

# **PHP**

**By**

## **Swamynaidu**

## **Naresh Technology**

**SRI RAGHAVENDRA XEROX**

*Software Languages Material Available*

Beside Bangalore Ayyangar Bakery, Opp. C DAC, Ameerpet, Hyderabad.

**Cell: 9951596199**



Client side : HTML, CSS

Server side : PHP, JSP, ASP

PHP is used to develop Web Applications.

PHP (Personal Home Page) (Hypertext Preprocessor)

PHP: PHP is a server side scripting language implemented by Rasmus Lerdorf in 1995 using 'C' and 'perl' Technologies. By using PHP we can develop Dynamic Web Application.

Static webpage: Only information displays. Ex: Newspaper.

Dynamic webpage: We can Login Ex: Gmail, Facebook.

PHP stands for Personal Home page and also it's contains it's alias name HtP i.e. Hypertext Preprocessor.

Rasmus Lerdorf implemented PHP 1.0 to find out how many companies accessing his Resume through Online. He used 'C' and 'perl' Technologies for this implementation.

Apache: Open Source free Server.

IIS: .NET used that server.

Note: MySQL is a Open source and only PHP can directly contact with MySQL, So there is a combination of MySQL & PHP.

PHP Application : Facebook

ASP Application : ORKUT

Note: PHP Maintained by Zend Organization.

Date: 14 AUG 2012

### \* Features of PHP :

- 1) Cross platform: It can run under any types of operating system.
- 2) Cross Server: It can run under diff. types of Webservers, like IIS, Apache, Tomcat etc.
- 3) Cross Database: It supports any type of Database server.
- 4) PHP 5.0 MySQL: PHP 5.0 is providing MySQL Library to interact with MySQL DB.
- 5) PHP supports no. of object Oriented programming concepts like Inheritance, Access Specifiers etc.
- 6) PHP is Open source.
- 7) PHP supports diff. types of content management sys like JOOMLA, DRUPAL, WORDPRESS etc. All CMS (Content Management System) are open source sys.

- 8) PHP providing number of security fns to apply security to applicat.
- i-Way Encrypt, 2-Way Encrypt, Authentication etc.
- 9) PHP supporting diff. types of Editors to develop prog, we can also use lightweight Editors (Notepad, Editplus etc)
- 10) PHP execute is very fast bcoz, of Zend Engine.
- 11) Zend organizat is maintaining the PHP script. They introduced number of tools to work with PHP & also they are providing complete support to PHP programmers thr. online.
- 12) PHP is very easy to understand & Develop the Applicat.

Date: 16 AUG 2012

### \* Versions of PHP :

- 1) PHP 1.0 (1995):
  - i) Rasmus Lerdorf implemented PHP 1.0 using 'c' & 'perl' S/W.
  - ii) It is not a server side scripting lang.
  - iii) Implementation from 'c' & 'perl' softwares.
- 2) PHP 2.0 (1997):
  - i) Rasmus Lerdorf implemented PHP 2.0 as partially server-side scripting language.
  - ii) PHP 2.0 supports different types of Database servers.
  - iii) We can implement some types of modules using 2.0 those are registration, login, etc.
- 3) PHP 3.0 (1998):
  - i) PHP 3.0 is fully server side scripting language.
  - ii) Zeev Suraski & Andi Gutmans rewritten the functionalities of PHP from scratch. on released it 3.0 as fully server side scripting lang.
  - iii) PHP 3.0 supports different platforms (Operating syst.)
  - iv) Personal Homepage is Renamed as Hypertext Preprocessor with this version.
- 4) PHP 4.0 (2000):
  - i) PHP 4.0 is cross server, it supports IIS, APACHE, TOMCAT etc.
  - ii) Zend Engine 1.0 is introduced in PHP 4.0, Zend Engine is Runtime Environment of PHP Applicat.
  - iii) Smarty templating syst. is introduced with this Version, we can simply develop the Applicat with predefined templates.
- 5) PHP 5.0 (2004-05):
  - i) The major upgrade of PHP 5.0 is Object Oriented programming support.
  - ii) Zend Engine 2.0 is introduced with PHP 5.0

iii) XML and Webservices related classes are introduced with PHP 5.0 3

### 6) PHP 6.0 :

- i) The main focus of PHP 6.0 is Unicode support means programmers can able to name their classes and methods with their native languages.
- ii) Some Advanced Object Oriented programming concepts introduced with PHP 6.0 Those are Namespaces, and Late Static Binding.

—x—x—

\* PHP is partially case sensitive lang. In <sup>fn</sup> point of view it is case <sup>sensitive</sup> and variables point of view, it is case sensitive.

\* Every statement in PHP, we should end with semicolon.

\* PHP is loosely typed language. That's why no need to provide datatypes at the time of variable declaration.

\* Variable Names in PHP should starts with '\$' (Dollar) symbol.

\* Every PHP file extension should be .PHP.

\* PHP script we should include within the Script Declaration style tags.

<?php ..... ?>

Date: 17 AUG 2013

### Webserver:

Webserver is a s/w used to run the Web Applicat<sup>n</sup>. It handles the request from client and sends the response to the client.

Ex: IIS, Apache, Tomcat etc.

### Web Browser:

It is a s/w used to open web applicat<sup>n</sup>s from Webserver.

Ex: Internet Explorer, Mozilla Firefox, etc.

### Request:

A Request is a Trip of Webpage from Browser to Server.

Ex: www.gmail.com.

### Response:

Server sends the output for client request in the form of Response.

### Client Side Script:

The script which executes client system with the help of

web browser can be called as Client-side Script.

Ex: ~~HTML~~ HTML, Javascript etc.

### Server Side Script:

The script executes <sup>in</sup> the webserver with the help of server engines. User can't see the original script of server side scripting languages.

Ex: PHP, ASP, JSP etc.

## \* Tools To Work With PHP:

### 1) WAMP:

(Stands for Windows APACHE MySQL and PHP), These tools supports Windows OS.

### 2) LAMP:

(LINUX APACHE MySQL and PHP). It can run under LINUX OS.

### \* 3) XAMPP:

(XPERT For any operating syst, APACHE, MySQL PHP & PERL)

All these tools are open source tools available with diff. extensions.

—x —x —x —  
How to Run a prog. in other Drive?

### \* Installat° of PHP: (XAMPP 1.7.2) → C:drive → XAMPP → apache → conf → httpd → edit F → ~~httpd.conf~~ → httpd.conf

After Installat° of XAMPP, we can find out XAMPP folder in the ~~apache~~  
Destinat° locat° contains number of subfolders. Press 9 Times → DocumentRoot  
Root "F:/PHP"

### htdocs:

It is the root directory of APACHE server, All php programs we should save at this locat°.

→ Press 9 Times  
Directory

### tmp:

It is temporary mem. locat° of server, All uploaded files and sessions will store at this locat°.

### php:

This folder contains php configurat° settings. php.ini is configurat° settings file of php.

### apache:

Apache contains apache configurat° settings httpd.conf is apache configurat° settings filename.

### mysql:

MySQL contains mysql configurat° settings my.ini is configurat° settings file.

## \* Steps to Create and Execute PHP program:

1) Open Editor and <sup>implement</sup> write PHP Script using Script Declarat° style tags.

2) Save this file in the root directory with extension: .php!

3) Open XAMPP control panel & start Apache server.

4) Open Browser and send the request to the server to get the O/P of PHP file.

\* Declaration style Tag of php: Date: 21-AUG-2013 (5)  
php is providing different types of declaration style Tags.

#### 1) Universal style Tag:

This Tag supports all functionality.

<?php.....?>

#### 2) Short open Tag:

This Tag supports very few functionality.

<?.....?>

#### 3) Asp style Tag:

We can add php script within asp declarat<sup>n</sup> syntax.

<%.....%>

#### 4) Script style Tag:

It is similar to javascript declaration syntax.

<script language = "php">

.....

.....

</script>

We have 2 Configurat<sup>n</sup> setting in php.ini related to declarat<sup>n</sup> style Tag.

The default value is "OFF" for short open Tags & asp tags. By changing the value we can execute this Tag. We need to restart the server to see the changes.

### \* Output functions in PHP:

#### 1) print:

This fn displays output on Browser & Returns a Boolean value "True", if output is printed successfully, otherwise False. We cannot print multiple statements with single print statement.

[Note: In PHP, the value of True is "1" and False means "No Value"]

Ex: <?php

```
$a = print "ABC";
print "<br/>";
print $a;
?>
```

#### 2) echo:

It same as print function. But we can print multiple statements with a single echo function.

Then print.

This fn doesn't return any value, that's why faster

<?php

```
echo "ABC", "DEF";
?>
```

## 2) Var\_dump:

This fn displays variable value along with its datatype.

Ex: <?php \$sno=111; var\_dump(\$sno); ?>

## 3) printf:

By using this fn, we can display output with the help of format specifier

Ex: <?php \$a=10; \$user = "scott"; printf("%s has %d cars", \$a, \$user)"

## 4) print\_r:

By using this fn, we can display all elements of array & property of object.

## \* Types of Errors in PHP

There are 4 types of errors.

### 1) Notice:

It is nothing but a small info to user. If we're trying to access undefined variables, the opp is "Notice". Ex: <?php  
\$x=100;  
echo \$a;

Notice doesn't stop script execution.

By default we cannot see the notice msg on browser. Bcoz, the config setting i.e. error-reporting value is "E\_ALL" & E\_NOTICE means display all error msgs except Notice. By removing E\_NOTICE, the notices are displayed by the browser. We can also display notice within the program by using ErrorReporting(E\_ALL);

<?php error\_reporting(E\_ALL);

```
$x=100;  
echo $x;  
echo $y;  
echo "Next";
```

?>

2) Warning: It is same as Notice does not

stop script execution. It occurs if we are trying to call undefined constants.

In PHP, we can declare constant by using define. Ex: <?php

```
define ("sno", 10);  
echo constant ("sno");  
echo constant ("a");  
?>
```

### 3) Fatal Error:

It stops the execution of webpage, from line where the error occurred. If we try to call undefined function Fatal Error occurs.

Ex: <?php

```
function f1()  
{  
    echo "hi";  
}  
f1()  
f2()  
?>
```

### 4) Parse Error:

It stops the execution of complete script if there is syntax mismatch.

Ex: <?php  
echo "line1";  
echo "12"  
echo "13";  
?>

#### Note:

If the last line doesn't contain a semicolon, No Error occurs.

Data types available in PHP:

datatype is used to specify the type of data

What variable can hold.

Basically, we have 3 types of datatypes in PHP.

- 1) scalar
- 2) compound
- 3) special

scalar datatypes again divided into diff. types.

1) Boolean:

These datatypes represents either True or False. In PHP, value of True is 1. and the value of False is Nothing.

Ex: <?php

```
$x=true;
echo $x;
?>
```

O/P: 1

2) is\_bool(variable):

By using this f<sup>n</sup>, we can check whether the variable is boolean variable or not.

Ex: <?php

```
$x=false;
echo is_bool($x); } // True: O/p 1
$y="false";
echo is_bool($y); } // False: O/p 0
?>
```

O/P: 1

3) (bool) variable, (boolean) variable:

To convert the datatype of variable

into boolean datatype.

<?php

```
$y="false";
$y=(bool)$y;
echo is_bool($y);
?>
```

O/P: 1

Integer:

This datatype stores numeric values.

Ex: <? php

```
$x = 123;
echo $x;
?>
```

O/P : 123

Is\_int, is\_integer:

By using this f<sup>n</sup>, we can check the \$x variable is Integer or Not.

<? php

```
$x = 123;
echo is_int($x);
?>
```

O/P : 1

Int (int) Variable, (integer) Variable, int val Variable:

By using this conversion f<sup>n</sup>, we can convert a variable datatype into integer datatype.

Ex: <? php

```
$x = "123";
$x = intval($x);
echo is_int($x);
?>
```

O/P : 1

Float:

This datatype represents decimal values.

is\_float:

By using this f<sup>n</sup>, we can check the \$x value is float or Not.

Float (variable):

By using this f<sup>n</sup>, we can convert a variable datatype into a floating pt. number.

<? php

```
$x = 123.45;
echo is_float($x);
?>
```

O/P : 1

precision:

It is a configurat<sup>n</sup> setting in php.ini used to specify total no. of digits displayed in floating pt. no.

String:

String is collect<sup>n</sup> of characters in php. In php, we can declare string variable in 3 ways.

- 1) Using single Quotat<sup>n</sup>s.
- 2) Using double Quotat<sup>n</sup>s.
- 3) Using Heredoc Syntax.

Ex: <?php

```
$user = "scott";
$str = "Welcome to $user"; //br
$str = "Welcome to $user";
echo $str;
echo $str;
?>
```

SAVE AS: String.php

O/P: Welcome to scott  
Welcome to scott

Note: If we place a variable within double Quotat<sup>n</sup>, It returns the value of that variable.

Heredoc:

By using the syntax we can display the HTML elements the .php script. It avoids the problems what were getting with open quotat<sup>n</sup>s of string. This syntax doesn't require any open quotat<sup>n</sup>s that's why for internal values we can use both single & double quotat<sup>n</sup>s.

The Syntax of Heredoc is : =>

~~opening~~      <<< nameofstring  
                         /// content  
                         nameofstring

Ex: <?php

```
echo <<<mystring =>
<input type="button" value='Go'>
mystring;
?>
```

SAVE AS: Heredoc.php

O/P:

If we not write these 2 stat. It gives Error  
Parse error: syntax error,  
unexpected '<' in  
prog

## \* compound Datatypes:

1) Array: collect<sup>n</sup> of Elements.

2) Object: Instance of a class.

## \* Special Datatypes:

### Resource datatype:

It refers the external Resources like Database Conn<sup>n</sup>, FTP Conn<sup>n</sup>, file ptrs etc.

Ex: <?php

```
$con=mysql_connect("local host","root","");
echo $con;br/>;
echo
get_resource_type($con);
?>
```

SAVE AS:

ResourceDatatype.php

O/P:

Resource id #2  
mysql link

### Null Datatype:

In php, null is not a value we can consider a variable as Null variable based on 3 cond<sup>n</sup>.

- i) IF the variable is not set with any value,
- ii) IF the variable is set with Null value.
- iii) IF the value of variable is unset.

### is\_null:

By using this f<sup>n</sup>, we can check whether the variable is Null or Not.

```
<?php
$x=100;
unset($x);
echo is_null($x);
?>
```

Note: We can unset the variable value by using unset f<sup>n</sup>.

SAVE AS: ISNULL.php

O/P: 1

### Array:

Array is collect<sup>n</sup> of heterogeneous (dissimilar) datatypes.  
php is loosely typed lang. That's why we can store any type of values in Arrays.

Normal variable can store single value, Array can store multiple values.

Array contains no. of elements, Each element is a combin<sup>n</sup> of element key & element value.

The key of 1<sup>st</sup> element is '0' and last element is 'total elements - 1'.

## Syntax of Array Declaration:

variableName = array (ele1, ele2, ele3, ...)

Ex: \$arr = array (10, 20, 30)  
 ↓      ↓      ↓  
 0<sup>th</sup> 1<sup>st</sup> 2<sup>nd</sup>

1) <?php

\$arr = array (10, 20, 30);

?>

↓  
O/P: Blank  
 For printing O/P use print\_r(\$arr)

If we want to see O/P on the screen  
 then prog. will be <?php

\$arr = array (10, 20, 30);  
 print\_r(\$arr);

?> SAVE AS: Array1.php  
O/P:

Array [0] ⇒ 10 [1] ⇒ 20 [2] ⇒ 30

2) <?php

\$arr = array (10, "scott", 30);

?>

↓  
O/P: Blank  
 For printing O/P use print\_r(\$arr)

3) <?php

\$arr = array (10, "scott", 30);

// print\_r(\$arr); or

echo \$arr[1];

?>

SAVE AS: Array2.php

O/P: scott.

We can create Array with Explicit Keys

23 AUGUST 2012

<?php

\$arr = array (0 ⇒ 10, 1 ⇒ "scott", 2 ⇒ 30);

print\_r(\$arr);

?>

SAVE AS: Array3.php

O/P: Array ( [0] ⇒ 10 [1] ⇒ scott [2] ⇒ 30 )

Note: If we don't provide an Explicit key to an Array Element, the new key is depending upon previous keys of that Element. First, It ~~considers~~ all previous keys finding max<sup>th</sup> key, then add it to the max<sup>th</sup> key & provide the resultant value as key to the next element.

