL is a language for expressing style sheets. An XSL style sheet is a file that describes the way to display an XML document.

Cascading Style Sheets is an answer to the limitations of HTML, where the structure of documents was defined and not the display. CSS formats documents for display in browsers that support it.

85) How do you display XML with XSLT?

First you need to declare the XSL style sheet:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

86) What is a CDATA section in XML?

CDATA - (Unparsed) Character Data

The term CDATA is used when you don't want some text data to be parsed by the XML parser.

A CDATA section starts with "<![CDATA[" and ends with "]]>":

87) What is a CDATA section in XML?

The CDATA section of XML is used to describe the text that should not be parsed by the XML parser.

The characters like "<" ">" are not supported in XML

"<" will cause an error by the parser. Because the parser identifies < as the starting character for an element.

Any text that is included in CDATA section is ignored by the parser.

Example :

```
<![CDATA[" < and > are used to enclose an element in XML "]]>
```

88) List the rules to be followed by an XML document.

Following rules need to be followed by an XML document:

They must have a root tag, the document should be well formed : the tags should be properly closed, since XML is case sensitive, one should take care that the documents are written with proper care and the attribute values should be inside ""

89) Describe the logical structure of XML.

XML documents comprise of declaration, elements and comments

XML Declaration

It identifies the version to which XML conforms

```
<?xml version = "1.0"?>
```

Document Type Declaration

It consists of markup code that indicates grammar rules or Document Type Definition (DTD) for the particular class of document.

```
<! DOCTYPE Car SYSTEM "cr.dtd">
```

This statement tells the XML processor that the document is of the class Car that conforms the rules specified in the DTD "cr.dtd".

Document element

The document element contains data of an XML document.

90) Why is XML so popular?

Due to the following advantages of using XML, it has become popular:

It supports Unicode. Therefore documents written in any human language can be communicated.

Data structures: records, lists and trees can be represented using XML.

Its format describes structure, field names and their specific values too. Its therefore called self-documenting.

Its syntax and parsing requirements make the necessary parsing algorithms very simple, efficient, and consistent.

It can be used as a document storage and processing format.

It is platform-independent.

91) Why is XML referred as self-describing data?

Text labels inside of XML's syntactic delimiters that cause most people to think that XML is self-describing. But these tags aren't part of XML.

Choosing the terms used for tags or naming anything is often a difficult and contentious activity. Everyone naturally creates names that make sense to them.

However, XML is not self describing.

92) Why is XML extensible?

Extensibility is another attribute of XML. XML is short of "eXtensible Markup Language. This is so because a developer may easily create his own XML syntax for any applications he wishes to use it for. Any other developer, once having learned how to use his own language's XML parsing routines, can use any XML-based format currently available.

93) What is HTML5?

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1

94) What is Web Forms 2.0?

Web Forms 2.0 is an extension to the forms features found in HTML4. Form elements and attributes in HTML5 provide a greater degree of semantic mark-up than HTML4 and remove a great deal of the need for tedious scripting and styling that was required in HTML4.

95) What do you mean by session storage in HTML5?

HTML5 introduces the sessionStorage attribute which would be used by the sites to add data to the session storage, and it will be accessible to any page from the same site opened in that window i.e. session and as soon as you close the window, session would be lost.

96) When a session storage data gets deleted?

The Session Storage Data would be deleted by the browsers immediately after the session gets terminated.

97) What is purpose of getCurrentPosition() method of geolocation object of HTML5?

This method retrieves the current geographic location of the user.

98) What is purpose of watchPosition() method of geolocation object of HTML5?

This method retrieves periodic updates about the current geographic location of the device.

99) What is purpose of clearPosition() method of geolocation object of HTML5?

This method cancels an ongoing watchPosition call.

100) What are Web Workers?

Web Workers do all the computationally expensive tasks without interrupting the user interface and typically run on separate threads.

101) What's PHP ?

The PHP Hypertext Preprocessor is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications.

102) What Is a Session?

A session is a logical object created by the PHP engine to allow you to preserve data across subsequent HTTP requests.

There is only one session object available to your PHP scripts at any time. Data saved to the session by a script can be retrieved by the same script or another script when requested from the same visitor.

Sessions are commonly used to store temporary data to allow multiple PHP pages to offer a complete functional transaction for the same visitor.

103) What Is a Persistent Cookie?

A persistent cookie is a cookie which is stored in a cookie file permanently on the browser's computer. By default, cookies are created as temporary cookies which stored only in the browser's memory. When the browser is closed, temporary cookies will be erased. You should decide when to use temporary cookies and when to use persistent cookies based on their differences:

- *Temporary cookies can not be used for tracking long-term information.
- *Persistent cookies can be used for tracking long-term information.
- *Temporary cookies are safer because no programs other than the browser can access them.
- *Persistent cookies are less secure because users can open cookie files see the cookie values.

104) What does a special set of tags <?= and ?> do in PHP?

The output is displayed directly to the browser.

105) How do you define a constant?

Via define() directive, like define ("MYCONSTANT", 100);

106) What is the difference between mysql_fetch_object and mysql_fetch_array?

MySQL fetch object will collect first single matching record where mysql_fetch_array will collect all matching records from the table in an array.

