Web Technology

→ HTML - Hypertext markup language ; Static web development

→ PHP [used for web development
Hypertext pre processor.

→ Static & dynamic web page
Static → Never Changes
dynamic → as per the user req process is done

→ HTML is used for developing web Pages

Structure :
```
<Html>
<head>
<title> Web Page </title>
</head>
<body> Welcome to html
</body>
</html>
```

→ In html the opening tags and closing tags are used.
→ HTML element start from the start tag to end tag.
→ Break tag is used for a single line break.

```
<html>
<body>
<p> My first paragraph<br>
    My second paragraph
</p>
</body>
</html>
```

→ <hr> is used to divide the web page, a horizontal line

Eg
```
<html>
<head>
</head>
<body> <center> <b> <u> <marquee>
Welcome to html </marquee> </u> </b> </center> </body>
</html>
```

→ SSL - Secure Socket layer
→ https - hypertext transfer protocol

https = http + SSL

```
<html>
<body>
<h1> MREC</h1>
<h2> MREC</h2>
<h3> MREC</h3>
<h4>MREC</h4>
<h5>MREC</h5>
<h6>MREC</h6>
</body>
</html>
```

o/p: MREC
!
MREC

Size decreases

```
<html>
<body>
<p> This is my first paragraph <br>
    This is my second paragraph</p>
</body>
</html>
```

o/p: This is my first paragraph
This is my second paragraph

```
<html>
<body>
<form>
<input type="text">
<input type="submit">
</form>
</body>
</html>
```

o/p: [ ] Submit

```
<html>
<body>
<h1> H<sub>2</sub>O</h1>
</body>
</html>
```

o/p: H_2O

3 Attributes
- face
- size
- color

Ex:
WELCOME TO HTML

.....

Red	green	Blue
00 to ff	00 to ff	00 to ff

color = #00ff00 → green
Color = #ff0000 → Red
Color = #0000ff → Blue

HTML Elements

Formatting tags

<body> ..... </body>
 MREC
<Strong> .....
 .....
<u> ..... </u>
<marquee> ..... </marquee>

^{.....}
_{.....}
<small> ..... </small>
<strike> ..... </strike>
<q> ..... </q>
<abbr> ..... </abbr>

.....

```
<html>
<body>
<h1> <abbr title="World Wide Web"> WWW </abbr> </h1>
</body>
</html>
```

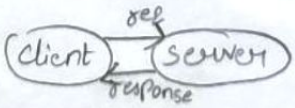
```
<html>
<body>
<h1><strike>Web page Creation </strike></h1>
</body>
</html>
```

Synchronization - imp optimization techniques

HTML Introduction

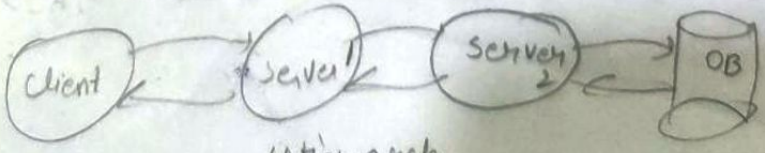
→ HTML is a markup language used to create web pages.
→ HTML consists of tags & elements.
→ We use HTML tags to markup the content of our website
→ So, next time when the browser makes a request to Server, the Server sends back some HTML which is then rendered by the browser on the Screen.
→ website is a collection of web pages.
→ In every website creation we will use these basic HTML tags.
→ Every HTML file ends with .html extention.
→ Image tag inserts a picture in a website.

Java Applications

1. Desktop Appl^n ← standard appln - C/C++/Java/Python related to particular appln for only that desktop
2. Web Appl^n - 

2 tier Archi
3. Enterprise Appl^n - distributed/Parallel, Scalable, reliable, more Se Consistency, accuracy, Transcation oriented eg: banking system.
4. Mobile Appl^n - Android 4 million +

(2) If one Server doesnot accept the request then the client goes to another server but the data is stored in database.



4 tier arch

→ Website → logical collection of web pages, 'n' no. of pages are present for user to access.

→ Bussiness logic - dynamic

① types
- static - The data will not change Ex Google info
- dynamic - The data will change / different for each other eg, filling a google sheet

→ for every minute 175 websites are created

Every day - 252,000

Every hour - 10,500

RESTful API

active websites - 200,756,193

→ website is a collection of publically accessable interlinked web pages that share a single domain name.

→ WWW - It is a large hyper media information service available on the Internet that allows user to browse information.

→ web browser is a software application for accessing information on WWW.

→ webpage is a document which is commonly written in html & translated by a browser.

→ A webpage can be identified by entering an URL (uniform resource locator)

A webpage can be static or dynamic type

Static website

→ It is a basic type of website i.e, easy to create, you don't need knowledge of web programming

Prime no → 2 to sq.rt(n) [condition]

"," - logical OR

→ Program - It is a set of logically related instructions that is arranged in a sequence that directs the computer in solving a problem.

→ Programming - The process of writing a program is called Programming. It is necessary & critical step in data processing.

→ Programming lang. - The language used in the communication of a computer instructions is known as Programming lang.

→ convinency, efficiency, Accurate - 3 major things to perform pgm.

→ factors of pgm - Time, Cost, Quality.

Agile methodology

① Programming language - C, C++, Java, Python

② Markup language - HTML, SGML (② standard generalized), XML (② Extensible)

③ Scripting language - validation done at client side is client S.L; validation done at server side is server S.L

③ → client - Java script, React Js, node JS, Ember JS, angular JS, UBScript

→ Server - *Servelates, JSP (③), Php (④). Structs; spring a spring boot, Hibernate -

* ASP - .Link

→ Postgre - Powerful & lang.

pgm = OS + Algorithm (DAA)
↓
data structure

→ HTML is used for creating web pages & web applications. HTML is developed and maintained by world wide web consortion.

→ In HTML, the term hyper signifies the navigation from one location to another in a Non-linear fashion. That is, clicking a hypertext on a web page takes you to the relevant page on the Internet or website, which is not necessarily the next page on the website.

→ A markup lang is a computer lang. that is used to apply layout & formatting conventions to a text document. Markup lang. makes text more interactive & dynamic.

→ Protocol - It is a set of rules & regulations that must be agreed & abide between 2 communication parties.

Introduction to HTML

→ In 1989, Tim Berners Lee invented HTML

→ HTML is subset of SGML

→ It is not compiled lang & is directly interupted by a brower.

→ CUI - character user Interface

→ GUI - Graphical user Interface

→ NUI - Natural user Interface - pulse, heart-rate, user is interacted with machine naturally.