<?php

\$arr = array (10, 100 ⇒ "scott", 30);

print\_r(\$arr);

?>

O/P: Array  
 ( [0] ⇒ 10  
 [100] ⇒ scott  
 [10] ⇒ 30  
 )

Ex: <?php  
\$arr = array(10, 20, 0 => 30);  
print\_r(\$arr);  
?>

Save As: Array 4.php  
O/P: Array ([0] => 30 [1] => 20)

Ex: <?php  
\$arr = array();  
\$arr[0] = 10;  
\$arr[1] = 20;  
print\_r(\$arr);  
?>

Save As: Array 5.php  
O/P: Array ([0] => 10 [1] => 20)

Ex: <?php  
\$arr = array();  
\$arr['long'] = 'PHP';  
\$arr['Manager'] = 'Scott';  
print\_r(\$arr);  
?>

SAVE AS: AssociativeArray.php  
O/P: Array ([Long] => PHP [Manager] => Scott)

We can provide string as Array Element <sup>key</sup>. This concept is called as Associative Array.

### COUNT()

This F<sup>n</sup> Returns total no. of Array Elements.

<?php  
\$arr = array(10, 20, 30);  
echo count(\$arr);  
?>

Save As: Count.php  
O/P: 3

### SORT()

This F<sup>n</sup> arranges the Array Elements in Ascending order with new keys.

Ex: <?php  
\$arr = array(10, 5, 20, 30);  
sort(\$arr);  
print\_r(\$arr);  
?>

Save As: ArraySort.php  
O/P: Array  
[0] = 5  
[1] = 10  
[2] = 20  
[3] = 30  
)

Ex: <?php  
\$arr = array(10, 'a' => 5, 20, 30);  
sort(\$arr);  
print\_r(\$arr);  
?>

O/P: same as above

**rsort():**

Arranges Array Elements Descending order. With the new keys.

Array: SAVE AS: SortDesc.php: Array [0] ⇒ 30

[1] ⇒ 20

[2] ⇒ 10

[3] ⇒ 5

↑  
prog:

<?php  
\$arr = array(10, 5, 20, 30);  
rsort(\$arr);  
print\_r(\$arr);  
?>

**asort():**

Ascending order with original keys.

O/P : 0 ⇒ 5  
1 ⇒ 10  
2 ⇒ 20  
3 ⇒ 30

**ksort():**

Ascending order based on Element keys.

<?php  
\$arr = array(10, 100 ⇒ 5, 20, 5 ⇒ 30),  
ksort(\$arr);  
print\_r(\$arr);  
?>

O/P : Array

**ksort():**

Descending order based on Array keys.

**array\_sum():**

To get the sum of Array Elements.

**Ex:** <?php

\$arr = array(10, 20, 30);  
echo array\_sum(\$arr);  
?>

O/P : 60

**Ex:** <?php

\$arr = array(10, 20, 30, 'Scott');  
echo array\_sum(\$arr);  
?>

**array\_product():**

Get the product of elements.

**Ex:** <?php

\$arr = array(10, 20, 30)  
echo array\_product(\$arr);  
?>

array\_push:

By using this f<sup>n</sup>, We can add new element at end pt. of Array & it returns total no. of Array Elements.

Ex: <? php

```
$arr = array(10, 20, 30);
echo array_push($arr, 40);
```

O/P : 4

Ex: <? php

```
$arr = array(10, 20, 30);
echo array_push($arr, 40);
print_r($arr);
?>
```

array\_pop:

Deletes the last element of Array and returns the value of the element.

Ex: <? php

```
$arr = array(10, 20, 30);
echo array_pop($arr);
```

O/P : 30

Ex: <? php

```
$arr = array(10, 20, 30);
echo array_pop($arr);
print_r($arr);
?>
```

array\_shift:

This f<sup>n</sup> removes the 1<sup>st</sup> element of an Array & Returns the value of that Element.

Ex: <? php

```
$arr = array(10, 20, 30);
echo array_shift($arr);
print_r($arr);
?>
```

array\_unshift:

By using this f<sup>n</sup> we can add an element at the beginning of Array & Returns the total no. of Array Elements.

Ex: <? php

```
$arr = array(10, 20, 30);
echo array_unshift($arr, 5);
print_r($arr);
?>
```

### array\_combine:

This fn creates a new array by combining the elements of 2 arrays. New array keys are values of 1<sup>st</sup> Array. New Array values are values of 2<sup>nd</sup> Array.

Ex: <? php

```
$arr1 = array(10, 20, 30),
$arr2 = array('a', 'b', 'c'),
$arr = array_combine($arr1, $arr2);
print_r($arr);
?>
```

O/P : Array  
(  
[10] = a  
[20] = b  
[30] = c  
)

### array\_diff:

This fn compares the values of 2 Array Elements & Returns the differences from 1<sup>st</sup> Array.

Ex: <? php

```
$arr1 = array(10, 20, 30);
$arr2 = array(10, 30, 40);
print_r(array_diff($arr1, $arr2));
?>
```

O/P :  
Array ([1] => 20)

### array\_diff\_assoc:

It compares Key and Values of 2 Arrays & Returns the differences from 1<sup>st</sup> Array.

Ex: <? php

```
$arr1 = array(10, 20, 30);
$arr2 = array(10, 30, 40);
print_r(array_diff_assoc($arr1, $arr2));
?>
```

O/P :

Array ([1] => 20)  
Array ([2] => 30)

### array\_diff\_key:

Compares 2 Array Keys & Returns the differences from 1<sup>st</sup> Array

Ex: <? php

```
$arr1 = array(10, 20, 30);
$arr2 = array(10, 20, 30, 40);
print_r(array_diff_key($arr1, $arr2));
?>
```

O/P :

Array ([1] => 20)

## Explode:

This fn divides a string as Array Elements based on I/p value. (16)

### Ex: <?php

```
$str = "Welcome";
$arr = explode("e", $str);
print_r($arr);
?>
```

O/P:

Array  
([0] => W  
[1] => lcom  
)

## implode:

This fn combines array elements as a string.

### Ex: <?php

```
$arr = array(10, 20, 30);
echo implode("@", $arr);
?>
```

O/P: 10@20@30

O/P: 10@/20@/30

## extract:

This fn divides the associative array elements as variables from Array.

### Ex: <?php

```
$arr = array('a' => 10, 'b' => 20, 'c' => 30);
extract($arr);
echo $a;
echo $b;
?>
```

O/P:

a = 10

b = 20

## list:

24 AUGUST 2012

By using this fn we can ascend array elements into the variables.

### Bx: <?php

```
list($x, $y) = array(10, 20);
echo $x;
echo $y;
?>
```

O/P:

10 20

## in\_array:

By using this fn we can check whether the \$/p value is existed in the specified Array or not. If the value is existed it returns True, otherwise it returns False.

### Ex: <?php

```
$arr = array(10, 20, 30);
echo in_array(20, $arr);
?>
```

O/P: 1

array\_search:

This fn searches the \$ip value is existed in specified Array or not & Returns key of that Element.

Ex: <?php

```
$arr=array(10,20,30);
echo array_search(20,$arr);
?>
```

Op:1

array\_rand:

By using this fn, We can get the keys of Arrays randomly.

Ex: <?php

```
$arr=array('b'=>10,'a'=>20,'c'=>30);
print_r(array_rand($arr,2));
?>
```

array\_slice:

By using this F, we can get some part of an Array.

Ex: <?php

```
$arr=array(10,20,30,40);
print_r(array_slice($arr,0,3));
?>
```

array\_count\_values:

This fn returns an Array with no. of occurrences for each value.

Ex: <?php

```
$arr=array(10,20,10,40);
print_r(array_count_values($arr));
?>
```

array\_key\_exists:

By using this fn we check whether the \$ip key is existed in the specified array or not.

Ex:

```
<?php
$arr=array(10,'a'=>20,40);
echo array_key_exists('a',$arr);
?>
```

array\_keys:

This fn returns all the keys of an array as a new array.

Ex:

```
<?php
$arr=array('x'=>10,'a'=>20,'b'=>40);
print_r(array_keys($arr));
?>
```

array\_merge:

To merge the elements of 2 Arrays.

Ex: <?php

```
$arr=array(10,20,30);
$arr1=array(40,50);
$na=array_merge($arr,$arr1);
print_r($na);
?>
```

array\_reverse:

This fn displays array elements in the Reverse dir.

Ex: <?php

```
$arr = array (10,5,20);
print_r(array_reverse($arr));
?>
```

array\_values:

Returns the all values of array as a new array.

Ex: <?php

```
$arr = array (10,'a'=>5,20);
print_r(array_values($arr));
?>
```

array\_shuffle:

This fn shuffles the elements of an Array.

Ex: <?php

```
$arr = array (10,'a'=>5,20);
shuffle ($arr);
print_r ($arr);
?>
```

★ Some Server Configuration Settings:★ document\_root directory:

By using this configuration setting we can change the document root path of Webserver. By default the root directory of Webserver is htdocs. We can change that locat using this configuration setting. This settings are available in server configuration setting file, i.e. httpd.conf.

★ listen, Servername, localhost:

By using this configuration setting, we can change the portno. of Apache server.

By Default, Server runs with portno.80.

★ directory index:

By using this configuration setting we can specify the startup filenames.

By default, Index, home & default comes under startup files.

★ isset:

By using this fn we can check whether the variable is set with any value or not.

```
<?php
$x=100;
echo isset($x);
$y;
echo isset($y);
?>
```

In php, Dot(.) is a concatenation operator used to join a string with another datatype value.

```
<?php
$x = " Scott";
$y = " John";
$x = $x . $y;
echo $x;
?>
```

http://localhost  
25 - AUGUST - 2018

### include :

By using this concept we can include a php script in another script. If we include an external script in the current php script we can access fns, classes, variables and constants of external script from current php script. To include the script, different types of fn are available those are

- 1) include
- 2) include\\_once
- 3) require
- 4) require\\_once

### 1) include :

This fn includes a file no.of times, if the included file is not available, it returns a warning msg and executes rest of the statements.

Ex: <?php Page1

```
echo "from Page1";
include "Page2.php";
echo $sno;
echo constant("city");
fun1();
?>
```

Page2:

```
<?php
echo "Script from page2";
$sno=100;
define("city", "hyderabad");
function fun1()
{
    echo "This is from function";
}
```

### 2) include\\_once :

It is same as include But it includes external file only one time. First it will check whether the specified file is already included or not. If it is not included, then only includes external file.

Ex: <?php

```
echo "from page1";
include_once "page2.php";
include "page2.php";
?>
```

### 3) require:

It's same as include but it returns fatal errors if external file is not available & stops the execution of script.

(20)

Ex: <php

```
echo "from Page1";
require "Page2.php";
echo "Next";
?>
```

### 4) require\_once:

It is same as require, But only one time it includes the external file

```
?> <?php
echo "from Page1";
require_once "Page2.php";
?>
```

## \* Types of Variables In PHP \*

A Variable is the name of mem. locat<sup>n</sup> used to store values at the time of program Execut<sup>n</sup>.

PHP is loosely typed lang. that's why we can create variables without datatypes.

### Types of Variables:

#### 1) Local Variable:

Variable declarat<sup>n</sup> within the f<sup>n</sup> comes under Local variable declarat<sup>n</sup>. Local Variables we can access within the f<sup>n</sup> where we declared. We cannot access from outside the f<sup>n</sup>.

Ex: <?php

```
error_reporting(E_ALL);
function fun1()
{
    $x=200;
}
function fun2()
{
    echo $x;
}
fun1();
fun2();
?>
```

O/P : Undefined variable x in  
F:\PHP\Local Variable.php  
on line 9.

## 2) Global Variable:

variable declared in global locat' means outside the all functions. Global Variables we can access from any f' within the script. By default, we can not access Global Variables from the f's directly. If u want to access Global Variables, Use "`$GLOBALS`" keyword or Redeclare the variable within the f' as Global Variable.

Ex:

```
<?php
//error_reporting(E_ALL);
$sn0=100;
function fun1()
{
    $sn0=111;
    echo $GLOBALS['sn0'];
    echo $sn0;
}
function fun2()
{
    global $sn0;
    $sn0=200;
    echo $sn0;
}
//fun1();
echo $sn0;
fun2();
echo $sn0;
?>
```

If we remove the  
Comment of fun1();  
Then O/P will be

100 111 100 200 200

O/P: 100 200 200

## Variables\_Variable:

If we assign Variable name as the value of another Variable, comes under variables\_variable f'.

Ex: <?php

```
$x="scott";
$y='x';
echo $$y;
?> $y=$x
∴ $x=scott
```

O/P: scott

## Static Variables:

static variables can maintain the previous values we can assign the values only one time into static variable.

Ex: <?php

```
function fun1()
{
    static $x=100; // * IF we don't write static here, Then the
    $x++;
    echo $x;
}
fun1(); //101
fun1(); //102
fun1(); //103
?
```

O/p: 101 102 103  
So, we use static keyword. \*

## Reference Variables:

It refers the value of another variable.

Actual Variable and Reference Variable refers the value of same address locat<sup>n</sup>.

Ex: <?php

```
$x=100;
$y=&$x; // O/p: 100
```

```
$y=200;
```

```
echo $x;
```

```
?
```



## Super Global Variables:

php is providing no. of Super Global Variables we can access this Super Global Variables from any locat<sup>n</sup> (from any webpage) within the project.

Different types of Super Global Variables are available. Those are  
\$\_GET, \$\_POST, \$\_REQUEST, \$\_SERVER, \$\_SESSION, \$\_COOKIE, \$\_FILES, \$\_ENV.

All Super Global Variables are array datatypes

## FORM

In HTML, Form is a container can hold S/I/P controls (Textbox, Radiobutton etc). If u want to x'ferr the values of S/I/P controls from 1 page to another page, When User click on SUBMIT Button we can go for FORM tag.

### Attributes of Form Tag:

27-AUG-2012

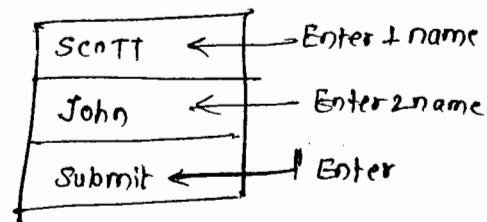
1) ACTION: By using this attribute we can specify the pagename in which page we can redirect form values.

2) METHOD: By using this attribute we can specify the type of method what we are using to x'ferr the form values. The possible methods are "GET" and "POST".

3) ENC TYPE: By using this attribute we can specify the encoding type of form data.

Ex: <form action="page2.html" method="POST">  
 <input type="text" name="t1">  
 <br>  
 <input type="text" name="t2">  
 <br>  
 <input type="submit" value="Send">  
 </form>

### Opp:



We need to provide name property to send the values of form from one page to another page.

Ex: <form action="page2.html" method="GET">  
 <input type="text" name="H1">  
 <br>  
 <select name='d1'>  
 <option> PHP </option>  
 <option> ASP </option>  
 <option> JSP </option>  
 </select>  
 <input type="Submit" value="Send" name="Sub">  
 </form>

## \* Difference Bet<sup>n</sup> GET and Post:

24

### GET

- i) GET x'fers the inform<sup>n</sup> thr. http head locat<sup>n</sup> & displays the data on URL address.
- ii) GET is unsecured.
- iii) GET x'fers limited amt of data.
- iv) ~~Head~~<sup>GET</sup> can't upload the file.

### Post

- i) POST x'fers the inform<sup>n</sup> thr. document body.
- ii) Post is highly secured.
- iii) Post x'fers <sup>huge</sup> ~~big~~ amt of data.
- iv) Post can upload the files.

### \$-GET:

By using \$-GET we can get the posted values of GET method. It is Array Variable the total no. of elements are equal to the total no. of posted values.

### \$-POST:

To Get the posted values of Post method.

Ex: <form method="get" action="page1.php">  
<input type="text" name='t1'>  
<br>  
<input type="text" name='t2'>  
<br>  
<input type="submit" name='sub' value='click'>  
</form>

```
<?php  
 1 print_r($_GET);  
 1 echo $_GET['t2'];  
 echo $_GET['sub'];  
 ?>
```

```

<?php
if (isset($_POST['sub'])) // if (isset
{
    echo "This is form PHP";
    $t1 = $_POST['t1'];
    $t2 = $_POST['t2'];
    echo "Value of first is: $t1";
    echo "Value of second is: $t2";
}
?>
<FORM METHOD="POST" action="pro.php">
<input type="text" name='t1'>
<br>
<input type="text" name='t2'>
<br>
<input type="Submit" name='sub' value='Click'>
</form>

```

28 Aug 2012

### ii) \$\_request:

By using this super global variable we can get the posted values of get method, and post Method. We can also get the values of COOKIES and query string

```

<form method="get" action="">
<input type="text">
<input type="text" name="method" value="click">
</form>

```

### iii) Query String:

It is a small amt of data on URL address followed by "?"