107) How do you pass a variable by value?

Just like in C++, put an ampersand in front of it, like \$a = &\$b

108) How can we register the variables into a session?

```
session_register($session_var);  
$_SESSION['var'] = 'value';
```

109) How can we destroy the session, how can we unset the variable of a session?

session_unregister() - Unregister a global variable from the current session
session_unset() - Free all session variables

110) How can we know the count/number of elements of an array?

a) sizeof(\$array) - This function is an alias of count()
b) count(\$urarray) - This function returns the number of elements in an array.
Interestingly if you just pass a simple var instead of an array, count() will return 1.

111) How can we find the number of rows in a table using MySQL?

Use this for MySQL
SELECT COUNT(*) FROM table_name;

112) How can I load data from a text file into a table?

The MySQL provides a LOAD DATA INFILE command. You can load data from a file.
Great tool but you need to make sure that:

- a) Data must be delimited
- b) Data fields must match table columns correctly

113) How can we know the number of days between two given dates using MySQL?

Use DATEDIFF()
SELECT DATEDIFF(NOW(), '2006-07-01');

114) What is Rails? And what are the components of Rails?

Rails is a extremely productive web-application framework written in Ruby language by David Hansson. Rails are an open source Ruby framework for developing database-backend web applications.

Rails include everything needed to create a database-driven web application using the Model-View-Controller (MVC) pattern.

115) Why Ruby on Rails?

There are lot of advantages of using ruby on rails.

DRY Principal(Don't Repeat Your Self): It is a principle of software development aimed at reducing repetition of code. "Every piece of code must have a single, unambiguous representation within a system"

Convention over Configuration: Most web development framework for .NET or Java force you to write pages of configuration code. If you follow suggested naming conventions, Rails doesn't need much configuration.

Gems and Plugins: RubyGems is a package manager for the Ruby programming language that provides a standard format for distributing ruby programs and library.

Plugins: A Rails plugin is either an extension or a modification of the core framework. It provides a way for developers to share bleeding-edge ideas without hurting the stable code base. We need to decide if our plugin will be potentially shared across different Rails applications.

If your plugin is specific to your application, your new plugin will be vendored plugin.

If you think, your plugin may be used across applications build it as a gemified plugin.

`$rails generate plugin -help` `//vendored plugin`

`$rails plugin -help` `//gemified plugin`

Most common plugin is ActiveSupport which helps to remove un-necessary whitespaces from Active Record or Active Model attributes. Its good for removing accidental spaces from user inputs.

Scaffolding: Scaffolding is a meta-programming method of building database-backend software application. It is a technique supported by MVC frameworks, in which programmer may write a specification, that describes how the application database may be used. There are two type of scaffolding:

-static: Static scaffolding takes 2 parameter i.e your controller name and model name.

-dynamic: In dynamic scaffolding you have to define controller and model one by one.

Rack Support: Rake is a software task management tool. It allows you to specify tasks and describe dependencies as well as to group tasks in a namespace.

Metaprogramming: Metaprogramming techniques use programs to write programs.

Bundler: Bundler is a new concept introduced in Rails 3, which helps you to manage your gems for application. After specifying gem file, you need to do a bundle install.

Rest Support: As explained above.

Action Mailer: As explained above.

116) What is MVC? and how it Works?

MVC tends for Model-View-Controller, used by many languages like PHP, Perl, Python etc. The flow goes like this:

Request first comes to the controller, controller finds and appropriate view and interacts with model, model interacts with your database and send the response to controller then controller based on the response give the output parameter to view, for Example your url is something like this:

`http://localhost:3000/users/new`

here users is your controller and new is your method, there must be a file in your views/users folder named new.html.erb, so once the submit button is pressed, User model will be called and values will be stored into the database.

117) GET and POST Method?

GET is basically for just getting (retrieving) data, whereas POST may involve anything, like storing or updating data, or ordering a product, or sending E-mail.

118) Does Ruby support multiple inheritance?

Ruby does not support multiple inheritance.

119) What are ruby gems?

- A gem is nothing more than a piece of ruby code packaged as a library so that it can be imported and used by others in their programs.

- A Ruby gem is therefore simply a library that is written in the ruby programming language.

120) What is the purpose of yield?

The interpreter essentially invokes a separate piece of code and places it in the location. You might say it is similar to a method calling another method. Let's understand a little bit of background about where yield might be useful first.

121) What are class variables? How do you define them?

Class variables are created using the @@ prefix to denote the variable as class level.

It works just like any other variable, however in the case of inheritance it works more like a static variable that is accessed across all variable instances.

122) How do you define instance variables?

Instance variables are defined using single @ symbol.

```
@foo = "Hello"
```

Within a class they can be declared as below:

```
class Animal  
  
  attr_accessor :name, :age  
  
end
```

123) How do you define global variables?

Global variables are defined using single \$ symbol.

```
$foo = 5
```

It can be declared anywhere and used anywhere.

124) Does Ruby support constructors? How are they declared?

Constructors are supported in Ruby. They are declared as the method initialize, shown below. The initialize method gets called automatically when Album.new is called.

```
class Album  
  
  def initialize(name, artist, duration)
```

@name = name

@artist = artist

@duration = duration

end

end