Features

→ Easy & simple lang

→ markup lang

→ Platform Independent

→ Effective & flexible

→ case insensitive

→ lang dependent - .net

→ .net is lang dependent & Java is lang independent, Platform dependent.. .net is Platform independent.

→ Each & Every tag maynot have attribute

→ macroon inc

Building blocks of HTML

① Tags - b/w <& >

② Attribute - provides Extra info

③ Elements

→ HTML Tags are like keywords, which defines that how web browser will format & display the content. with the help of tags, a web browser can distinguish b/w an HTML content & a simple content.

→ List, table form tags

→ HTML formatting tags - <b>, <I>, <U>, <strike>, <sup>, <sub>

image tag

→

anchor tag

→ html basic programs

→ < a href = "header.html"> header tags <a> < br/>

anchor & image tag

→ <a href = "https://www.javatpoint.com/what-is-robotics">
<img src = "robot.jpg" height = "200" width = "300"
alt = "robot image"> → by clicking picture the data is seen. (they are linked)

In output: 1st picture is displayed

TABLE TAGS

```
<table>
<tr><th> Firstname</th><th> Lastname</th> <th>Marks</th></tr>
<tr><td> Sanjay </td><td> Kumar </td><td> 60 </td> </tr>
<tr> <td> Vijay </td> <td> pratap</td> <td> 50</td> </tr>
</table>
```

o/p:

Firstname	Lastname	Marks
Sanjay	Kumar	60
Vijay	pratap	50

→ <rowspan> - the row gets divided into 2.

→ <td> - table data ; <tr> - table row ; <th> - table name.

```
<table>
<tr><th> Name </th><td> Srinivas </td></tr>
<tr><th rowspan="2"> MobileNo</th><td> 9412345678</td></tr>
<tr><td> 9876543214 </td></tr>
</table>
```

o/p:

Name	Srinivas
MobileNo	9412345678
	9876543212

→ <Colspan> - the columns gets divided

```
<table>
<tr><th> Name</th> <th colspan="2"> MobileNo.</th></tr>
<tr><td> Vijay </td><td> 9142687948</td>
<td> 9876543214 </td></tr>
</table>
```

o/p:

Name	Mobile no	
Vijay	9142687948	9876543214

```
<html>
<head>
<title> <h2> TIMETABLE </h2> </title>
<body>
<caption>
<table>
<tr> <th> align = "Center" DAY/TIME </th>
    <th> 9:45 - 10:35 </th>
    <th> 10:35 - 11:25 </th>
    <th> 11:25 - 12:15 </th>
    <th> 12:15 - 12:55 </th>
    <th> 1:00 - 1:50 </th>
    <th> 1:50 - 2:40 </th>
    <th> 2:40 - 3:50 </th>
</tr>
<tr>
    <th> MONDAY </th>
    <td align = "center"> WT </td>
    <td align = "Center"> DBMS </td>
    <td align = "center"> ACA </td>
    <td rowspan Colspan = "4"> LUNCH </td>
    <td align = "Center" colspan rowspan = "3"> WTLAB </td>
</tr>
<tr>
    <th> TUESDAY </th>
    <td align = "Center"> WT </td>
    <td align = "Center"> OWDM </td>
    <td align = "Center"> ACA </td>
    <td align = "Center"> DBMS </td>
    <td align = "Center" Colspan = "2"> LIB </td>
</tr>
</body>
</head>
</html>
```

HTML lists

① Numbered list → ordered list - follows numbering format; ol tag is used
② unordered list →
③ Description list

①

Type = "1" → default type, numbers
Type = "I" → uppercase Roman Numbers
Type = "i" → lowercase Roman Numbers
Type = "A" → uppercase letters
Type = "a" → lowercase letters

Ex* <ol type="i">
HTML Java SQL

O/p: i. HTML ii Java iii. SQL

* <ol type="A" start="5"> - starts from "E"

* <ol reversed>
 HTML   Java   SQL

O/p: 3. HTML 2. Java 1. SQL

②

- desc; ○ - circle; □ - square; none

Type = "disc" → default type, marked with bullets
Type = "circle" → marked with circles
Type = "square" → marked with square
Type = "none" → no mark

→ <dl> - description list
→ <dt> - data items
→ <dd> - description or definition

Ex <dl>
<dt> HTML </dt> <dd> is a Markup lang. </dd>
<dt> Javascript </dt> <dd> is a Scripting lang. </dd>
</dl>

Code

Q. MREC offered Courses

A. UG Course

a) CSE
   i, CSE Core
   ii, CSE DS
   iii, CSE AIML
   iv, CSE IOT

b) IT

c) ECE

d) EEE

B. PG Course

1. CE
   i, SE
   ii, GES

2. ECE
   i, ES ii, VLSI

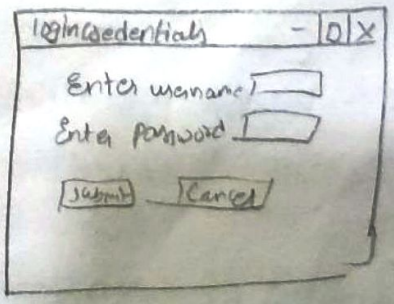
<Html>
<head>

HTML form

→ HTML form is a section of a document which contains controls such as text fields, submit buttons, passwords etc.

→ Label - read only text, info that need to entered is based on label.

Eg: <form action = "https://www.javatpoint.com/html-tutorial">
<label> Enter username </label>
<input type = "text" name = "username">
<label> Enter password </label>
<input type = "password" name = "password">
<input type = "submit" value = "Submit">
<input type = "reset" value = "Cancel">
</label>
</form>



radio tag – only one option will be selected

Eg <input type="radio" name="color" value="Red"> Red
<input type="radio" name="Color" value="Blue"> Blue

Checkbox – multiple options can be selected

eg <input type="checkbox" name="sport 1" value="cricket">Cricket
<input type="checkbox" name="sport 1" value="football"> Football

Button – Pop up window with given msg.

