UNIT-1

PHP

1. Introduction, overview, characteristics, Advantage, disadvantages of PHP
2. Declaring Variables in PHP
3. Data types in PHP [Integer, Boolean, Double, String, Array, object, NULL, Resource]
4. Operators in PHP [precidence & Associativity]
5. Expressions in PHP.
6. Session handling in PHP
7. Cookies in PHP
8. Control Structures [Conditional statements (if, ifelse, elseif, Switch) Control loops (while, do while, for, for each)]
9. file handling in PHP (open, read, write, close)
10. handling file uploads
11. Form handling (or) form Controls (Get and post method)
12. Listing Directories
13. Scope and lifetime of a Variable in PHP
14. Connecting to database [MYSQL as a reference]

UNIT-1:

1) Introduction to PHP.

⇒ PHP was developed in the year 1994 by Apache group.

⇒ It is a widely used open Source general purpose Scripting language that is Especially used for web development

⇒ PHP Stands for hyper text preprocessor Earlier it is called as personal homepage.

⇒ PHP is a Server Side Scripting language. It is mainly used for form handling, database Access, web development.

⇒ It is free to download & use.

Syntax:-
```
<?php
// code is written here
?>
```

Example:-
```
<?php
echo "Hello";
?>
```
it is Statement which is used to generate output

* A php Script Starts with <?php and ends with ?>
* In above example we use built in PHP function "echo" to output "Hello" on web page
* php statements ends with Semicolon (;)

Overview of php:-

⇒ php is a Server Side Scripting language Embedded in XHTML. It is an Alternative to CGI, ASP, ASP.NET, JSP.

- JSP → Java Server Page
- CGI → Common Gateway Interface
- ASP → Active Server page

⇒ The Extension to php files are .php, .php3, phtml

⇒ php makes use of dynamic typing that means there is no need to declare Variable type in php.

⇒ php has large number of library functions which makes it flexable to develop code of php

⇒ php processor works in two modes

(a) If php processor finds XHTML tags in the php Script then code is Simply Copied to output file.

(b) But when php processor finds php code in Script then code is Simply understand and output is Copied to output file.

characteristics:-

(1) open Source: we Can download php with free of Cost in Interne

(2) Simplicity: Since php donot include library functions like C, C++, So its Structure is Simple.

(3) Efficiency: It uses Resource Allocation mechanisms, object oriented programming, Session management features, So it eliminates the unnecessary memory Allocation.

(4) Security: It Support Encryption functions for Security

(5) Flexability: It is very flexable language because it can be embedded with HTML, CSS, JavaScript, XML and many other languages. and the php code can be run on any devices like phone, Tabs, laptops etc..

(6) object Oriented: php Support oop's features.

Advantages:

(1) It is Supported by All operating Systems like windows, linux, etc.

(2) It is Integrated with other programming languages like HTML, CSS, Java Scripts etc..

(3) It is easy to Connect with the database to store and retrieve data from database

(4) It is fast programming language compared to other programming languages.

(5) php Frameworks & tools are used to protect web Applications from Attacks & threats

Disadvantages

① It is not Suitable for large Applications because its maintanance is difficult

② Error handling of php framework is not good.

② Declaring Variables in PHP :-

⇒ Variables are the Entities that are used for storing the values.

⇒ PHP is a dynamically typed language ie, PHP has no type declaration.

⇒ The value Can be Assigned to the Variable in following manner.

Syntax: `$variable_name = value;`

⇒ If the value is not Assigned to the Variable then by default the value is "NULL". and this unassigned Variables are called unbound Variables and its null value is converted to the value 0.

Example:

```
<?php
$txt = "hello";
$x = 5;
$y = 10.5;
?>
```

→ when you Assign text value to a Variable then put Quotes around it.

Rules that must be followed while using Variables:

(1) php Variables starts with $ sign by Variable name and php Variable names are Case Sensitive: ie php treat Capital letters and Small letters differently.

Example: `$x = 10;`

`$a = 10;` `$A = 10;`

both are different

(2) Variables must start with letter or under Score.

Examples { `$x = 10;` `$_b = 10;`

(3) Variables must not start with number.

(4) Variables must Contain only Alpha numeric Values.

(5) there should not be Space in the name of the Variable.

Example 2:

```
<html>
<body>
<?php
$x = 5;
$y = 10;
echo $x + $y;
?>
</body>
</html>
```

output :- 15

```
<!DOCTYPE html>
<html>
<body>
<?php
$txt = "Avanthi College";
echo "I love" . $txt;
?>
</body>
</html>
```

output

I love Avanthi College.

(3) DATATYPES IN PHP :-

there are four Scalar types that are used in PHP.

they are

(a) Integer type
(b) Boolean type
(c) Double type [(or) float]
(d) String type

Scalar types

(a) ~~Integer~~ ⇒ Variables Can store data of different types, and different datatypes Can do different things.

⇒ PHP also Supports datatypes Such as Compound type and Speval type

① Array
② object
> Compound types

③ NULL
④ Resource
> Special types

lege.

① Integer type:

It is used for displaying Integer values. It is similar to long datatype in 'c'. Its size is 32 bit.

Example: $x = 5;

⇒ An integer must have at least on digit. and it must be non decimal and it can be either positive or negative.

② Double type:

It is used for displaying Real values.

Example: $x = 10.5;

It includes number with decimal point. It is not compulsory to have ~~numbers~~ digits before & after decimal point.

③ Boolean types:

there are two types of values that can be defined by Boolean type they are true & false

true is Represented by 1 and false is Represented as 0.

Example: $x = true;
$x = false;

⇒ If Boolean Values are used in Context of double type then False is Represented as 0.0 and true is Represented as 1.1

String type:

⇒ there is no character datatype in PHP. If character has to be represented then, it is Represented using String datatype.

⇒ String is Collection of characters. In php String is ~~denoted~~ defined by Single or double Quotes

⇒ The Simple php Script that Stores the String in a Variable is given below.

```
<?php
$A = "I like php";
echo $A;
?>
```

o/p: I like php

following are the various functions used for string handling

① strlen(string) :- find length of string

② strcmp(string1, string2) :- this function compares string1 and string2

③ strtolower(string1) :- converts characters in string1 to lower case

④ strtoupper(string1) :- " " " " upper case

⑤ trim(string1) :- this function Eliminates the white space from both ends of string.

① NULL :- NULL is a special datatype in PHP which can have only one value : NULL.

⇒ A variable of data type NULL is a variable that has no value assigned to it.

⇒ if a variable is created without a value, it is automatically assigned a value of NULL.

⇒ Variables can also be emptied by setting the value to NULL

Example:

```
<?php
$x = "sai";
$x = null;
var_dump($x);
?>
```

It is used to dump information about variable.

output: null

Resource in PHP:-

⇒ The Special resource type is not an Actual datatype. it is the storing of a reference to functions and resources external to PHP.

Object type:-

⇒ php is object oriented programming language ie, we can create objects in php. object is Real world Entity. and this object is Compound data type.

⇒ class is a template for object and object is an instance of class. In php we can create number of classes in a class and we can include number of objects in it. This object Contains properties and Action. In class we need to write class name starting with Capital letter

Example:

```
<?php
class Fruit {
    == } code is written here
}
?>
```

Capital letter

To declare an object of a class we need to use new statement

```
class Myclass
{ ==
}
$obj = new Myclass;
```

```php
<?php
$obj = new stdclass;
$obj -> name = "Deepak";
$obj -> age = 21;
$obj -> marks = 75;
print_r ($obj);
?>
```

} object of stdclass is null to begin with. we can add properties dynamically

output:-
```
Std Class object (
    [name] => Deepak
    [age] => 21
    [marks] => 75
)
```

Arrays:-

⇒ Arrays in php is a type of data structure that allows us to store multiple elements of similar datatype under a single variable there by saving us the effort of creating a different variables for every data.

⇒ Array is a collection of similar type of elements and in Php we can include elements of mixed type together in single Array.

There are two ways to Create Arrays in php

using Construct Array:-

```php
$mylist = array [10, 20, 30, 40, 50];
```

Assigning value directly to Array

```php
$mylist[0] = "10";
```

An empty array can also be created using the array construct.

Example: $mylist = array(); → array function is used to create Array

we can have mixed type of elements in the Array.

Example:- $mylist = array ("Archana", 519, 89.23);

there are three types of arrays in PHP

① Numeric Array :- Arrays with a numeric index.

② Associative Array :- Arrays with named keys.

③ multidimensional Array :- Arrays containing one or more Arrays.

Example to Count Size of Array :- (or) Length of array :-

```
<!DOCTYPE html>
<html>
<body>
<?php
$bikes = array ("KTM", "Bullet", "Duke");
echo count($bikes);
?>
</body>
</html>
```

o/p : 3

Example 2:

```
<!DOCTYPE html>
<html>
<body>
<?php
$bikes = array("KTM", "Bullet", "Duke");
echo "I like". $bikes[0]. ", ". $bikes[1]. "and".
$bikes[2]. ".";
?>
</body>
</html>
```

O/P:- I like KTM, Bullet and Duke.

3. OPERATORS IN PHP :- 7 types

1. Arithmatic operators
2. Assignment operators
3. Logical operators
4. Equality operators
5. Comparision operators
6. Increment & decrement operators
7. Bitwise operator

① **Arithmatic operators** :- there are 5 arithmatic operators that are used to perform mathematic operators they are +, −, *, /, %.

operators	Name	Example
+	Addition	$x + $y
−	Subtraction	$x - $y
*	multiplication	$x * $y
/	Division	$x / $y
%	modulus	$x % $y

(2) Assignment operators (denoted by Symbol "=")

operator	Name	Example
=	Assignment	$x = 2
+=	Addition Assignment	$x += 2
-=	Subtraction "	$x -= 2
*=	multiplication "	$x *= 2
/=	Division "	$x /= 2
%=	modulus "	$x %= 2

(3) Logical operator:

Used to perform decision i.e, Used to take decision

operator	Name	Example
&&	AND	$x && $y (or) $x AND $y
||	OR	$x || $y (or) $x OR $y
!	NOT	$x ! $y (or) $x NOT $y
XOR (Exclusive OR)	Exclusive OR	$x XOR $y

(4) EQuality operator:

Used to compare two Values

operator	Name	Example
==	Equal to	$x == $y
!=	Not Equal to	$x != $y
===	is identical to	$x === $y

(5) Comparision operator :-

<, >, <=, >=, ?:

Example: $x < $y ternary

(6) Increment & decrement operators:

++, --

Increment Decrement

Example: ++$x ⟹ pre Increment

$x++ ⟹ post Increment

(7) Bit wise operator:

&, ^, |, ~, <<, >>

AND XOR OR NOT Shift left Shift Right

Example: $x & $y

Precidence and Associativity:

Associativity ⟶ means order of execution

Precidence ⟶ means priority i.e, it represent which Symbol has high priority and which Symbol has low priority

from highest priority to lowest

highest ⟶ [ ]

*, /, %.

+, -

>, >=, <, <=

==, !=

&& [AND]

|| (OR) ← Lowest

operator	Associativity
+, −	R to L
*, /, %	L to R
<, <=, >, >=	L to R
==, !=	L to R

4. EXPRESSIONS IN PHP:

⇒ Expressions are the most important building blocks of PHP, almost anything we write is an Expression.

⇒ An Expression is defined as "Anything that has a Value"

Ex: $a = 5

⇒ Anything that appears to the right of assignment operator (=) is an expression.

Syntax:

```
$x=100; // 100 is an expression
$a = $b+$c; // $b+$c is an expression
$c = add ($a, $b); // add ($a, $b) is expre
$var =
$var = $x != $y; // $x != $y is expressio
```

⇒ Expression with ++ and -- operators are called increment and decrement operators. Both prefix and postfix ++ operators increment value of operand by 1, whereas -- operators decrement by 1.

pre increment

```
<?php
$x=10;
echo ++$x;
?>
```

o/p: 11

post increment

```
<?php
$x=10;
echo $x++;
echo $x;
?
```

o/p: 10
11
12

pre decrement

```
<?php
$x=10;
echo --$x;
?>
```

o/p: 9

post decrement

```
<?php
$x=10;
echo $x--;
echo $x;
?>
```

o/p: 10
9
8
7

⇒ php Regular Expression are Sequence of characters. Using Regular Expression you can find a particular string, Replace one string by another, we can split single string into many parts etc..

⇒ Ternary operator has three operands. First one is logical expression. if it is TRUE, Second operand expression is evaluated otherwise third one is evaluated. Syntax: (condition) ? "true" : "false";

```
<?php
$x=10;
echo ($x<20) ? pass : fail;
?>
```

5. SESSION HANDLING IN PHP :-

⇒ when you work with an Application, you open it, do some changes and then you close it. This entire process is known as Session.

⇒ the Computer knows who you are, It knows when you start the Application and when you end. But in the internet there is one problem. the web Server doesnot know who you are, or what you do, because the http address doesnot maintain State.

⇒ Session Variables Solve this problem by storing user informati to be used across multiple pages by default, Session variables are present until user closes the browser. So, Session Variables hold information about one Single user, and are available to all pages.

⇒ A Session is Started with Session_Start() function. this function checks if a Session is already Started and if non is Started then it start Session. This Session_Start() is put at the beginning of page.

⇒ A php Session Can be destroyed by using Session_destroy() function. this function doesnot need any arguments and Single Call Can destroy a Single Session Variables.

```
<?php
session_start();
?>
```
→ to Start Session

=> Unset function is used to unset a Session Variables

```
<? php
Unset ( $_SESSION['counter']);
?>
```

```
<? php
Session_destroy();
?>
```

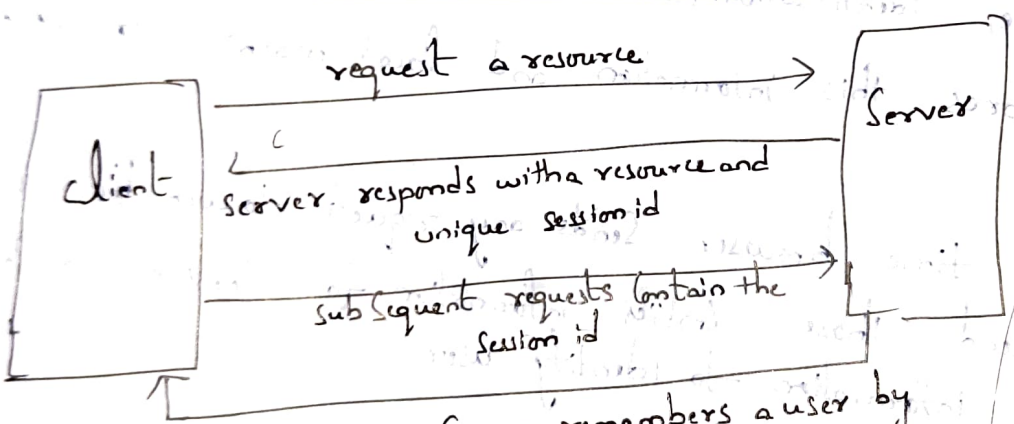


fig: Session handling

6. COOKIES IN PHP :-

=> A Cookies is often used to identify a user. A Cookies is a small file that Server embeds on the user's Computer. It is used for tracking purpose. Each time the Same Computer requests a page with a browser, it will Send the Cookie too. with PHP, you Can both Create and retrieve Cookie values.

⇒ A Cookie is created with the SetCookies() function

Syntax: SetCookie(name, value, expire, path, domain);

⇉ only name parameter is required. All other parameters are optional.

⇒ PHP transparently support HTTP Cookies. there are three steps involved in identifying & Returning users

(a) Server Script Sends Set of Cookies to the browser. For example name, age, identification number etc..

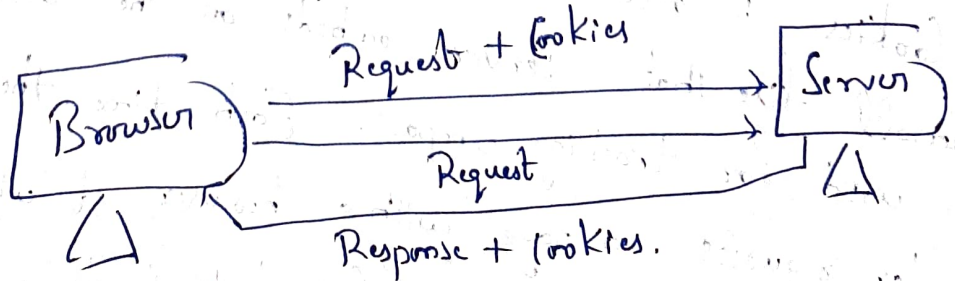
(b) Browser Stores this information and local machine for future user

(c) when next time browser Sends any request to web Server then it Send those Cookies information to Server and Server uses that information to identify user.

Example:

```
<?php
$ Cookie_name = "user";
$ Cookie_value = "John";
SetCookies ( $Cookie_name, $ Cookie_value,
             time() + (86400 * 30) );
?>
```

30 day



7. CONTROL STRUCTURES IN PHP:

allows us to control the flow of execution of code. Generally code is executed Sequentially.

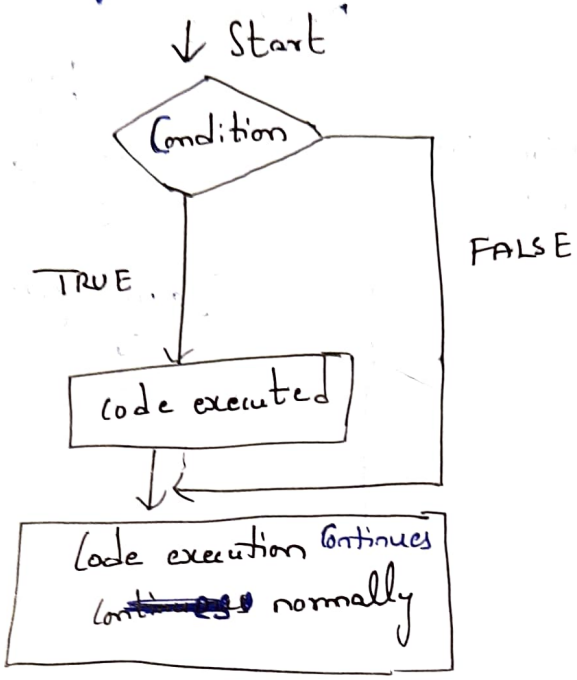


fig:- Structure of Control structures.

⇒ there are two types of Control Structures in PHP they are

(a) Conditional Statements

(b) Control loops

(a) Conditional Statements:

Conditional Statements are used to perform different actions based on different conditions. There are four Conditional Statements in php they are

1. If Statement
2. if else Statement
3. If... elseif... else Statement (elseif statement)
4. Switch Statement.

① If Statement:

It executes some code if condition is true.