If u want to xfer some xtra data along with the form data we can go for Origin. We can pass query string in 2 ways.

- 1) query string with name and value
- 2) query string with only value.

By using \$\_REQUEST we can read the query string which contains name & value.

Ex: <?php

```

if(isset($_REQUEST['sub']))
{
    echo "value is";
    $_REQUEST['t1'];
    echo $_REQUEST['sno'];
}
echo $_REQUEST['uname'];

```

1) <form method="post" action="abc.php? sno=100 & uname=scott">  
<input type="text" name="t1" value="Enter name">  
<br>  
<input type="submit" value="click" name="sub">  
</form>

2) <?php  
if (isset(\$\_REQUEST['sub']))  
{  
val = \$\_POST['t1'];  
echo \$val;  
}  
?  
<form method="post" action="abc.php? sno=100 & uname=scott">  
<input type="text" name="t1" value="<?php echo \$val?>">  
<br>  
<input type="submit" value="click" name="sub">  
</form>

3) <?php  
if (isset(\$\_POST['sub']))  
{  
echo \$\_POST['sub'];  
}  
?  
<form method="post" value="">  
<input type="submit" name="sub" value="sub1">  
<input type="submit" name="sub" value="sub2">  
</form>

Assignment:Expected  
Op:No1: No2: Res:  +  -  \*

```

<FORM Method="post" Value=" "
<input type="text" name='t1'>
<br>
<input type="text" name='t2'>
<br>

<form action=" " type="post">
<input type="text" name='t1' value="php echo $val1 ?&gt;"&gt; &lt;br&gt;
&lt;input type="text" name='t2' value="<?php echo $val2 ?&gt;"&gt; &lt;br&gt;

&lt;form method="post" action=" "&gt;
No1: &lt;input name='t1' value="<?php
&lt;br&gt;    echo $no1 ?&gt;"&gt;
No2: &lt;input name='t2' value="<?php
&lt;br&gt;    echo $no2 ?&gt;"&gt;
Res: &lt;input type='Res' value="<?php
&lt;br&gt;    echo $res ?&gt;"&gt;

&lt;input type='submit' name='sub'
      value='+'&gt;
&lt;input type='submit' name='sub'
      value='-'&gt;
&lt;input type='submit' name='sub'
      value='*'&gt;
&lt;br&gt;
</pre

{



php</p

if (isset($_POST['sub']))



{



$no1 = $_POST['t1'];



$no2 = $_POST['t2'];



if ($_POST['sub'] == "+")



$res = $no1 + $no2;



if ($_POST['sub'] == "-")



$res = $no1 - $no2;



if ($_POST['sub'] == "*")



$res = $no1 * $no2;



}



?>


```

## \* `$_FILES`:

iii) By using this super global variable we can get the info<sup>n</sup> of uploaded files. It is 2 dimensional array variable providing 5 elements. Every element used the complete info<sup>n</sup> about uploaded file.

Each element first dimension is name of upload control

Q: What to create 2-Dimensional Array.

29-AUGUST-2018

```
→ <?php
$arr = array(100 => array(1,2,3), 20);
echo $arr[100][1];
?>
```

O/P : 2

If user upload any file from Browser to server first that file will upload into server temporary mem. locat<sup>n</sup>. We need to implement server-side script to move that file from temporary mem. locat<sup>n</sup> to permanent locat<sup>n</sup>.

In Temporary mem. locat<sup>n</sup> it saves the file with another name.

### \* Elements of `$_files`

→ `$_files` contains 5 elements. Each element's first dim is name of the upload control.

1) `$_files['nameofuploadcon']['size']`:

By using this element we can get the uploaded file size in bytes.

2) `$_files['nameofuploadcon']['name']`:

To get the uploaded filename

3) `$_files['nameofuploadcon']['type'] (MIME)`:

To get the uploaded filetype.

4) `$_files['nameofuploadcon']['tmp_name']`:

To get the temporary filename,

5) `$_files['nameofuploadcon']['error']`:

To get error no. if any error occurred at the time of file uploading.

MIME:

(MIME stands for Multipurpose Internet Mail Extension)  
It is a type of extension, used to xfer the files from one locat<sup>n</sup> to another locat<sup>n</sup>. User click on submit button.

By default FORM can xfer text format HTML data.  
If you want to xfer another format file, we need to specify MIME type of that file.

Some of the MIME types are:

exe: application/octet-stream

jpg: image/jpeg

pdf: application/pdf

<form> tag contains "enc" type attribute. By using this we can specify the MIME tag of input file.

multipart/form-data: This <MIME> Tag supports any type of file

\* is\_uploaded\_file(\$FILES['Fileupcon']['tmp\_name']):

By using this f<sup>n</sup>

We can check whether the file is uploaded from client syst. To Server Temporary locat<sup>n</sup> or not.

\* move\_uploaded\_file:

By using this, We can move the uploaded file from temporary locat<sup>n</sup> to permanent locat<sup>n</sup>. It contains 2 Arguments temporary filename and permanent locat<sup>n</sup> path

Ex: → <Form method="post" action="upload.php" enctype="multipart/form-data">  
<input type="file" name="file1">  
<br>  
<input type="submit" value="Send">  
</form>

<?php

?> if (is\_uploaded\_file(\$FILES['~~file1~~file1']['tmp\_name'])) save: upload.php  
{  
\$fname = \$FILES['file1']['name'];  
?> if (move\_uploaded\_file (\$FILES['file1']['tmp\_name'], "uploads/\$fname"))  
echo "File is Moved";  
?> else  
?> echo "Not moved";

```
else  
echo "NOT";  
?>
```

(30)

## \* Configurat' settings Related to files-uploads:

C:\xampp\tmp\ini

### 1) File-uploads:

By using this configurat' setting, we can allow and stop the file uploads. The default value is "ON". By changing this values as OFF we can stop the file-uploads.

### 2) upload\_max\_filesize:

By using this configurat' setting, we can specify the max<sup>m</sup> filesize to upload.

### 3) upload\_tmp\_dir:

By using this configurat' setting, we can specify the Temporary locat' for uploaded files.

The default locat' is tmp folder.

== x ==

30-AUG-2012

### \$\_FILES['uploadconname'][error]:

Super Global Variable

By using this configurat', we can get the error msg if any occurred at the time of file upload.

If the <sup>Error</sup> no. is zero, file is uploaded successfully. If Error is 1, one, file size is max<sup>m</sup> than server configurat' settings value. If Error is 2, file size is max<sup>m</sup> than Browser configurat' settings value. If Error is 3, there is some NW problem at the time of file uploading. If Error is 4, User selected the submit button without any file select.

### \$\_SERVER: (Get Server Configurat' setting)

By using this Super Global Variable, we can get the inform' about server & Browser. If you want to get server S/W, server IP address, Browser IP address etc, we can go for \$\_SERVER.

### (\$\_SERVER['SERVER\_SOFTWARE']):

By using this we can get the S/W inform' what is available in Webserver.

### \$\_SERVER['SERVER\_NAME']:

To get the name of Webserver

### \$\_SERVER['SERVER\_ADDR']:

To get the IP address of Webserver.

### \$\_SERVER['SERVER\_PORT']:

To get the port no. of Web server.

(By default port no. is '80')

\* `$_SERVER['REMOTE_ADDR']`:

To get the IP address of Browser.

\* `$_SERVER['DOCUMENT_ROOT']`:

By using this super Global variable, we can get the document root of current Application.

(c:/XAMPP/htdocs)

\* `$_SERVER['SERVER_ADMIN']`:

By using this we can get the Admin name ~~@~~ @ server-name,

\* `$_SERVER['SCRIPT_FILENAME']`:

To get the current Scriptname with Complete Script Path. (c:/xampp/htdocs/upload.php)

\* `$_SERVER['REMOTE_PORT']`:

To get the port no. of Browser

By default  
(1317)

\* `$_SERVER['SERVER_PROTOCOL']`:

To get the protocol Inform what we are using to xiferr the data. (http/1.1)

\* `$_SERVER['REQUEST_METHOD']`:

To get the method what we are using to xiferr data with the Browser and Server. (POST)

\* `$_SERVER['QUERY_STRING']`:

To get the query string value

\* `$_SERVER['REQUEST_URI']`:

To get the current Scriptname with query string value.

\* `$_SERVER['SCRIPT_NAME']` & `$_SERVER['PHP_SELF']`:

By using it's Element

We can get the currently executing Scriptname.

\* `$_SERVER['REQUEST_TIME']`:

By using this Element We can get the exact time inform of last Request, as Timestamp value.

~~\$\_SERVER['REQUEST\_TIME']~~

\* `$_SERVER['argc']`:

By using this element we can get the total no. of query string arguments.

Ex: page1.php

```
<?php  
$sno=100;  
echo "Value is ", $sno;  
?>
```

Page 2.php

```
<?php
```

(32)

## \* Protocols:

A set of instructions to transfer the data between Browser & server.

Protocols are divided into 2 Types.

- 1) Statefull protocols and
- 2) Stateless protocols.

### 1) Stateful protocols:

These protocols can maintain the state of Application means we can access the all previous Request and Response values from current request.

In Windows Application we are using these protocols.

Ex: TCP/IP, FTP etc.

### 2) Stateless protocols:

These protocols can't maintain the state of Application means we cannot get previous request and Response values from current request.

In Web Application we are using these protocols Bcoz, They don't carry the previous page values that's why the performance is very fast.

Ex: http, https, etc.

The main Drawback of Stateless protocols is maintaining the state of Application to overcome this drawback State Management Cookies Concept was introduced.

## \* COOKIES \*

(33)

The state management object using to maintain the state of Application. COOKIE stores the data in clients mem. locat<sup>n</sup>, These data we can access from any webpage within the Applicat<sup>n</sup>.

COOKIES stores the inform<sup>n</sup> in 2 locat<sup>n</sup>s. Either Hard Disk or RAM Memory locat<sup>n</sup> of Client System.

COOKIES are divided into 2 Types:

- 1) In Memory cookie.
- 2) Persistence cookie.

1) Inmemory Cookie: (Stores inform<sup>n</sup> in RAM)

If we create any cookie without Explicit Expiry tag comes under Inmemory cookie. Inmemory cookie stores the inform<sup>n</sup> in clients RAM mem. locat<sup>n</sup> and destroys the data when user closed the Browser.

2) Persistence cookie: (Stores inf. in Hard Disk)

If we create any cookie with Explicit Expiry tag comes under Persistence cookie. Persistence cookie stores the data in Hard Disk and Deletes the inform<sup>n</sup> when the Lifetime cookies completed.

### SET\_COOKIE:

By using this f<sup>n</sup> we can create the cookies in php.

### \$-COOKIE:

By using this Super Global Variable we can get the value of cookie.

31-AUG-2017

Cookies stored in Browser mem. locat<sup>n</sup> that's why we cannot access the cookies from one browser to another browser.

Cookies are browser dependent.

Ex: Page1.php

```
<?php  
Setcookie('x',100);  
echo "Cookie created";  
?>  
<a href="page2.php">GO</a>
```

Page2.php

```
<?php  
echo "Value is ".$_COOKIE['x'];  
?>
```

\$-COOKIE  $\Rightarrow$  To access Cookies

O/P: Cookie is created Go

Click on it, it displays  
Value is

## \* steps to create persistence cookie:

(34)

- i) Get the current Date and Time Inform<sup>d</sup> when user send the request to access the file where we created cookies.
- ii) Add Lifetime to the current date and time to get Expiry time.
- iii) Create cookies with that Expiry tag.

Ex: <%php

```
setcookie ("un", "scott");
SETCOOKIE ("sno", 1001, time() + 3600);
echo $_COOKIE ['un'];
echo $_COOKIE ['sno'];           ↗ If we set time like 3600sec means 1 hour  
the o/p will be same upto 1 hour.
?>
```

O/P : Scott 1001

Persistence Cookies will store in Hard Disk in a file of Browser Mem. locat?  
The name of that file is username@domain\_name

We can Delete the cookie from client Syst. by Recreating the cookie with completed Mozilla → Tools → option → privacy → custom settings → show cookies for History

## \* Disadvantages of Cookies:

- i) Cookies stores the inform<sup>d</sup> in client syst. that's why client can ~~not~~ delete the inform<sup>d</sup> OR client can modify cookie data.
- ii) Cookie is storing limited amt of data.
- iii) Cookie can store only Text data.

Internet Explorer → Tools → Internet → settings → options  
Select view files →

## \* SESSIONS \*

(35)

Sessions can store the data in Webserver. This data we can access from any webpage in the Browser. By using `$_SESSION`, we can create & access the sessions.

\* Differences Bet<sup>n</sup> COOKIES And SESSIONS.

### → COOKIES

### SESSIONS

- |  |  |
|--|--|
| i) COOKIES stores the info in Client-system.       | ii) SESSIONS stores the data in Server-system. |
| ii) COOKIES stores limited amt of data.            | iii) SESSIONS stores huge amt of data.         |
| iii) COOKIES <sup>can</sup> stores only Text data. | iv) SESSIONS can store any type of data.       |
| iv) COOKIES are unsecured.                         | iv) SESSIONS are Highly secured.               |

By Default, we cannot access the sessions of one page from another page. If you want to access the sessions we need to initialize the sessions when the request is started.

By using SESSION.AUTO-START we can initialize sessions. The Default value is zero. By changing its value as '1', we can start the sessions when the request is started. Otherwise use session\_start function in the application from where you want to access SESSIONS.

```
<?php  
session_start();  
$_SESSION['x']=100;  
echo $_SESSION['x'];  
?  
<a href="page2.php">Go To Next</a>
```

```
<?php  
session_start();  
echo $_SESSION['x'];  
?>
```

O/P: 100 GO TO NEXT ←  
100

### Session\_id:

`session_id` is an unique value generated by the server when user sends the 1<sup>st</sup> request to the server. It is alphanumeric string.

When user ~~connected~~ connected with server, without `session_id`, server creates new `session_id` for the users. This `session_id` ~~stores~~ in client syst. as ~~inmemory~~ cookie. The name of cookie is `PHPSESSID` and value is the `SESSION-ID`. At the same time, server temporary mem.locat<sup>n</sup>, a new file will create to store the session data. filename is same as `SESSION-ID` with prefix word `SESS`.

This file creates in Temporary Mem. Locat' of Server.

01-SEPT-2019 (36)

When user sending the 1<sup>st</sup> request without session\_id, New session will create. The same session\_id X'fers Bet' Browser and Server, which subsequent request and responses. If user closing the Website, The inmemory cookie will destroy. Again ~~will~~ if we connect with server, the Request will go without session\_id, Then server creates the new session\_id for the user again.

### \* SESSION\_id :

By using this f', we can get the session id what is generated by the Server.

```
<?php  
session_start();  
echo session_id();  
?  
<a href="page2.php">GOTONext</a>
```

```
<?php  
session_start();  
echo session_id();  
?>
```

Note: and starting  
After closing Browser each time,  
We can see diff. o/p. on Browser.

Ex:

```
<?php  
session_start();  
if (isset($_POST['sub']))  
{  
    $pro = $_POST['drpt'];  
    $qtn = $_POST['t1'];  
    $_SESSION[$pro] = $qtn;  
}  
?  
<form method="Post" action=" " >  
products:<select name='drpt'>  
<option> Nokia 1100 </option>  
<option> Samsung 1100 </option>  
</select>  
<br>  
Quantity:<input name='t1'>  
<br>  
<input type="submit" name="sub" value="Submit" >  
</form>  
<a href="bill.php"> show Bill </a>
```

```
<?php  
session_start();  
foreach($_SESSION as  
$k => $v)  
{  
    echo $k, " --- ". $v;  
    echo "<br>";  
}
```

O/p:

Products : Nokia 1100  
Quantity :  
Submit  
Show Bill

### \* SESSION\_UNSET:

By using this fn, we can delete the data of session.