<form>
<input type="button" ~~click~~ value="Click me" onclick="alert ('You are using HTML')">

</form>

File

<form>
<label> select file to upload: </label>
<input type="file" name="newfile">
<input type="submit" value="submit">
</form>

image – used to represent a submit button in form of image

eg,
<input type="image" alt="submit" src="login.jpg" width="100px">

→ <required> – if the data that is to be entered is mandatory then it is used.

```
<form align="center">
<h1> Login Details </h1>
<label> Enter user name</label>
<input type="text" name="username" Placeholder="username"
required> <br><br>
<label> Enter password: </label>
<input type="Password" name="passwd" minlength="8" <required>
<br> <br>
<label> Gender: </label>
<input type="radio" name="gender" value="male"> MALE
<input type="radio" name="gender" value="female"> FEMALE
<br><br>
<label for="Email"> Enter your email: </label>
<input type="email" name="email"><br><br>
<label> Enter your phone no: </label>
<input type="tel" id="phone" name="phone" pattern="[0-9]{3}-
[0-9]{3}-[0-9]{4}"><br>
<br>
<label align="top"> Enter your Address: </label>
<textarea rows="5" cols="20"> Enter Here</textarea><br><br>
<label> Birthday: </label>
<input type="date" name="birthday"><br><br>
<label> Your Department</label>
<select name="depts">
<option value="CSE"> CSE</option>
<option value="EEE"> EEE</option>
<option value="ECE"> ECE</option>
<option value="IT"> IT</option>
</select><br><br>
<label> B. Tech Joined (month & year): </label>
<input type="month" id="monthyear" name="monthyear">
<br><br>
<label for="quantity"> Programming skills (between 1 & 5):
</label>
```

```
<input type="number" id=name="quantity" min="1" max="5">
<br><br>
<label>Enter your fav programming lang:</label>
<input type="checkbox" name="cse" value="C language">C Language
<input type="checkbox" name="cse" value="Python">Python
<br><br>
<label>Age limit for the Job is between 18 & 32</label><br><br>
<label>Enter a date after 1990-0-01:</label>
<input type="date" name="datemin" min="1990-01-2"
max="2003-12-1">
<br><br>
<label>Attach your resume:</label>
<input file="file" name="resume"><br><br>
<label (for="homepage" optional)>Add your homepage:</label>
<input type="url" (id="homepage") name="homepage"><br><br>
<label>Quantity (1 to 100):</label>
<input type="number" name="quantity" min="0" max="100" step="10"
value="20"><br><br>
<label>Volume (between 0 & 100):</label>
<input type="range" name="vol" min="0" max="100"><br><br>
<label>Select a time:</label> <input type="time" name="appt">
<br><br>
<label for="birthdaytime" (current)>Birthday (date and time):</label>
<input type="date-time-local" id="birthday-time" (current date)
name="birthday time" (current date)><br><br>
<label for="week">Select a week:</label>
<input type="week" id="week" name="week"><br><br>
<label>Select your favourite Color:</label>
<input type="Color" name="#ff00ff"><br><br>
<input type="button" value="click me" onclick="alert("Hello")">
<input type="submit" name="submit">
<input type="reset" name="reset"><br><br>
</form>
</body>
</html>
```

LOGIN DETAILS

Enter username

Enter password:

Gender: ○MALE ○FEMALE

Enter your email:

Enter your phone number:

Enter your Address:

Birthday: 08/16/2021

Your Department: CSE (CSE, ECE, EEE, IT)

B.Tech Joined (month and year): December 2020

Programming Skills (between 1 & 5):

Enter your favorite Programming lang. ☐ C language ☐ Python

Age limit for the Job is between 18 & 32

Enter a date after 1990-01-01 mm/dd/yyyy

Attach your resume: Choose file No file chosen

Add your homepage:

Quantity (1 to 100): 30

Volume (between 0 & 100):

Select a time: --:-- --

Birthday (date and time): mm/dd/yyyy --:-- --

Select a week: Week --, ----

Select your favourite Color:

Submit Reset

```
<optgroup label="Emerging CSE">
<option value="Data Science">DS</option>
<option value="IOT">IOT</option>
<option value="AIML" selected="selected">AIML</option>
<option value="CS">CS</option>
</optgroup>
```

O/P: Emerging CSE
Data Science
IOT
AIML
CS

```
<input type="tel" name="phone" pattern="[1-9]{1}[0-9]{9}">
```

Email

Pattern="/^[a-zA-Z0-9.!#$%&'*+/=?^'{|}~-]+@

Try to simplify [a-zA-Z0-9-]+(?:\b[a-zA-Z0-9]+)*$/"

Regex Quantifiers

x? → x occurs once or not at all

x+ → x occurs once or more times

x* → x occurs zero or more times

x{n} → x occurs n times only

x{n,} → x occurs n or more times

x{y,z} → x occurs at least y times but less than z times

Regex Character classes:

[abc] → a, b, or c (simple class)

[^abc] → Any character except a, b or c (negation)

[a-zA-Z] → a through z or A through Z, inclusive (range)

[a-d[m-p]] → a through d, or m through p: [a-dm-p] (union)

[a-z&&[def]] → d, e, or f. (intersection)

[a-z&&[^bc]] → a through z, except for b & c: [ad-z] (subtraction)

[a-z&&[^m-p]] → a through z, & not m through p: [a-lq-z] (subtraction)

. → it must contain

? → atleast once

pattern = "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[*.!@$%
^&(){}[]:;<>,.?/~_+-=|\]).{8,32}$"

HTML Comments

Syntax: <!-- Comment here -->

Stringstr = "/"hi"/"; - strings, when they are need to be Pointed on screen

Ex: <!-- Remember to add information here -->

htmlcolorCodes.com → website used for colors.

HTML Colors

Ex: rgb(255,0,0) ⇒ red, rgb(0,255,0) ⇒ green, rgb(0,0,255) ⇒ blue,
rgb(0,0,0) ⇒ black & rgb(255,255,255) ⇒ white

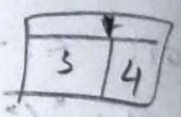
#ff0000 ⇒ red, #00ff00 (green), ~~rgb(~~ #0000ff (blue),
#000000 ⇒ black

```
<html>
<body>
<marquee width="100%" behaviour="scroll" bgcolor="Pink">
This is an example of a scroll marquee..
</marquee>
<marquee width="60%" direction="right" height="100px">
This is a Sample Scrolling text that has Scroll texts to right.
</marquee>
<marquee width="60%" direction="left">
This is a Sample Scrolling text that has scroll texts to left
</marquee>
<marquee behaviour="scroll" direction="left">
This text scrolls from left
</marquee>
</body>
</html>
```

HTML frames

①
```
<html>
  <frameset rows="35%, 65%">
    <frame src="frame1.html" />
    <frame src="frame2.html" />
    <frameset cols="60%, 50%">
      <frame src="frame3.html" />
      <frame src="frame4.html" />
    </frameset>
  </frameset>
</html>
```

O/P:



Cascading style sheet (CSS):

→ Explains (described by) how elements are displayed on screen, paper or in other media

→ We can add new looks to our HTML documents.