Syntax:

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

Program:

```
<?php
  $age = 50;
if ($age > 30)
  {
echo "your age is more than 30";
  }
?>
```

o/p:- your age is more than 30

② If else Statement:

It executes some code if a condition is true and another code if that condition is false

Syntax:

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

Program:

```
<?php
  $t = 10;
if ($t < "20")
  {
echo "Have a good day!";
  }
else
  {
echo "Have a good night!";
  }
?>
```

③ if ..... elseif .... else Statement:

Use this Statement to Specify a new condition to test, if first condition is false

Syntax:

```
if (condition)
  {
code to be executed if condition is true;
  }
else if (condition)
  {
code to be executed if first condition
 is false and this is true;
  }
  else
  {
code to be executed if all conditions
  are false
  }
```

Program

```
<?php
$t = "10";
if ($t < "10")
{
echo "good morning!";
}
else if ($t < "20")
{
echo "good Afternoon!";
}
else
{
echo "good night!";
}
?>
```

④ Switch Statement:

⟹ Switch Statement is a multiple choice Selection Statement

Syntax:

```
Switch (n)
  {
  case label1:
  code to be executed if n=label1;
    break;
  case label2:
  code to be executed if n=label2;
    break;
    :
  default:
  code to be executed if n is different
        from all labels;
  }
```

Program:

```
<? php
$colour = "red";
Switch ( $colour)
  {
case "red";
echo " your colour is red! ";
break;
case "blue":
echo " your colour is blue!";
break;
case "green":
echo " your colour is green;
break;
default:
echo " your colour is not present;
  }
?>
```

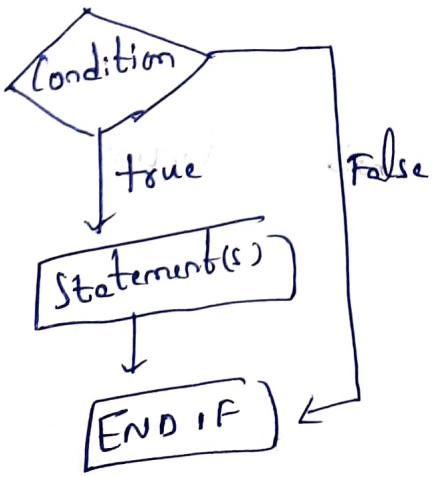


fig: if statement flowchart

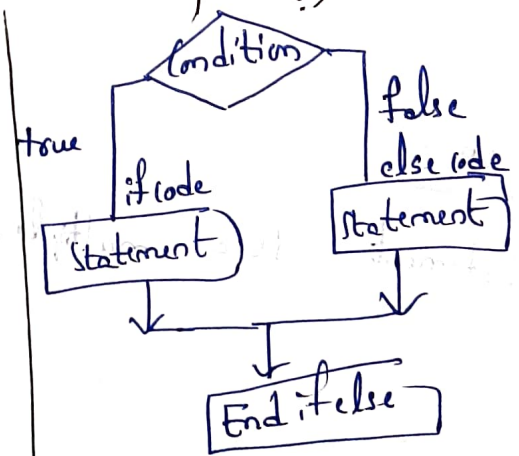


fig: if else flowchat

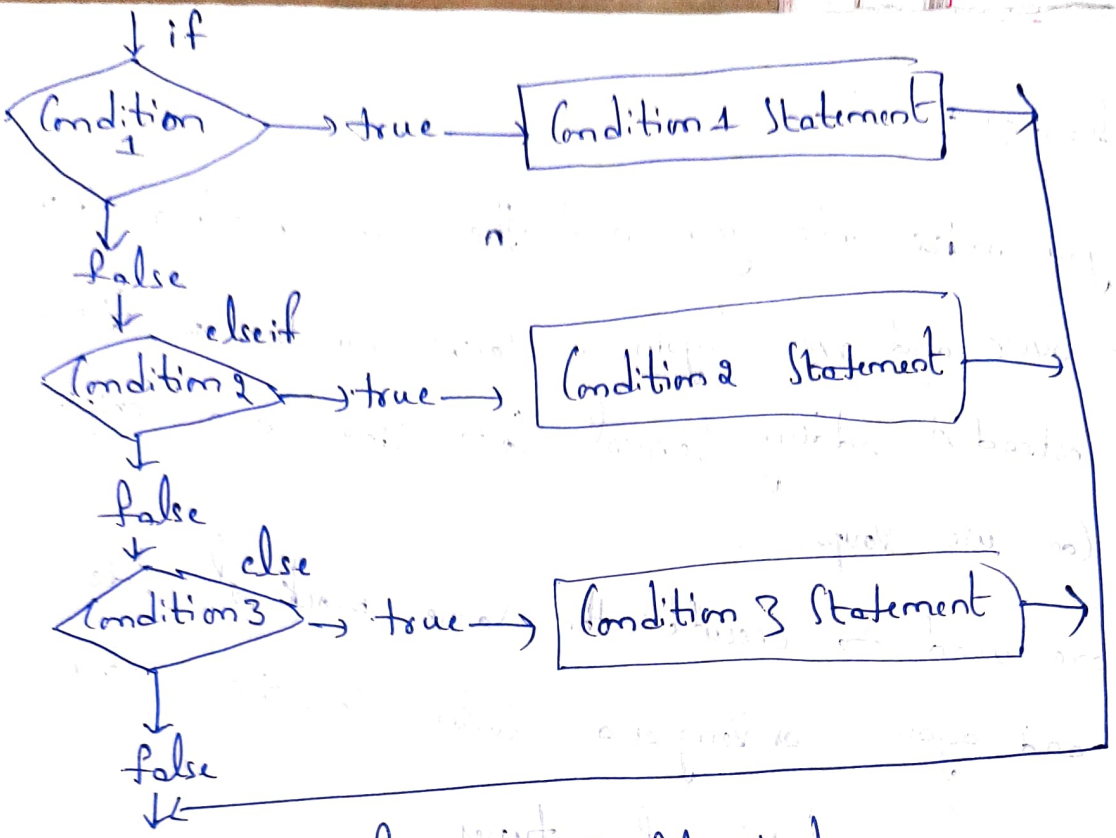
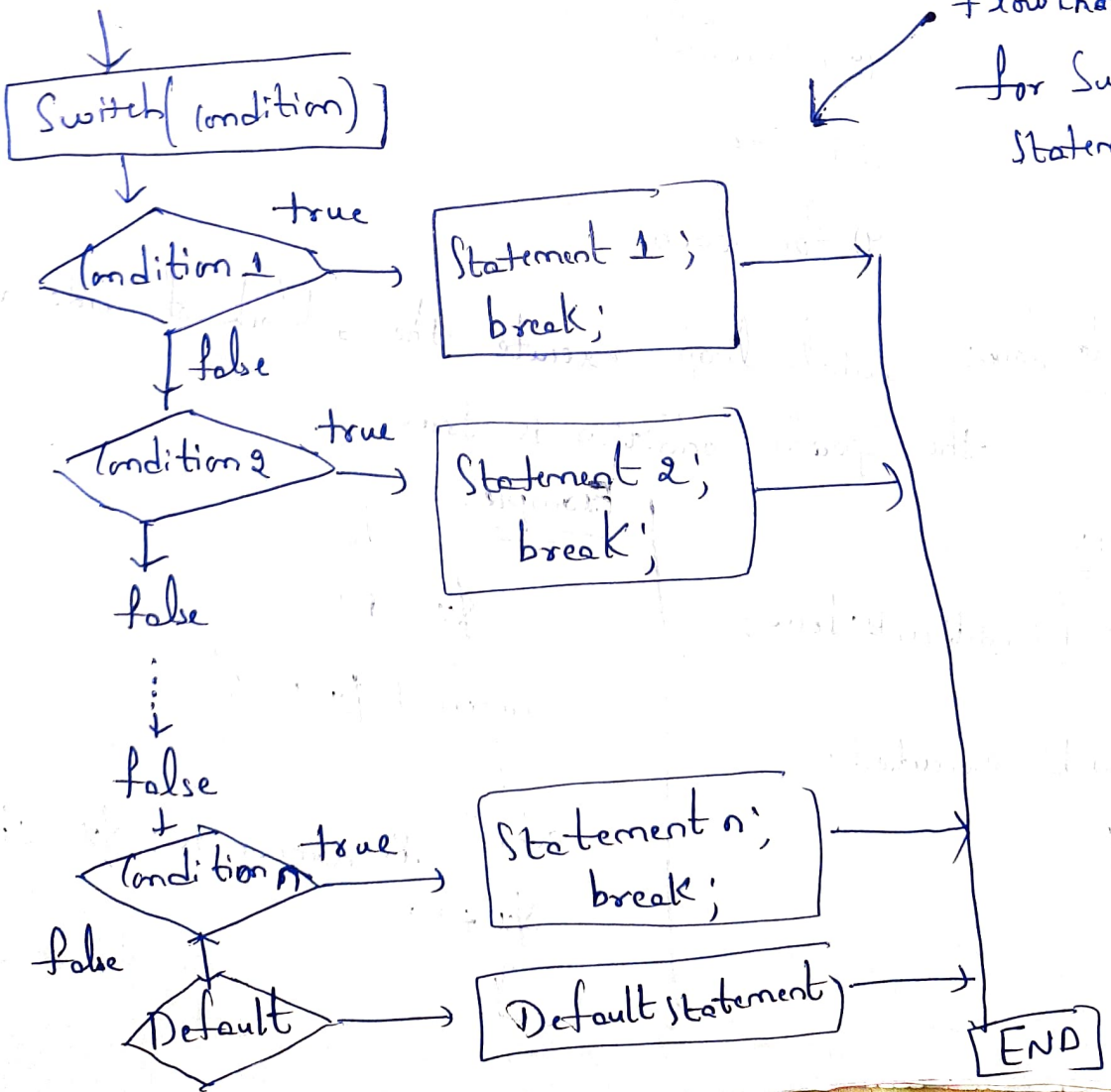


fig: elseif flowchart

flowchart for Switch Statement.



(2) Control loops:

⇒ when you write code, you want the same block of code to run over and over again a certain number of times. So, instead of adding several code lines in a script, we can use loops.

⇒ loops are used to execute the same block of code again and again, as long as a certain condition is true.

⇒ loops are classified into four types

1. while
2. do while
3. for
4. for each

(1) while loop:- while loops execute the a block of code as long as the specific condition is true

Syntax:

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

Example:

```
<?php
   $x = 1;
while ($x <= 5)
  {
echo "The number is: $x <br>";
$x++;
  }
?>
```

output: 1,2,3,4,5

② do-while loops:

do while loops through a block of code once, and then repeats the loop as long as the specific condition is true

Syntax:

```
do
{
condition to be executed;
}
while(condition is true);
```

Program:

```
<?php
$x = 1;
do
{
echo "The number is $x <br>";
$x++;
}while($x<=5);
?>
```

output:

The number is 1
" " " 2
" " " 3
" " " 4
" " " 5

③ For loop:

For loop through block of code a specific number of times

Syntax:

```
for(init counter ; test counter ; increment counter)
{
code to be executed for each iteration;
}
```

Example:

```
<?php
for($x=0 ; $x<=10 ; $x++);
{
echo "the number is $x: <br>";
}
?>
```

output:

the number is 0
" " " 1
" " " 2
" " " 10

(4) for each:

⇒ for each loops through a block of code for each element in an array

⇒ for each loop works only on Array, and is used to loop

= through each key/value pair in an array

Syntax:

```
for each ($array and $value)
  {
  code to be executed;
  }
```

Program:

```
<?php
$colors = array ("red", "green",
                 "blue", "yellow");
for each ($colors as $value)
  {
  echo "$value <br>";
  }
?>
```

O/P: red
green
blue
yellow

8 FILE HANDLING IN PHP

File handling is an important part of any web Application

For different tasks we need to open and process a file.

⇒ php has several functions for Creating, reading, uploading, editing etc..

File handling in PHP is similar as file handling done by using any programming languages like C.

PHP has many functions to work with normal files. Those functions are

① fopen():

↓

fileopen.

⇒ PHP fopen() function is used to open file. It takes two parameters.

function

Contains name of file to be opened.

mode

It includes in which mode file should be opened

Example:

```
<?php
$file = fopen("f.txt", "w");
?>
```

modes:

r → read only
w → write only
r+ → read/write
w+ → Read/write.

a → Append
a+ → Read/Append

② fread():

After file is opened using fopen() the contents of data are read using fread().

It takes two arguments

① pointer to file

② file size [in bytes]

Syntax: $filedata = fread ($file, $size);

⇒ we can also use other functions like fgets() & fgetc() to read functions from file

③ fwrite()

⇒ New file can be created or text can be appended to an existing file using fwrite() function.

⇒ fwrite() and fputs() are functions used to write data into file

modes: w, w+, a, a+, x, x+..... etc.,

④ fclose():

file is closed using fclose() function. Its argument is file which needs to be closed.

Example: fclose ($file);

⑤ ~~Delete~~ Unlink():

Used to delete the file. only one parameter is Accepted i.e, file name. Example: unlink ($file)

if file is Successfully deleted it display TRUE else it Return false.

9 HANDLING FILE UPLOADS :

⇒ PHP allows you to upload Single and multiple files through few lines of code

⇒ PHP file upload features allows you to upload binary and text files. both moreover, you Can have the full Control over the file which you have uploaded by using PHP.

⇒ PHP also Supports authentication & file operation functions

PHP $_FILES :-

The PHP global $_FILES contains all the information of file By the help of $_FILE global, we Can get filename, filetype, filesize, temp file name and errors associated with file.

Here, in this example, we are assuming the file name is filename.

$_FILES ['filename'] ['name'] :- It is used to return the name of the file

$_FILES ['filename'] ['type'] :- " " " MIME type " " (multipurpose Internet mail extensions)

$_FILES ['filename'] ['size'] :- " " " size " "

$_FILES ['filename'] ['tmp_name'] :- " " " temporary " "

$_FILES ['filename'] ['error'] :- " " " errors associated with the file

move_uploaded_file() function:

⇒ the move_uploaded_file() function moves the uploaded file to a new location. the move_uploaded_file() function checks internally if the file is uploaded through POST Request. It moves the file if it is uploaded through post Request.

Syntax: move_uploaded_file (String $filename, String $destination)

10. FORM HANDLING (OR) FORM CONTROLS :-

⇒ we can Create and use forms in PHP.

⇒ To get the data present in the form, we need to use PHP Superglobals. they are

① $_GET i.e, GET method
② $_POST i.e, POST Method

} used to collect form data.

⇒ we can Send form request by using get() or post() methods.

⇒ To retrieve data by using get method we need to use $_GET

⇒ " " post " " $_POST

⇒ " " "

PHP post form:-

post request is widely used to submit form that have large amount of data such as file uploads, image uploads, login forms, registration forms etc.

⇒ The data passed through post request is not visible on the URL browser. So it is secured. you can send large amount of data through post request.

⇒ A simple example to receive data from post request in php is file.html.

```
<html>
<body>
<form action = "Data.php" method = "post">
Name: <input type = "text" name = "name"> <br>
Email: <input type = "text" name = "email"> <br>
<input type = "submit" value = "Submit">
</form>
</body>
</html>
```

```
<html>
<body>
welcome
<?php
echo $_POST["name"];
?>
<br>
your email address is :
<?php
echo $_POST["email"];
?>
</body>
</html>
```

← Data.php file.

o/p:-

Name: Sai

E-mail: nsai305@gmail.com

Submit

welcome Sai
your email address is :
nsai305@gmail.com

PHP Get method:

This Get Request is the default form Request. the data passed through get request is visible on the URL browser, So it is not Secure. you need to Send limited amount of data through get Request.

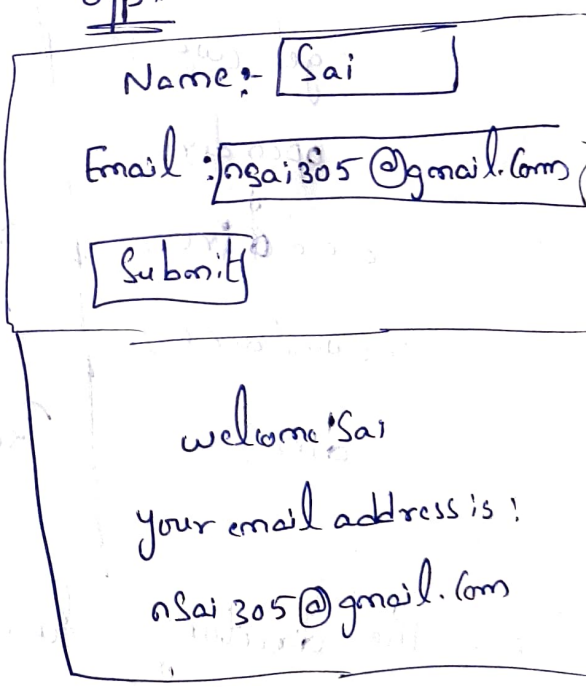
file.html.

```
<html>
<body>
<form action = "Data.php"  method = "get">
Name: <input type = "text"  name = "name"> <br>
Email: <input type = "text"  name = "email"> <br>
<input type = "Submit" action = "Submit">
</form>
</body>
</html>
```

Data.php file.

```
<html>
<body>
welcome
<?php
echo $_get["name"];
?>
<br>
your email address is :
<?php
echo $_get["email"];
?>
</body>
</html>
```

O/p:



11. LISTING DIRECTORIES

⇒ PHP provides a Support of directory functions that allows you to retrieve information about directories and its contents

⇒ Various directory functions are given below.

Function	description
chdir()	change Current directory
chroot()	" " root "
closedir()	close Current working directory
getcwd()	Return " "
opendir()	open directory handle
readdir()	read an entry of directory
rewinddir()	Reset a directory handle.
dir()	Returns an instance of the directory class

⇒ The directory functions are part of PHP Core. No installation is required to use these functions.

⇒ the opendir and readdir are the two important directory functions that are commonly used while listing the contents of a directory. Let us discuss Syntax of the

① opendir:

Syntax: opendir (path, Context);

parameters

path → denotes path of Specific directory

Context → (optional) used to denote behaviour of stream

② Readdir:

Syntax: readdir (handle);

optional parameter.

It denotes directory handle resource which is opened with opendir()

③ closedir:

Syntax: closedir (handle);

optional parameter.

It denotes directory handle resource which is opened with opendir()

12. SCOPE AND LIFETIME OF VARIABLES IN PHP

Scope of a Variables:-

Variables are of ~~two~~ three types

1. local Variables :- Variables declared inside a function.
2. global Variables :- " " outside " "
3. Static

⇒ Global Variables can be accessed anywhere in the program. To access global Variables in a function, it must be declared in a global decleration of function using global Keyword. otherwise, the Variable acts as a local Variable.

```
<?php
$a=1;
$b=2;
function Sum()
{
global $a, $b;
$b = $a + $b;
}
Sum();
echo $b;
?>
```

output :- 3

if you want to define global Variable inside a function we use global Keyword.

Static

Normally, when a function is completed / executed. all of its variables are deleted. however, Sometimes we wanted local variable NOT to be deleted. To do this we use Static Keyword

Eg: Static $x=10;

lifetime of Variable:

⇒ The life time of local Variable in php function begin when Variable is first used and ends when the function execution terminates i.e after termination of function execution, the value of local Variable is lost.

⇒ The life time of a Static local Variable in a function begin when the Variable is first used in function, and ends when the Script Execution terminates.

This Static local Variables is Specified using 'Static keyword'

13. CONNECTING TO DATABASE [MYSQL as a reference]

⇒ MYSQL is open Source database product and Can be downloaded from the internet.

⇒ MYSQL is a kind of database in which the records are Stored in an entity Called table. In the table the data is arranged in the rows and Columns. we Can query a database to retrieve particular information. Query is a request or a question for database.

⇒ PHP is a Server side Scripting language and it has an ability to Create dynamic pages with Customized features

Using PHP- MYSQL user friendly and interactive web sites can be created.

⇒ Both PHP and MYSQL are openSource technologies that work hand-in-hand to create rich internet Applications. and it is highly Secure.

⇒ PHP-MYSQL are Stable technologies and have Cross platform Capability. Hence the web application developed using these technologies become portable.

⇒ Since HTML is embedded with PHP, there is no need to write Separate Code for web-Scripting.

⇒ PHP has got a Strong Support for developing e-Commerce applications using the technologies Such as ECommerce, Drupal and Soon.

⇒ The most popular websites being developed using PHP and MY SQL technologies are

1. Facebook
2. wordpress
3. wikipedia
4. Yahoo.

⇒ mysql_connect() function is used to establish a connection between PHP and MYSQL.

Functions in MYSQL:

to Create data base :- function mysql_query is used

for errors in Commands :- " mysql_error() " "

Selecting database :- " mysql_Select_db() " "

Listing Database :- " mysql_list_db() " "

⇒ Tables are present inside the database.

function mysql_list_table function is used to display tables present in database, we can also use mysql_query. we can perform various operations like

1. Create a table
2. Inserting data into the table
3. Altering table
4. Deleting table.

UNIT-2

HTML

1. HTML ? Features ? Advantages ? Disadvantages ? Example ?
2. HTML tags (common html tags)
3. HTML lists (ordered list, unordered list, Description list)
4. html tables
5. HTML forms
6. Attributes of form tag
7. HTML frames ? Disadvantages ? Attributes of frameset tag ?
8. HTML images
9. HTML Cascading Style sheets (HTML CSS), properties, Inline, Internal, External CSS.

XML

1. XML ? features and advantages ? Difference between HTML & XML
2. XML tags
3. XML Attributes and values
4. XML Schema (or) XML schema definition (XSD)
5. XML DOM (Document object model)
6. XHTML parsing XML Data., Difference between HTML and XHTML
7. XML DTD (Document type Definition), Internal DTD and External DTD
8. DOM and SAX parsers , Difference between DOM and SAX parser.

HTML: html Stands for "Hyper text markup language"

⇒ HTML is the Standard markup language, for Creating web pages. ie, it is not a programming language but it is a markup language.

A markup language is a Set of markup tags. HTML uses markup tags to Represent web pages. ie, markup tags tells the web browser, Such as mozilla firefox, Google chrome, how to display the page. An HTML file must have an htm or html file extension.

HTML Elements and tags:

⇒ An HTML element is defined by Start tag, Some Content, end tag. ie, HTML element is everything from Start tag to end tag. HTML uses predefined tags.

Example: <tagname> Content </tagname>

Features of HTML:

1. It is easy to learn and easy to use.
2. platform independent
3. Images, videos, audio's Can be added to a web page
4. It is a markup language

Advantages:

1. HTML is used to build websites
2. It is Supported by all browsers.
3. it is integrated with other languages like CSS, JavaScript etc

Disadvantages:

A large amount of code has to be written to create simple web page and Security feature is not good.

BASIC STRUCTURE OF HTML CODE:

```
<!DOCTYPE html>        ← tells version of html
<html>                 ← root element of html
<head>
<title> my web page </title>
</head>
<body>
<h1> web technologies </h1>
<p> what is html </p>
</body>
</html>
```

o/p: webtechnologies
what is html

where,

<head>: used to contain page HTML metadata

<title> title of HTML page

<body> hold content of HTML

<h1>: HTML heading tag

<p>: HTML paragraph tag.

HTML VERSIONS

1991	1993	1995	2000	2014	2017
HTML	HTML+	HTML2.0	XHTML1.0	HTML5	HTML 5.2

HTML TAGS:

⇒ HTML tags are used to Create HTML documents and each HTML tags have different properties. HTML tags Contain three main parts ie, Starting tag, Content, ending tag. when a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML file must have some essential tags So that web browser Can differentiate between a Simple text and html text. you Can use as many tags you want as per your Code requirement.

⇒ All HTML tags must enclosed with <> these Angle brackets

⇒ Every tag in HTML perform different tasks. If you have used an open tag <tag>, then you must use a close tag </tag>. (except Some tags).

⇒ HTML tags are always written in lower Case letters.

BASIC HTML TAGS:

① html tag: It represents root of an html document.