### \* SESSION\_DESTROY:

To destroy the sessions.

```
Ex: <?php
    SESSION_START();
    $_SESSION['abc'] = "SCOTT";
    SESSION_UNSET();
    SESSION_DESTROY();
?>
```

### Ex: <?php      Login page

```
SESSION_start();
if(isset($_POST['sub']))
{
    $uname = $_POST['txtname'];
    $pwd = $_POST['txtpwd'];
    if($uname == "SCOTT" and $pwd == "SCOTT123")
    {
        $_SESSION['aut'] = true;
        echo "<script>location='Welcome.php'</script>";
    }
    else
        echo "Invalid";
}
?>

<form method="post" action=" " >
    username: <input type='text' name='txtname'>
    <br>
    password: <input type='text' name='txtpwd'>
    <br>
    <input type='submit' value='login' name='sub'>
</form>
```

### Welcome page

```
<?php
    session_start();
    if(isset($_SESSION['aut']))
    {
        echo "Welcome to user";
    }
    else
        echo "<script>location='login.php'</script>";
?>
<a href="logout.php"> logout </a>
```

### Logout page

```
<?php
    session_start();
    session_destroy();
    echo "Logged out successfully";
?>
```

## \* Configuration settings to work with sessions.

C:\xampp\php.ini

### 1) session.auto\_start:

By using this configuration setting, we can start the sessions when the request is started.

The default value is zero. By changing this value as "1", we can start the sessions.

### 2) session.save\_handler:

By using this configuration setting, we can specify the handler where u want to save sessions.

### 3) session.save\_path:

By using this, we can specify the path `local`, where u want to save sessions.

The default `local` is temporary folder of Xampp folder, C:\xampp\tmp

### 4) session.use\_cookies:

By using this configuration setting, we can use cookies or not to store the sessions. The default value is '1'. If we change value to '0', then we can see diff. session at each Refresh.

### 5) session.name:

By using this configuration setting, we can change the session\_cookie name.

The Default name is ~~PHPSESSID~~ "PHPSESSID".

### 6) session.cookie\_lifetime:

By using this configuration setting, we can increase or decrease the lifetime of cookie, what we r using to store sessions.

The default lifetime is '0' that's why it is creating Inmemory cookie.

By increasing this value we can create persistence cookie to store the sessions in client syst.

Ex:

```
<?php
session_start();
if (isset($_SESSION['aut'])) {
echo "<script>location='Welcome.php'</script>";
}
?>
<?php
```

```
<?php
Session_start();
Session_destroy();
Setcookie("PHPSESSID", " ", time() - 1);
echo "Logged out successfully";
?>
```

04-SEPT-2012

### session.gc\_maxlifetime:

By using this configuration setting, We can increase or decrease the lifetime to collect the unreferenced sessions by Garbage Collector.

The default value is 1440sec (24 min)

### \$-ENV:

(Environment Variables) By using this we can get Operating Syst. variables.

```
<?php
```

```
print_r($-ENV);
?>
```

Opp: ~~Array~~

### getenv():

By using this f<sup>n</sup>, We can get the values of Environment variables.

```
<?php
```

```
echo getenv("Computer name");
phpinfo();
?>
```

Opp: ~~XP~~

~~Windows 7 Version~~

### phpinfo():

By using this f<sup>n</sup>, we can display all super global variables or browser.

```
<?php
```

```
echo getenv("COMPUTER NAME");
phpinfo(16);
?>
```

Opp: Display all Environment variables.

### ini\_get():

By using this f<sup>n</sup>, We can get a Configuration settings value.

### ini\_set();

To change the Configuration settings value,

```
<?php
```

```
echo ini_get("precision"); //16:Opp
ini_set("precision", 14); //14:Opp
echo ini_get("precision");
?>
```

## \* FILE HANDLING \*

(40)

FILE HANDLING is the concept of reading the file contents as well as writing the file contents.

PHP is providing no. of fns, to read and write the file contents.

If we want to Read and write the contents of file, first we need to open the file with the specific file mode.

Different Types of file modes are available:

### 1) r(read):

To Read the file contents. If u open the file with read mode, file ptr locates at the beginning of file.

### ~~2)~~ w(write):

To write the contents in a file. If u open any file, with write mode it deletes the previous contents of files & locates the file ptr. at starting posn. If the file is not available, it creates a New file.

### 3) a(append):

To append some text with the existing content of file.

The file ptr. locates at the end of the file.

### 4) r+(read/write):

To Read and Write the contents of a file. It is same as Read mode. We can also write file contents as same as Read mode.

### 5) w+(write/read):

It is same as Write mode, we can also read file contents.

### 6) a+(append/read):

It is same as Append mode, we can also read file contents.

### 7) ~~fopen()~~ fopen():

By using this fn, we can open a file with specific file mode.

It contains 2 Arguments. filename and filemode.

### 8) fread():

To Read the file contents, arguments or file ptr. and file size

### 9) fwrite():

To write the contents in a file, arguments or file ptr. & ~~new content~~

### 10) filesize():

To get the total no. of Bytes occupied by a file. Argument is filename.

Ex: <?php

```
$fp = fopen("myfile.txt", "r");
$size = filesize("myfile.txt");
$cont = fread($fp, $size);
echo $cont;
?>
```

Ex: <?php

```
$fp = fopen("myfile.txt", "w");
fwrite($fp, "John123");
?>
```

Ex: <?php

```
$fp = fopen("myfile.txt", "r");
fwrite($fp, "smith");
echo fread($fp, filesize("myfile.txt"));
?>
```

Ex: <?php

```
$fp = fopen("myfile.txt", "a");
fwrite($fp, "John123");
?>
```

★ rewind():

By using this f<sup>n</sup>, we can locates the file ptr. at the starting locat<sup>n</sup>.

Ex: <?php

```
$fp = fopen("myfile.txt", "r+");
fwrite($fp, "smith");
rewind($fp);
echo fread($fp, filesize("myfile.txt"));
?>
```

★ fseek():

By using this f<sup>n</sup>, we can locate the file ptr. ~~on~~ <sup>on</sup> specified locat<sup>n</sup>

Ex: <?php

```
$fp = fopen("myfile.txt", "w+");
fwrite($fp, "Hello Scott");
fseek($fp, 2);
echo fread($fp, filesize("myfile.txt"));
?>
```

★ fgets():

By using this f<sup>n</sup>, we can read a line from a string. & it locates

the file ptr. at the begining of Next line.

Ex: <?php

```
$fp = fopen("myfile.txt", "r");
$line = fgets($fp);
echo $line;
$line = fgets($fp);
echo $line;
?>
```

O/P:

Line1  
Line2

NotePad

Line1  
Line2  
Line3  
Line4  
Line5

### \* fgetss():

It is same as fgets, But it ignores the html elements.

42

### \* fgetc():

By using this f<sup>o</sup>, we can get a character from a file.

Ex:

```
<?php  
$fp = fopen ("myfile.txt", "r");  
$line = fgetc ($fp);  
echo $line;  
$line = fgetc ($fp);  
echo $line;  
?  
?>
```

O/P : Li

05-SEPT-2012

### \* file\_get\_contents():

By using this f<sup>o</sup>, we can read the contents of a file without file ptr.

```
<?php  
echo file_get_contents ("myfile.txt");  
?>
```

### \* file\_put\_contents():

By using this f<sup>o</sup>, we can write new content in a file.

```
<?php  
echo file_put_contents ("myfile.txt", "abcd");  
?>
```

### \* readfile():

By using this f<sup>o</sup>, we can read the contents of a file at the same time we can write those contents in current buffer locat?

```
<?php  
readfile ("myfile.txt");  
?>
```

### \* file:

This f<sup>o</sup> reads each & every line of a file & returns every line as array elements.

```
<?php  
$arr = file ("myfile.txt");  
print_r ($arr);  
?>
```

O/P : Array [0] = Line1

Array [1] = Line2

Array [2] = Line3

### \* filectime:

By using this f<sup>o</sup> we can get the created date & time inform<sup>d</sup> of a file the O/P it returns as timestamp value.

Ex: <?php  
echo ~~gettime~~ date('d/m/y', filetime("myfile.txt"));  
?>

(43)

O/P : 04/09/12

### fileatime:

To get the last access date & time info of a file.

### filemtime:

To get the last modified date and time info of a file.

### ~~unlink~~ unlink:

By using this f, we can delete a file permanently from Hard Disk.

```
<?php  
unlink("index.html");  
?>
```

### Copy ():

To copy a file from one locat to another locat.

```
<?php  
copy("index.php", "C:/ind.php");  
?>
```

### rename ():

```
<?php  
rename("demo.php", "demoxyz.php");  
?>
```

## \* Directory Functions: \*

### 1) mkdir:

By using this f, We can create new directory.

```
<?php  
mkdir("c:/abcd");  
?>
```

### 2) rmdir:

To remove existing directory

```
<?php  
rmdir("C:/abcd");  
?>
```

### 3) opendir:

To open a directory.

```
<?php  
$dh = opendir("E:/abcd");  
?>
```

### readdir:

To read the files of a directory.

### closedir:

To close the opened directory.

Ex: <?php

```
$dh = opendir("C:/xampp");
while ($file = readdir($dh))
{
    echo $file;
    echo "<br>";
}
?>
```

### scandir:

By using this f<sup>n</sup> we can scan the all files of a directory, & Returns ~~the~~ the op<sup>n</sup> in the form of an Array.

<?php

```
$arr = scandir("C:/xampp");
print_r($arr);
?>
```

### getcwd:

This f<sup>n</sup> used to get the current working directory.

Ex: <?php

```
echo getcwd();
?>
```

### chdir:

This f<sup>n</sup> used to change the current working directory locat<sup>n</sup>.

Ex: <?php

```
chdir("C:/");
chdir("C:/");
$fp = fopen("1.jpg", "r");
?>
```

FAQ: 6: How can we submit the form (44)

without submit button?

→ By using "Javascript submit"

→ <form method="post" action="page.php">  
<select onchange="submit()">  
<option> PHP </option>  
<option> Asp </option>  
</select>  
</form>

Q: What does a special set of tags

<? = and ?> do in PHP?

→ <?

"hi"  
?>

Op: hi  
=

06 SEPT 2012

### Webserver:

Webserver is a s/w used to run Web applicat<sup>n</sup>s.

Once, We taken the space in the Webserver, We have to upload the project files from our local syst. to External servers.

By using the Control Panel, We can upload the files into external servers.

Diff. Types of External servers are available to upload project. We can release registered the Space to the websites.

|                   |
|-------------------|
| www.bethost.com.  |
| www.godaddy.com   |
| www.noads.biz     |
| www.rackspace.com |
| www.ipage.com     |

## \* File Manager:

(45)

By using this option we can upload the files from our local sys. to external servers.

## \* Database Manager:

By using this option, we can create the DB tables in External Webservers.

## \* Email Manager:

To create the email\_ids with our domain name.

Every External server contains Root Directory, we should upload a project files into ~~external~~ Root Directories.

## \* Index:

It is a start up of a file, Every server should contains start up filename with index.

www.byethost.com → Free hosting → sign up → Registration Form → After activation →  
online File Manager → htdocs

<http://nileshsoftpure.byethost15.com>.

## \* Header:

It is the small amt of data packet bet<sup>n</sup> Browser and Server.

Headers are divided into 2 Types.

1) Request header and

2) Response header.

### 1) Request Header:

Data transmission Bet<sup>n</sup> Browser & Server is called as Request Header.

### 2) Response Header:

Data transmission Bet<sup>n</sup> Server & Browser is called as Response Header.

In PHP, we are using header() to declare headers. The available headers are location, content-type, content-length, etc.

## \* Location:

07. SEPT. 2012

By using this header, we can redirect from one page to another page.

```
<?php
if(isset($_POST['sub']))
{
    if($_POST['sub'] == "page1")
        header("location: Page1.php");
    else
        header("location: Page2.php");
}
```

```
<form method="post" action=" " >
<input type="submit" name="sub" value="Page1">
<input type="submit" name="sub" value="Page2">
</form>
```

Save! Headerlocation.php

O/P:  
Blank

Page1 Page2

## \* content-disposition: attachment:

By using this header we can display the download dialogue box or Browser.

```
<?php
header("Content-Disposition: attachment;
    filename=abc.txt")
?> Note: We see the window of downloading as on O/p
```

```
<?php
header("Content-Disposition: attachment;
    filename=abc.txt");
readfile("myfile.txt");
?>
```

~~Script~~ save select.htm

Query String

```
<a href="frm.php?fname=drupal-6.1.zip">
    Download</a>
<br>
<a href="frm.php?fname=get.php">Download2</a>
<br>
<a href="frm.php?fname=myfile.txt">Download3</a>
```

```
<?php
    Form.php
    $qs = $_REQUEST['fname'];
    $size = filesize($qs);
    header("Content-length: $size");
    header("Content-disposition: attachment;
        filename=$qs");
    readfile($qs);
?>
```

## \* WWW-Authenticate: Basic Realm=<sitename>

By using this header, we can display the Authentication dialogue Box.

Every Browser contains Authentication Dialogue Box. To activate this dialogue box, we are using this header.

## \* \$\_SERVER['PHP\_AUTH\_USER']:

By using this super global variable we can get the username textbox value.

## \* \$\_SERVER['PHP\_AUTH\_PW']:

To get the password Textbox value.

Authentication

Dialogue box refers the information headers that's why the data transmission performance is very fast.

If we are using Authentication dialogue box, By default the Textbox values will store in Super Global variables. That's why no need to use session variables to get the control values from one page to another page.

Ex: <?php save: WWWAuthBasicRealm.php

```
header("WWW-Authenticate: Basic
    Realm=abcd");
```

```
echo
$_SERVER['PHP_AUTH_USER'];
echo
```

```
$_SERVER['PHP_AUTH_PW'];
?> O/P: 

```

Ex: <?php save: Authusernamepwd.php

```
$auth = 0;
if(isset($_SERVER['PHP_AUTH_USER']) and isset($_SERVER['PHP_AUTH_PW']))
```

```
{ $uname = $_SERVER['PHP_AUTH_USER'];
    $pwd = $_SERVER['PHP_AUTH_PW'];
    // $fp = fopen("logins.txt", "r");
    while ($line = fgets($fp))
```

```
{ list($un, $pw) = explode(":", $line);
    $un = trim($un);
    $pw = trim($pw);
    if($uname == $un and $pwd == $pw)
```

```
{ $auth = 1;
    break;
}}
```

```
if($auth == 0)
    header("WWW-Authenticate: Basic Realm="MyWebsite");
```

```

else
{
    header("location:Welcome.php");
}
?

```

O/P:

Authentication Required		X
Username :	<input type="text"/>	
Password :	<input type="password"/>	
<input type="button" value="Login"/> <input type="button" value="Cancel"/>		

47

? <?php  
echo "Welcome to ".\$\_SERVER['PHP\_AUTH\_USER'];

? If we remove dot O/P: Welcome to (.), Then it shows Error.

10-SEPT-2012

## \* MAIL \*

By using this f<sup>n</sup>, we can send a mail from our Applicat<sup>n</sup> to Destinet<sup>n</sup> mail id. It contains 4 arguments address, subject, body, and collect<sup>n</sup> of headers.

Mail f<sup>n</sup> Fourth argument is collect<sup>n</sup> of headers, the separator bet<sup>n</sup> 1 header and another header is "\r\n".

We need to use ~~enter~~ smtp server to send the mails from one locat<sup>n</sup> to another locat<sup>n</sup>.

Ex: <?php

```

if(mail("abc@gmail.com","hi","hello","From:xyz"))
    echo "sent";
else
    echo "Not sent";
?

```

## \* Different Types of Mail Related Headers:

Diff Types of Mail Related

headers are available like Cc (Carbon Copy), Bcc (<sup>Blank</sup> Carbon Copy), content-type etc.

<?php

```

if(mail("abc@gmail.com","hi","hello","From: xyz\r\nCc:scott@gmail.com"))
    echo "sent";
else
    echo "Not sent";
?

```

### Ex 2:

```

<?php
if (isset($_POST['sub1']))
{
    $to = $_POST['to'];
    $sub = $_POST['sub'];
    $from = $_POST['from'];
    $body = $_POST['body'];
    if (mail($to, $sub, $body, "From: $from"))
    {
        echo "Mail sent";
    }
    else
        echo "Not sent";
}
}

<form method="post" action=" " >
To:<input name='to'>
<br>
Sub:<input name='sub'>
<br>
Body:<input name='body'>
<br>
From:<input name='from'>
<br>
<input type="submit" value="click" name="sub1">
</form>

```

~~Ex 2~~

### Ex 3

(48)

```

<?php
$str = <<<abc
<form method="post" action="http://swamysolutions.com/geto.php">
<span style="color: Red"> Username </span>
<input type="text" name='t1'>
<br>
<input type="submit" value="click" name="sub">
abc;
if (mail("abc@gmail.com", "HTML email",
        $str, "From:smith@gmail.com/r/nContent
        type:text/html"))
{
    echo "Mail sent";
}
else
    echo "Not sent";

```

## \* String fns:

String is a collect<sup>n</sup> of characters, no. of string fns are available to work with string

1) strlen: By using this fn we can get the length of all characters of a string.