→ It can also be used with HTML to change the style of web pages & user interfaces. We can completely change the look of our website with only a few changes in CSS code

Syntax: `h1 { Color : blue ; font-Size : 12px ; }`

(h1 → selector; Color → property; blue → value; font-Size → property; 12px → value; Color & font-Size: Declaration)

→ Multiple property & values can be added after the semicolon

eg,
```
p
{
  Color: red; /* this is a Comment */
  textalign: center;
}
```

→ Comments are ignored by browsers.

CSS Selectors

→ CSS Selectors are used to select the content you want to style. Selectors are part of CSS rule set.

→ There are several different types of selectors in CSS. They are:

1. CSS Element Selector
2. CSS Id Selector (#)
3. CSS class selector (.)
4. CSS universal selector (*)
5. CSS Grouping selector

1. CSS Element Selector

→ It selects HTML elements based on the element name.

```
<html>
<head>
<style>
body {
    background-color:green;
}
h1{
    color:red;
    margin-left:80px;
}
</style>
</head>
<body>
<h1> the internal style sheet is applied on this heading </h1>
<p> This paragraph will not be affected </p>
</body>
</html>
```

2. CSS Id Selector

→ The id selector is attribute based.

```
<html>
<head>
<style>
#para1 {
```

```
    text-align : Center;
    Color: blue;
}
</style>
</head>
<body>
<p id="para1">Hello MREC CSE</p>
<p>this paragraph will not be affected</p>
<h1 id="para1">Hello MREC CSE2222</h1>
</body>
</html>
```

3. CSS class Selector

```
<html>
<head>
<style>
.Center {
        text-align : center;
        Color: blue;
}
</style>
</head>
<body>
<h1 class="center"> this heading is blue & center-aligned.</h1>
<p class="center"> this para is blue & " </p>
</body>
</html>
```

4. CSS universal Selector

<style> * {Color: green; font-size : 20px ;} </style>

→ * → works for entire program

5. CSS grouping Selector

<style> h1, h2, p {text-align: center ; Color : blue;} </style>

→ groups all elements & is applicable for elements mentioned

How to Add CSS

3 types — ① Inline CSS - applicable for that particular tag
tags are inserted - ② Internal CSS - used to add a unique style for a single doc.
tags are inserted - ③ External CSS through link.

① <h2 style="color:red; margin-left:40px;"> Inline CSS is applied on this heading </h2>

② your element (selector pgm

③

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

---

mystyle.css

```
body {
    background-color: green;
}
h1 {
    color: navy;
    margin-left: 20px;
}
p
{
  color: red;
  font-size: 24px;
}
```

@import style sheets

```
<style> @import url("mystyle.css"); </style>
```

→ It is an alternative for external CSS

A. CSS Background.

① CSS Background color - specifies background color of the element.

```
h2,p {
    background-color: #00ff00;
}
```

② CSS background image -

```
body {
    background-image : url("flower1.jpg");
    margin-left: 100px;
}
h1 {
    color: white; margin-left: 20px;
}
```

③ CSS background repeat

background-repeat : repeat-x; [as per requirement x and y axis is taken].

④ CSS background attachment

⑤ CSS background position

Eg: body { background : #ffffff url("img.png") no-repeat fixed right top; }

B. CSS Border

① CSS Border styles

```
<style>
  P.dotted { border-style: dotted; }
  P.dashed { border-style: dashed; }
  P.double { border-style: double; }
</style>
<body>
  <P class="dotted"> A dotted border </P>
</body>
```

② CSS Border width

```
<style>
  P.one {
      border-style : solid;
      border-width: 5px; / medium / 1px
  }
</style>
```

③ CSS Border Color

border-color: red;

Eg: P { border: 5px solid red; }

C. CSS font [fonts.google.com]

① css font color

```
body {
    font-size: 100%
}
h1 { color: red; }
h2 { color: #9000A1; }
P { color: rgb(0,220,98); }
}
```

② font size - Small, medium, Large

PHP

→ Hypertext preprocessor - PHP

→ Open Source, interpreted (→ no need of compilation), object-oriented scripting language that can be executed at Server side.

→ used to develop web applications

→ faster than other scripting lang.

Server side scripting lang., used to manage

Features

→ Performance

→ Control

→ familiarity with syntax

→ Embedded

→ Platform independent

→ Database Support

→ Error Reporting

→ Loosely Typed lang.

→ Web servers support

→ Security

Prerequisite

→ HTML - HTML is used to design static webpage

→ CSS - CSS helps to make the webpage content more effective & attractive

→ Java script - Java script is used to design an interactive website.

→ XAMPP (Cross, Apache, MYSQL, PHP, Perl)

①
```
<doctype!html>
<html>
<body>
<? php
 echo "<h2>Hello First PHP </h2>";
 ?>
</body>
</html>
```

② Save the file with hello.php in ~~htdocs~~ mdec folder, which resides inside the xampp folder.

③ localhost/mdec/hello.php

```
<html>
 <body>
 <? php
     echo "Hello echo 1 <br> ";
     ECHO "Hello ECHO </br>";
     EcHo "Hello EcHo <br>";
  ?>
</body>
</html>
```

Syntax of echo:

Void echo (string $arg1 [, string $...])

echo → used to print the string, multiline strings, escaping characters, variable, array, etc.
- → faster than Print statement
- → Can be used with or without paranthesis
- → doesnot return any value

```
<?php
echo "Hello by PHP echo1";
echo "Hello by PHP echo 2
  this is multiline
  text printed by
  PHP echo statement
  ";
echo "Hello php3 escape \sequence\ characters";
$msg = "Hello PHP4";
echo "Message is: $msg";
?>
```

Syntax of PHP

1. int print (String $arg)

```php
<?php
echo "Hello by PHP echo1";
echo "Hello by PHP echo 2
this is multiline
text printed by
PHP echo statement
";
echo "Hello Php3 escape \sequence\ characters";
$msg="Hello PHP4";
echo "Message is: $msg";
?>
```

Syntax of PHP

1. int Print (String $arg)

```php
<?php
$str="hello string";
$x=200;
$y=44.6;
echo "string is: $str <br/>";
echo "integer is: $x <br/>";
echo "float is: $y<br/>";
?>
```

Add of 2 no's

```php
<?php
$x=5;
$y=6;
$z=$x+$y;
echo $z;
?>
```

→ PHP Variables are Case Sensitive but keywords are not as

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

```
<? php
$a="hello";    // valid
$_b="hello"; // valid
echo "$a<br>$_b";
$4a="hello1"; // Invalid
$*b="hello1"; // Invalid
echo "$4a<br>$*b";
?>
```

PHP: loosely typed language → automatically converts the data

PHP: Variable Scope

① Local variable
② Global Variable
③ Static Variable

①

```
<?php
  function localVar()
  {
    $lang="PHP";
    echo "web development language ".$lang;
  }
  localVar();
// echo $lang; // Error is generated
?>
```

(2)

```
<?php
$name = "Sanaya Sharma";
function global_var()
{
 global $name;
 echo "Variable inside the function: ".$name;
 echo "<br>";
}
global_var();
echo "Variable outside the function: ", $name;
?>
```

→ Without using the global keyword, if you try to access a global variable inside the function, it will generate an error that variable is undefined.