Syntax: <head> ... </head>

② title ~~head~~ tag: It defines the title or name of an HTML document

Syntax: <title> ... </title>

③ body tag: It is used to define body Section of an HTML document.

Syntax: <body> ... </body>

(4) heading tags:

It is used for making heading in web pages. there are 6 types of heading tags. all of these tags are automatically in bold form.

① <h1> ... </h1>
② <h2> ... </h2>
⋮
⑥ <h6> ... </h6>

Bigger to [text size: 17]
Smaller

Some HTML tags are not closed, for example br and hr

 tag:- br stands for break line, it breaks the line of code

<hr> tag:- hr stands for horizontal Rule. This tag is used to put a line across the web page

⇒ <!DOCTYPE> tag is used to define html document type

⇒ tag:- you can use tag to put any content in the center of the page

Other frequently used tags in html:-

<P> paragraph Tag </P>
<h2> Heading Tag </h2>
 BOLD Tag
 Italic tag
<u> Underline tag </u>

HTML LISTS :-

A list is a record of short pieces of related information or used to display the data or any information in web pages in the ordered or unordered form. HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists.

1. Ordered List (or) Numbered List (ol)
2. Unordered List (or) Bulleted List (ul)
3. Description List (or) Definition List (dl)

(1) Ordered List:

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

Example

```
<!DOCTYPE>
<html>
<body>
<ol>
<li> Samsung </li>
<li> mi </li>
<li> vivo </li>
<li> iphone </li>

</ol>

</body>
</html>
```

output:

1. Samsung
2. ~~Bing~~ mi
3. vivo
4. iphone

There Can be different types of numbered list. they are

1. Numeric Numbers (1, 2, 3)
2. Capital Roman Numbers (I II III)
3. Small Roman Numbers (i ii iii)
4. Capital Alphabet (ABC)
5. Small Alphabet (abc)

To represent different ordered lists, there are 5 types of attributes in <ol> tag.

Type	Description
Type "1"	This is the default type, in this list items are numbered with numbers
Type "I"	In this list items are numbered with upper case roman numb
Type "i"	In this .. .. .. with lower case Roman numbers
Type "A"	.. .. .. with upper case letters
Type "a"	" " .. .. lower case letters

Example to Display list in roman numbers lower case:

```
<!DOCTYPE html>
<html>
<body>
<ol type="i">
<li> Samsung </li>
<li> mi </li>
<li> vivo </li>
</ol>
</body>
</html>
```

output:

i. Samsung
ii. mi
iii. vivo

Example of Start Attribute :-

the Start Attribute is used with ol tag to Specify from where to start the list items.

Example: Syntax: `<ol type = "1". Start = "5">` :- it shows numeric value starting with "5".

Example:

```
<! DOCTYPE html>
<html>
<body>
<ol type="1" Start = "5">
  <li> SamSung </li>
  <li> VIVO </li>
  <li> mi </li>
</ol>
</body>
</html>
```

output:

5. SamSung
6. VIVO
7. mi

② Unordered list :-

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

Example:

```
<!DOCTYPE>
<html>
<body>
<ul>
<li> SamSung </li>
<li> Sony </li>
<li> LG </li>
</ul>
</body>
</html>
```

output

- SamSung
- Sony
- LG

(3) Description list:

HTML also Supports description lists. A description list is a list of terms, with a description of each term.

=> the <dl> tag defines description list

=> " <dt> tag " the term (name)

=> " " <dd> tag describes each term

Example:

```
<!DOCTYPE html>
<html>
<body>
<dl>
<dt> c language </dt>
<dd> developed in the year 1972 </dd>
<dt> php </dt>
<dd> developed by Apache group </dd>
</dl>
```

```
</dl>
</body>
</html>
```

output:

c language
- developed in the year 1972

php
- developed by Apache group

where,

<dl> tag defines description list
<dt> tag defines data term
<dd> tag defines data definition (description)

---

3. HTML TABLES :-

=> HTML tables allow web developers to arrange data into rows and columns. There can be many columns in a row. We can create a table to display data in tabular form, using <table> element, with the help of <tr>, <td> <th> elements.

where <tr> tag defines table row
<th> tag defines table header
<td> tag defines table data.

Other HTML table tags include:

<caption> tag defines table caption

<thead> Groups the header content in a table

<tbody> " " body " " "

<tfoot> " " footer " "

Example:

```
<!DOCTYPE html>
<html>
<head>
<Style>
 table, th, td {
  border : 1px solid black;
   }
</style>
</head>
<body>
<h1> Marks of Student </h1>
<table>
 <tr>
  <th> Student </th>
  <th> marks </th>
 </tr>
  <tr>
   <td> Ramesh </td>
   <td> 100 </td>
  </tr>
```

```
<tr>
<td> shiva </td>
<td> 75 </td>
</tr>
</table>
</body>
</html>
```

output:

Marks of Student

Student	marks
Ramesh	100
shiva	75

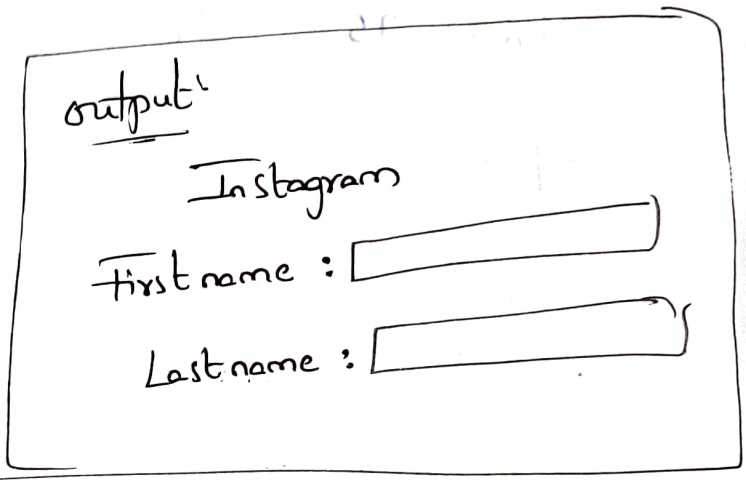
HTML FORMS :-

An HTML forms is used to collect user input. the user input is most often Sent to a Server for processing

Basic Example:

```
<!DOCTYPE html>
<html>
<body>
<h2> Instagram </h2>
<form action = "/action_page.php">
</form>
First name : <input type = "text" name = "first_name" />
<br>
```

```
Last name : <input type = "text" name = "last_name" />
</form>
</body>
</html>
```



⇒ An HTML form is a section of document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

⇒ An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number etc..

⇒ HTML forms are required if you want to collect data from the site visitor. for example, If a user want to purchase some items on internet, he/she must fill the form such as shipping address, Credit card/ debit card details so that item can be sent to given address.

there are four primary elements is used with in form tag:

① <input>:- it is whatever message that we want to insert in database.

② <Select>:- when ever we want to select particular data from data base we use Select tag.

③ <text area>:- Used to write data in the text boxes

④ <button>:- we can use various kinds of buttons like radio button, checkboxes etc..

Example 2: Designing of text field in web page:

```
<html>
<head>
<title> facebook </title>
</head>
<body> <h1> facebook </h1>
<form>
Enter name < input type = "text" size = "30" /> <br>
Enter password < input type = "text" size = "30" /> <br>
<input type = "button" value = "Send" />
<input type = "button" value = "Exit" />
</form>
</body>
</html>
```

output:

facebook

Enter name

Enter password

Send Exit

5. ATTRIBUTES OF FORM TAG:

there are three Attributes of form tag they are.

1. Action
2. method
3. Encryption type.

(1) Action:

⇒ It is used to determine, where to Send data.

⇒ It Specify url (Uniform resource locator) to which form data will be Submitted

⇒ we would Specify url of a program on a Server or an email

Example: <form action = "data.asp">

where asp - active Server page
jsp - java " "
CGI - Common gateway interface

(2) method:

It is an Attribute of form tag. It determines how form data will be Submitted.

the 2 options of these attribute is

Get & post method

Get: we get data from Server, no url is required. data passed through get request is visible on url browser

post: it is used to Submit form that have le amount of data. data passed through url is not visible on browse So it is Secure

(3) Encryption type:

It Specify the format of data being Submitted. It Specify an Encoding protocol known as multipurpose internet mail Extension (MIME) for Security.

⇒ MIME Ensure that data doesnot become Corrupted when transmit across the internet.

Example:- <form action = "data.asp" method = "post" Encrypt = "plain/text">

HTML FRAMES

HTML frames are used to divide your browser window into multiple Sections where each Section Can load a Separate HTML document. A Collection of frames in the in the browser window is Known as a frameset. The window is divided into frames in a Similar way the tables are Organized into rows and Columns

Disadvantages of Frames:

There are few drawbacks with using frames, So it's never recommended to use frames in your webpage

(1) Some Smaller devices Cannot cope with frames because their Screen is not big enough to be divided up.

(2) Sometimes your page will be displayed differently on different Computers due to different Screen Resolution

(3) the browser's back button might not work as the user hopes

(4) there are Still few browsers that do not Support frames technology.

⇒ It Can display one or more than one html document in Same browser window. Each html document is called FRAME each frame is independent of other.

<frameset> tag is used to divide browser window.

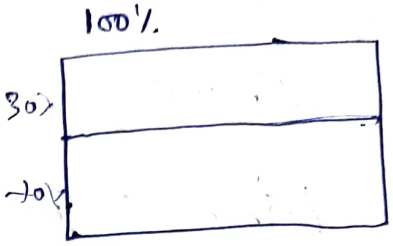
<body> tag is not required.

Attributes of frameset tag :-

- → Rows
- → Columns
- → frame border
- → Border Color
- → name

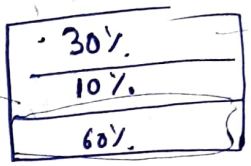
Rows:

It divides browser into row wise



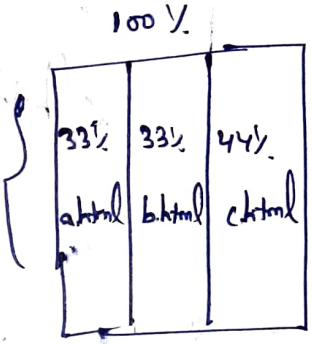
A.html.

```
<html>
{ <frameset rows = "30%, 70%">
  </frameset>  (or)
{ <frameset rows = "30%, 10%, *">
  </frameset>
</html>
```



② Columns:

It divides browser window column wise




```
<html>
 <frameset cols = "33%, 33%, 34%">
<frame src = "a.html">
<frame src = "b.html">
<frame src = "c.html">
 </frameset>
</html>
```

Example: To design an Indian flag:

a.html

```
<html>
<body bgcolor = "orange">
</body>
</html>
```

background

b.html.

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

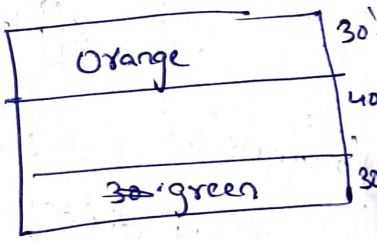
c.html

```
<html>
<body bgcolor = "green">
</body>
</html>
```

main.html:

```
<html>
<frameset rows = "30%, 40%, 30%">
<frame src = "a.html">
<frame src = "b.html">
<frame src = "c.html">
</frameset>
</html>
```

o/p:-



HTML IMAGES

⇒ Images can improve the design and the apperance of the web page.
The <img> tag is used to embed (place) an image in an HTML page.
we cannot import image directly to the web page, we need to use <img> tag to link images to web pages. This <img> tag creates a holding space for the referenced image.

the <img> tag has two required attributes.

① src:- Specifies the path to the image

② alt:- Specifies an alternate text for the image; if the image for some reason cannot be displayed

other attributes include

(1) Size: Specifies size of the image

(2) Title: It adds title to the image

(3) Alignment: where to place an image, like left, right, top or bottom

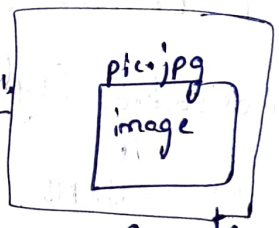
(4) Border Size: Images can be appeared with the border and we can increase size or the border or " " " decrease " " " "

Example:

```
<html>
<body>
<img src = "pic.jpg" align = "right">
</img>
</body>
</html>
```

Source file

o/p:

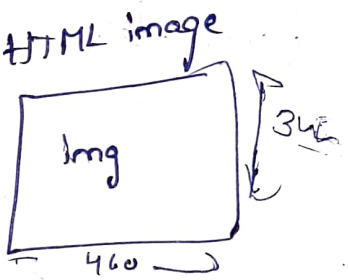


⇒ Images are not part of webpage file. They are separate files which are inserted into the page where it is used by the browser. We can also add source of image file to the html web page.

Example 2:

```
<!DOCTYPE html>
<html>
<body>
<h2> HTML Image </h2>
<img src = "image.jpg" alt = "Flowers" , width = "460"
                                height = "345" >
</img>
</body>
</html>
```

o/p:



HTML CSS :→ Stands for Cascading Style sheets.

⇒ Cascading Style sheets (CSS) is the language we use to Style an HTML document. CSS describes how HTML elements should be displayed. CSS Saves lot of work. it can control the layout of multiple web pages all at once.

⇒ with CSS you can control the color, font, size of text the spacing between elements, how elements are positioned and laid out., what background ~~colours~~ images and background colors are to be used etc~

⇒ The word Cascading means that a style applied to a parent element will also applied to all children elements within the parent. So, if you set the color of body text to "blue", all headings, paragraphs, and other elements with in the body will also get the same color.

⇒ CSS Can be added to HTML document in 3 ways.

① Inline:-
② Internal
③ External

① Inline CSS:

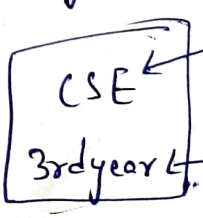
An inline CSS is used to apply a unique style to a single HTML element. An inline CSS uses the style attribute of an HTML element.

The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red.

Example:

```
<html>
<body>
<h1 style = "color: blue;"> CSE </h1>
<p style = "color: red;"> 3rdyear </p>
</body>
</html>
```

o/p:



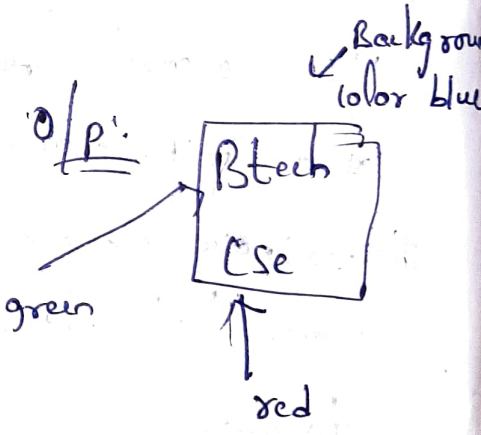
② Internal CSS:

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the <head> Section of an HTML page, within a <Style> element

Example:

```
<!DOCTYPE html>
<html>
<head>
<Style>
body { background-color: blue; }
h1 { color: green; }
p { color: red; }
</Style>
</head>
<body>
<h1> Btech </h1>
<p> CSE </p>
</body>
</html>
```



(3) External CSS:

An external Style Sheet is used to define the Style for many HTML pages

Example:

```
<html>
  <head>
    <link rel = "stylesheet" href = "style.css">
  </head>
  <body>
    <h1> Btech </h1>
    <p> CSE </p>
  </body>
</html>
```

o/p:

Btech — Blue

CSE — Red

style.css

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

h1{
    color : blue;
    }

p{
    color: red;
    }
```

(2) CSS properties.

CSS color property defines the text color to be used

CSS font-family property defines the font to be used

CSS font-size " " " text size " "

CSS Border " " " border around an HTML element

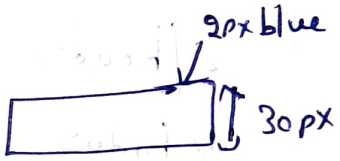
CSS padding " " " a space between text and border

Ex: p{

border : 2px solid blue; →

padding: 30px;

}



CSS margin property defines space outside the border

Xml :-

⇒ Xml Stands for Extensible Markup language. Xml donot Contain any predefined tags we Can Create and define our own tags in Xml. Xml was designed to Store and transport data.

⇒ XML was designed to be both human and machine readable. XML plays an important role in many different IT Systems. it is often used for distributing data over the internet. So it is important for all Software developers to have a good understanding of XML.

⇒ XML was designed to be Self descriptive. ie,

1. It has Sender information
2. It has receiver "
3. It has a heading
4. It has a message body

⇒ XML became a W3C Recommendation as early as in February 1998 (world wide web Consortium)

Features (or) Advantages :

1. Separate data from HTML
2. Simple data sharing
3. Simple data transport
4. Increases data availability
5. Simple platform change

(2)

⇒ Example:

```
<?xml version = "1.0" encoding = "UTF-8" ?>
  <email>
    <to> Raju </to>
    <from> Kishore </from>
    <heading> Important message </heading>
    <body> Come to College on monday </body>
  </email>
```

Output.

email

To : Raju

from : Kishore

heading : important message

body : Come to College on monday

⇒ XML is designed to Carry data not to display data.

HTML VS XML :-

HTML	XML
(1) To display data	(1) to store and transport data
(2) markup language	(2) provide framework to define markup language
(3) non Case Sensitive	(3) Case Sensitive
(4) Static	(4) Dynamic

(5) HTML Example

```
<html>
<body>
<p> Student 1 </p>
</body>
</html>
```

(6) HTML Stands for hypertext markup language

(7) HTML focus on ~~Appea~~ Appearance and presentation

(8) Closing tags are not necessarily needed

(5) XML example.

```
<college>
<class 1>
<Student> Raju </Student>
</ class 1>
</ college>
```

(6) XML Stands for extensible markup language

(7) XML focus only on the exchange of information.

(8) Closing tags are used mandatory

Example 2: XML

```
<? xml version = "1.0" encoding = " UTF-8 " ?>
<college>
<class 1> CSE </class 1>
<Student> Shiva </Student>
<Roll no> 912 </ Roll no>
</college>
```

o/p:

```
college
  class 1 : CSE
  Student : shiva
  Roll no : 912
```

XML tags:

⇒ XML tags are important features of XML document. It is similar to HTML but XML is more flexible then HTML. It allows to create new tags (user defined tags). the first element of XML document is called root element. the Simple XML document contains opening tag and closing tag. XML tags are case Sensitive. :<root> and <Root> both are different

Properties of XML tags:

① Every XML document must have a root tag which enclose the XML document. it is not necessary to name root tag as root. the name of root tag is any possible tag name.

Example1:

```
<root>
<name> Sai </name>
<age> 22 </age>
</root>
```

Example-2

```
<class>
<name> Rushi </name>
<section> A </section>
</class>
```

② The starting tag is known as root tag. and we need to include tag in angle brackets <name>.

Example: <name>

③ the tag which is started by start tag must end

with the same tag with forward slash. in other words every XML document must be ended with end_tag. the end tag start with '<' followed by / and ends with >.

Example: <Name> Sai </Name>

- Both must be same
- forward slash at the ending

(3) XML tags are Case Sensitive. It means that <Root> and <root> both are different.

Example1: <Name> Sai </name>

it will display error because N and n both are different

Example2:

<Name> Sai </Name>

<name> Ravi </name>

(4) The text that appears between start-tag and end-tag is called Content. An element which has no content is known as empty.

Example: <name> </name>

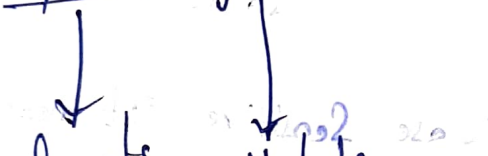
it is empty as there is no content in between tags

(5)

Attributes and Values in XML:

XML elements can have attributes just like HTML. Attributes are designed to contain data related to a specific element and attribute values must always be quoted. Either single or double quotes are used.

Example: `<person gender = "female"> </person>`



(or)

`<person gender = 'female'>`

Things to be considered while using attributes:

1. attributes cannot contain multiple values (where elements can)
2. attributes cannot contain tree structures (where elements can)
3. attributes are not easily expandable for future change. If we want to change attributes in feature that may be complicated

(4) Attributes are more difficult to be manipulated by program code.

Example: Same information can be Represented in two ways!

1st way:

<book publisher = "SIA"></book>

2nd way:

<book>
<publisher> SIA </publisher>
</book>

in first example publisher is used as an Attribute and in the second example publisher is an element. Both examples provide the same information but it is good to avoid attribute in XML and use elements instead of attributes.

XML Schemas: (or) XML Schema definition (XSD):

⇒ An XML Schema describes the structure of an XML document.

XML Schema language is also referred to as XML Schema definition (XSD)

⇒ XML schemas are written in XML and they are extensible. XML schemas support datatypes and name spaces. It is like DTD but provide more control on XML structure.

By using XML schema:

1. It is easier to describe document content
2. " " define restrictions of data.
3. " " Validate Correctness of data.
4. " " convert data between different datatypes

⇒ Another great strength about XML schema is that they are written in XML and you don't have to learn a new language and you can use your XML editor to edit your schema files.

⇒ you can use XML parser to parse your schema files.

we can verify data with XML Schema.

D)!

XML document.

XML

ey are

name Spaces

structure

types

s that

learn

editor

files.

Example:

Syntax: <XS:Schema xmlns:xs="">

Prefix

it is name Space

we define URI in it

Definition types

Simple type:

Used only in the Context of text i.e, Simple type allows you to have text based element. It Contains less attributes, child elements and it Cannot be left empty.

Complex type

The Complex type allows you to hold multiple attributes and elements. and it Can also Contain additional Sub elements. It Can be left empty.

⇒ you Can manipulate your Schema with the XML DOM. and you Can transform your Schema with XSLT.

⇒ with XSD (extensible Schema definition) you can :-

(a) Reuse your Schema in other Schemas
(b) Create your own datatype derived from the Standard types
(c) Reference multiple Schemas in Same document.

⇒ with XML Schemas, the Sender Can describe the data in a way that the receiver will understand.

⇒ In the XML world, hundreds of standardized XML formats are in daily use, many of these XML standards are defined by XML Schemas, and this XML Schema is an XML based and more powerful alternative to DTD

Example: Following Example shows how to use Schema:

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

<xs:schema xmlns:xs = "https://www.------" >
<xs:element name = "Contact" >
<xs:complexType >
<xs:Sequence >

<xs:element name = "name" type = "xs:String" />
<xs:element name = "company" type = "xs:String" />
<xs:element name = "phone" type = "xs:int" />

</xs:Sequence>
</xs:complexType>
</xs:element>

</xs:Schema>
```

Description of XML Schema:-

<xs:element name = "contact"> :- It defines the element name contact.