Ex: <?php

```
$str = "Welcome";
echo strlen($str);
?>
```

O/P: 7

### 2) strtoupper:

By using this fn, we can convert the all characters of a string into uppercase characters.

Ex: <?php

```
$str = "Welcome";
echo strtoupper($str);
?>
```

O/P: WELCOME

### 3) strtolower:

By using this funct<sup>n</sup>, we can convert the all characters of a string into lowercase characters.

<?php

```
$str = "WELCOME";
echo strtolower($str);
?>
```

O/P: welcome.

### 4) chr:

By using this fn

we can get the character of an ASCII value.

Ex: <?php
echo chr(97);
?>

### 5) ord:

To get the ASCII

value of \$/p character.

Ex: <?php

```
echo ord('A');
?>
```

O/P: 65

### 6) ucfirst():

By using this fn we can convert the 1<sup>st</sup> character of a string into uppercase character.

Ex: <?php

```
$str = "welcome";
echo ucfirst($str);
?>
```

O/P: Welcome.

### 7) ucwords:

Converts

the first character of all words into uppercase.

<?php

```
$str = "Welcome To Scott";
echo ucwords($str);
?>
```

O/P: Welcome To Scott

### 8) nl2br:

By using this fn

we can break new lines of a string.

<?php
\$str = "Welcome
to
PHP";
echo nl2br(\$str);
?>

O/P: Welcome
to
PHP.

String where

single quotations & double quotations are occurred.

Ex: <?php

```
$str = "Welcome";
$str1 = addslashes($str);
echo $str1;
?>
```

O/P: Wel'come

### 9) addslashes:

By using

this fn, we can add backslashes within the single quotations & double quotations are occurred.

Ex: <?php

```
$str = "Wel'come";
$str1 = addslashes($str);
echo $str1;
?>
```

O/P:

### 10) stripslashes:

By using this fn, we can remove the slashes what we added with add slashes.

### 11) addslashes:

Adds the backslashes in front of Specified Character.

### 12) stripslashes:

To Strip the slashes what we added with addslashes

Ex: <?php

```
$str = "We\l'come";
$str1 = addslashes($str, "e");
echo $str1;
echo "<br>";
echo stripslashes($str1);
?>
```

O/P:

### 13) str\_word\_count:

By using this f<sup>n</sup>, we can get the total no. of words of a string. We need to pass mode value as second argument. If mode is zero (0), It returns total no. of words.

If mode is one (1), It returns each word as an Array. Array values of words and keys are 0,1,2,...

If mode is 2, it returns the all words as Array, values are the words of string & keys are the index no. of the string,

Ex: <?php

```
$str = "Welcome to scott";
print_r(str_word_count($str, 2));
```

O/P:

### 14) similar\_text:

(50)

By using this f<sup>n</sup>, we can get the similarities Betw 2 ~~2~~ string.

<?php

```
$str = "Welcome to scott";
$str1 = "Welcome to scott";
echo similar_text($str, $str1);
```

IF we write like that  
similar\_text(\$str, \$str1); Then o/p will be 1.2.  
?>

O/P: 1.4

### 15) join():

By using this f<sup>n</sup>, we can join Array elements as a string. It is same as implode().

```
<?php
$arr = array("scott", "smith");
echo join("/", $arr);
?>
```

O/P:

### 16) trim:

Removes the LHS & RHS spaces of a string.

### 17) ltrim:

Removes the LHS spaces of a string.

### 18) rtrim:

Removes the RHS spaces of a string.

<?php

```
$str = "Welcome ";
$str1 = ' scott';
echo trim($str), ltrim($str1);
?>
```

O/P:

### 19) chop:

It is an alias of rtrim.

### 20) str\_shuffle:

This f<sup>n</sup> randomly shuffles a String.

<?php

```
$str = "Welcome";
echo str_shuffle($str);
?>
```

## 21) str\_replace:

Replaces a part of string with a New string.

## 22) str\_ireplace:

It is same as str\_replace, But it is case Insensitive.

<?php

```
$str = "Welcome";
echo str_ireplace("Come", "go", $str);
?>
```

## 23) str\_repeat:

Repeats a string, with a specified no. of times.

<?php

```
$str = "Welcome";
echo str_repeat($str, 3);
```

G/P: WelcomeWelcomeWelcome  
3 Times.

## 24) split:

Converts a string as an Array.

<?php

```
$str = "Welcome to Scott";
$arr = str_split($str, 3);
print_r($arr);
?>
```

## 25) strcmp:

This f<sup>n</sup> compares 2 strings and Returns '0', If Both are same, It Returns >0, If 1<sup>st</sup> string > 2<sup>nd</sup> string. It Returns <0, If 1<sup>st</sup> string < 2<sup>nd</sup> string.

Ex: <?php

```
$str = "Smith";
$str1 = "smith";
echo strcmp($str, $str1);
?>
```

Ex: 2 → <?php

```
$str = "Smith";
$str1 = "smith";
echo strcmp($str, $str1);
?>
```

## 26) strcasecmp:

(5)

It is same as strcmp.

But it is case Sensitive.

\* G/P \*

## 27) strchr:

By using this f<sup>n</sup>, we can get all characters of a string from Specified character.

<?php

```
$str = "SmithHello";
echo strchr($str, "t");
?>
```

## 28) strrchr:

It is same as strchr,

But it gets the string from reverse dir.

Ex: <?php

```
$str = "SmithHello";
echo strrchr($str, "t");
?>
```

## 29) strstr:

same as strchr.

## 30) striistr:

same as strstr, But case Insensitive.

## 31) strpos:

Find the positt<sup>n</sup> of a character in a string.

## 32) stripo<sub>s</sub>:

same as strpos, But case Insensitive.

Ex: <?php

```
$str = "Welcome";
echo strpos($str, "e");
?>
```

## 33) substr:

To get the substring of a string.

<?php

```
$str = "Welcome";
echo substr($str, 3, 4);
?>
```

### 34) strip\_tags:

By using this fn, we can strip the html tags.

```
<?php
    $str = "Welcome <input type='button' value='click'>";
    echo strip_tags($str);
?>
```

### 35) strrev:

By using this fn, we can convert the string into Reverse dir.

```
<?php
    $str = "hello";
    echo strrev($str);
?>
```

### Encryption:

Date : 12. SEPT. 2019

It is a concept of Encoding & Decoding the data. Basically, we have 2 Types of Encrypt in PHP. Those are

- 1) One-way Encryption
- 2) Two-way Encryption

### I) One-way Encryption:

By using

this, we can encode the data. But, we can not decode encoded data.

#### 1) md5 : (Msg. Digest 5)

By using this fn, we can encode the data as 32 characters length, alphanumeric string.

(combination of Alphabets & nos.)

Ex: <?php

```
$str = "Scott";
echo md5($str);
?>
```

O/P

### 2) crc32 (cyclic Redundancy check) (52)

This fn converts \$/p string as Numeric encrypted data.

Ex: <?php

```
$str = "scott";
echo crc32($str);
?>
```

### 3) SHA1:

Converts \$/p string as 40 character length alphanumeric string.

Ex: <?php

```
$str = "scott";
echo sha1($str);
?>
```

### 4) crypt:

Converts \$/p string as alphanumeric string with special characters ^

Ex: <?php

```
$str = "scott";
echo crypt($str);
?>
```

### II) Two-way Encryption:

By using

this concept, we can encode & decode the data.

#### 1) base64\_encode:

By using this fn, we can encode a string as 64-bit encrypted data.

Ex: <?php

```
$str = "Welcome";
?>
```

#### 2) base64\_decode:

To decode the encoded data.

Ex: <?php

```
$str = "Welcome";
$str1 = base64_encode($str);
echo $str1;
echo base64_decode($str1);
?>
```

## \* Date And Time Functions:

### 1) time:

By using this f<sup>n</sup>, we can get the current date and time inform<sup>n</sup> as timestamp value.

A timestamp value is nothing but a total no. of seconds from ~~Saturn~~ ~~Sunday~~ 1970 Jan 1<sup>st</sup>, 12.00am to upto now.

Ex: <?php

```
echo time();
?>
```

### 2) date:

Converts the f/p timestamp

Value as date, month and year form<sup>n</sup> string.

Ex: <?php

```
$t = time();
echo date('d/m/y', $t);
?>
```

O/P:

### 3) getdate:

By using this f<sup>n</sup>, we can get the current date and time inform<sup>n</sup> as an Array.

Ex: <?php

```
print_r(getdate());
?>
```

### 4) sleep:

By using this f<sup>n</sup>, we can stop the exec<sup>n</sup> of script upto specified time intervals.

Ex: <?php

```
echo "hi";
echo "hello";
sleep(10);
echo "scott";
?>
```

O/P: hi  
hello  
scott

Note:

scott display after 10 sec.

\* characters what we can used

With date function are as follows: →

i) d: 2 digit day with leading 0.

ii) j: 2 digit day without 0.

iii) W: day no. from 0 to 6.

iv) N: ISO representor<sup>n</sup> of day number (1-7)

1) L: Full day name

2) D: Three letters day name.

3) Z: Day no. in the year  
(small)

4) W: Week no. in the year.

5) m: month no. with 0.

6) n: month no. Without 0.

7) F: full month name,

8) M: Three letters month name.

9) t: No. of days in the given month.

10) l: Checks the givn yr is leap or not.

11) y: 2 digit year no.

12) Y: 4 digit year no.

13) a: Lower case antimeridiem/post  
(am) meridiem  
(pm)

14) A: Upper case antimeridiem/post  
(AM) meridiem.  
(PM)

15) g: 12 hour format without leading 0.

16) G: 24 hour format without leading 0.

17) h: 12 hour format with zero.

18) H: 24 hour format with zero.

19) s: seconds,

20) i: minutes.

Ex: <?php

```
echo date("i");
?>
```

checkdate:

By using this f<sup>n</sup>, we can  
check the given date is existed or not.

Ex: <?php

```
echo checkdate("2", "29", "2012");  
?>
```

Taqueria

(56)

१

२

३

४

५

६

७

८

९

०

१

२

३

४

५

६

७

८

९

०

१

२

३

४

५

६

७

८

९

०

१

२

३

४

## JQuery

20. SEPT. 2019 (57)

- \* JQuery is an Open Source JavaScript library providing no. of predefined functionalities.
- \* JQuery is available with different extensions like JQuery.js, JQuery-1.js, etc.
- \* JQuery functionalities we can apply on any type of element.
- \* JQuery supports any type of Web browser.
- \* JQuery methods we can apply on html elements using '\$' 21. SEPT. 2019

```
<Script src = "jquery-1.js">
</script>
<style>
p{
    background-color : green;
    color : white;
}
</style>
<Script>
function funshow()
{
    $("#P2").show("slow")
}
function funhide()
{
    $("#P2").hide ("slow")
}
</script>
<p id = "P1">
    some content . . .
</p>
<p id = "P2">
    some content . . .
</p>
<input type = "button" value = "show"
       onclick = "funshow()">
<input type = "button" value = "hide"
       onclick = "funhide()">
```

- \* show(): Display elements in Horizontal dir?
- \* Hide(): Hides the elements in Horizontal dir?
- \* slideUp(): Hides the elements in vertical Dir?
- \* slideDown(): Displays the elements in vertical dir?

SAVE AS: ~~funshowhide.htm~~ funshowHide.htm  
Op:

Ex:

```
<Script>src = "jquery-1.js">
</script>
<Script>
function fun1(t)
{
$(t).fadeOut("slow")
$(t).fadeIn ("slow")
}
</script>


```

SAVE AS: ~~funny.htm~~  
FadeInFadeOut.htm

(58)

### \* fadeTo:

To fade the element upto the specify value.

SAVE AS: ~~funny3.htm~~  
FadeTo.htm

```
<Script>src = "jquery-1.js">
</script>
<Script>
function fun1()
{
$(".div1").fadeTo("slow", 0.2)
}
</script>
<Style>
.div1{
background-color: black;
width: 200;
height: 200;
position: absolute;
top: 0;
left: 0;
}
</Style>

<div class = "div1" onmouseover = "fun1()">
</div>
```

## \*corner()

By using this f<sup>o</sup> of JQuery, we can display contents with round corners.

It is available with "JQuery.corner.js" library. The supporting library is jquery.js, jquery-1.js etc.

```
<script src="jquery-1.js">
</script>
<script src="jquery.corner.js">
</script>
<script>
function fun1()
{
    $("*").corner();
}
</script>

<br>
<div id="div1" style="width:100; height: 100; background-color: blue"></div>
<button onclick="fun1()> click </button>
```

## \*addClass():

By using this method, we can add the class selector properties on the elements.

## \*removeClass():

To Remove the class selector properties of an Elements.

```
<script src="jquery-1.js">
</script>
<style>
.class {
    background-color: yellow; color: red;
}
</style>
<script>
function fun2()
{
    $("#p1").removeClass("class");
}
function fun1()
{
    $("#p1").addClass("class")
}
</script>
<p id="p1" onmouseover="fun1()" onmouseout="fun2()>
Content ...
</p>
```

## \* Cycle :

By using this fx, we can apply sliding effect on image correctly. This fx is available in jquery. cycle.js library.com, jquery.cycle.min.js etc.

### fx:

By using this property we can specify the type of sliding effect what u want to apply on image correctly.

(Zoom, Fades, Wipe, Curtainx, Curtainz, Shuffle etc. fx's)

Ex: `<script src='jquery-1.9.js'>`

`</script>`

`<script src='jquery.cycle.all.js'>`

`</script>`

`<script>`

`function func()`

`{`

`$("#div1").cycle({fx:'zoom'})`

`}`

`</script>`

`<style>`

`#div1{`

`position: absolute;`

`top: 100;`

`left: 100;`

`}`

`</style>`

`<div id="div1">`

``

``

``

`</div>`

`<input type="button" value="click" onclick="func()" />`

- i) blindx
- ii) blindy
- iii) blindz
- iv) cover
- v) curtainx
- vi) curtainy
- vii) fade
- viii) fadeZoom
- ix) growx
- x) growy
- xi) none
- xii) scrollUp
- xiii) scrollDown
- xiv) scrollLeft
- xv) scrollRight
- xvi) scrollHorz
- xvii) scrollVert
- xviii) shuffle
- xix) slideX
- xx) slideY
- xxi) toss
- xxii) turnUp
- xxiii) turnDown
- xxiv) turnLeft
- xxv) turnRight
- xxvi) UnCover
- xxvii) wipe
- xxviii) zoom

## \* Validate.js :

By using this fn, we can validate the form controls. This fn is available in Validatejquery.validate.js library. This library is providing no. of properties to apply the validation those r required, email, number, etc.

By using errorClass property, we can apply the class selector on ~~library~~ Validate library. When we r applying validations on the elements, we need to provide name property to the elements.

```

<script src="jquery-1.js">
</script>
<script src="jquery.validate.js">
</script>
<style>
.cls1{
    color: blue;
    border-color: red;
}
</style>
<script>
    function fun1()
    {
        $('#frm1').validate({errorClass : 'cls1'})
    }
</script>
<body onload="fun1()">
<form id="frm1" action="page1.html">
    Username: <input type="text" name="txtnuser" class="required">
    <br>
    Password: <input type="password" name="pwd" class="required"
                    minlength=10 maxlength=20>
    <br>
    Email: <input type="text" class="required_email" name="Em">
    <br>
    Age: <input class="required_number" min=20 max=99 name="ag" id="fg">
    <br>
    URL: <input type="text" name="Ur" class="Url">
    <br>
    <input type="submit" value="Click" name="Sub">
</form>
```

## \* Cookie):

By using this fn, we can create the cookies in a client-syst.  
It is available in cookie.js library. To create the persistence cookie we can use expires properties.

If u want to destroy the cookie before lifetime, Recreate that cookie with null value.

```
<script src="jquery-1.js"></script>
<script src="cookie.js"></script>
<script>
function fun1()
{
$.cookie("un", "scott")
$.cookie("city", "hyd", {expires:1})
alert($.cookie("un"))
alert($.cookie("city"))
}
</script>
<input type="button" value="create" onclick="fun1()">
```

## \* Strong Password:

```
<script>
function fun1(val)
{
x=false;
for(i=0; i<val; length; i++)
{
ch=val.charAt(i)
if(ch=="@" || ch=="#")
{
x=true
break;
}
}
if(x==false)
alert("Not strong");
else
alert("strong")
}
</script>
<input onblur="fun1(this.value)">
```

## † Webcam.js : (library)

By using this library we can display the video capture control on Browser. we can activate camera control using this library. It is providing no. of methods.