→ Local variables has higher priority than the global variable.

(3)

→ Static value will be stored globally & value gets changed i.e., incremented after function call.

→ Non-Static is Const. value

```
<?php
 function static_var()
 {
  Static $num1=3; // Static variable
  $num2=6;        // Non-Static
  $num1++;        // increment on non-static
  $num2++;        // increment on static
  echo "Static: ".$num1."<br>";
  echo "Non static: ".$num2."<br>";
 } Static_var(); // 1st function call
  Static_var(); // 2nd function call
?>
```

→ $var (single dollar) - normal variable [int, float, str, etc]
→ $$var (double dollar) - reference variable [value of $var]

```
<?php
$name="cat";
$($name)="Dog";
$($($name))="Monkey";
echo $name."<br>";          - cat
echo $($name)."<br>";       - Dog
echo $cat."<br>";           - Dog
echo $($($name))."<br>";    - Monkey
echo $Dog."<br>";           - Monkey
?>
```

PHP Data types

① Scalar Types (Predefined)
② Compound Types (user defined)
③ Special Types

① → boolean, integer (decimal, octal, hexadecimal), float, String - 4 types
② (1) TRUE, FALSE (0)
→ Array, object - 2 types
③
→ resource, null - 2 types

Integer

```
<?php
$dec1=34; $oct1=0243; $hexa1=0x45;
echo "Decimal no: ".$dec1."<br>"; //34
echo "octal no: ".$oct1."<br>"; //163
echo "hexa no: ".$hexa1."<br>"; //69
?>
```

Php Array

An array is a compound datatype, stores multiple values of same datatype in single variable.

```
<?php
    $bikes = array ("Royal Enfield", "Yamaha", "KTM");
    var_dump ($bikes); // var_dump() func returns the datatypes & values.
    echo "<br>";
    echo "Array Element 1: $bikes[0] <br>"; // Royal Enfied
    echo "Array Element 2: $bikes[1] <br>"; // Yamaha
    echo "Array Element 3: $bikes[2] <br>"; // KTM
?>
```

PHP object

Objects are the instances of user-defined Classes that can store both values & functions. They must be explicitly declared.

```
<?php
    class bike {
        function model() {
            $model_name = "Royal Enfield";
            echo "Bike model: ".$model_name;
        }
    }
    $obj = new bike();
    $obj -> model();
?>
```

```
<?php
    $a = 40; $b = 5;
    function arithmetic()
    {
        global $add;
        echo "add value is: ".$a+$b;
        global $sub;
        echo "Sub Value is: ".$a-$b;
```

```
echo<br>
}
arithmetic()
?>
```

```
<? php
 $a=4; $b=5;
```

→ ascii value of NULL is 0

→ String is a static, contiguous memory location

→ ~~array is~~

→ Every String should be calculated by null value

```
<? php
  $nl=NULL;
  echo $nl;  //it will not give any output
?>
```

PHP operators

→ Operator is a symbol used to perform operations on operands.

→ 3types
- unary – operation performed on one operand
- Binary – operation performed on 2 operands
- Ternary – operation performed on 3 operands

1. Arithmetic operators

+, -, *, /, %, **

% → Modulus

** → Exponent

17%4 = 1

-17%4 = -1

17%-4 = 1

-17%-4 = -1

2. Assignment operators

1. Arithmetic operators

+, -, *, /, %, **
% → Modulus, ** → Exponent

17%4 = 1
-17%4 = -1
17%-4 = 1
-17%-4 = -1

$a & $b

2. Assignment operator

=, +=, -=, *=, /=, %=

→ In Rhs, the value, variable, constants are taken; In Lhs only variables are used.

3. Bitwise operator

→ Bit level manipulation

&, |, ^, ~
AND, OR, XOR, Not

4. Comparing operator (Relational)

==, ===, !==, !=, <>, <, >, <=, >=, <=>
Equal, Identical, Not identical, Not equal, Spaceship

→ Return -1 if $a is < $b, Return 0 if $a = $b, Return 1 if $a > $b

5. Increment / decrement

++, --

int.

6. Logical operator (Boolean operator)

and, or, xor, !, &&,

```
if(m<40 || p<40 || c<40)
 {
  Pf("Fail")
 }
Else
 {
  Pf("Pass")
 }
```