<xs:complexType> :- It defines that the element 'contact' is complex type.

<xs:sequence> :- It defines that the complex type is a sequence of elements.

<xs:element name = "name" type = "xs:string" /> :-
It defines that the element "name" is of string/text type.

XML DOM [Document object model]

⇒ The XML DOM defines a standard way for accessing and manipulating XML documents. It presents an XML document as a tree-structure. So, understanding the DOM is a must for anyone working with HTML or XML.

⇒ DOM stands for Document object model. DOM is a programming API (Application programming Interface) for HTML and XML document. DOM provides standard programming interface that can be used in a wide variety of environment and various Applications. This DOM can be used with any programming language.

=⇒ XML DOM makes a tree-structure view of an XML document we can Access all elements through the DOM tree. we Can modify or delete their Content and also Create new elements. The elements, their Content (text and attribut are all Known as nodes.

```
<?xml version="1.0" ?>
<books>
  <book>
    <author> Rahul </author>
    <price> 256 </price>
    <publishdate> 05/10/21 </publishdate>
  </book>
  <publication>
    <publisher> Nagendra </publisher>
    <State> telangana </State>
  </publication>
</books>
```

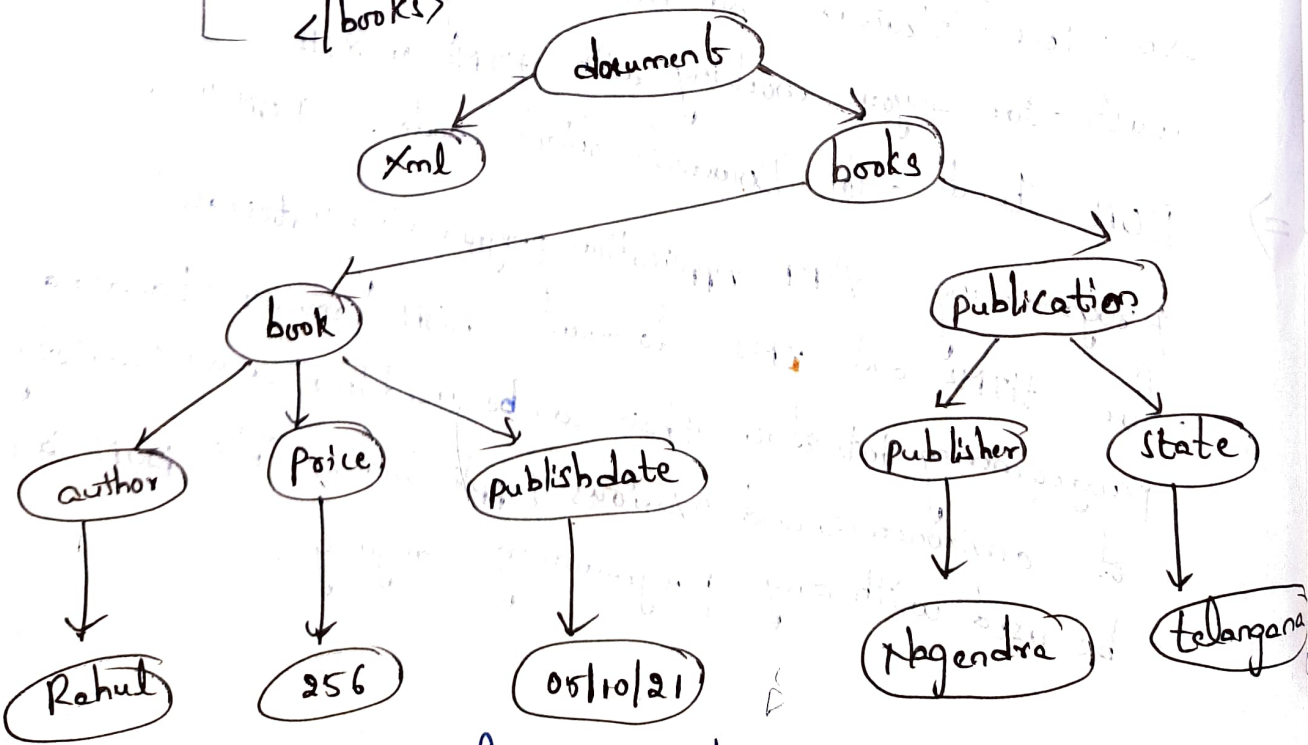


fig: XML document structure

⇒ Within the XML document structure, each circle represents a node, which is called an XML Node object. The XmlNode object is the basic object in DOM tree. Nodes have a single parent node; a parent node being a node directly above them. The only nodes that do not have a parent is the Document root, as it is the top level node and contains the document itself.

⇒ most nodes can have multiple child nodes, which are nodes directly below them. The DOM document is a collection of nodes or pieces of information organized in a hierarchy. The hierarchy allows a developer to find specific information easily by looking into the tree.

Advantages of XML DOM:-

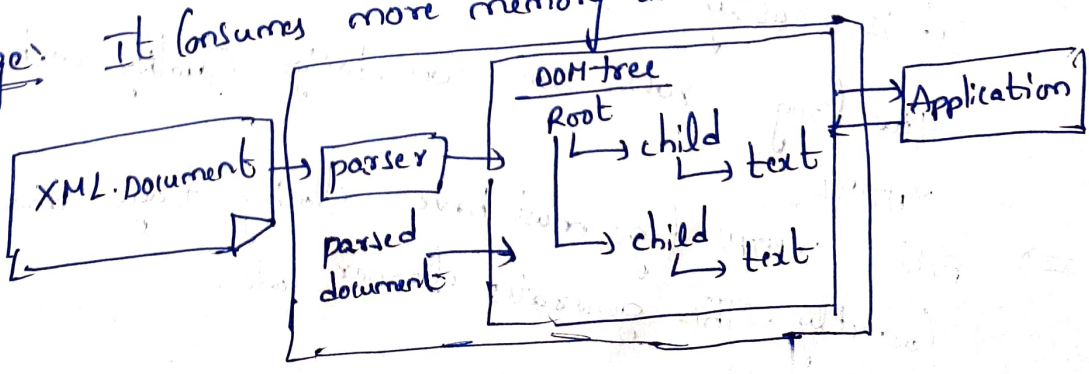
(1) XML DOM is language and platform independent

(2) XML DOM is traversable ie, information is arranged in hierarchy which makes us easy to find information

(3) XML DOM is modifiable ie, it is dynamic in nature providing the developer a scope to add, edit, move or remove nodes at any point on tree.

Disadvantage: It consumes more memory and it is slow.

XML DOM Structure :-



XHTML Parsing XML Data:-

⇒ XHTML Stands for Extensible Hypertext markup language. It is the next step to evolution of internet. XHTML was developed by world wide web Consortium (W3C). It helps web developers to make the transition from HTML to XML. Using XHTML, developers can enter the XML world with all the features of it,

⇒ The XHTML 1.0 is the first document type in the XHTML family and it is Recommended by W3C in 26 january 2000. The XHTML 1.1 is Recommended by W3C in 31 may 2001. at present we are using HTML 5. The XHTML document contains three parts, They are.

1. DOCTYPE :- It is used to declare a DTD.
2. head :- the head Section is used to declare title and other attributes.
3. body :- The body tag contains the content of web pages. It contains many tags.

⇒ XHTML documents are Validated with Standard XML tools.

⇒ It is easily to mantain, Convert, edit document in the long run. we can design Quality web pages in XHTML.

⇒ All XHTML tags must have closing tags and are nested correctly. This generates cleaner code.

⇒ XHTML works in association with CSS to create web pages that can easily be updated.

HTML	XHTML
⇒ HTML or hypertext markup language is the main markup language for creating web pages.	⇒ XHTML (Extensible hypertext markup language) is a family of XML markup languages that mirror or extend versions of the widely used hyper text markup language (HTML)
⇒ not Supported by all browsers	⇒ Supported by all major browsers.
⇒ proposed by Tim Berners lee in 1987	⇒ it is world wide web Consortium (w3c) Recommendation in 2000.
⇒ It is Application of Standard Generalized markup language (SGML)	⇒ It is Application of XML
⇒ Extended from SGML	⇒ Extended from XML, HTML
⇒ changes needed inorder to work in Various devices	⇒ works in all devices without any changes.

⇒ XHTML is almost identical to HTML but it is stricter than HTML. XHTML is stricter than HTML in Syntax and Case Sensitive. XHTML documents are well-formed and parsed using Standard XML parsers.

⇒ many pages on the internet contains "bad" HTML i.e not follow the HTML rule. HTML code works fine in most browsers even if it does not follow HTML rule.

```
<html>
<head>
<title> example of bad html </title>
<body>
<h1> Bad HTML
<p> This is Bad html
</body>
```

⇒ the above code doesn't follow the HTML rule although it runs and this HTML is not supported in smaller devices. Unlike HTML, XHTML doesn't facilitates you to make badly formed code to be. where simple errors like missing out a closing tag are ignored by browser. but XHTML strictly follows code rules, code must be exactly how it is specified to be.

⇒ In XHTML → <!DOCTYPE> is mandatory.
→ Xmlns attribute in <html> is mandatory
→ <html>, <head>, <title>, <body> are mandatory.
⇒ → Elements must always be properly nested and properly closed
→ Elements and attributes must be in lower case
→ Attribute names must be in lower case and Attribute values Quoted.

XML DTD (~~Data~~ Document Type Definition)

⇒ DTD Stands for Document type definition. It is a document that defines the structure of an XML document. It is also used to define elements and attributes of XML document.

⇒ This XML DTD is also used for performance Validation.

we can also called DTD as Document type declaration.

DTD Syntax:

```
<! DOCTYPE element DTD ;identifier
[ declaration 1
  declaration 2 ] >
```

Types of DTD

Internal DTD

elements are declared with in the XML files.

Syntax:

```
<! DOCTYPE root-element
[element-declaration] >
```

External DTD

elements are declared outside XML file

Syntax:

```
<! DOCTYPE root-element
SYSTEM "file-name" >
```

Internal DTD Example:

```
<? xml version = "1.0" encoding = "UTF-8" ?>
```

DTD

```
<! DOCTYPE Address [                    → root element
   <! Element Address(Name, Company, phone) >
   <! ELEMENT name (#PCDATA) >
   <! ELEMENT company (#PCDATA) >
   <! ELEMENT phone (#PCDATA) >
]>
```

xml.

```
<Address>
   <name> Sai </name>
   <company> DELL </company>
   <phone> 92488.... </phone>
</Address>
```

External DTD Example:

```
<?xml version="1.0" ?>
<!DOCTYPE Address SYSTEM "Add.dtd">
```

(Address = root element)

```
<Address>
<name> Sai </name>
<Company> DELL </Company>
<phone> 9248.. </phone>
</Address>
```

Add.dtd file

```
<!ELEMENT Address (Name, Company, phone)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Company (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
```

Characteristics:

⇒ It defines the Compulsory and optional elements in XML document.

⇒ It Validates the Structure of XML document.

⇒ It check for the grammar of XML document.

Advantages:

⇒ we Can define our own format for the XML files by DTD

⇒ It helps in Validation of XML file

⇒ It provides us with a proper documentation.

⇒ It enables us to describe a XML document efficiently

Disadvantages:

⇒ DTD's are hard to read and maintain if they are large in size

⇒ it is not object oriented

⇒ The Documentation Support is limited.

⇒ DTD doesnot Support namespaces.

DOM and SAX parsers in java :-

An XML parser is a software library or package that provides interfaces for client applications to work with an XML document. The XML parser is designed to read the XML and create a way for programmers to use XML.

XML parser validates the document and check that the document is well formatted.

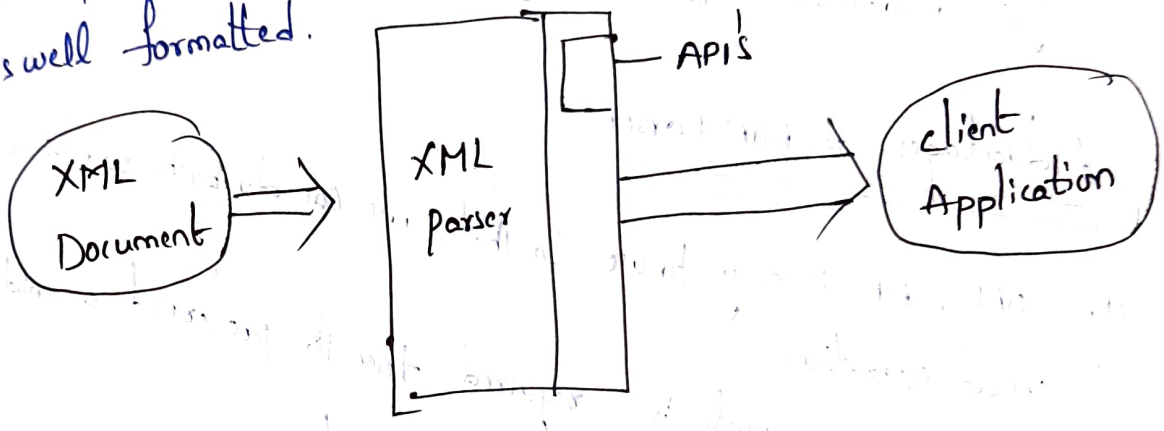


fig: working of XML parser

There are two main types of XML parsers :-

1. DOM
2. SAX

1) DOM (Document objects model) :-

A DOM document is an object which contains all the information of an XML document. it is composed like a tree structure. The DOM parser implements a DOM API. This API is very simple to use. DOM reads an entire document. It is useful when reading small to medium size XML files. The DOM API

Provides the classes to read and write an XML file.
we can insert and delete nodes using DOM API.

Features of DOM parser:

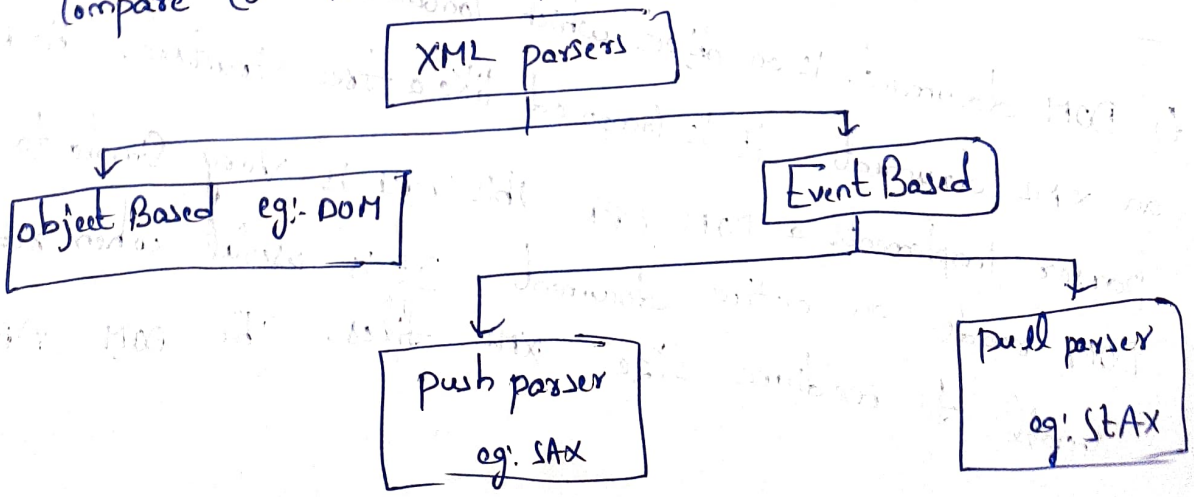
(1) The internal structure can be created by DOM parser.
(2) Because of these internal structure, the client can get information about original XML doc

Advantages of DOM parser:

(1) DOM API is easy to use so that we can do both write and read operations
(2) when a document is required then it preferred a wide part that can be randomly accessed.

Disadvantages of DOM parser:

(1) Its efficiency of memory is not too good, it takes more memory because XML docs needed to load in there.
(2) Compare to the SAX parser, it is too slow.



SAX parser:

SAX represents Simple API for XML. and a SAX API is implemented by SAX parser. This API was called event-based API which provides interfaces on handlers. there are four handler interfaces.

1. Content handler
2. DTD handler
3. Entity Resolver
4. Error Handler interface.

⇒ It doesnot ~~not~~ Create any internal structure rather it takes the occurrences of Components of an input document as events, and then it tells the client what it reads through the input document.

It is Suitable for large XML files because it doesn't requires loading the whole XML file.

Features of SAX parser:

1. The internal structure Cannot be Created by SAX parser
2. these event-based SAX parsers work the same as event handler in Java.

Advantages of SAX parser:

1. Very Simple to use and good efficient of memory.
2. Its runtime is too fast and it Can be work for a bigger document or file System.

Disadvantages of SAX parser:

1. Its Ability to understand API's is too less than an event-based API
2. we Can't Know the full information because of lot of piece of data

SAX PARSER

1. It is called as Simple API for XML parsing
2. It's an event based parser
3. SAX parser is slower than DOM parser
4. Best for smaller size files
5. It is suitable for making XML files in java
6. The internal structure cannot be created by SAX parser
7. It is readonly
8. In SAX parser backward navigation is not possible
9. Suitable for efficient memory
10. A small part of the XML file is only loaded in memory

DOM PARSER

1. It is called as Document object model
2. It is tree structure.
3. DOM parser is faster than SAX Parser.
4. Best for larger size of file.
5. It is not good at making XML files in lower memory
6. The internal structure can be created by DOM parser.
7. It can insert or delete nodes
8. In DOM parser backward and forward search is possible
9. Suitable for large XML document.
10. It loads whole XML document in memory

UNIT-3

SERVLETS

1. Introduction to Servlets ? Advantages ? Applications ?
2. Common Gateway interface (CGI) ? Difference between Servlets and CGI ?
3. Life cycle of Servlet [Architecture of Servlet]
4. deploying a Servlet [steps for deploying Servlet], Tomcat Server, Difference between tomcat and JDSK
5. The Servlet API [interfaces and classes]
6. Reading Servlet parameters [or] form Data]
7. Reading Initialization parameters
8. Handling Http Request & Responses
9. Using Cookies and Sessions
10. Connecting to a database using JDBC.

① Java Servlet

⇒ Servlets are the Java programs that runs on the java enabled web Server or Application Server. They are used to handle the request obtained from the web Server, process the request, produce the response, then send a response back to web Server.

Properties of Servlets:-

1. Servlets work on Server Side.
2. Servlets are capable of handling complex requests obtained from the web Server.

⇒ fig below shows Servlet Architecture.

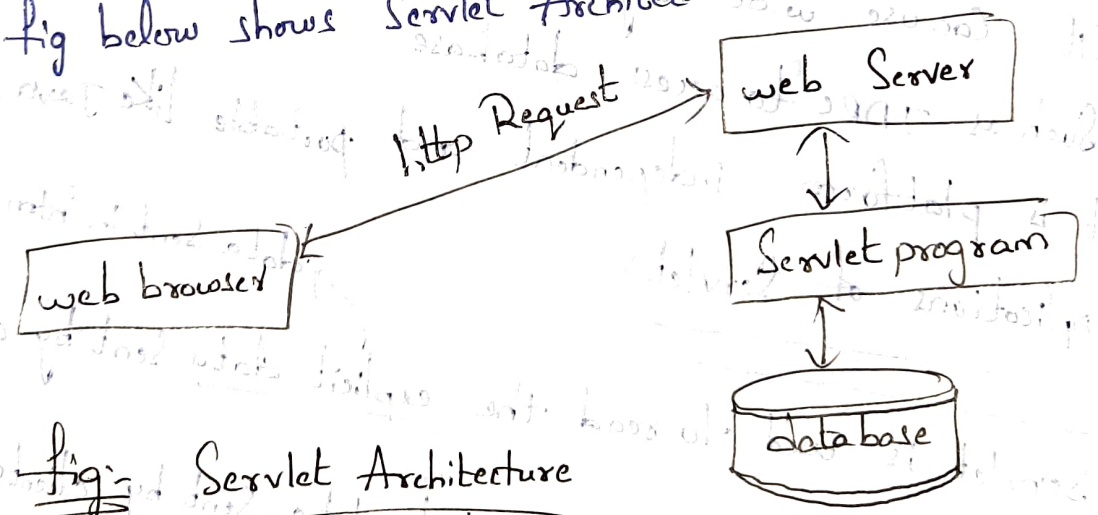


fig:- Servlet Architecture

⇒ Execution of Servlets basically involves Six basic steps:-

1. client Send the request to the web Server.
2. The web Server receives the request.
3. web Server passes the request to the corresponding Servlet.