### i) Webcam.get\_html :

By using this method, we can get HTML code Webcam control on Browser. It contains 4 Arguments width & height of video Recorder control and width & Height of Image control.

### ii) Webcam.snap :

By using this, we can take a snap from Webcam control.

### iii) Webcam.reset :

To Reset the Webcam control.

### iv) Webcam.configure :

To Configure the cam control.

### v) Webcam.set\_api\_url :

This f<sup>n</sup> is used to execute php script when user take a snap from a Webcam control.

### vi) Webcam.set\_quality :

(1-100) By using this f<sup>n</sup>, we can change the quality of jpeg image

### vii) Webcam.set\_shutter\_sound :

By using this f<sup>n</sup>, We can enable and disable sound

### Ex:

```
<script src="Webcam.js"></script>
<script>
document.write(Webcam.get_html(400,400,500,500));
Webcam.set_api_url("store.php");
Webcam.set_quality(100);
Webcam.set_shutter_sound(true);
</script>
<body>
<br> <input type="button" value="snap" onclick="Webcam.snap()">
<input type="button" value="Reset" onclick="Webcam.reset()">
<input type="button" value="configure" onclick="Webcam.configure()">
</body>
```

```

<?php
$filename = time(). ".image.jpg";
$content = file_get_contents("php://input");
file_put_contents($filename, $content);
?>

```

no space

### \* datepicker :

\* By using this f<sup>n</sup>, we can display the calendar control on browser.  
It is available in datepicker.js library. It requires some supporting libraries those are jquery.ui.core.js, jquery-1.6.2.js.

\* It requires a supporting CSS library i.e. jquery.ui.css

\* This library is providing no. of methods ~~different~~

#### dateFormat() :

By using this method, we can change the format of Date & Time.

#### minDate() :

To specify minimum Date.

#### maxDate() :

To specify the Max<sup>m</sup> Date.

#### onSelect() :

To execute a JavaScript, when a date is changed.

Ex<sup>e</sup>

26. SEPT. 2012

<head>

```

<link rel="stylesheet" href="jquery-ui.all.css">
<script src="jquery-1.6.2.js"></script>
<script src="jquery.ui.core.js"></script>
<script src="jquery-ui-datepicker.js"></script>
<script>
    function fun1()
    {

```

```

        $("#txt1").datepicker({dateFormat:"yy/mm/dd", minDate:0, maxDate:03,
```

```

        onSelect : function(data)
        {

```

```

            alert(data)
        }
    }
}
```

}

</script>

<head>

<body onload="fun1()">

FromDate:<input type="text" id="txt1">

</body>

Ex:

```

<head>
  <link rel="stylesheet" href="jquery.ui.all.css">
  <script src="jquery-1.6.2.js"></script>
  <script src="jquery.ui.core.js"></script>
  <script src="jquery.ui.datepicker.js"></script>
  <script>
    function fun1()
    {
      $("#txt1").datepicker({dateFormat:"yy/mm/dd", minDate=0, maxDate=3,
        onSelect: function(data){$("#txt2").datepicker({dateFormat:"yy/mm/dd"})}
        $("#txt2").datepicker("option", "minDate", data)}
      })
    }
  </script>
</head>
<body onload="fun1()"> FromDate:<input type="text" id="txt1"> <br>
  ToDate:<input type="text" id="txt2">
</body>

```

\* draggable :

By using this library we can drag the elements on webpage.  
The library name is jquery.ui.draggable.js

The supporting libraries are jquery-1.6.2.js, jquery.ui.core.js,  
jquery.ui.widget.js, jquery.ui.mouse.js

```

<script src="jquery-1.6.2.js"></script>
<script src="jquery.ui.core.js"></script>
<script src="jquery.ui.widget.js"></script>
<script src="jquery.ui.mouse.js"></script>
<script src="jquery.ui.draggable.js"></script>
<script>
  $(document).ready(function(){
    $("#img1").draggable()
    $("#div1").draggable()
  })
</script>
</head>
<body>
  
  <div id="div1" style="border:1px solid silver; width:100; height:100; background-color: #green">

```

Hello  
<br>

Hi  
</div>  
</body>

Ex:

`<script src="jquery-1.6.2.js"></script>  
<script src="jquery.ui.core.js"></script>  
<script src="jquery.ui.widget.js"></script>  
<script src="jquery.ui.mouse.js"></script>  
<script src="jquery.ui.draggable.js"></script>  
<script>  
 sec=60;  
 min=2;  
 var t;  
 function start()  
 {  
 t=setInterval("ctime()",50);  
 }  
 function ctime()  
 {  
 sec--  
 if(min==0 && sec==0)  
 {  
 document.getElementById('div1').innerHTML=min+":"+sec  
 alert("Game over")  
 $("img").hide("slow")  
 clearInterval(t)  
 }  
 if(sec<0)  
 {  
 sec=59  
 min--  
 }  
 document.getElementById('div1').innerHTML=min+":"+sec  
 }  
 function fun1(){  
 $("img").draggable()  
 start();  
 }  
</script>`

```

<body>
  <div id="div1" style="color:blue; font-size:40">
    3:00
  </div>
  
  
  
  
  
  <input type="button" value="play" onclick="fun1()">
</body>

```

27. SEPT. 2013

### toggle():

By using ~~effect f~~, we can execute 2 f's parallelly, one after another. It is available in jquery.effects.core.js library. The supporting library is jquery-1.6.2.js library.

```

<script src="jquery-1.6.2.js"></script>
<script src="jquery.effects.core.js"></script>
<script>
$(function()
{
  $('#button').toggle(function()
  {
    $('#img1').hide("slow") $('#button').val("show")
  })
  function()
  {
    $('#img1').show("slow")
    $('#button').val("Hide")
  });
});
</script>
<body>
<input type='button' value='Hide' id="button">
<br>

</body>

```

jquery effects library is providing diff. types of Effects, like pulse, slide, etc. We can apply this effects on Elements ~~to hide them~~. The available effects are ~~slide~~ blind, bounce, fold etc. For Every type of Effect ~~there~~ library is available. We need to include the library in our webpage.

The supporting library is jquery-1.6.2.js

Ex:

```

<Script src = "jquery-1.6.2.js"></Script>
<Script src = "jquery.effects.core.js"></Script>
<Script src = "jquery.effects.blind.js"></Script>
<Script src = "jquery.effects.bounce.js"></Script>
<Script src = "jquery.effects.clip.js"></Script>
<Script src = "jquery.effects.drop.js"></Script>
<Script src = "jquery.effects.explode.js"></Script>
<Script src = "jquery.effects.fold.js"></Script>
<Script src = "jquery.effects.highlight.js"></Script>
<Script src = "jquery.effects.pulse.js"></Script>
<Script src = "jquery.effects.scale.js"></Script>
<Script src = "jquery.effects.shake.js"></Script>
<Script src = "jquery.effects.slide.js"></Script>
<del><Script>
        function fun1()
    {
        var val = document.getElementById('drop1').value;
        $('#img1').hide(val);
    }
</Script>
<head>
<Select id = "drop1">
    <option value = "blind"> blind </option>
    <option value = "bounce"> bounce </option>
    <option value = "clip"> clip </option>
    <option value = "drop"> drop </option>
    <option value = "explode"> explode </option>
    <option value = "fold"> fold </option>
    <option value = "highlight"> highlight </option>
    <option value = "pulse"> pulse </option>
    <option value = "scale"> scale </option>
    <option value = "shake"> shake </option>
    <option value = "slide"> slide </option>
</Select>

```

```
<input type="button" value="click" onClick="fun1()"><br/>

```

### \* \$.get():

By using this `fD`, we can execute a Webpage <sup>from server</sup>, without submitting a current page. It contains 3 Arguments.

1) Server side script name

2) Query String collect<sup>D</sup>

3) Function Def<sup>D</sup> what u want to Execute when the server side script execu<sup>D</sup> is completed.

It is using `get()` method to x'ferr the data.

### ~~\*\*~~ \$.post():

It is same as `$.get`, But it is using post method to x'ferr the data Bet<sup>D</sup> Browser and Server.

```
<script src="jquery-1.8.js"></script>
```

```
<script>
```

```
function fun1()
```

```
{
```

```
$.get ("page2.php", {qs : 'scott', qsl : 'smith'}, function (x){alert(x)})}
```

```
<input type="button" value="Click" onClick="fun1()"><br/>
```

```

```

```
<?php save:page2.php
```

```
$un = $_REQUEST ['qs'];
```

```
$sq = $_REQUEST ['qsl'];
```

```
echo $un, $sq;
```

```
?>
```

### Ex:

28. SEPT. 2012

```
<script src="jquery-1.8.js"></script>
```

```
<script>
```

```
function fun1(pname)
```

```
{
```

```
$.get (pname, {}), function(data){document.getElementById ('div1').innerHTML = data})}
```

```
<table border='2' width="100%" height="100%">
```

Exapmle

```
<tr><td style='width:100' onmouseover="fun1('page1.html')">Page1.html <td rowspan="3"
```

```
valign="top"><div id="div1"> Hi </div>
```

```
</td><td onmouseover="fun1('page2.html')"> Page2.html
```

```
</td><td onmouseover="fun1('page3.html')"> Page3.html
```

```
</tr>
```

70

```
<h1 style='color:green'> Page1 </h1>
<img src='file:///e:/15.jpg' width='200'>
<hr style='border-top: 2px solid black;'>
```

```
<h1 style='color:red'> Page2 </h1>
<img src='>

<h1 style='color:yellow'> Page3 </h1>
<img src='>
```

```
<h1 style='color:pink'> Page4 </h1>
<img src='>
```

```
<h1 style='color:purple'> Page5 </h1>
<img src='>
```

- \* jquery is providing jquery.ui.tabs.js library used to display tab control on the webpage, requires some supporting libraries those are jquery.ui.core.js, jquery.ui.widget.js and jquery-1.6.2.js, it requires the css file i.e. jquery.ui.all.css.
- \* To display the tab control, first we need to take a container like div, span, etc. Tabs we can display by using list items.

```
<link rel="stylesheet" href="jquery.ui.all.css">
<script src="jquery-1.6.2.js"></script>
<script src="jquery.ui.core.js"></script>
<script src="jquery.ui.widget.js"></script>
<script src="jquery.ui.tabs.js"></script>
<script>
$(function(){$("#tabs").tabs();}),
</script>
<body>
<div id="tabs">
<ul>
<li><a href="#tabs-1">Content</a></li>
<li><a href="#tabs-2">Content2</a></li>
<li><a href="#abc">MyPage</a>
</li>
</ul>
```

```

<div id="tabs-1">
  <h2> Content Heading 1 </h2>
  <p> This is a Paragraph </p>
</div>
<div id="tabs-2">
  <h2> Content Heading 2 </h2>
  <p> This is a Paragraph </p>
</div>
<div id="tabs-3">
  <h2> Content Heading 3 </h2>
  <p> This is a Paragraph </p>
</div>

```

### \* size :

By using this fn we can get the total no. of elements of a type.

```

<script src="jquery-1.js"></script>
<script>
  function func()
  {
    alert($("#div").size())
  }
</script>
<div> Hi </div>
<div> Nilesh </div>
<input type="button" value="click" onclick="func()">

```

### \* html () :

By using this fn, We can display some html content on browser container.

Ex:

```

<script src="jquery-1.js"></script>
<script>
  function func()
  {
    $("#div1").html("<img src='file:///e:/3.jpg' width='100'>")
  }
</script>
<div id="div1"> Hi </div>
<input type="button" value="click" onclick="func()">

```

## \* Append():

By using this fn we can append some text on html elements.

```
$("#div1").append("<img src='file:///e:/3.jpg' width='200'>")
```

## \* val():

To get and set the value of input control.

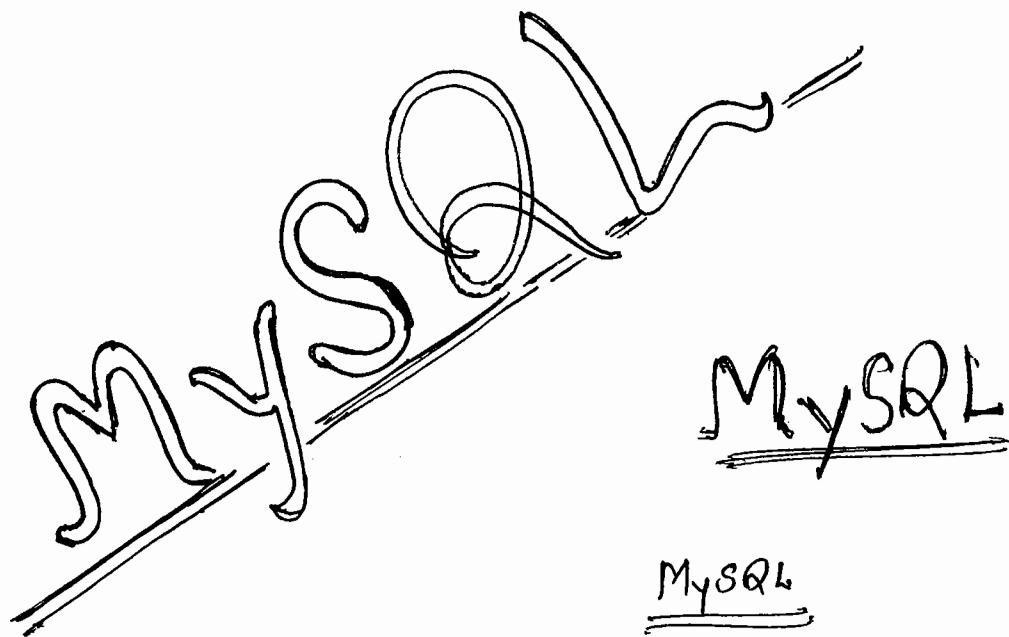
Ex: <script src="jquery-1.js"></script>

```
<script>
function fun1()
{
    alert($("#t1").val())
    $("#t1").val("scott")
}
</script>
<input type="text" id="t1">
<input type="button" value="click" />
```

## \* css():

By using this property we can apply styles to the elements.

```
$("#t1").css("color", "green")
```



Path: `http://localhost/phpmyadmin`

My son

# \* MySQL \*

28. SEPT. 2019 (75)

- \* PHP is providing `php_mysql.dll` library with no. of functionalities to connect with MySQL database.
- \* MySQL is an open source relational Database management syst. supporting number of objects like Tables, views, etc.
- \* MySQL is implemented by MySQL AB Corporation in 1995. Now, it is maintaining by Oracle Corporation.
- \* The default username of mysql db is "root" and it does not contain any "password".

~~MySQL administration~~

02. OCT. 2019

## \* Database :

A Database is a collect<sup>n</sup> of database objects. Database we r using to maintain tables, stored procedures, etc.

## \* Table:

Table is a collect<sup>n</sup> of Rows and columns.

## \* MySQL Interaction With Command Prompt:

We can connect with MySQL database

through a command prompt by Executing `mysql.exe` file what is available in XAMPP/MySQL/Bin Folder. `C:/xampp/mysql/bin>mysql.exe -u root`  
`C:/xampp/mysql/bin>mysql -u root & create database dbnew;`  
`mysql> Create database dbnew;` `mysql> Use dbnew;`  
`mysql> Use dbnew;`

Columns : fields.

Rows : Records.

## \* phpMyAdmin:

It is a Graphical User Interface used to connect with MySQL database. It is available with XAMPP download. The URL address to open phpMyAdmin is `http://localhost/phpmyadmin`.

### 1) Insert:

By using this option we can insert records in Table.

### 2) Browse:

By using this option, we can Browse the Table Records.

### 3) Structure:

To change the structure of a Table, If u want to add new ~~fields~~ fields, delete existing fields, Add constraints, delete constraints and to rename columns

### 4) SQL:

By using this option, we can execute our SQL statements.

### 5) Search:

By using this option, we can search the Records in Ascending or Descending order based on ~~any~~ columns (columnnames)

### 6) Operations:

By using this option, we can change the Tablename, To move Table into another database, to change the table engine etc.

### 7) Export:

By using this option we can Export database Tables as sql, pdf, etc. file formats.

### 8) Import :

By using this option, we can Import the exported file.

### 9) Empty:

By using this option, we can delete the table Records.

### 10) Drop:

By using this option, we can ~~drop the table structure~~ delete the Table Structure.

## \* MySQL Interaction with PHP:

PHP 5.0 is providing php-mysql.dll library to connect with mysql database. This library is providing more functions.