7. String operators

. → concatenation of 2 strings

.= → Concatenation assignment

$a = $a.$b

8. Array operators

+, ==, !=, ===, !==, <>

union, Notequal, Identity, Not equal

Type Operators

→ instance of : used to determine whether an object,

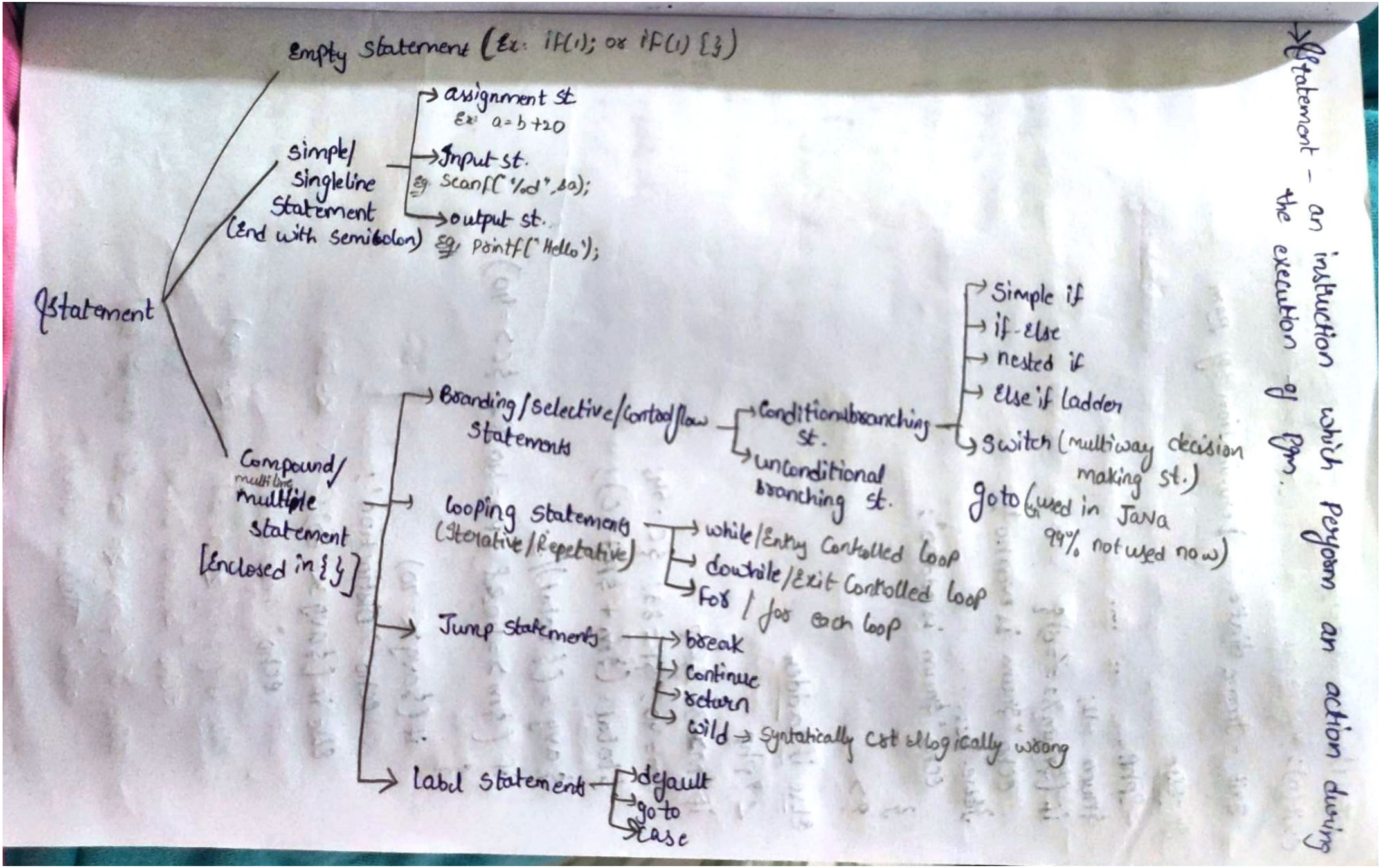
```
<?php
//class declaration
class Developer
{}

class programmer
{}
//creating an object of type developer
$charu=new Developer();
//testing the type of object
if($charu instanceof Developer)
{
  echo "charu is a developer";
}
else
{
echo "charu is a programmer";
}
echo "</br>";
var_dump($charu instance of Developer);
var_dump($charu instanceof Poogrammer);
?>
```

o/p: charu is a developer

bool(true) bool(false)

→ Statement - an instruction which perform an action during the execution of pgm.



exit(1); - normal termination → C → <stdio.h>
exit(0); - abnormal " → C++ <process.h>

→ Exit - Immediate termination of Pgm

If-Else

```
<?php
$num =12;
if ($num%2==0){
  echo "$num is even no";
}else {
  echo '$num is odd no';
}
?>
```

Else-if ladder

```
<?php
if(m>=35 && p
  $m=40; $p=83; $c=54;
  $total=($m+$p+$c);
  $avg=($total)/3;
  if ($m>=40&& $p>=40 && $c>=40)
  {
    if($avg>=70)
      echo "Distinction";
    else if ($avg<=60&& avg<=70)
      echo "first class";
    else if ($avg<=50 && avg<=60)
      echo "second class";
    else if ($avg<=40 && avg<=50)
```

Switch Case

→ The default is an optional statement. It must be last statement.

→ There can be only one default in a switch case.

→ PHP allows you to use number, character, string as well as func in switch expression.

→ Nesting of switch st. is allowed, but it makes the pgm more complex & less readable.

→ You can use semicolon (;) instead of colon(:). It will not generate any error.

PHP switch statement with string

```
<? php
$ch = "B. Tech";
switch ($ch)
{
  case "BCA":
    echo "BCA is 3yos Course";
  break;
  case "BSC":
    echo "BSC is 3 yos Course";
  break;
  Case "B.Arch",
    echo "B.Arch is 5yrs Course";
    break;
default:
    echo "wrong choice";
break;
}
?>
```

}

PHP Switch Statement using fall through

In the Switch case the break Statement should be used. If the break Statement is not found then the corresponding Statements are executed continuously.

```
<?php
$ch='c';
switch($ch)
{
  case 'a':
    echo("a is variable");
  break;
  case 'c':
    echo "c is variable";
```

```
case'd':
    echo "d is variable";
break;
default:
    echo" Wrong choice";
}
?>
```

For loop

```
<?php
for($n=1; $n<=10; $n++) {
  echo "$n<br/>";
}
?>
```

Palindrome

```
<?php
n=153;
while(n>0)
{
  r= n%10;
  s = (s*10) +r;
  n=n/10;
}
```

Step ① n=153, s=0

while (153>0)

{

r= 153%10 = 3

s = (0*10)+3 = 3 =3

n= 153/10 =15

}

Step ② n=15 , s=3

while (15>0)

{

r= 15%10 = 5

s=(3*10)+5 = 35

n= 15/10 = 1

}

Step ③ n=1, s=3

while (1>0)

{

r= 1%10 = 1

s = (35*10) +1 = 351

n= 1/10 = 0

}

Step ④ n=0 , s= 351

while (0<0)

// loop ends

}

Count of numbers

14572

```
<?php
n=14572; count=0
if(n>0)
{
  r=n%10;
  count++;
}
echo "count
```

~~①~~ (14572)>0

{

r= 14572/10 = 1457

count++ ⇒ 1

} n=r;

② 1457 >0

{

r= 1457/10 = 145

count++ ⇒ 2

n=145;

}

③ 145>0

{

r=145/10= 14

count++ ⇒3

} n=14;

④ 14>0

{

r=14/10 = 4

count++ ⇒4

} n=4;

⑤ 4>0

{

r= 4/10 =0

count ++ ⇒ 5

}

⑥ 0>0

// loop Ends

Nested loops

foreach loop

```
<?php
$season = array("Summer", "winter", "Autumn", "Rainy");
foreach ($season as $element) {
    echo "$element";
    echo "<br>";
}
?>
```

Associative array element through foreach loop

```
<?php
$employee = array{
        'Name' => "Alex",
        "Email" => "alex_jtp@gmail.com",
        "Age" => "21",
        "Gender" => "Male"
    };
foreach ($employee as $key => $element) {
    echo $key." : ".$element;
    echo "<br>";
}
?>
```

function el method diff

PHP Functions

without argument

```
<?php
function sayHello(){
  echo "Hello PHP function";
}
 sayHello();
?>
```

with argument

```
<?php
 function sayHello($name){
   echo "Hello $name<br>";
}
 sayHello("Sonoo");
 SayHello("John");
?>
```