④ The Servlet processes the request and generates Response in the form of output.

⑤ The Servlet Sends the response back to the web Server.

⑥ The web Server Sends the response back to client and client browser display it on the Screen.

Advantages of Java Servlet:

⇒ Servlet is faster than CGI as it doesn't involve the creation of a new process for every new request received.

⇒ Servlets, as written in Java, are platform independent.

⇒ need less memory with good Security

⇒ It Can use wide range of APIs Created on java platform Such as JDBC to access database

⇒ It is platform independent and portable like Java.

Applications of Servlet:

① Servlet is used to read the explicit data Sent by clients (data Sent in html form browser.

② Servlets is used to read implicit data Sent by clients (brows This includes cookies, media files, etc.

③ process the data and generate results

④ we can Send explicit data and Implicit data to browser.

⇒ Servlet techn at Server si

⇒ Servlet is an including d be implemen

② COMMON

CGI is actual using any of this is resp dynamic Cont

In CGI A access dynamic following operat

① when user Sen web page and Client reques

② for each client and it destroy client.

⇒ Servlet technology is used to create a web application resides at Server side and generates a dynamic web page.

⇒ Servlet is an API that provides many interfaces and classes including documentation. Servlet is an Interface that must be implemented for Creating any Servlet.

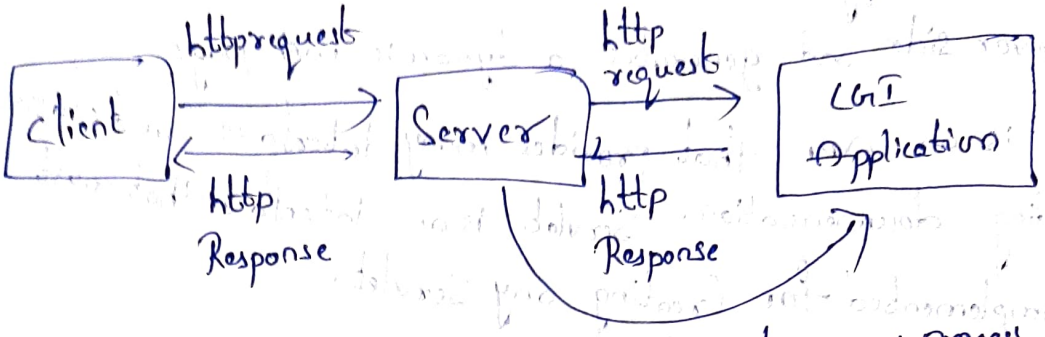
(2) COMMON GATEWAY INTERFACE (CGI)

, CGI is actually an External Application that is written by using any of the programming languages like C or C++ and this is responsible for processing clients request and generating dynamic Content.

In CGI Application, when a clients makes a request to access dynamic web pages, the web Server performs the following operations.

1) when user Send request then CGI first locates the requested web page and then Creates a new process to Service the Client request.

2) For each clients request CGI Application Creates new process and it destroy process after providing HTTP response to the client.



For each Request a new process will be created

fig: Common gateway Interface (CGI) for processing a clients Request

⇒ So, in CGI Server has to create and destroy the process for each request. It's easy to understand that this approach is applicable for handling few clients but as the number of clients increases, the workload on the Server increased and so the time is taken to process requests increased.

Difference between Servlet and CGI

(or)

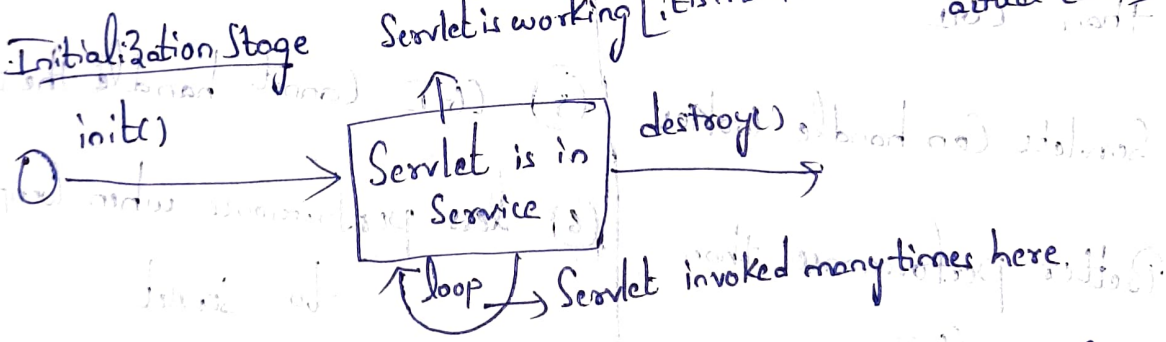
Advantages of Servlet over CGI.

Servlet	Common Gateway Interface (CGI)
① Servlets are portable and efficients	① CGI is not portable
② In Servlets, sharing data is possible	② In CGI, sharing data is not possible.
③ Servlets Can directly Communicate with web Server	③ CGI Cannot directly Communicate with the web Server
④ Servlets are less expensive than CGI	④ CGI is more expensive than Servlets.
⑤ Servlets Can handle cookies	⑤ CGI Cannot handle the cookies.
⑥ Better performance than CGI	⑥ less performance when Compare to Servlet
⑦ platform independent	⑦ platform dependent
⑧ we Can Communicate with other Applications like RMI (Remote method Invocation)	⑧ Cannot Communicate with other Applications
⑨ more Secure than CGI	⑨ less Secure Compared to Servlet.

(3) Life Cycle of Servlet :-

A Servlet lifecycle can be defined as the entire process from its Creation to till the destruction. The following are the paths followed by a Servlet.

(1) The Servlet is Initialized by calling the init() method.
(2) The Servlet calls service() method to process a client's request.
(3) The Servlet is terminated by calling the destroy() method.
(4) Finally, Servlet is garbage Collected by the garbage Collector of JVM.



As displayed in the above diagram, there are three states of a Servlet: new, ready, end. The Servlet is in new state if Servlet instance is created. After invoking the init() method Servlet comes in the ready state. in ready state, Servlet performs all the tasks. when the web container invokes the destroy() method, it shifts to end State.

(1) Servlet class is loaded:

The classloader is responsible to load the Servlet class. the Servlet class is loaded when the first request for the Servlet is received by web Container.

(2) Servlet instance is created:-

The web Container Creates the instance of a Servlet after loading the Servlet class. the Servlet instance is created only once in Servlet life Cycle

(3) init method is invoked:-

The web Container Calls the init() method only once after Creating the Servlet instance. The init method is used to initialize the Servlet.

Syntax:- public void init(ServletConfig Config) throws ServletException

(4) Service method is invoked:

The web Container Calls the Service method each time when request for the Servlet is received. if Servlet is not initialized, it follows the first three steps as described above and then Calls the Service method. if Servlet is initialized,

it Calls the Service method. Servlet is initialized only once.

The Syntax of Service method is

```
public Void Service (ServletRequest request, ServletResponse response)
        throws ServletException, IOException
```

(5) destroy method is invoked :-

The web Container Calls the destroy method before removing the Servlet instance from the Service. It gives the Servlet an opportunity to cleanup any resource for example memory, thread etc. The Syntax of destroy method is given below.

Syntax :- public Void destroy()

Following fig represent a typical Servlet life Cycle Scenario.

⇒ First the http request Coming to the Server are sent to the Servlet Container. The Servlet Container loads the Servlet before invoking the Service() method. Then the Servlet Container handles multiple requests by Sending multiple threads, each thread executing the Service() method of a Single instance of Servlet.

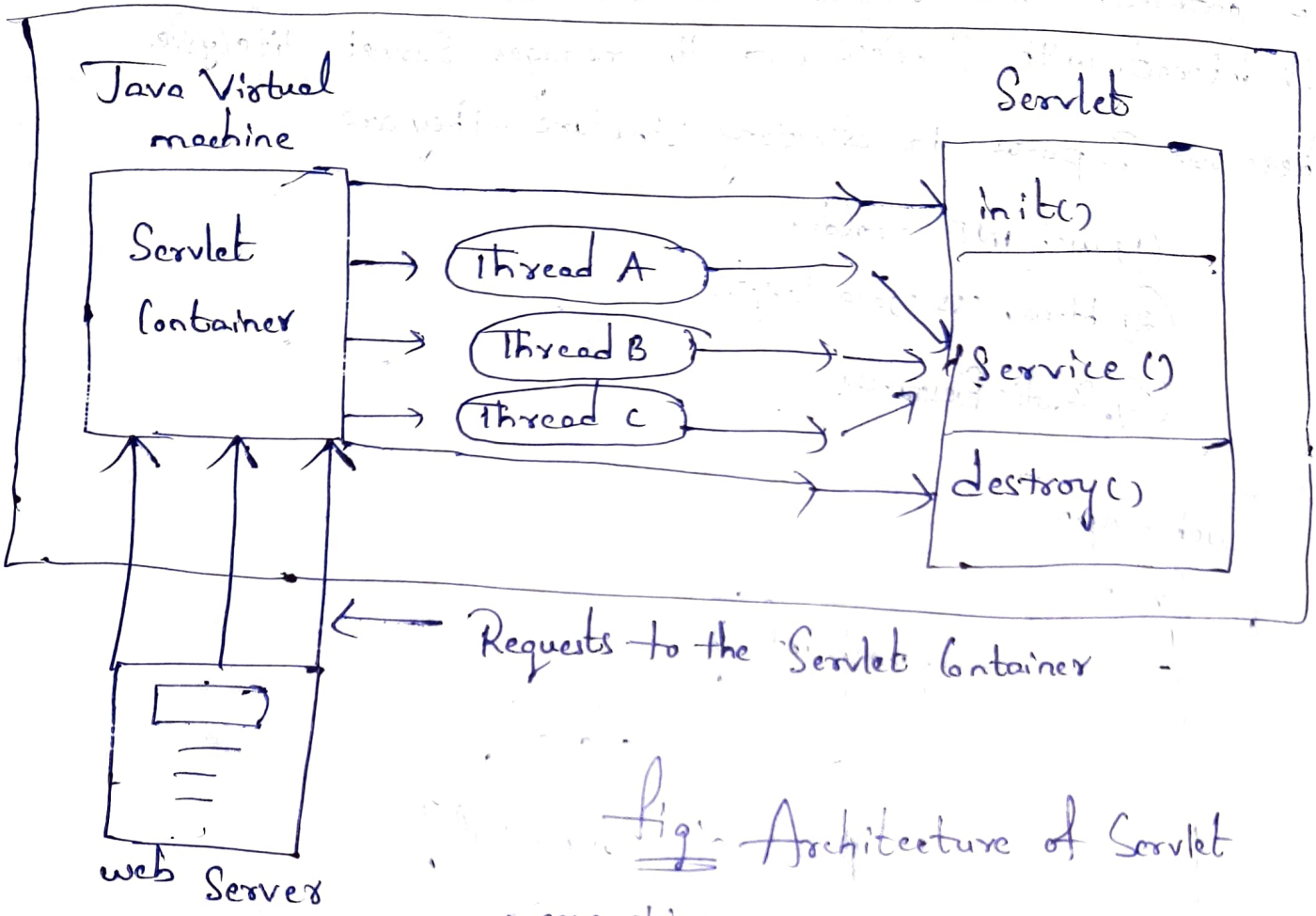


fig:- Architecture of Servlet

DEPLOYING A SERVLET

executing

Servlet deployment Steps:- There are Six steps in deploying a Servlet.

① Creating a directory Structure:-

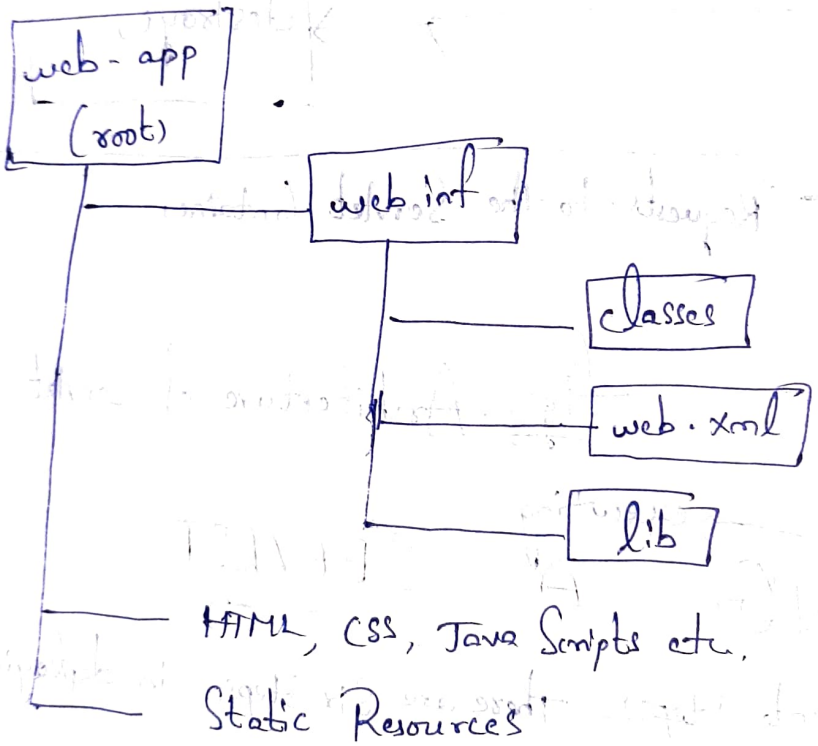
⇒ This directory Structure will define where to Store different kinds of files that are present in web Applications.

⇒ web Container will take information from the web Application and this web Container will Respond to the client.

web Container is a Component of web Server. this Container will interact with Servlets and it manages Servlet lifecycle

there are 3 parts in directory Structure. they are

- (1) web INF folder
- (2) Html, CSS, Java Scripts
- (3) Static Resources



(2) Creating a Servlet :-

we can create Servlet in 3 ways

(1) By implementing Servlet interface

(2) By inheriting Generic Servlet class

(3) By " Http Servlet class → this is most widely used

⇒ to handle Http Request, Servlets will provide ~~two~~ some method such as doget() method; dopost() method etc. to communicate with the client and Server

Syntax :- public class Demoservlet extends Http Servlet.

- Demo is name of Servlet we are extending
- we are using extends because we are inheriting
- Demo Servlet from Http Servlet.

3) Compiling a Servlet :-

inorder to compile Servlet we use 'jar file' with '.jar extension (Java archive)

this jar file can be loaded in 2 ways

① by Setting class path

② paste jar file in JRE/lib/ext (Java runtime environment)

Different Kinds of Servers uses different Kinds of Jar files

⇒ in Apache Tomcat Server → we use api.jar file

4) Creating a deployment descriptor.

this deployment descriptor is a Xml file (web.xml). It Contains all the information related to a Servlet.

it This xml file contains <servlet>, <servletname>, <servletclass>
<url-pattern> etc.
↓
name of Servlet

⇒ web container uses parsers to get information from (web.xml) file

(5) Start Server and deploy the project :

(5) to Start Server Goto → Apache tomcat/bin.directory
and then click on Startup.bat . This is the process to Start the Server.

to deploy the project, Copy and paste the project in webapps folder under tomcat.

(6) Accessing the Servlet:

⇒ http://hostname:portno/contextroot/urlpattern
↓ ↓ ↓
localhost 9999 demo

Tomcat Server, Difference between JDSK and Apache tomcat

⇒ Servlets are the best choice if we want to go for serious Server side programming. These Servlets needs a special environment in which they can be executed. For that we use Servlet Container. ie Tomcat.

⇒ Tomcat is open source product. It is maintained by Jakarta project of Apache Software foundation. Tomcat is basically a Servlet Container that contains the class libraries, documentation, runtime Support which is useful for executing and testing the Servlet.

For executing of Servlet following Software must be installed on Computer.

① JDK (Java development kit)

Servlets are basically Java files we should have JDK installed in our Computer.

② Tomcat :- Tomcat is Servlet Container using which Servlet can be executed.

Features of Tomcat:

① reduces Garbage Collection
② It improves performance & Scalability

it

③ parses Jsp efficiently

④ platform changability.

⇒ Tomcat is an opensource web Server ~~and Servlet container~~ developed by Apache Software foundation (ASF).

Tomcat 4.X released with following features:

5

① Cataline:- It is Servlet container. It helps to execute Servlets and Jsp. when you startup tomcat, you actually Startup Cataline.

② Coyote: It is a tomcat web Connector Component. It listen for Incoming Connectors to Server on Specific Tcp port and forwards the request to tomcat engine. to process the request and send back the response to client

③ Jasper: It is basically a tomcat Jsp engine. It parses the Jsp files & compile them to Java code as Servlet. Jasper detect the changes to Jsp files and recompiles them

JDSK :-

(Java development Standard Edition Kit)

It is available for Sun microSystem inorder to develop & deploy Java Applications on desktops and Servers. It is freely available in internet

JDSK	Apache tomcat
platform for developing the core Java Applications	It is web Server using which Server Side programs Such as Servlets or JSP's Can be executed
It is openSource product developed by Sun microSystem	Apache

To execute any Servlet or JSP we require both JDSK as well as Tomcat installed on our Computer.

SERVLET API

To represent interfaces and classes for Servlet api, we use two packages they are

1. javax.servlet
2. javax.servlet.http

it ⇒ The javax.Servlet package contains many interfaces and class
that are used by the Servlet or web container. these are not
Specific to any protocol

⇒ The javax.Servlet.http package contains interfaces and classes
that are responsible for http requests only

Interfaces in javax.Servlet package:

(5)

There are many interfaces in javax.Servlet package they are as follows.

1. Servlet
2. Servlet Request
3. Servlet Response
4. RequestDispatcher
5. Servlet Config
6. Servlet Context
7. Single thread model
8. Filter
9. Filter Config
10. Filter chain
11. Servlet Request Listener
12. " " Attribute listener
13. Servlet Context listener
14. Servlet Context Attribute listen

classes in javax.servlet package

There are many classes in javax.servlet package. They are

1. Generic Servlet
2. Servlet input Stream
3. Servlet output Stream
4. Servlet Request wrapper
5. Servlet Response wrapper
6. Servlet Request Event
7. Servlet Context Event
8. Servlet Request Attribute Event
9. Servlet Context Attribute Event
10. Servlet Exception
11. Unavailable Exception.

Interfaces in javax.servlet.http package they are :-

1. Http Servlet Request
2. Http Servlet Response
3. Http Session
4. Http Session Listener
5. Http Session Attribute Listener
6. Http Session Binding Listener
7. Http Session Activation Listener
8. Http Session Context

classes in javax.servlet.http package:

1. HttpServlet
2. Cookies
3. Http Servlet Request wrapper
4. Http Servlet Response wrapper
5. Http Session Event
6. Http Session Binding Event
7. Http Utils

Reading Servlet parameters:

⇒ ~~As we know~~, when you need to pass some information from your browser to web Server and ultimately to your backend program. the browser uses two methods to pass this information to web Server. these methods are

1. GET method
2. POST method

⇒ GET method is the default method to pass information from browser to web Server. The GET method sends the enloded user information added to page request

⇒ POST method Send information to backend program. It is same as get method except that the data which we have entered on form will appear on url.

Reading Form Data using Servlet:

Servlet handles form data parsing automatically using the following methods depending on Situation

① getparameter()

you call request.getparameter() method to get the value of a form parameter.

② getparametervalues()

you call this method if the parameter appears morethan once and returns multiple values, for example:- checkbox.

③ getparameterNames()

you call this method if you want a complete list of all parameters in the current request.

Example:

index.html

```
<form name = "welcome" method = "get"
action = "http://localhost/servlet/welcome">
Enter yourname <input type = "text" name = "name"><br>
<input type = "Submit" value = "login">
</form>
```

DemoServ.java

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
public class DemoServ extends HttpServlet {
public void doGet (HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException
{
```

```
res.setContentType("text/html");
PrintWriter pw = res.getWriter();

String name = req.getParameter("name");
pw.println("welcome" + name);

pw.close();
}
}
```

in the above example, we are displaying the name of the user in the Servlet. For this purpose, we have used the getParameter method that returns the value for the given request parameter name.

Reading Initialization Parameters:

most of the time, data (ex- admin email, database Username and password) need to be provided in production mode (client). initialization parameters can reduce the Complexity and maintenance,

we can pass Some parameters to the Servlet using web.xml file. using <init-~~parameter~~ Para> tag we can Specify name and

value of parameters with the help of <para-name> and <paravalue> tags.

Example of ServletConfig to get initialization parameter

In this example, we are getting the one initialization parameter from the web.xml file and printing this information in servlet

DemoServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class DemoServlet extends HttpServlet
    {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException
        {
response.setContentType("text/html");
PrintWriter out = response.getWriter();

ServletConfig config = getServletConfig();
String driver = config.getInitParameter("driver");
```

```
        out.print("Driver is :" +driver);
        out.close();
        }
    }
```

web.xml

```
<web-app>
  <servlet>
    <servlet-name> DemoServlet </servlet-name>
    <servlet-class> DemoServlet </servlet-class>
    <init-param>
      <param-name> driver </param-name>
      <param-value> Sun.jdbc.odbc.jdbcodbc driver </param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name> DemoServlet </servlet-name>
    <url-pattern> /Servlet1 </url-pattern>
  </servlet-mapping>
</web-app>
```

Handling Http Request & Responses:

We know that the client can make the request to the web-server using HTTP protocol. there is HTTP Servlet class in which there are some special methods which can be used to handle HTTP requests. These methods are.