### 1) mysql\_connect:

By using this fn, we can create a conn' Bet' php prog. and mysql db. It contains 3 Arguments. i) ServerName, ii) Username, And iii) Password.

### 2) mysql\_select\_db :

To select a db from mysql server, arguments are database name And connection id.

### 3) mysql\_query:

To execute a sql query in my\_sql database. Arguments are sql statement And connection id.

### 4) mysql\_error:

To Get the Error Message if Any occurred at the time of mysql stat. Execution.

### 5) mysql\_errno:

To Get the Error number if any occurred at the time of mysql stat. Execut'.

\* Program: To create a conn' Bet' MySQL and PHP.

Ex: <?php if(\$con=mysql\_connect("localhost", "root", "")) {

    echo "Connected"; } else {

    echo mysql\_error(); }

?>

Prog. To add New Record

Ex: <?php

    mysql\_connect ("localhost", "root", "");

    mysql\_select\_db ("db\_barepm");

    if(mysql\_

file, document, file, Excel, etc.

sql

1

delete the Table Structure

~~drop the Table Structure~~

PHP 5.0 is providing php-mysql.dll library

to connect with mysql database. This library is providing more functions.

### 1) mysql\_connect:

By using this fn, we can create a conn' Bet' php prog. and mysql db. It contains 3 Arguments. i) ServerName, ii) Username, And iii) Password.

### 2) mysql\_select\_db :

To select a db from mysql server, arguments are database name And connection id.

### 3) mysql\_query:

To execute a sql query in my\_sql database. Arguments are sql statement And connection id.

### 4) mysql\_error:

To Get the Error Message if Any occurred at the time of mysql stat. Execution.

### 5) mysql\_errno:

To Get the Error number if any occurred at the time of mysql stat. Execut'.

\* Program: To create a conn' Bet' MySQL and PHP.

Ex: <?php if(\$con=mysql\_connect("localhost", "root", "")) {

    echo "Connected"; } else {

    echo mysql\_error(); }

?>

03.OCT.2014

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

    echo mysql\_error(); }

?>

    echo mysql\_error(); } else {

Ex: <?php  
 \$con=mysql\_connect("localhost","root","");
 if(mysql\_query("create database db-new",\$con))
 echo "Database is created";
 else
 echo mysql\_error();
 ?>

SAVE AS: DatabaseCreated.php (77)

↓ O/P : Database is created  
Connect id ON Double Click (Refresh)  
O/P :

Ex: <?php Prog. to create a Table in a Database

```
$con=mysql_connect("localhost","root","");
mysql_select_db("db-new",$con);
if(mysql_query("Create Table tb-new(sno int)",$con))
    echo "Created";
else
    echo mysql_error();
?>
```

SAVE AS: TableCreated.php  
O/P : Created  
 ON Double click (Refresh)  
 Table 'tb-new' already exists.

## \* Datatypes Available in MySQL Server:

Basically, we have 3 types of Datatypes

- 1) Text Types
- 2) Number Types
- 3) Date Types

i) TextTypes: occupies ~~too~~ bytes

i) CHAR: By using this type, we can store ~~a~~ fixed set of characters from 0 to 255 characters long.

ii) VARCHAR: By using this type, we can store a variable set from 0 to 255 characters long.

iii) TINYTEXT: A string with a max<sup>m</sup> length of 255 characters.

iv) TEXT: A string with a max<sup>m</sup> length of 65535 characters.

v) BLOB: A string with a max<sup>m</sup> length of 65535 characters. (same as TEXT)  
 But can also store, Binary Data.

vi) MEDIUMTEXT: A string with a max<sup>m</sup> length of 16777215 characters.

vii) MEDIUMBLOB: It is same as ~~tiny~~ MEDIUMTEXT. But, it can also store Binary Data.

viii) LONGTEXT: String with a max<sup>m</sup> length of 4294967295 characters.

ix) LONGBLOB: Same as ~~tiny~~ LONGTEXT. We can also store Binary Data.

## II) Number Types:

### i) TINYINT:

Store the signed values from -128 to 127, and 0 to 255 Unsigned values.

### ii) SMALLINT:

Signed: -32768 to 32767 &  
Unsigned: 0 to 65535.

### iii) MEDIUMINT:

Signed: -8388608 to 8388607  
Unsigned: 0 to 16777215.

### iv) INT:

Signed: -2147483648 to 2147483647  
Unsigned: 0 to 4294967295.

### v) BIGINT:

Signed: -9223372036854775808  
To 9223372036854775807  
(9.2e+17)

### vi) FLOAT:

A small no. with a floating Decimal pt.

### vii) DOUBLE:

A Large no. with a floating Decimal pt.

### viii) DECIMAL:

A Double stored as a string, allowing for a fixed decimal point.

## III) Date

1) DATE YYYY-MM-DD

2) DATETIME YYYY-MM-DD HH:MM:SS

3) TIMESTAMP YYYYMMDDHHMMSS

4) TIME HH:MM:SS  
f: www.googleAnalytics.com To see website ranking

### \* Function Supported By MySQL:

#### 1) Abs:

By using this function we can get the absolute value of \$ip value.

(Converts -value to +value & +value to -value)

Ex: select abs(-10) Opp: 10

#### 2) Sin:

To get the sin value of \$ip value

Ex: select sin(0) Opp: 0

Ex: select sin(90) Opp: 0.89399666360056

3) Cos: To get the cosine value. (78)

4) Tan: To get the Tangent value

8) TRIM:  
By using this f we can delete the LHS & RHS spaces of a string.

9) LTRIM:  
To delete LHS spaces of a string.

10) RTRIM:  
To delete RHS spaces of a string.

Ex: Select concat(trim("Scott"), "smith")  
Opp: Scottsmith

11) SQRT:  
To get the square Root of \$ip value

Ex: Select sqrt(16) Opp: 4

Date: 04 Oct 2012 ASCII!  
By using this f, we can get the ASCII value of \$ip value.

Ex: Select ascii('a')

\* Count: To get total no. of records

13) CHAR:  
Ex: Select count(empid) from emp  
To get the character of an ASCII value.

Ex: Select char(98) Opp: b

14) MOD:  
To Get the Remainder of \$ip value

Ex: Select mod(10,5) Opp: 0

#### 15) ADDDATE:

With By using this, we can add some dates to \$ip date.

Ex: Select ADDDATE('2012-01-02', INTERVAL 31 DAY);

#### 16) CURDATE, CURRENT\_DATE:

By using this we can get the current date inform?

17) CURRENT\_TIME(): Opp: 2012-10-15

To get the current time inform?

18) CURRENT\_TIMESTAMP(): Opp: 19:24:38

It is combination of current\_date & current\_time used to display Both Date & Time Inform?

Ex: i) Select curdate()

ii) Select curtime()

iii) Select current\_timestamp()

Opp: 2012-10-15

Opp: 19:24:38

03.OCT.2013  
5) round:

### 27) MDS (Message Digest 5)

By using this f<sup>n</sup>, we can convert the fip value as 32 characters length alphanumeric string. Ex: Select MDS("scott") 14 AUGUST 2012 03.OCT.2013

Rounds

~~Ex~~ a floating point number to its nearest integer value.

Ex: Select round(10.40) O/P: 10 \* now(): some as current timestamp

Select round(10.90) O/P: 11

Select round(10.50) O/P: 11

By using this f<sup>n</sup> rounds a floating point number to its nearest lowest integer value.

Ex: Select floor(10.99) O/P: 10

### 7) ceil, ceiling:

Rounds a floating point number to its nearest highest integer value.

Ex: Select ceil(10.40)

O/P: 11

### \* sum:

To get all records of specified column.

### 19) rand:

Returns a random number bet<sup>n</sup> 0 and 1. Ex: select sum(empid) from emp

Ex: select rand

O/P: 0.58637....

### \* pow:

To get the power of fip value

### 20) min:

To get minimum value from group of values.

Ex: select min(sno) from tab-user

Ex: select pow(2,2)

### 21) max:

select max(empid) from emp

O/P: 4

To get max<sup>n</sup> value from group of values. Ex: select max(empid) from emp

### 22) concat: → Join two or more than two string values.

Ex: select concat(cname, username) from fos-user O/P:

Ex: Select concat(uname, pwd) from new-tab O/P:

Ex: Select concat("scott", "smith", "john"). O/P: scottsmithjohn

### 23) CHAR\_LENGTH, CHARACTER\_LENGTH:

By using this f<sup>n</sup>, we can get the total number of characters of a string.

Ex: i) select char\_length(uname) from new-tab O/P:

ii) SELECT CHARACTER\_LENGTH("NILESH") O/P:

### 24) Reverse:

Display the string in Reverse direction.

Ex: select reverse("scott") O/P: tcoes

### 25) Substring:

We get the substring of a string contains 2 arguments.

First character number & second is total number of character.

Ex: Select substr("scott", 2, 3) O/P: cot

Select substr("welcome", 4, 4) O/P: Come

26) CRC32 (Cyclic Redundancy check): By using this f<sup>n</sup>, we can convert the fip String as 32 bit Encrypted Numeric Value.

Ex: Select crc32("scott") O/P:

Program: 1

```
<? PHP
if ($con = mysql_connect("localhost", "root", ""))
    echo "Connected";
else
    echo mysql_error();
echo $con;
?>
```

```
<?php
if(isset($_POST['submit']))
{
    $db_name = $_POST['db'];
    mysql_connect("localhost", "root", "");
    mysql_select_db($db_name);
    if(mysql_query("create database $db_name"))
        echo "Created";
    else echo mysql_error();
}
```

Program: 2

```
<? PHP
$con = mysql_connect ("localhost", "root", "");
if (mysql_select_db ("test", $con))
    echo "Selected";
else
    echo mysql_error();
?>
```

Op: DBName:   
Create

Program: 3

```
<? PHP
$con = mysql_connect ("localhost", "root", "");
if (mysql_query ("create database dbl", $con))
    echo "DB created";
else
    echo mysql_error();
?>
```

Program: 4

```
<? PHP
//mysql_connect("localhost", "root", "");
if(mysql_query("create database dbl", $con))
    mysql_select_db("test");
mysql_query("create table tb1_newtab (uname varchar(100), pwd varchar(100))");
$sqlstt = "insert tb1_newtab values('scott09','scott123')";
mysql_query($sqlstt);
```

```
<?php
$con = mysql_connect ("localhost", "root", "");
mysql_select_db ("test");
mysql_query ("create table tb2_batch (uname varchar(100), sno int)", $con);
$sqlstt = "insert tb2_batch values ('Ni1', '101')";
mysql_query($sqlstt);
echo "Row is added";
```

mysql\_fetch\_row:

By using this fn we can fetch a record from resultset and returns the o/p as Numeric Array. At the same time it locates the resultant pointer at the begining of Next Record.

```

<table border='0' cellpadding='0' cellspacing='0'>
<tr>
<th style='background-color: lightyellow; color: red'>sno</th>
<th style='background-color: lightyellow; color: red'>password</th>
<?php
    $no=1;
    mysql_connect("localhost", "root", " ");
    mysql_select_db("test");
    $data=mysql_query("select * from new_tab");
    while($rec=mysql_fetch_row($data))
    {
        if($no%2 == 0)
            $color="lightblue";
        else
            $color="lightgreen";
        $no++;
        echo '<tr> $rec[0]<td> $rec[1]<td> $rec[2]</td>';
    }
?>
```

id: satyagam.php@gmail.com  
 pwid: phppassword  
 id: stbatch11@gmail.com  
 pwid: stbatch

Run -

<http://localhost/xyz.php>

rahulpanddy.stn@gmail.com

Father: Ramdars Singh  
 Entry Singh

8109332193 Mob No:

C/o Sona Devi

Nayee Dadhi

Karvate, Chandauli, UP.

232104

DOB: 06. April. 1989

provesh26@gmail.com

Database Engines in MySQL: Database Engine supported by MySQL internal process  
A DB engine is an underlying program using to run the SQL statements in DB languages. Diff. types of DB engines are available in MySQL

## \* Different Types of DB Engines

### i) myisam:

It is a default storage engine supports MySQL. Every table by default creates with this engine.

If we create any table with myisam engine, it creates 3 files to maintain the table information? The filenames are same as table name contains diff. types of extensions.

### ii) .frm:

This extension file maintains table format.

### iii) .myd:

It maintains Table Data.

### iv) .myi:

It maintains Table Index.

All these files are available in data folder of MySQL folder.

~~SQL stat. to create table with myisam engine~~

→ Create Table HBL-new1(sno int) engine=MyISAM

Create Table <tablename> engine=MyISAM

MyISAM stores the information in Low Level Binary format. That's why we can use this tables with any operating system.

To see the DB engines available with MySQL we need to execute Show Engines statement: Show engines

in form low level binary format that's why it is compatible with any type of OS.

MyISAM can store max<sup>n</sup> limit upto 4GB data. It supports max<sup>n</sup> dataspace and also it executes select statement very fast.

### \* Drawbacks of MyISAM:

i) This engine does not support relationships b/w the tables that's why we don't use this engine with complex enterprise level applications.

ii) We can not get the data back easily if database server is crashed.

iii) It is faster at the time of retrieving data. But it is slow while inserting & updating data.

### \* INNODB:

i) It is transaction safe engine supports the relationships b/w the tables.

ii) These tables we can use to develop complex enterprise level application.

iii) This engine supports the data recovery option, if the database server is crashed.

iv) It is very faster at the time of inserting & updating the records.

v) We can easily retrieve the data back, if db server is crashed.

### \* SQL stat. to create table with innodb engine:

Create Table HBL-new1<tablename>(col1, col2, ...) engine=innodb

Create Table HBL-new1(sno int) engine=innodb

### 3) MEMORY (heap):

This Engine can stores the data in Memory locat<sup>n</sup> of database server. It is 30% more faster than MyISAM Engine. But the Data is deleted when we stop the Database server.

### 4) BERKELEY: ~~Memory table~~

By using this Engine we can maintain the Relationships Bet<sup>n</sup> the Tables. It is same as innodb But this Engine Occupies more memory locat<sup>n</sup> and we cannot move this Tables easily from one locat<sup>n</sup> to Another locat<sup>n</sup>. That's why we don't use these engines in applicat<sup>n</sup>s.

### ~~constraint supported file MySQL DB~~

#### \* Constraints Supported file MySQL DB:

05.OCT.2019

Constraints are nothing but the cond<sup>n</sup> what we can apply on Database Columns.

MySQL is supporting different types of Constraints

##### 1) NOT NULL:

By using this constraint, we can Restrict the Null values in Database columns. This Constraint doesn't allow Null-values.

Ex: Create Table tbl\_new(sno int not null)

##### 2) Unique:

By using this constraint, we can Restrict the duplicate values in DB columns.

Ex: Create Table tbl\_new4(sno int unique)      Ex: Create Table tbl\_new11(sno int unique, uname varchar(100))

##### 3) primary key:

By using this constraint, we can Restrict a column with Null & Duplicate values. It is a combinat<sup>n</sup> of not Null & Unique.

Ex: Create Table tbl\_new4(sno int primary key)

##### 4) auto\_increment:

By using this constraint we can increase the column value with a new Record without passing any explicit value. The auto increment column should be primary key column. Specifying variable

Ex: Create Table ~~tbl~~\_new5(sno int primary key auto\_incremet, uname varchar(100))

##### 5) default:

By using this constraint, we can set the default value to the Table column.

Ex: Create Table ~~tbl~~\_new7(sno int default 100, uname varchar(100) default 'scott')

##### 6) foreign key:

By using this constraint, we can refer a table column with another table primary key column. (Refers values of another column from primary key column.)

To create the Relationship Bet<sup>n</sup> primary key & foreign key column, The Table should be innodb ~~table~~.

Ex: create table tbl-parent (pid int primary key, pname varchar(100)) engine=innodb  
 create table tbl-child (cid int, cname varchar(100), pid int, foreign key (pid) references tbl-parent(pid) engine=innodb

(84)

tbl-parent		tbl-child	
pid	pname	cid	cname
1	-		
2	-		

\* { Unique key allows only one Null value }

### SQL Statements:

\* SQL statements supported by MySQL Database.

\* SQL statements are divided into 4 different types.

- 1) DML (Data Manipulation Language)
- 2) DCL (Data Control Language)
- 3) TCL (Transact Control Language)
- 4) DDL (Data Definition Language)

### I) DML Statements:

By using this statement, we can manage the data of Database objects

#### a) SELECT :

By using this statement, we can select Table Records.