PHP Call by Reference el Call by value

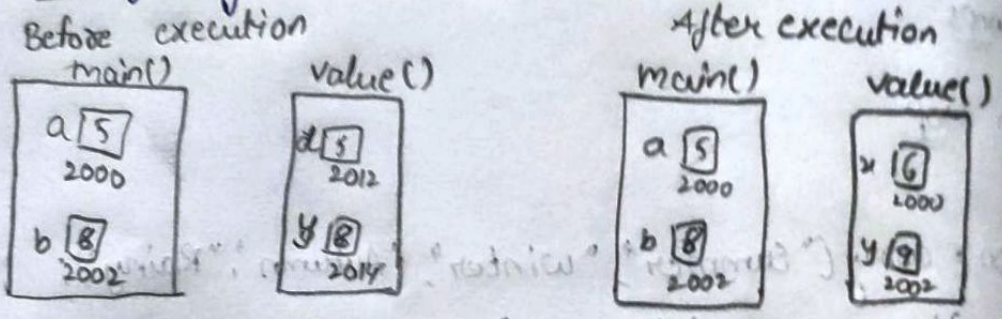


Fig: call by value [only the formal value gets incrementation of value changed actual value is const.]

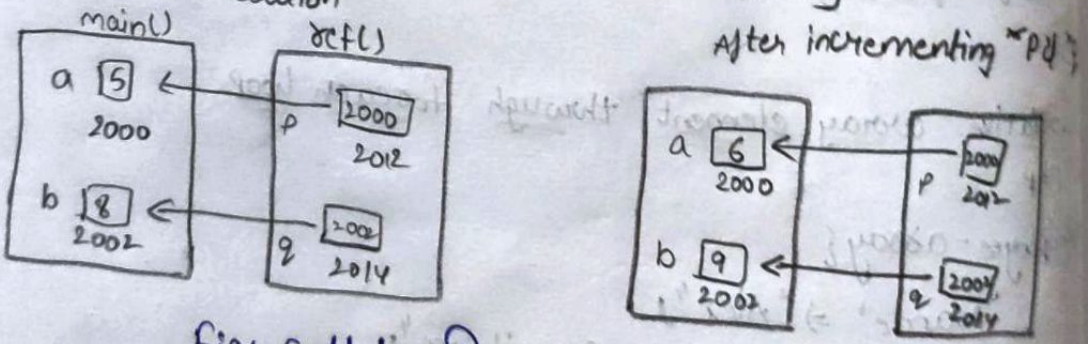


Fig: Call by Reference [both formal el actual value gets changed]

```
<?php
function adder(&$str2)
{
  $str2.= "Call by Reference";
}
$str = "Hello";
adder($str);
echo $str;
?>
```

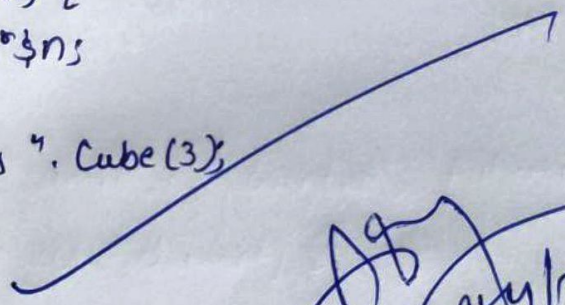
o/p: Hello Call by Reference

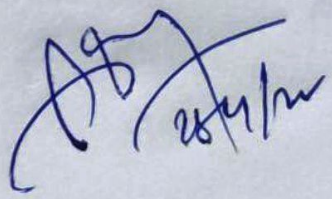
Default argument

```
<?php
function sayHello ($name="sonoo"){
  echo "Hello $name <br>";
}
  Say Hello ("Rajesh");
  say Hello ();
?>
```

function that returns value

```
<?php
function cube($n) {
  return $n*$n*$n;
}
echo "Cube of 3 is ". Cube(3);
?>
```





NAWR

```
#include<stdio.h>
int add( )
{
  int a=4, b=5;
  s=a+b;
}return(s)
void main()
{
 int x=add();
  printf(x);
}
```

NANR

```
Void add()
{

}
void main()
{
  add();
}
```

NAWR

```
void add(int x, int y)
{
 pf(x+y)
}return
Void main()
{
 int a=10, b=20;
 add(a,b);
}
```

PHP Parameterized function

```
<!DOCTYPE Html>
<html>
<head>
<title> Parameter Addition & Subtraction Example </title>
</head>
<body>
<?php
  // Adding 2 numbers
 function add ($x, $y)
 {
   $sum = $x + $y;
   echo "sum of two numbers is =$sum <br>";
}
add (467,943);
// subtracting 2 numbers
```

```
function sub($x,$y)
{
 $diff = $x-$y;
 echo " Difference between 2 numbers is = $diff ";
}
sub(943,467);
?>
</body>
</html>
```

PHP Variable length Argument function

It means you can pass 0,1, or n number of arguments in function. we need to use 3 ellipse (dots) before the argument name.

```
<?php
function add (... $numbers)
{
 $sum=0;
 foreach ($numbers as $n)
 {
  $sum += $n;
 }
 return $sum;
}
echo add(1,2,3,4);
?>
```

Generic Servlet class

→ Handle any type of request so it is protocol independent

→ Methods of Generic Servlet class

1. Public void init (ServletConfig Config) is used to initialize the servlet.
2. Public abstract void service (ServletRequest request, ServletResponse response)
3. Public void destroy()
4. public ServletConfig getServletConfig() - returns object of ServletConfig
5. Public String getServletInfo()

→ Garbage collection — Mark & sweep - data can be removed [unused data]

→

```
Public class first extends GenericServlet
{
  Public void service (ServletRequest req, ServletResponse res) throws
  IOException, ServletException,
  {
    res.setContentType ("text/html");
    PrintWriter out = res.getWriter();
    out.print("<html><body>");
    out.print ("<b>Hello generic servlet</b>");
    out.print ("</body></html>");
  }
}
```

http Servet class

→ Extends the generic servlet class

Methods

① Public void service (ServletRequest req, ServletResponse res)

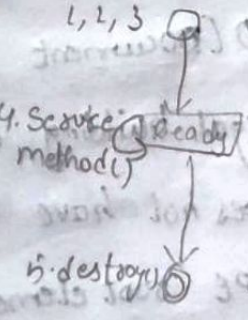
② protected void service(HttpServletRequest req, HttpServletResponse res)

③ protected void doGet(HttpServletRequest req, HttpServletResponse res)

④ protected void doPost(HttpServletRequest req, HttpServletResponse res)

② → receives the request from the service method, & dispatches the request to the doXXX() method depending on the incoming http request type.