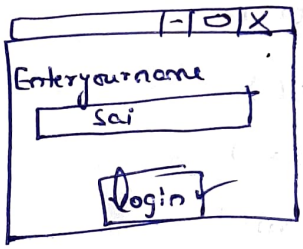
1. doDelete()
2. doGet()
3. doPost()
4. doPut()
5. doHead()
6. doOption()
7. doTrace()

For handling the input, the HTTP request makes use of two commonly used methods. Such as GET and POST In HTTP GET request the doGet method is used. In HTTP-post request the doPost request method is used. when user Sumbits his request using doGet method then URL String displays the request Submitted by user. But if doPost method is used then URL String does not show the Submitted Content. For handling httpRequest and Response here we write two programs.

1. HTML
2. Servlet.

ColorGet.html

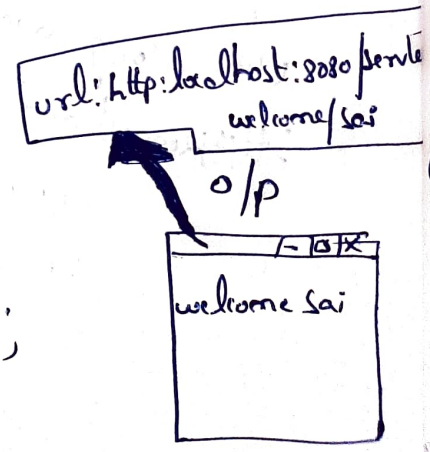
```
<html>
<body>
<center>
<form name = "welcome" method = "get" action =
http://localhost:8080/servlet/welcom
Enteryour name < input type = "text" name = "name"> <br>
<input type = "Submit" value = "login">
</form>
```



DemoServ.java

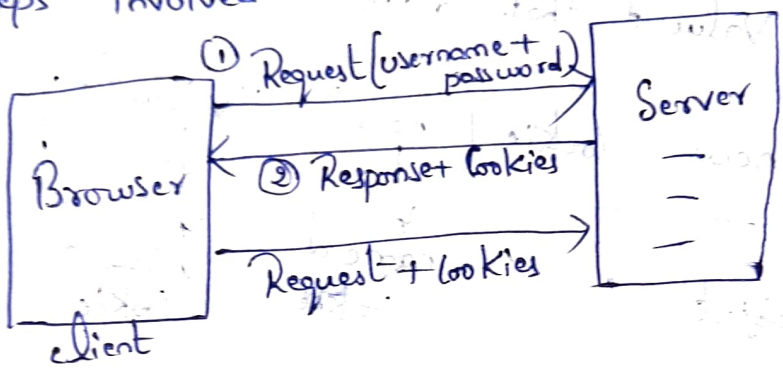
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Demoserv extends HttpServlet {
public void doGet (HttpServletRequest request, HttpServletResponse respo
        throws ServletException, IoException
{
res.setContentType("text/html");
printwriter pw = res.getwriter();
String name= req.getparameter("name");
pw.println("welcome" + name);
pw.close(); } }
```



Cookies in java Servlet:

- Cookies are the text file Stored on the client computer. they are used for tracking purpose.
- Steps involved in identification



- whenever user Send Request to Server by Entering username and password. then Server Respond back by Sending Cookies along with Response. when ever user ~~again~~ Send Request again to the Server. then Server Receive Request and as well as Cookies.
- Cookies are classified into 2 types

Non-persistent Cookie

Non-persistent Cookies are valid for Single Session only. ie Cookie is Removed each time when user closes the browser.

Persistent Cookie

Persistent Cookies are valid for multiple Session. ie Cookies are not removed from the browser even user closes the browser

Steps to Set Cookies in java Servlet:

① Creating Cookie object:

inorder to Create Cookie object we call Cookie Constructor with Cookiename and Value of that Cookie

```
Cookie c = new Cookie("name", "value");
```

- c → name of Cookie object
- "name" → name of Cookie
- "value" → value of Cookie

② Set maximum Age:

⇒ how long a Cookie should be valid (in Seconds).

that Can be Achieved by using Set maxage method.

Syntax

```
c.SetMaxAge(60*60*24);
```

- 60 → 60min
- 60 → 60seconds
- 24 → 24 hours (1day)

60 * 60 * 24 = 86400 Seconds in a day

③ Send Cookie into http response header:-

next we need to Send Cookie in to http response header. So that it Can be Stored on client side.

```
response.addCookie(c);
```

- c → name of Cookie object

⇒ A Cookie has a name, value, optional attributes such as a comment, path, domain, maximum age, version number.

Advantages:

① Simple technique of maintaining the state
② Cookies are maintained at client side

Disadvantages:

① It will not work if Cookie is disabled from the browser
② only textual information can be set in Cookie object.

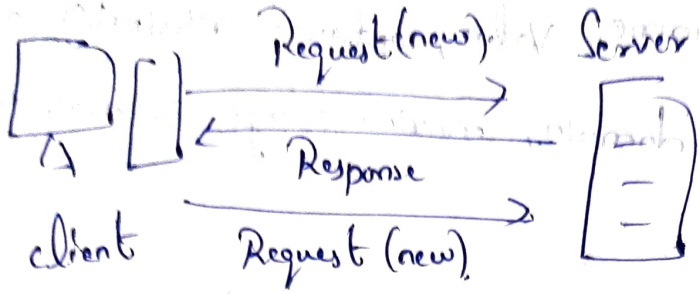
Sessions in java Servlet:

⇒ Session simply means a particular interval of time.

⇒ Session tracking is a way to maintain state(data) of an user. It is also known as Session management in Servlet.

⇒ Http protocol is a stateless so we need to maintain state using Session tracking technique. Each time user requests to the server, server treats the request as the new request. So we need to ~~maint~~ maintain the state of an user to recognize to particular user.

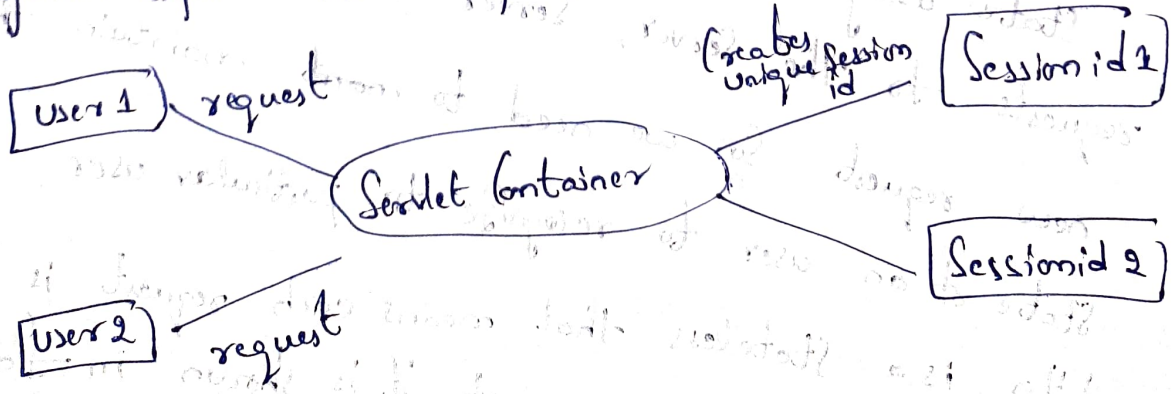
⇒ Http is a stateless that means each request is considered as the new request. it is shown in fig.



Inorder to achieve Session tracking in Servlets, Cookies have been one of the most commonly used technique. however, they have the following disadvantages.

1. they can keep only textual information
2. they are browser dependent. Hence, if client disable them, your web application cant make use of them
3. Individual Cookie can contain not morethan 4 kb of information

⟹ we can create Sessions with a unique Session id for each user. in java Servlet. For this, Servlets provide an interface called "Http Session interface". the following diagram explain how Http Sessions work in Servlet.



Advantages of http Sessions in Servlets:

① Any kinds of object can be stored into a session, be it a text, database, dataset etc.

② Usage of Sessions is not dependent on clients browser.

③ Sessions are secure and transparent

Disadvantages of http Session:

① performance overhead due to Session object being stored on Server.

② overhead due to Serialization and deserialization of data

there are four techniques used in Session tracking.

① Cookies
② hidden form field
③ URL Rewriting
④ http Session object

① Cookies:

A web Server can assign a unique Session ID as a Cookie to each client and for subsequent requests from the client they can be recognized using the received Cookie.

② Hidden form fields:

A web server can send a hidden HTML form field along with a unique session ID as follows

```
<input type = "hidden" name = "sessionid" value = "12345">
```

the entire meaning is that, when the form is submitted, the specific name and value are automatically included in the GET and POST data. Each time when web browser sends request back, then session_id value can be used to keep the track of different web browsers.

③ URL Rewriting:

you can add some extra data on the end of URL, that identifies the session.

Example: https://www.....com/file.htm;sessionid=123

sessionid to identify us

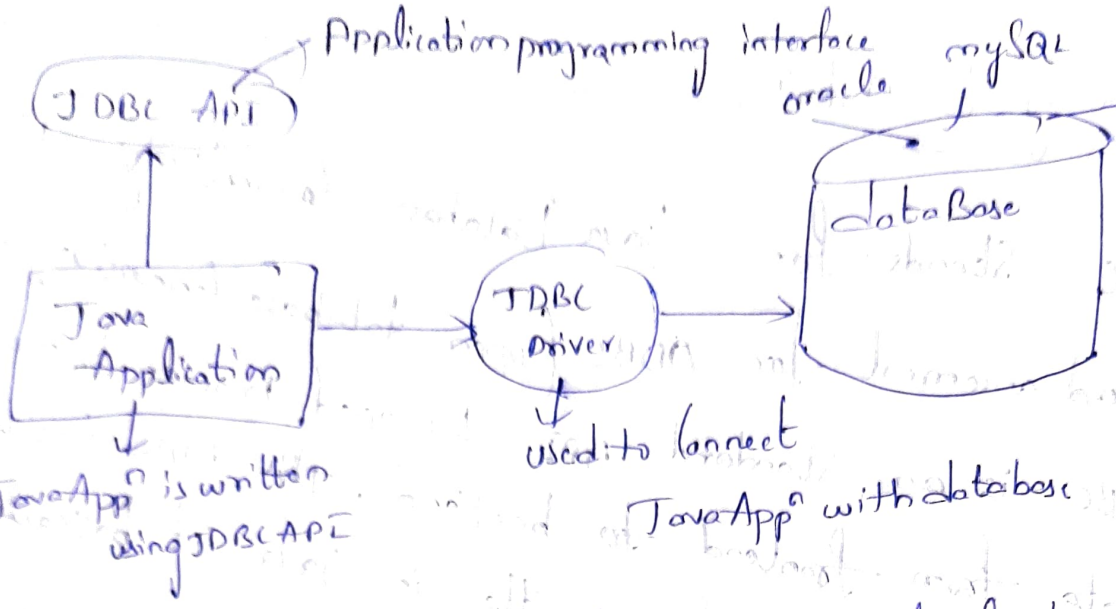
④ Http Session object:

Servlet provides Http Session interface which provides a way to identify a user across more than on page request or visit to a website and to store information about user.

Connecting to DataBase using JDBC

JDBC Stands for java Database Connectivity. It's an advancement for ODBC (open database Connectivity). JDBC is an Standard API developed in order to move data from frontend to backend. this API Consists of classes and interfaces written in java. It basically Acts as an interface or channel ~~or channel~~ between your Java program and databases. ie, it establishes a link between the ~~two So that a prog~~ Java program and database So that the programmer could send data from java Code and store it in the database for future use.

⇒ JDBC Came into Existence because, ODBC being platform dependent had a lot of draw backs. ODBC API was written in C, C++, python, Core java, as we know abov these languages (except java and python) are platform dependent. therefore to remove dependence, JDBC was developed by a database Vendor which Consisted of classes and interfaces written in java.



there are 5 steps to connect any java Application with the database using JDBC.

1. Register the Driver class
2. Create Connection
3. Create Statement
4. Execute Queries
5. close Connection

1. Register the driver class.

the forname() method of class class is used to register the driver class. this method is used to dynamically load the driver class.

Syntax: public static void forName( )

2) Create the Connection object:

the getConnection() method of DriverManager class is used to establish Connection with the database

example:

```
Connection con = DriverManager.getConnection("path", "username", "password");
```

Connection → class Connection
con → objects

3) Create Statement object:

the createStatement() method of Connection interface is used to create Statement. the object of Statement is responsible to execute queries with database.

```
Statement Stmt = con.createStatement();
```

Stmt → Statement name
con → Connection

4) Executing the Query:

the executeQuery() method of Statement interface is used to execute queries to the database. this method returns the object of ResultSet that can be used to get all the records of a tables

```
ResultSet rs = Stmt.executeQuery("select * from Emp");
```

rs → name of ResultSet
Stmt → Statement
"select * from Emp" → display table of Employees

④ ~~Close the Connection object~~.

```
while(rs.next())
    rs.getInt(), rs.getString();
```

⑤ Close the Connection.

By closing Connection object Statement and Result set will be closed automatically. The close() method of Connection interface is used to close the Connection.

```
con.close();
```

↓

Connection.

UNIT - 4

JSP (Java Server page)

(1) Introduction to JSP ? Advantages of JSP over Servlet ?
Architecture (or) processing JSP, lifecycle of JSP ?
Simple JSP Code Writing and Executing ?

(2) Cookies and Sessions in JSP ?

(3) Connecting to DATABASE in JSP ?

(4) Using Beans in JSP ?

(5) Anatomy (or) Components of JSP ? [Scriptlet tag, Expression tag, Declaration tag, Action tag, Custom tag Directives]

(6) Directives [page directives, include directives, taglib directive]

(7) Implicit objects.

1) Introduction to JSP

↓

Java Server pages

⇒ Java Server pages (JSP) is a technology that helps Software developers to create dynamic web pages based on HTML, XML, (or) other document types.

⇒ Using JSP, one can easily Separate presentation logic and business logic and java developers can write Server side complex computation code without concerning the web design.

⇒ Jsp's are released in the year 1999 by Sun micro Systems JSP is Similar to the java programming language. To deploy and run java Server pages, a compatible web Server with a Servlet Container such as Apache tomcat is required

JSP = Html + java code embedded

Advantages of JSP over Servlet:

⇒ JSP technology is used to create web Application just like Servlet technology. JSP is an extension to Servlet because it provides more functionalities than Servlet. Such as expression language, JSTL etc..

A JSP page Consists of HTML tags and JSP tags. the JSP pages are easier to mantain than Servlet because we can Separate designing and development. It provides some additional features Such as Expression language, Custom Tags etc..

There are many advantages of JSP over the Servlet. they are as follows.

① Extension to Servlet

JSP technology is the extension to Servlet technology. we can use all the features of the Servlet in JSP. in addition to, we can use implicit objects, predefined tags, expression language, Custom tags in JSP, that makes JSP development easy.

② Easy to maintain:

JSP can be easier to manage because we can easily Separate our business logic with presentation logic. In Servlet technology, we mix our business logic with presentation logic.

③ Fast Development :- No need to recompile and redeploy

If JSP page is modified, we don't need to recompile and redeploy the project. the Servlet code needs to be updated and recompiled if we have to change the look and feel of an application

④ Less code than Servlet:

In JSP we can use many tags Such as action tags, JSTL, Custom tags etc. that reduces the code.

Life cycle of a JSP page & Architecture of JSP (or) JSP processing

The JSP pages follow these phases :

1. Translation of JSP page
2. Compilation of JSP page
3. class loading [the class loader loads class file]
4. Instantiation [object of Generated Servlet is created]
5. initialization [the container calls jsp init() method]
6. Request processing [the container invokes / calls jsp Service() method]
7. Destroy [the container invokes / calls jsp Destroy() method]

(1, 2, 3, 4, 5, 6, 7 — life cycle methods)

the web Server needs a JSP Engine or JSP container to process JSP pages. the JSP container is Responsible for taking request of JSP page.

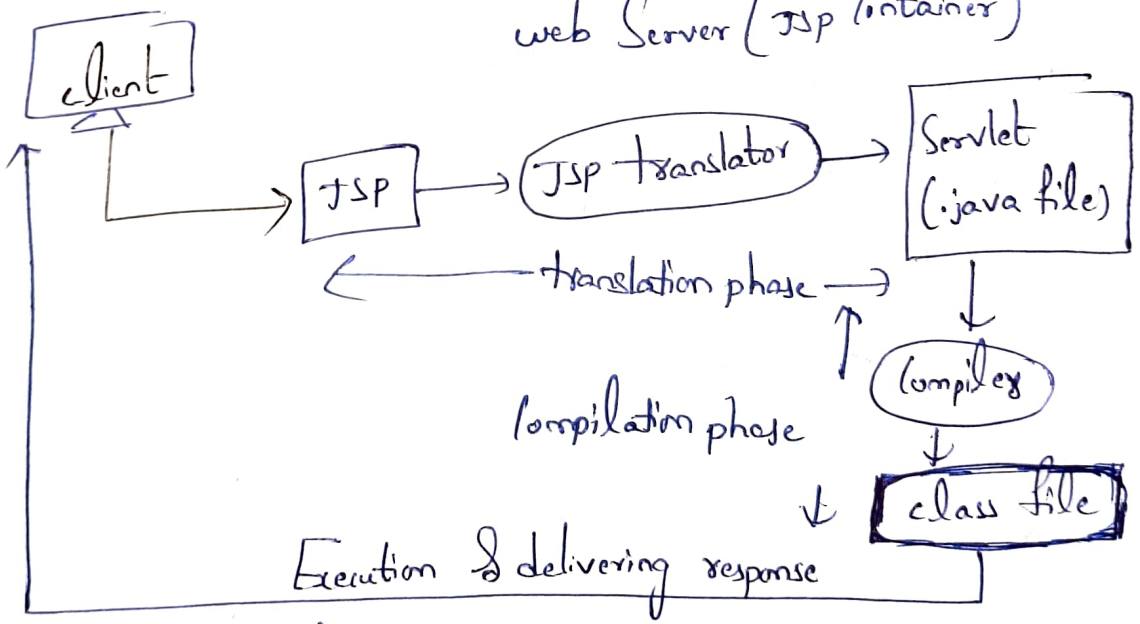


fig: JSP Architecture

As shown in the above diagram, JSP page is translated into Servlet by the help of JSP translator. The JSP translator is a part of web Server which is responsible for translating

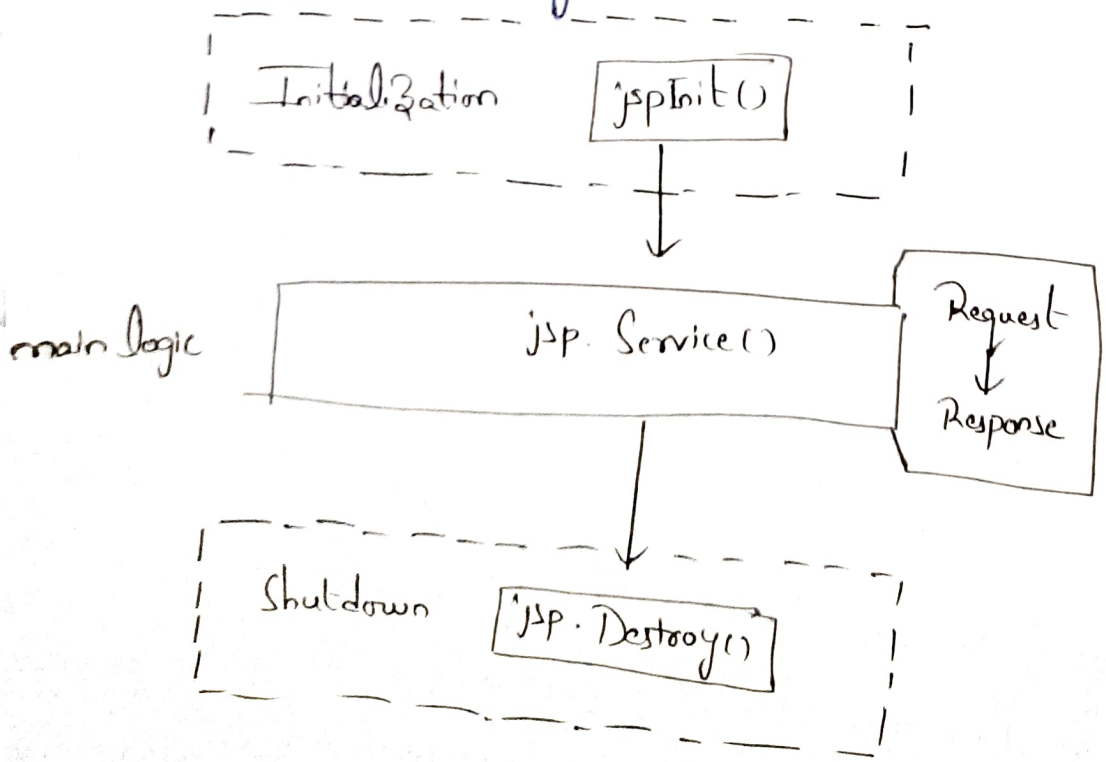
the JSP page into Servlet. After that, Servlet page is compiled by the compiler and gets converted into the class file. the class file is executed and output of execution is sent to client as response.

Now lets see the life cycle of Java Server page (JSP)

JSP life cycle:

A JSP life cycle can be defined as the entire process from its creation to till its destruction which is similar to the Servlet life cycle with a additional step which is required to compile a JSP into Servlet.

the major phases of JSP life cycle are very similar to Servlet life cycle and they are as follows.



① JSP Compilation:

when a browser asks for a JSP, the JSP engine first check to see whether it needs to compile the page. If the page has never been compiled, or if the jsp has been modified since it was last compiled, the JSP engine compiles the page.

② JSP Initialization:-

When a Container loads a JSP it invokes / calls the jspInit() method before servicing any requests. If you need to perform JSP-specific initialization use jspInit() method

```
public void jspInit()
    {
    // initialization code ----
    }
```

Typically initialization is performed only once ~~and as with~~

③ JSP Execution:-

JSP Service() method is used to serve the raised requests by JSP. It takes request and response objects as parameters ie, It takes HttpServletRequest and HttpServletResponse as its parameters.

The JSP Service() method is ~~called~~ invoked once per each request and is responsible for generating one response for the request.

JSP

```
Void JspService (HttpServletRequest request,
                 HttpServletResponse response)
{
  // Service handling code
}
```

(4) JSP cleanup

⇒ The destroy phase of JSP life cycle ~~represents~~ is used when we want to remove JSP from container. This method is called only once, if you need to perform cleanup tasks like closing file releasing database connection etc..

⇒ jspDestroy() method is used inorder to destroy the method

```
public void jspDestroy()
{
  // your cleanup code goes here....
}
```

SIMPLE JSP CODE WRITING AND EXECUTING

for executing JSP code we must have

(1) JDK installed

(2) Apache tomcat installed

Step 1: open text editor and type following code

hello.jsp

```
<html>
<body>
<% out.println ("this is my first JSP page!"); %>
</body>
</html>
```

o/p:- this is my first JSP page

Create a Separate directory at the path

C:|your-tomcat-directory|webapps|jsp-examples & store above code in newly created directory.

I have created a directory named HelloDemo in which I have stored above program by name hello.jsp. Note that while Saving file using Notepad editor with .jsp extension, you must Select all files option. If you donot do that then file may Saved as text file because the default extension for notepad is txt.

Step-2:-

Start Tomcat web Server by typing command Startup at command prompt or by clicking Startup file.

Step-3:

Open Some ~~web~~ web browser like firefox or IE. Type path for JSP page prefix http://localhost:8080.

Note that localhost is default DNS for tomcat webServer

http://localhost:8080/foldername/yourpgm.extension ie,

http://localhost:8080/examples/hello.jsp.

2. Using Cookies and Session for Session tracking

JSP Cookies handling:

Cookies are text files stored on client computer and they are kept for various information tracking purposes. JSP Supports HTTP Cookies using Servlet technology.

There are three steps involved in identifying and returning users.

① Servlet Script Sends a set of Cookies to browser.
For example, name, age, id number etc.

② Browser stores this information on local machine for future use.

③ when the next time the browser Sends any request to the web Server then it Sends those Cookies information to Server and Server uses that information to identify user or may be for Some other purpose as well.

Setting Cookies with JSP.

Setting Cookies with JSP involves three steps

Step①: Creating a Cookie object

you call the Cookie Constructor with a Cookie name and a Cookie value, both of which are Strings. Keep in mind, neither the name nor the value should Contains white Space or any of the characters like [], (), =, ,, /, ?, @, :, ;.

```
Cookie cookie = new Cookie("Key", "value");
```

Step②: Setting the maximum age

you use SetMaxAge to Specify how long (in Seconds) the Cookie should be valid. the following Code will setup a Cookie for 24 hours.

```
cookie.SetMaxAge(60*60*24);
```

Step③: Sending the Cookie into the HTTP response header

you use response.addCookie to add Cookies in the HTTP response header as follows.

```
response.addCookie(cookie);
```

Types of Cookies in JSP

2 types

Persistent Cookies (or) Permanent Cookies

they remain on hard drive and present until the user delete them or they expire themselves

Session Cookies (or) temporary Cookie

They get deleted themse as soon as the Session ends or browser close

Structure of Cookies in JSP :

A Cookies sent by a JSP page in HTTP header looks like this

```
HTTP/1.1 200 OK
Date: Sat, 25 Nov 2021 10:03:38 GMT
Server: Apache tomcat /9.0.34 (windows)
Set-Cookie : name = my_name ; expires = Sun, 26-Nov 2021, 10:03:38 GM
  path = // ;  domain = Sai.com
Connection: close
Content-Type = text / html.
```

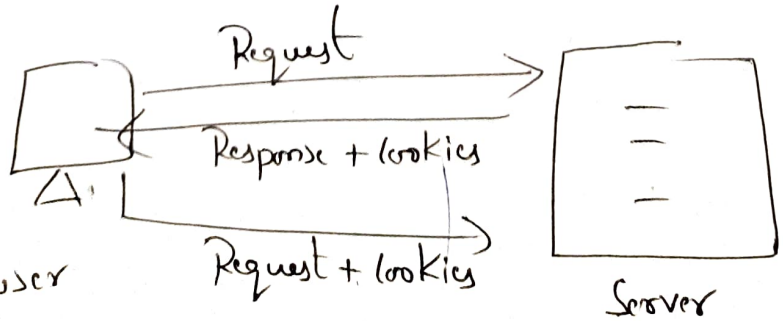
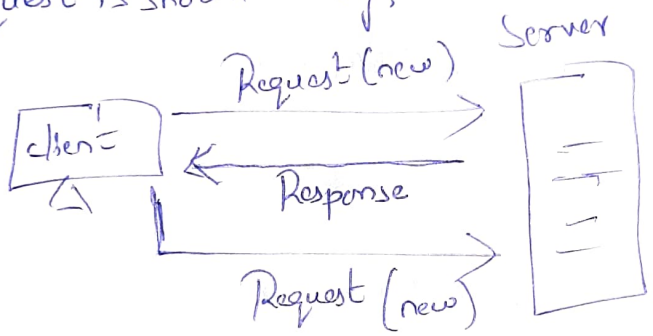


fig: Structure of Cookies.

Sessions in JSP :-

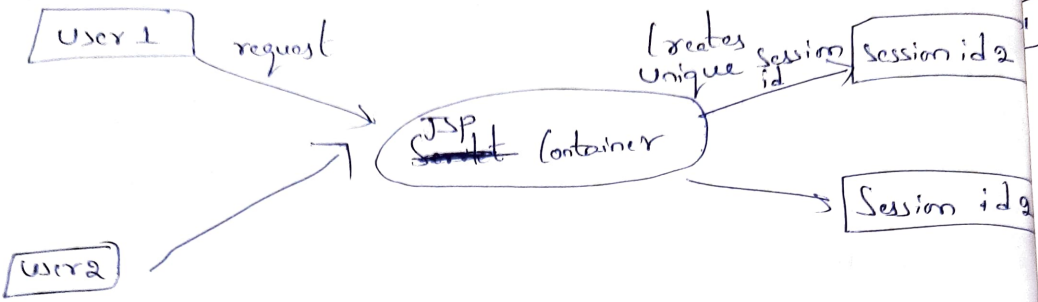
Session Simply means a particular interval of time. Session tracking is a way to maintain State(data) of an user. It is also known as Session management in JSP. Http protocol is a stateless So we need to maintain State using Session tracking technique. Each time user requests to the server. Server treats the request as the new request. So we need to maintain the State of an user to recognize to particular user.

http is a stateless that means each request is considered as the new request is shown in fig.



Inorder to Achieve Session tracking in JSP, Cookies have been one of the most commonly used technique however they have the following disadvantage

1. They Can keep only textual information
2. They are browser dependent. Hence, if client disable them, your web application Cant make use of them
3. Individual Cookies Can Contain not morethan 4kb of information.



Advantages of http Sessions in JSP

1. Any kinds of object can be stored into a Session, be it a text, database, dataset etc.
2. Usage of sessions is not dependent on clients browser
3. Sessions are Secure and transparent.

Disadvantages of http Session:

1. performance overhead due to Session object being Stored on Server.
2. overhead due to Serialization and deserialization of data

there are four techniques used in Session tracking.

1. Cookies
2. hidden Form field
3. URL Rewriting
4. http Session object.

(1) Cookies:-

web Server Can assign a unique Session ID as a Cookie to each client and for Sub Sequent requests from the client they Can be recognized using the received Cookies.

(2) Hidden Form Fields:-

A web Server Can Send a hidden HTML form field along with a Unique Session ID as follows.

```
<input type = " hidden name = "sessionid" value = "12345" >
```

the entire meaning is that, when the form is Submitted, the Specific name and value are automatically included in the GET and POST data. Each time when web browser Sends requests back, then Sessionid value, Can be used to keep the track of different web browser.

(3) URL Rewriting :-

you Can add Some extra data on the end of URL that identifies the Session.

Example: Https:// www. . . . .Com/ file.htm ; Sessionid=123

Sessionid to identify user

(4) Http Session object:

Servlet provides http Session interface which provides a way to identify a user across more than one page request or visit to a web Site and to store information about use

Connecting to DATABASE in JSP :-

The database is used for storing various types of data which are huge and has storing capacity in giga bytes. JSP can connect with such databases to create and manage the records.

while accessing JSP database ~~we~~ from JSP page we should have some DB packages installed. In this section we will discuss the connectivity of JSP with mySQL database.

Prerequirements :-

1. Tomcat web Server
2. MySQL Server
3. JDK

Step (1) :- Creating a database named Students in MySQL using following commands.

```
mysql > CREATE DATABASE Students ;
```

then create a table named Students_table in the Students database as follows:

```
mysql > use Students ;
mysql > CREATE TABLE Students_table (
        roll_no INT(4) NOT NULL AUTO_INCREMENT
```

name VARCHAR(50) NOT NULL,

address VARCHAR(50) NOT NULL,

phoneno VARCHAR(15) NOT NULL,

PRIMARY KEY (roll_no)

Student database:

Students table

name	address	phoneno	rollno

Step②: for establishing the Connectivity between JSP & MySQ using JDBC driver. what we need is to download MySQL JDBC Connector. Download this Connector from http://www.mysql.com/products/connector/j/. Just pickup the mirror and download ZIP file. then from the extracted folder just copy jar file name mysql-connector-java-xx-bin.jar to C:/your-tomcat-directory/common/lib. then just set CLASS using environmental Variables for that purpose goto Control panel → Syst System properties → environmental Variables & Set classpath

Variable name : CLASSPATH

Variable value : c:/your-directory/common/lib/mysql-connector-java-3.1.19-bin

OK Cancel

Using Beans in JSP :-

Java beans are reusable Components. we Can use Simple java bean in JSP. this helps us in keeping the business logic Seperate from presentation logic. Beans are used in JSP as instance of class. we must Specify Scope of the Bean in JSP page. Here Scope of bean means. how much time bean exists in JSP. when the bean is present in Scope its id is available in that Scope.

there are Various Scopes using which the bean Can be used in JSP. They are.

1. **page Scope** :- It is default Scope. the bean object gets disappear as Soon as Current page gets closed.
2. **Request Scope** :- the bean object remains in existence as long as the request object is present
3. **Session Scope** :- Bean object remains from starting to ending time of user in internet or Browser.
4. **Application Scope** :- It is broad cast Scope provided by JSP. It should be used only once when it is necessary.

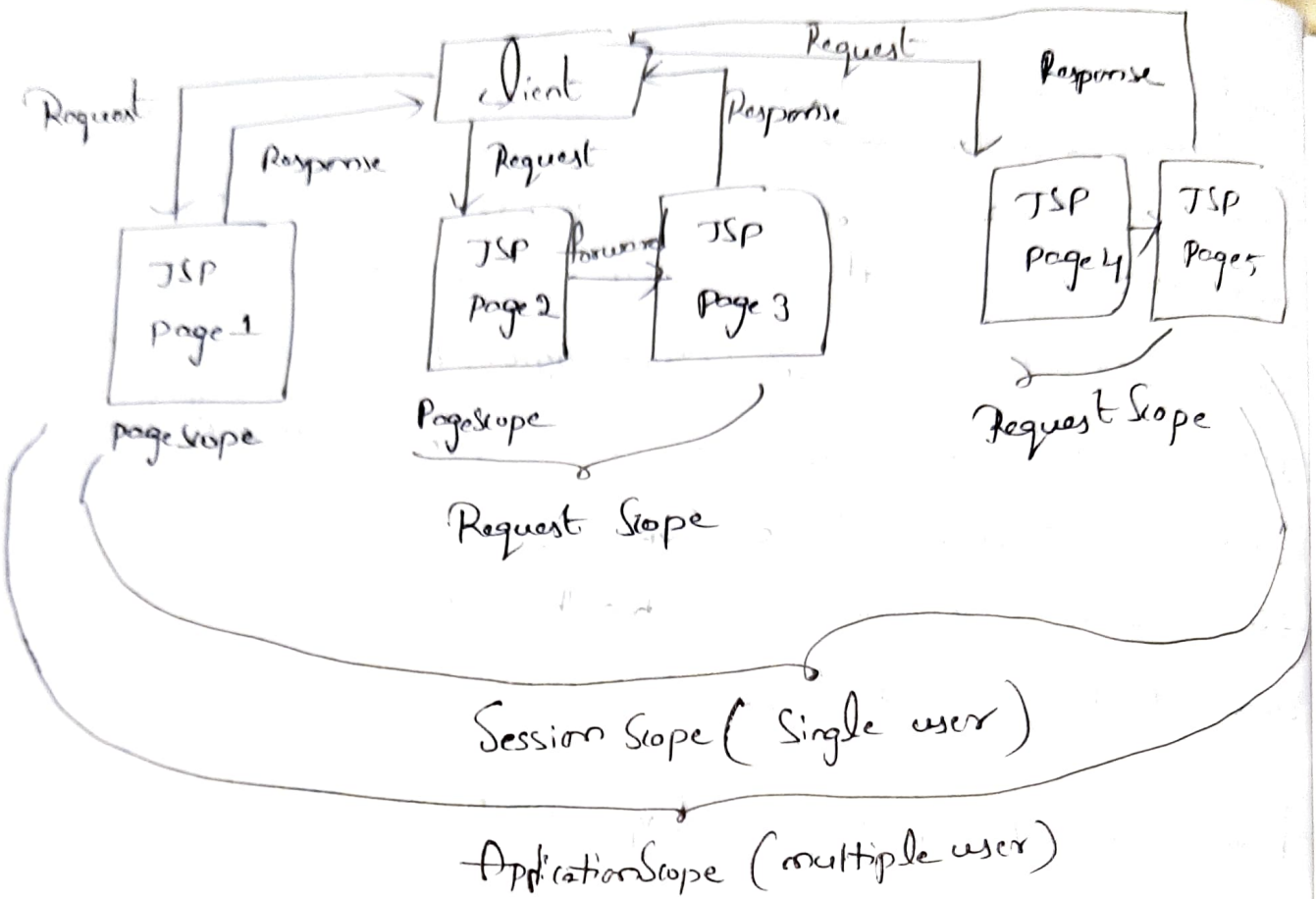


fig: Various Scopes.

Components of JSP Bean :-

① <jsp:usebean> It is nothing but a reusable software component
It is used to specify scope of bean.

② <jsp:Set property> this action tag is used to set the values to bean class properties by calling setter methods.

<jsp:Set property> tag must be used inside of <jsp:useBean> tag

③ <jsp:getproperty>
this tag is used to read the values of a bean class property by calling getter method.
this tag must be used out side of <jsp:UseBean> tag.

Anatomy of JSP / Components of JSP :-

JSP is built using Components Such as

1. Scriptlets tag
2. Expressions tag
3. Declarations tag

} Scripting elements → provides ability to insert java code inside the jsp.

4. Action tags
5. Custom tags
6. Directives

(1) Scriptlet tag :-

⇒ In JSP, java code can be written inside the jsp page using the Scriptlet tag. A scriptlet tag is used to execute java Source code in jsp.

Syntax :- `<% java Source code %>`

Example of JSP Scriptlet tag :-

```
<html>
<body>
<% out.print("JSP is equal to HTML + Java"); %>
</body>
</html>
```

Example of JSP Scriptlet tag that prints the user name

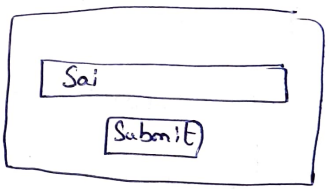
in this example we have created two files

1. index.html
2. welcome.jsp

the index.html file gets the username from the user and the welcome.jsp file prints the username with welcome message

index.html

```
<html>
<body>
<form action = "welcome.jsp">
<input type = "text" name = "username">
<input type = "submit" value = "go" > <br/>
</form>
</body>
</html>
```



welcome.jsp

```
<html>
<body>
<%
String name = request.getParameter("username");
out.print("welcome" + name);
%>
</body>
</html>
```

o/p: welcome Sai

(2) Expression tag :

the code placed within Jsp expression tag is written to generate output as response. So you need ~~to write~~ out not to write out.println() to write data. it is mainly used to print the values of variable or method

Syntax :- <%= Statement %>

Example :

in this example of jsp expression tag, we are simply displaying a welcome message

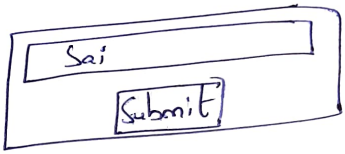
```
<html>
<body>
<%= "welcome to jsp" %>
</body>
</html>
```

Example of jsp expression tag that prints the user name

In this example, we are printing the username using the expression tag. the index.html file gets the user name and sends the request to welcome.jsp file. which display user name.

index.html

```
<html>
<body>
<form action = "welcome.jsp">
<input type = "text" name = "username">
<input type = "Submit" value = "go">
</form>
</body>
</html>
```



welcome.jsp:

```
<html>
<body>
<%= "welcome" + request.getParameter("username") %>
</body>
</html>
```

o/p: welcome Sai

(3) Declaration tag:-

⇒ The JSP Declaration tag is used to declare fields and methods.

⇒ the code written inside the jsp declaration tag is placed outside the Service() method of Servlet So it doesn't occupy memory for each request.

Syntax: `<%! field or method declaration %>`

Difference between JSP Scriptlet tag and Declaration tag

JSP Scriptlet tag	JSP Declaration tag
it can declare only Variables not methods	it can declare Variables as well as methods
the declaration is placed inside jspService() method	the declaration is placed outside the Service method

Example:

```
<html>
<body>
<%! int data = 50; %>
<%= "Value of Variable is" + data %>
</body>
</html>
```

o/p: Value of Variable is 50

JSP Directives:

The jsp directives are messages that tells the web container how to translate a JSP page into the corresponding Servlet.

There are three types of directives.

1. page directives
2. include directive
3. taglib directive

Syntax of JSP directive : `<%@ directive attribute = "value" %>`

1. JSP page directives:

The page directive defines attributes that are applied to an entire jsp page.

Syntax: `<%@ page attribute = "value" %>`

Attributes in JSP page directives are:

1. import
2. content type
3. extends
4. info
5. buffer
6. language
7. isELIgnored
8. ~~at~~ autoFlush
9. Session
10. page Encoding
11. errorpage
12. & Error page

1) import:-
The import attribute is used to import class, interface or all the packages. It is similar to import keyword in java class or interface.

Example:- <%@ page import = "java.util.Data" %>

2) Content type:-
The ContentType attribute defines the MIME (multipurpose internet Mail Extension) type of the HTTP response.

Example:- <%@ page Contenttype = application/msword %>

3) extends:-
The extends attributes defines the parent class that will be inherited by Servlet. It is rarely used

1) info:-
This attribute simply sets the information of JSP page.

Example:- <%@ page info = "Composed by Nagendra" %>

4) buffer:-
The buffer attribute sets the buffer size in kilobytes to handle output generated by JSP page. The default size of buffer is 8 Kb.

Example:- <%@ page buffer = "16 Kb" %>

⑥ language:

the language attribute Specify the Scripting language used in JSP page. default value is "Java".

⑦ is EL Ignored:

we can ignore the Expression Language (EL) in jsp by the is EL Ignored Attribute.

⑧ errorpage:

the error page attribute is used to define the error page if exception occurs in Current page, it will be redirected to error page.

Example: <%@ page errorpage = "myerrorpage.jsp" %>

⑨ is 'Error page':

the is Errorpage attribute is used to declare that the Current page is the error page.

Example: <%@ page is Error page = "true" %>

(2) Jsp include Directive:

the include directive is used to include the contents of any resource it may be jsp, html, text file etc.

Advantage of include Directive is Code Reusability

Syntax: <%@ include file = "resource name" %>

Example: <%@ include file = "Sai.html" %>

(3) JSP Taglib directive:

the jsp taglib directive is used to define a tag library that defines many tags. we use the tag Library Descriptor (tld) file to define the tags.

Syntax: <%@ taglib uri = "uri of tag library" prefix = "prefix of tag lib" %>

Example:

<%@ taglib uri = "https://www.nslectures.com/tags" prefix = "mytag" %>

Implicit objects:

The implicit objects are predefined Variables used to access request and application data. these objects are used by Scripting elements.

Variable name	class/Interface name	meaning	Sample methods
① application	javax.Servlet. ServletContext	this object provides resources shared with in a web application	log( ) getServerInfo( )
② Config	javax.Servlet. ServletConfig	It helps in passing information to Servlet or JSP page during initialization	getInitParameter( ) getServletName( )
③ request	javax.Servlet.http HttpServletRequest	It provides the method for Accessing information made by Current Request	getContentLength( ) getLocalAddress( ) getServerName( )
④ out	javax.Servlet.jsp JSP writer	It provides the method related to Info	clear( ) newline( )
⑤ response	javax.Servlet.http Http ServletResponse	It provides method related to adding Cookies, Sessions or Setting	addCookie( ) addHeader( ) flushBuffer( ) getContentType( ) setContentType( )
⑥ page	java.lang.Object	this variable is assigned to instance of JSP implementation class It is rarely used	
⑦ pageContext	javax.Servlet.jsp. pageContext	It provides access to several JSP page attributes	getPage( ), getRequest( ) getResponse( ) getSession( )
⑧ Session	javax.Servlet.http Http Session	this Variable is used to access the Current client's session	getId( ) getCreationTime( )

UNIT-5

JAVASCRIPT

Introduction to Javascript :

Javascript is a light weight, cross-platform and interpreted Scripting language. It is well-known for development of webpages many non-browser, environments also use it. Java Script can be used for client-side developments as well as server-side developments. Javascript contains a standard library of objects like Array, Date and Math, and also contains elements like operators control structures and statements.

If Javascript is client-side :-

It supplies objects to control a browser and its Document object Model (DOM). Like if client-side extensions allow an application to place elements on. An HTML form and respond to user events such as mouse clicks, form input and page navigation. Useful libraries for the client-side are Angular Js, React JS, VueJs and so many others.

1. Server side : It supplies objects relevant to running Javascript on a server, Like if the server-side extensions allow an application to communicate with a database and produce continuity of information from one invocation to another of the applications, perform file manipulations on a server, The useful framework which is the most famous these days in node.Js

Javascript can be added to your HTML file in 2 ways:

Internal JS :- We can add Javascript directly to our HTML file by writing the code inside the <Script /> tag. The

<script> tag can either be placed inside the <head> or the <body> tag according to the requirement.

External JS :- We can write Javascript code in other file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

Syntax:

```
<script>
    // Javascript code
</script>
```

Example:

HTML

```
<!DOCTYPE html>
<html lang = "en">
  <head>
  <title> Javascript Example </title>
 </head>
 <body>
 <script>
Console log ("welcome to CSE");
 </script>
 </body>
 </html>
```

Output :-

Welcome to CSE

History of Javascript :-

It was created in 1995 by Brendan Esch while he was an engineer at Netscape. It was originally going to be named liescript but was renamed. Unlike most programming languages, the Javascript language has no concept of input or output. It is designed to run as a scripting language in a host environment to provide mechanisms for communicating with the

Outside world. The most common host environment is the browser.

Features of Javascript :- According to a recent survey conducted by stack overflow. Javascript is the most popular language on earth.

With advances in browser technology and Javascript having moved into the server with the Node.js and other frameworks, Javascript is capable of so much more. Here are a few things but we can do with Javascript:

1) Earlier websites were mostly static, after Javascript was created dynamic websites were made.
2) Functions in JS are objects. They may have properties and methods just like another object. They can be passed as arguments, in other functions.
4) Can handle date and time.
5) Perform form validation although the forms are created using HTML.
6) No compiler is needed.

Applications of Javascript:

1. Web Development

We can add interactivity and behavior to static sites. By using Angular JS that can be achieved so easily.

2. Web Applications :-

By using this Javascript, we can create robust web applications. for Example, when we explore a map in Google

maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of Javascript. It uses API (Application programming interface) that provides extra power to code.

3) Server Applications :-

With the help of Node.js, Javascript made its way from client to server and node.js is the most powerful on server side.

4) Games :-

Not only in websites, but javascript also helps in creating games. The combination of javascript and HTML5 makes javascript popular in game development as well. It provides the base JS library which provides solutions for working with rich graphics.

5. Smart Watches

Javascript is being used in all possible devices and Applications. It provides a library pebble JS which is used in smart watch applications. This framework works for applications that require the internet for its functioning

6. Art :-

Artists and designers can create whatever they want using Java Script.

7. Machine Learning :-

This Javascript library can be used in web development by using machine learning.

Limitations of Java Script :-

1) Performance :

Java Script does not provide the same level of performance

as offered by many traditional languages. So it is not suitable to handle complex tasks.

8) Complexity :

To master a scripting language, programmers must have a good knowledge of all the programming concepts, core language objects, client and server side objects. Otherwise it would be difficult for them to write advanced scripts using Javascript.

3) Weak error handling and type checking facilities :-

It is weakly typed language as there is no need to specify the datatype of variable. So wrong type checking, is not performed by compiler.

Java Script Variables :-

A Javascript variable is simply a name of storage location. There are 2 types of variables in Javascript.

1) Local variable
2) Global variable

There are some rules while declaring a Javascript variable

1) Name must start with a letter (a to z or A to z), underscore (_) or dollar sign ($)
2) After first letter, we can use digits (0 to 9), for example value 1.
3) Javascript variables are case sensitive, for example x and X are different variables.

Var X=10;
Var-value = "Sonoo"; } Correct Javascript variable

Var 123= 30;
Var *aa = 320; } In correct Javascript variable.

Example of Javascript Variable :-

```
<html>
<body>
<script>
var x=10;
var y=20;
var z=x+y;
document.write(z);
</script>
</body>
</html>
```

Output = 30

Javascript Local Variable :-

A Javascript local variable is declared inside block or function.
It is accessible within the function or block only for ex :-

1.
```
<script>
function abc(){
var x=10; // Local variable
}
</script>
```

2.
```
<script>
if(10<13){
var y=20; // Javascript local variable
}
</script>
```

Javascript Global Variable

A Javascript global variable is accessible from any function. A variable i.e. declared outside the function or declared with window object is known as global variable. for example:

```
<html>
<body>
<script>
var data=200; //global variable
function a(){
document.write In(data);
}function b(){
document.write In(data);
}
a(); // calling Javascript
b();
</script>
</body>
</html>
```

output: 200 200

3. Javascript Data Types:

Javascript provides different datatypes to hold different types of values. There are two types of data types in Javascript.

1) Primitive Type 2) Non-primitive (reference) data type.

Javascript is a dynamic type language, means you dont need to specify type of variable because it is dynamically used by Javascript engine, you need to use variable here to specify the data type. It can hold any type of values such as numbers,

strings etc.. for example:

1. var a=40; // holding number
2. var b="Rohitha"; //holding string

Javascript primitive data types:-

There are 5 types of primitive data types in Javascript. They are as follows:

Data Type	Description
string	Represents sequence of characters eg: "hello"
Number	Represents numeric values eg: 100
Boolean	Represents boolean value either false or true
undefined	Represents undefined value
Null	Represents null i.e; no value at all.

Javascript non-primitive datatypes:-

The non-primitive data types are as follows:

Datatypes	Description
Object	Represents instance through which we can access members.
Array	Represents group of similar values
Reg Exp	Represents regular expression

Javascript conditional statement:

1) if statement
2) if else statement
3) if else if statement
4) switch statement

Javascript if Statement :-

It evaluates the Content only if expression is true

Syntax:- if (expression) {
    // Content to be evaluated
}

Ex:-
```
<script>
Var a=20;
if(a>10) {
document.write("value of a is greater than 10");
}
</script>
```

Javascript if---else Statement :-

It evaluates the Content whether condition is true or false.

Syntax:-
```
if (expression) {
// Content to be evaluated if condition is true
}
else{
// Content to be evaluated if condition is false
}
```

Ex:-
```
<script>
Var a=20;
if (a%2==0) {
document.write("a is even number");
}
else{
document.write("a is odd number");
}
</script>
```

Javascript if---else if statement :-

It evaluates the content only if expression is true from several expressions.

Syntax:

```
if (expression 1) {
// Content to be evaluated if expression 1 is true
} else if (expression 2) {
// Content to be evaluated if expression 2 is true
} else if (expression 3) {
// Content to be evaluated if expression 3 is true
} else {
// Content to be evaluated if no expression is true
}
```

Ex:-

```
<script>
var a=20;
if (a==10){
document.write("a is equal to 10");
  else if (a==15){
document.write("a is equal to 15");
}
else if (a==20){
document.write("a is equal to 20");
}
  else{
document.write("a is not equal to 10,15 or 20");
}
  </script>
```

Javascript Switch :-

The Javascript switch statement is used to execute one code from multiple expressions.

Syntax :-

```
Switch (expression) {
```

```
Case value 1:
Code to be executed
  break;
  Case value 2:
  Code to be executed
   break;
   -----
   default:
  Code to be executed if above values are not matched;
}
```

Example of switch statement in Javascript:

```
<Script>
Var grade = 'B';
 Var result;
  Switch (grade){
     Case 'A':
      result = "A Grade";
       break;
      Case 'B':
      result = "B Grade";
       break;
       Case 'C';
      result = "C Grade";
       break;
       default:
     result = "No Grade";
    } document . write (result);
      </Script>
```

Javascript Loops :

The Javascript loops are used to iterate the piece of code using for, while, do while or for-in-loops. It makes the code compact. It is mostly used in array.

There are 4 types of loops in Javascript

1. for loop
2. while loop
3. do-while loop
4. for-in loop

1. Javascript for loop :-

The Javascript for loop iterates the elements for the fixed number of times. It should be used if number of iteration is known, The syntax of for loop is given below.

```
for (initialization ; Condition; Increment)
{
  Code to be executed
}
```

Ex :-
```
<script>
for (i=1; i<=5; i++)
{
document write (i+ "</br>")
}
</script>
```

2. Javascript while loop :-

The Javascript while loop iterates the elements for the infinite number of times. It should be used if number of iteration is not known. The Syntax of while loop is given below.

```
while ( condition)
{
  Code to be executed
}
```

Ex: <script>
var i=11;
while (i<=15)
{
document.write (i+"
");
i++;
}
</script>

3) Javascript do while loop :-

The Javascript do while loop iterates the elements for the infinite number of times like while loop. But, code is executed atleast once whether condition is true or false. The syntax of do while loop is given below.

```
do
{
Code to be executed
}while (condition);
```

Let's see the simple example of do while loop in javascript

```
<script>
var i=21;
do{
document.write (i+"<br/>");
i++;
}while (i<=25);
</script>
```

4) Javascript for in loop :-

The Javascript for in loop is used to iterate the properties of an object.

Functions in Javascript :-

Javascript functions are used to perform operations, we can call Javascript function many times to reuse the code.

Advantage of Javascript function :-

There are mainly 2 advantages of Javascript functions.

1. Code reusability :- We can call a function several times so it saves coding.
2. Less Coding : It makes our program compact. we don't need to write many lines of code each time to perform a common task.

Javascript function Syntax :-

The Syntax of declaring function is given below

```
function functionName([arg1, arg2, .... argN])
{
  // Code to be executed
}
```

Javascript functions can have 0 or more arguments.

Ex :-
```
<html>
<body>
<h2> JavaScript functions </h2>
<p id="demo"> </p>
<script>
var x = myFunction(4,3);
document.getElementById("demo").innerHTML = x;
function myfunction(a,b)
{
return a*b;
}
</script>
```

→ when java script reaches return Statement it stops executing

op :

Javascript function

12

→ A Javascript function is defined by with the function keyword, followed by a name, followed by parantheses

→ Function names can contain letters, digits, underscores and dollar signs.

→ The parantheses may include parameter names separated by commas

Ex:- (parameter 1, parameter 2, ---)

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

Function invocation :-

The code inside the function will execute when "Something" calls the function.

a) when an event occurs (when a user click a button)

b) when it is invoked (called) from javascript code.

c) Automatically (Self invoked)

Javascript Events :-

HTML events are "things" that happen to HTML elements when Javascript is used in HTML pages.

Javascript can "react" on these events.

HTML Events :-

An HTML Event can be something the browser does or Something a user does.

Here are some examples of HTML events

- An HTML webpage has finished loading
- An HTML input field was changed.
- An HTML button was clicked.

often, when events happen, you may want to do something. Javascript lets you execute code when events are detected. HTML allows event handler attributes, with Javascript code, to be added to HTML elements.

Ex:-

```
<html>
<head>
<script type = "text/javascript
<!...
function Say Hello() {
alert ("Hello world");
}
//-->
</script>
</head>
<body>
<p> click the following button </p>
<form>
<input type = "button" on click = "Say Hello()"
value = "Say Hello"/>
</form>
</body>
</html>
```

output:

click the following button

[Say Hello] → Hello world.

Some of the events & their event handlers are:

Mouse Events

Event performed	Event handler	Description
Click	onclick	when mouse click on an element.
mouse over	on mouse over	when the cursor of the mouse comes over the element
mouse out	on mouse out	when the cursor of the mouse leaves an element
mouse up	on mouse up	when the mouse button is released over the elements
mouse move	on mouse move	when the mouse movement takes place

Keyboard events :-

Event performed	Event Handler	Description
Key down and key up	on keydown & on key up	when the user press then release the key.

Form events :

Event performed	Event Handler	Description
focus	on focus	when the user focuses on an element
Submit	on submit	when the user submit the form
blur	on blur	when the focus is away from a form element
change	on change	when the user modifies or changes the value of a form element

Window / Document events :

Event performed	Event handler	Description
load	onload	when the browser finishes the loading of the page
unload	on unload	when the visitor leaves the current web page, the browser unloads it
resize	on resize	when the visitor resizes the window of the browser

Javascript form validation :-

It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate user.

Javascript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer Javascript form validation.

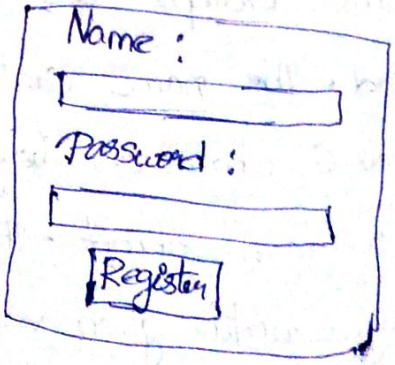
Through Javascript, we can validate name, password, email, date, mobile numbers & more fields.

Javascript form validation Example :-

In this example, we are going to validate the name and password. The name can't be empty and password can't be less than 6 characters long. Here we are validating the form on form submit. The user will not be forwarded to the next page until given values are correct.

```
<html>
<body>
```

```
<script>
function validate form()
{
  var name = document.myform.name.value;
  var password = document.myform.password.value;
  if (name == null || name == "")
  {
    alert("Name can't be blank");
    return false;
  }
  else
    if (password.length < 6)
    {
      alert("password must be atleast 6 characters long");
      return false;
    }
  }
</script>
<form name="myform" method="post"
action="http://----" onSubmit="return validate form()>
Name: <input type="text" name="name"> <br/>
Password: <input type="password" name="password"> <br/>
<input type="Submit" value="register">
</form>
</body>
</html>
```



Javascript provides a way to validate form's data on the client's computer before sending it to the web server. Form validation generally perform two functions

Basic validation :-

First of all the form must be checked to make sure are the mandatory fields are filled in. It would require just a loop through each field in form and check for data.

Data format validation :-

Secondly, the data that is entered must be checked for correct form and value. Your code must include appropriate logic to test correctness of data.

Document object Model (DOM)

The document object represents the whole html document. When html document is loaded in the browser, it becomes a document object. It is the root element that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our webpage.

As mentioned earlier, it is the object of window. So

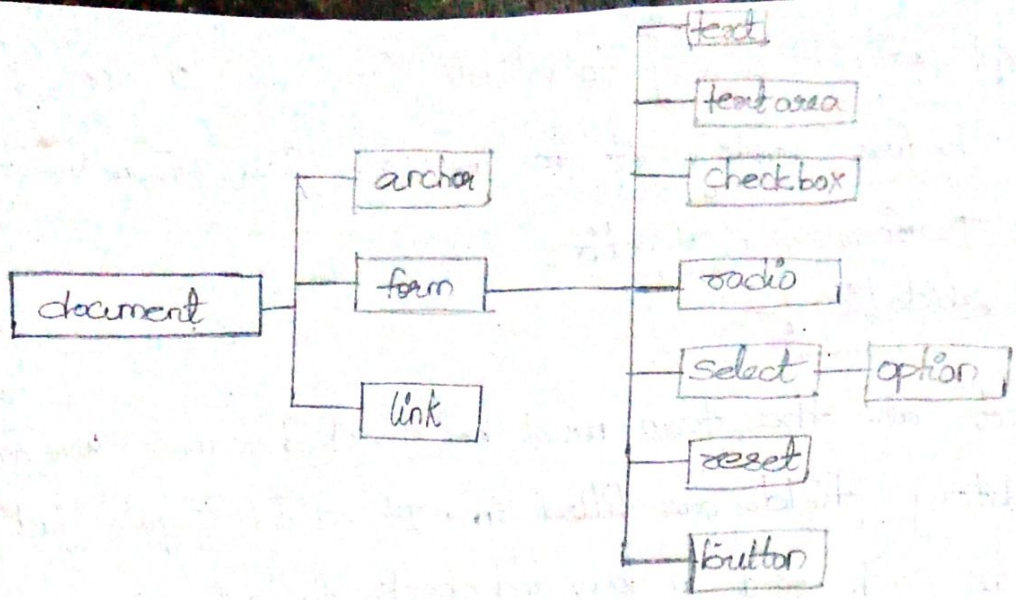
1. window.document

Is same as

1.document

Properties of document object :-

Let's see the properties of document object that can be accessed and modified by the document object.



Methods of document object

We can access and change the contents of document by its methods, The important methods of document object are as follows:

Method	Description
write ("string")	writes the given string on the document
writeln ("string")	writes the given string on the document with newline character at the end
getElementById()	returns the element having the given id value
getElementsByName()	returns all the elements having the given name value
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

Accessing field value by document object

In this example, we are going to get the value of input text by user. Here, we are using document.form1.name.value to get the name field.

Here the document is the root element that represents the

html document.

form 1 is the name of the form

name 1 is the attribute name of the input text

value is the property, that returns the value of the input text.

Let's see the simple example of document object that prints name with welcome message.

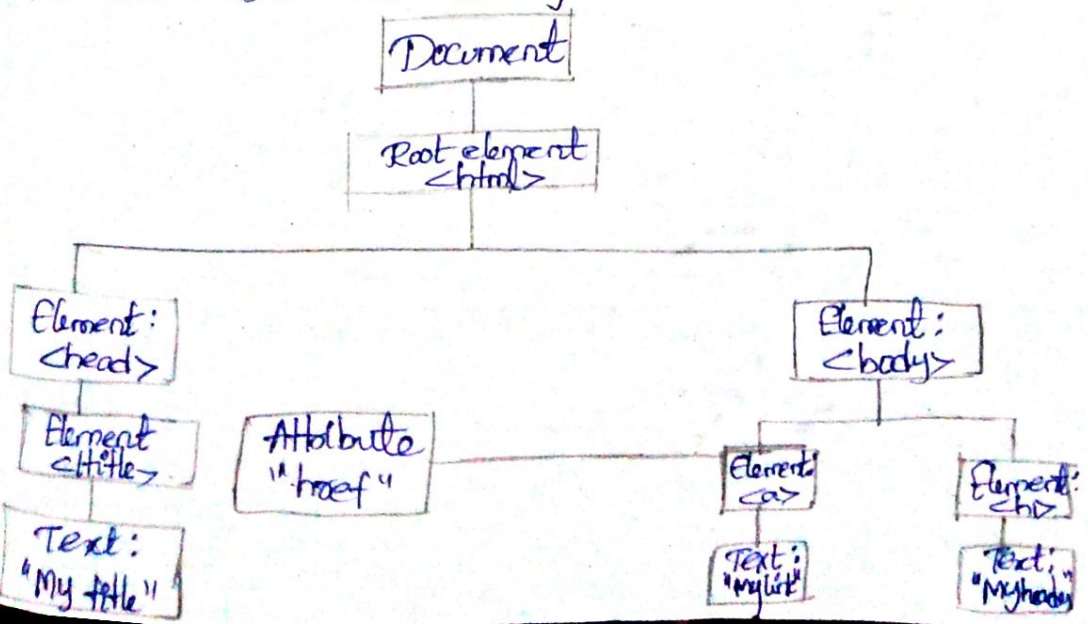
```
<script type = "text/javascript">
  function printvalue() {
  var name = document.form1.name.value;
  alert ("welcome:" + name);
  }
  </script>
  <form name = "form1">
Enter Name: <input type = "text" name = "name"/>
<input type = "button" onclick = "printvalue()" value = "print name">
  </form>
```

The HTML DOM (Document object Model)!-

When a webpage is loaded, the browser creates a Document object Model of the page

The HTML DOM model is constructed as a tree of objects:

The HTML DOM Tree of objects.



With the object Model, Javascript gets all the power it needs to create dynamic HTML:

- Javascript can change all the HTML elements in the page
- Javascript can change all the HTML attributes in the page.
- Javascript can change all the CSS styles in the page
- Javascript can remove existing HTML elements and attributes
- Javascript can add new HTML elements and attributes
- Javascript can react to all existing HTML events in the page.
- Javascript can create new HTML events in the page