Life cycle of servlet



1. Servlet class is loaded
2. Servlet instance is created
3. init method is invoked
4. service method is invoked
5. destroy method is invoked.

XML

→ Extensible Markup lang.
→ designed to store & transport data
→ designed to be self descriptive
→ designed to carry data, not to display data
→ No predefined tag, HTML have predefined tags
→ Except XML, all other words are user defined tags
→ case sensitive tags
→

<	- lessthan
>	- greater than
&	- ampersand
'	- apostrophe
"	- quotation mark

→ white space is preserved in XML
→ Avoid "-", ".", ":", ";" as the names of functions
→ Snake case <first_name>
Pascal Case <FirstName> - class [C lang]
Camel case <firstName> - methods [lang]
→ Rules
* one unique element
* begin with XML declaration
* Attribute values must be quoted.

XML DTD (Document Type Definition)
→ defines legal building blocks of an XML document
→ DTD does not have datatype.
→ <!DOCTYPE root element - mandatory in DTD/~~XML~~

CRUD
C → create (Insert a Record)
R → Read (Retreival of record)
U → update (updation of existing record)
D → delete (Deletion of existing record)
→ ODBC - Open database Connectivity.
→ How to create users w.r.t database (Toy)
MySQL - 8 (GUI & CUI); Notepad++;
Structured Connection

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$con = mysqli_connect("$servername", "$username", "$password");
if($con)
{
```

```
die("sorry we failed to connect: ". mysqli_connect_error());
}
else
{
echo "Connection was successful";
}
echo "<br>";
?>
```

localhost/mrec/

files are displayed

Creation of table

```
<?php
$host = 'localhost';
$user = 'root';
$pass = '';
$dbname = 'testdb';
$conn = mysqli_connect($host, $user, $pass, $dbname);
if(!$conn) {
    die('Could not Connect: '. mysqli_connect_error());
}
echo 'Connected successfully <br/>';
$sql = 'create table emp1 (id INT AUTO_INCREMENT, name
VARCHAR(20) NOT NULL, salary INT NOT NULL, primary key(id))';
if (mysqli_query($conn, $sql)) {
    echo "Table emp1 created successfully ....!!! ";
}else {
echo "could not create table: ". mysqli_error($conn);
}
mysqli_close($conn);
?>
```

Insertion of value in table

```
$sql='INSERT INTO empl(name,salary) VALUES('Raju',8000);
if(mysqli_query($conn,$sql)){
  echo"Record inserted successfully";
}else{
  echo" could not insert record:".mysqli_error($conn);
}
mysqli_close($conn);
?>
```

update

```
$id=3;
$name="Rahul";
$salary=8000;
$sql='update empl name="$name", salary=$salary where
id=$id';
if(mysqli_query($conn,$sql)){
  echo" Record update successfull";
}
else{
```

```
echo "Couldnot Update record : ". mysqli_error($conn);
}
mysqli_close($conn);
}
```

Delete

```
$id=2;
$sql="delete from emp1 where id=$id";
```

From

```
$sql = SELECT *FROM emp1;
$retval = mysqli_query($conn, $sql);
if(mysqli_num_rows($retval)>0) {
  while($row=mysqli_fetch_assoc($retval)) {
      echo "EMP ID: {$row['id']} <br>";
        "EMP NAME : {$row['name']} <br>",
        "EMP SALARY: {$row['salary']} <br>",
        "..... <br>";
  } //end of while
}
else {
  echo "0 results";
}
mysqli_close($conn);
}
```

26/22

Life cycle of computer - Input, Processing, Output, Storage

How to run Java project

Step 1: open Eclipse and click file > New > Java point

ctrl + spacebar – the content of text will be expanded (full form)

ctrl + f11 – Execution

ctrl + shift + → Zoom in & zoom out

Step 2: Provide project Name & Click on Finish Button

Step 3: In the Package Explorer (Left side of window) Select the project which you have created.

Step 4: Right-click on the src folder, select New > class from submenu.

Step 5: write the program and save it

Step 6: Now, press ctrl + f11 or click on the Run menu & Select Run or Click on Run button ▷

Creating Servlet Example in Eclipse

* Create a Dynamic web project
* Create a Servlet
* add Servlet-api.jar file
* Run the Servlet

Step ①: For creating a dynamic web project click on file Menu New → Project → web → dynamic web project → write your project name e.g. first → Finish.

Step ②: for creating a Servlet, explore the project by clicking the + icon → explore the Java Resources → right click on src → New → Servlet → write your Servlet name eg. Hello → uncheck all the checkboxes except doGet() → next → finish

Step ③: For adding a jar file, right click on your project → Build Path → Configure Build path → (To active click Class path) click on libraries tab in Java Build Path → click on Add External JARs. button → Select

the Servlet-api.jar file under tomcat/lib → ok

Step ①:
For Starting the Server & deploying the project in one Step,
Right click on your project → Run as → Run on Server →
choose tomcat Server → next → add All → finish.

DemoServlet.java

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
Public class DemoServlet extends HttpServlet {
  Public Void doGet (HttpServletRequest req, HttpServletResponse res)
   throws ServletException, IOException
  {
   res.setContentType("text/html");
   PrintWriter pw=res.getWriter();
   // writing html in the stream
   pw.println("<html><body>");
   Pw.println("welcome to servlet");
   pw.println("</body></html>");
   Pw.close();
  }}
```

→ Streams < Byte < I/P Stream, O/P Stream
Character < Reader, Writer

→ Concrete class - Each & every method Should be implemented in code / simple method

→ Package < classes < Concrete, Abstract, Wrapper -
Interface - gives

→ int class is a wrapper class

→ String to array - charArray
Array to string - toString

XML data

```
<sname> Raju </sname>
<age> 36 </age>
<dob> 1970-03-27 </dob>
```

XML Schema

```
<xs:element name="sname" type="xs:string"/>
<xs:element name="age" type="xs:integer"/>
<xs:element name="dob" type="xs:date"/>
```

default

```
<xs:element name="color" type="xs:string" default="red"/>
```

Restrictions

```
<xs:element name="age">
 <xs:simpleType>
  <xs:restriction base="xs:integer">
  <xs:minInclusive value="0"/>
  <xs:maxInclusive value="100"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

Indicator → all - the values should be given
→ choice - one value should be specified compulsorily
→ sequence