Ex: select \* from tbl-user → '\*' means all columns

i) select uname, pwd from tbl-user

ii) select \* from tbl-user where uname='scott'

iii) select \* from tbl-user where uname like 's\_'

That 2 underscore means it shows O/P Starting with S & 2 underscores after 's'!

iv) select \* from tbl-user limit 0,2

v) select \* from tbl-user where uname like 'sy%'

#### b) INSERT :

By using this statement, we can insert Records in Table.

Ex:

i) insert tbl-user values (123, 'abc', 'abc')

ii) insert tbl-user (sno) values (1234) → insert into tbl-user (uname) values ('smith')

iii) insert into tbl-user values (123, 'abc', 'abc'), (456, 'xyz', 'xyz')

iv) insert into tbl-user values (100, 'scott'), (200, 'smith')

L8.php

```
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$sqlstmt = "insert into tbl-user values (333, 'x', 'y')";
mysql_query($sqlstmt);
echo "Row Added";
?>
```

### MySQL\_Fetch\_Row():

This fn fetches a Record from Resultset returns as an Array.  
 Total number of Array Elements are equal to the number of table columns.  
 This fn returns the arr as Numeric Array.

Key starts from "0" and Ends with

Total no. of Elements values are the column values of the Records.

<?php

```
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data = mysql_query("select * from table_user");
while($row = mysql_fetch_row($data))
```

=x=

<style>

```
th {
  background-color: lightblue;
  color: red;
}
```

</style>

```
<table border='2'>
<th> Sno </th>
<th> Uname </th>
<th> Pwd </th>
```

<?php

```
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data = mysql_query("select * fromtbl_user");
$count = 1;
```

while (\$row = mysql\_fetch\_row(\$data))

{

```
if ($count % 2 == 0)
  $color = "lightyellow";
else
```

```
$color = "lightgreen";
$count++;
```

// print\_r(\$row);

```
echo "<tr style='background-color: $color'><td>$row[0]</td><td>$row[1]</td><td>$row[2]</td>";
```

}

}

86

```

<?php
if(isset($_POST['sub']))
{
    $sno=$_POST['sno'];
    $uname=$_POST['un'];
    $pwd=$_POST['pwd'];
    mysql_connect("localhost", "root", "");
    mysql_select_db("test");
    if(mysql_query("insert into new
                    values('$sno','$uname','$pwd')"))
        echo "Row is inserted";
    else
        echo mysql_error();
}

```

<form method="post" action=" " >

sno:<input name="sno"> <br> Username:<input name="un"> <br>
<input type="text" name="pwd" value="insert">

</form>

  
**Ex:**  
<?php  
mysql\_connect("localhost", "root", "");  
mysql\_select\_db("test");  
\$data=mysql\_query("select \* from  
tbl-user");  
echo mysql\_num\_fields(\$data);

\* mysql\_num\_fields (Resultset) ~~function~~

→ By using this f<sup>n</sup> we can get the total no. of Resultset fields. ~~function~~

\* mysql\_field\_name ~~(resultset, fieldIndexNumber)~~:

By using this f<sup>n</sup> we can get the name of a field from a Resultset

The Arguments are resultset, fieldIndexNumber

Ex: <table border="2">

```

<?php
if (isset($_POST['sub']))
{
    $tname=$_POST['t1'];
    mysql_connect("localhost", "root", "");
    mysql_select_db("test");
    $data=mysql_query("select * from $tname");
    $fc=mysql_num_fields($data);
    for ($i=0; $i<$fc; $i++)
    {
        $fname=mysql_field_name($data, $i);
        echo "<th>$fname</th>";
    }
    while ($rec=mysql_fetch_row($data))
    {
        echo "<tr>";
        for ($i=0; $i<$fc; $i++)
        {
            echo "<td>$rec[$i]</td>";
        }
    }
}

```

</table>  
<form method="post" action=" " >  
<Table name:<input name="t1"> <br>
<input type="Submit" name="sub"
value="Get Records">  
</form>

Ex: <?php  
mysql\_connect("localhost", "root", "");  
mysql\_select\_db("test");  
\$data=mysql\_query("select \* from  
tbl-user");  
echo mysql\_field\_name(\$data, 2);

### \* mysql\_num\_rows()

By using this f<sup>n</sup>, we can get the Total no. of Records from mysql resultset. Contains 1 arguments name of Resultset and mysql resultset.

```
<?php
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data = mysql_query("select * from tbl-user");
echo mysql_num_rows($data);
?>
```

### \* mysql\_fetch\_assoc():

By using this f<sup>n</sup>, We can fetch the records from Resultset & Returns the

Opf as an Associative Array. Array keys are columnnames, Array values are column values.

Ex:

```
=<?php
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data = mysql_query("Select * from tbl-user");
$rec = mysql_fetch_assoc($data);
//print_r($rec);
echo $rec['uname'];
?>
```

### \* mysql\_fetch\_array():

To read the records from resultset & returns the Opf as both numeric Array & associative array. It is a combination of mysql\_fetch\_row(), mysql\_fetch\_assoc(), ~~mysql\_fetch\_array()~~

### \* mysql\_fetch\_object():

<sup>8th</sup> Reads a Record from Resultset & Returns the Opf as an object

This object belongs to a Base class i.e. ~~stdClass~~. StdClass.

Ex: <?php

```
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data = mysql_query("select * from tbl-user");
$rec = mysql_fetch_object($data);
print_r($rec);
echo $rec->uname;
?>
```

Ex: \$data = mysql\_query("select \* from tbl-user");
\$rec = mysql\_fetch\_array(\$data);
print\_r(\$rec);

## \* mysql\_fetch\_field():

By using this f<sup>n</sup>, we can fetch a field from Resultset  
and returns the O/P ~~as an object~~. (To fetch complete inform<sup>n</sup> of a field as  
Array.

Ex: <?php

```
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$field=mysql_fetch_field($data);
print_r($field);
//field mysql_fetch_field($data);
Unacademy
```

```
<?php
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data=mysql_query("select * from
tbl_user");
print_r(mysql_fetch_field($data));
print_r(mysql_fetch_field($data));
?>
```

Note:

Other way of that  
prog. see

On last page on  
Dated 30. Oct. 2013

```
<table border='1'>
<?php
mysql_connect("localhost", "root", "");
mysql_select_db("information_schema");
if(isset($_REQUEST['qs']))
$index=$_REQUEST['qs'];
else
{
$index=0;
$data=mysql_query("select * from character_sets");
$_SESSION['tr']=mysql_num_rows($data);
}
$tabrec=$_SESSION['tr'];
$totrec=5;
$data=mysql_query("select * from character_sets limit $index, $totrec");
while ($rec=mysql_fetch_row($data))
{
echo "<tr><td>$rec[0]<td>$rec[1]<td>$rec[2]</td>";
}
echo "</table>";
$previousindex=$index-$totrec;
if ($index!=0)
{
echo "<a href='prog.php?qs=$previousindex'> previous </a>";
}
$nextindex=$index+$totrec;
if ($nextindex < $tabrec)
echo "<a href='prog.php?qs=$nextindex'> NEXT </a>";
```

delete:

By using this stat, we can delete a Record or set of Records from a Table.

Ex: delete from tbl-user where sno=100;

update:

To update the Records of a Table.

Ex: i) update tbl-user set uname='scott123' where sno=10

ii) update tbl-user set uname='scott123', pwd='abcd' where sno=10;

DDL statements:

By using these stat. we can define the structure of database Objects, To create, alter, drop etc. we can go for ~~these~~ DDL statements.

1) create:

By using this statement, we can create DB objects.

09.OCT.2013

Ex: i) create user scott@localhost

ii) create user smith@localhost identified by 'smith123'

Ex: 5) alter table tbl-new drop primary key  
Ex: 4) alter table tbl-new add constraint primary key(sno)

2) alter:

By using this statement, we can change the structure of a Table.

Ex: 1) alter table tbl-new add column uname varchar(100)

Ex: 2) alter table tbl-new add column address varchar(100) first

3) drop: By using this statement, we can drop db objects like Table, user, etc.

We can also drop the table columns

Ex: drop table tbl-new

4) truncate:

By using this statement, we can delete the all Records of Table.

Ex: truncate table tbl-new

5) rename:

By using this option, we can change the table name.

Ex: rename table tbl-new to abcd

mysq>list\_dbs();

This fn Returns list of databases as a Resultset.

Ex:

<select> → select tag is used for drop-down list.  
<?php

mysql\_connect("localhost", "root", "");

\$data=mysql\_list\_dbs();

while(\$db=mysql\_fetch\_row(\$data))

{  
echo "<option>\$db[0]</option>";

}

</select>

## 2) mysql\_list\_tables:

To get the list of Tables available in mysql database.

(9)

Ex: <select>

```
<?php
    mysql_connect("localhost", "root", " ");
    $data = mysql_list_tables("db_new");
    while ($table = mysql_fetch_row($data))
    {
        echo "<option>$table[0]</option>";
    }
}>
</select>
```

Ex: 2) <form method="post" action=" " >

```
Databases :<select name="drpdb">
<?php
    $txt = $_POST['drpdb'];
    mysql_connect("localhost", "root", " ");
    $data = mysql_list_dbs();
    while ($db = mysql_fetch_row($data))
    {
        if ($txt == $db[0])
            echo "<option selected>$db[0]</option>";
        else
            echo "<option>$db[0]</option>";
    }
}>
</select>
<input type="submit" name="sub" value="GetTables">
<?php
if (isset($_POST['sub']) || isset($_POST['sub2']))
{
    $pname = $_POST['drptable'];
    mysql_connect("localhost", "root", " ");
    $dbname = $_POST['drpdb'];
    $tables = mysql_list_tables($dbname);
    echo "<br>Tables :<select name='drptable'>";
    while ($rec = mysql_fetch_row($tables))
    {
        if ($pname == $rec[0])
            echo "<option selected>$rec[0]</option>";
    }
}
```

```

    else
        echo "<option> $rec[0] </option>";
    }
}

<input type="submit" value="Get Records" name="sub2" />

<?php
if (isset($_POST['sub2']))
{
    mysql_connect ("localhost", "root", " ");
    $dbname = $_POST['drpdb'];
    $tname = $_POST['drptable'];
    mysql_select_db ($dbname);
    $data = mysql_query ("select * from $tname");
    echo "<table border='2'>";
    $fc = mysql_num_fields ($data);
    for ($i=0; $i < $fc; $i++)
    {
        $fname = mysql_field_name ($data, $i);
        echo "<th>$fname";
    }
    while ($rec=mysql_fetch_row ($data))
    {
        echo "<tr>";
        for ($i=0; $i < $fc; $i++)
        {
            echo "<td>$rec[$i]</td>";
        }
    }
}
</form>

```

Ex:3

```

<?php
if (isset($_POST['sub1']))
{
    $tname = $_POST['txtable'];
    mysql_connect ("localhost", "root", " ");
    mysql_select_db ("test");
    $data = mysql_query ("select * from $tname");
    echo "<table border='2'>";
    $fc = mysql_num_fields ($data);
    for ($i=0; $i < $fc; $i++)
    {
        $fn = mysql_field_name ($data, $i);
        $fn = ucfirst ($fn);
        echo "<th>$fn </th>";
    }
    while ($rec=mysql_fetch_row ($data))
    {
        echo "<tr>";
        for ($i=0; $i < $fc; $i++)
        {
            echo "<td>$rec[$i]</td>";
        }
    }
}

```

```

<?><Form method="POST" action=" " >
<Table name:><input type="text" name="txtable">
<br/><input type="submit" value="Get Records" name="sub">
</form>

```

# A Script to Delete Multiple Records

```
<script>src="jquery-1.3.js"></script>
<style>
  .cls1 {
    background-color: lightyellow;
  }
</style>
<script>
  function applystyle(t)
  {
    t.style.backgroundColor = "lightyellow"
  }
  function removesyles(t,col)
  {
    t.style.backgroundColor = col
  }
  function chkfun(x)
  {
    cons = frm1.elements
    for(i=0, i < cons.length; i++)
    {
      if(cons[i].type == "checkbox")
        cons[i].checked = x
    }
  }
  function fun1(sno)
  {
    "MySQL"
    location = Del.php?ss=" + sno
  }
</script> "MySQL"
<form action="DelAll.php" method="post"
      id="frm1">
  <table border='2'>
  <%php
    mysql_connect("localhost", "root", " ");
    mysql_select_db("test");
    $dataq = mysql_query("select * from tb1");
    $count = 0;
    while($rec = mysql_fetch_row($dataq))
    {
      if(($count%2) == 0)
        $color = "lightblue";
      else
        $color = "silver";
      $count++;
  
```

SAVE AS: MySqlDelMulRec.php

11.OCT.2012

92

# is used for  
all the styling  
the same page.  
and within the same  
page.

```
echo "<tr style='background-color: $color'
  'onmouseover='applystyle(this)'
  onmouseout='removesyles(this,
    '$color')'><td><input type='checkbox'
    name='$rec[0]'>
  <td>$rec[0]<td>$rec[1]<td>$rec[2]
  <td><img src='buttonup.png' close.png'
    onclick='fun1($rec[0])'>";
}
</table>
<input type="submit" name="sub"
      value="DeleteAll">
</form>
<a href="#" onclick="chkfun(true)">
  Check All        &ampnbsp </a>
<a href="#" onclick="chkfun(false)">
  Uncheck All </a>
```

delall.php

```
<input type=" SAVE AS: MySqlDelAll.php
<%php
  mysql_connect("localhost", "root", " ");
  mysql_select_db("test");
  foreach($_POST as $k => $v)
  {
    mysql_query("delete from tb1
      where sno = $k");
  }
  header("location: MySqlDelMulRec.php");
  } // for go to previous page
```

```

<?php      SAVE AS: MySqlDel.php    del.php
$x=$_REQUEST['sn'];
mysql_connect("localhost", "root", "");
mysql_select_db("test");
mysql_query("delete from Hb2_City where sno=$x");
header("location: MySqlDel.php");
```

? ?

## \* program to delete multiple Records.

Ex: <script>

```

function funchk(yo2)
{
arr=document.getElementById('frm1').elements
for(i=0; i<arr.length; i++)
{
if(arr[i].type="checkbox")
{
arr[i].checked=yo2
}
}

Function fundel(sn)
{
location='del.php?' +sn
}

</script>
<Form method="post" action="delall.php" id="frm1">
<Table border='1'>
<?php
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data=mysql_query("select * from Hb2_city");
while($rec=mysql_fetch_row($data))
{
echo "<tr><td>input type='checkbox' name=$rec[0]>
      <td>$rec[1] <td>$rec[2] <td>
      <img src='b_drop.jpg' onclick='fundel($rec[0])'>";
}
?>
</Table>
<input type="Submit" value="Delete All">
</form>
<a href="#" onclick='funchk(true)'>Check All</a>
<a href="#" onclick='funchk(false)'>Uncheck All</a>
```

```

<?php
$q = $_SERVER['QUERY_STRING'];
mysql_connect ("localhost", "root", " ");
mysql_select_db ("test");
mysql_query ("delete from tb1_city where cid=$q");
header ("location : get.php");
?>

del all

<?php
mysql_connect ("localhost", "root", " ");
mysql_select_db ("test");
foreach($_POST as $k => $v)
{
    mysql_query ("delete from tb1_city where cid=$k");
}
header ("location : get.php");
?>

```

### \* mysql\_pconnect:

It is the same as mysql\_connect, it creates ~~Persistent~~ conn.

First, It checks whether any conn is created or not. If any conn is already created, it uses the conn. Otherwise, it creates a new conn. The conn will close when we stop the server.

Ex: <?php  
if(mysql\_pconnect ("localhost", "root", " "));  
echo 'Connected';  
else  
echo "Not Connected";  
?>

```

revoke insert on *.* from scott@localhost
revoke insert, select on *.* from scott@localhost
revoke all privileges on *.* from scott@localhost

```

\* <meta http-equiv="refresh" content="5" /> <body bgcolor="lightblue">  
 By using this, ~~will~~ Refresh same page after every 5 seconds automatically  
 means reloading after every 5 seconds.

~~Registration With Client-side Validation~~ ← Page AutoReload with Metatag.

\* Registration with Client-side Validation & Redirect to Another page when

Javascript is Disabled:

Reg.php

Step I

```

→ <noscript>
  <meta http-equiv="refresh" content="0; error.html">
</noscript>

<script>src = "jquery-1.3.js"></script>
<script src = "jquery.validate.js"></script>
<style>
  .cls1 {
    color: red;
  }
</style>
<script>
  function fun1()
  {
    $('#frm1').validate({errorClass:'cls1'})
  }
</script>
<body onload="fun1()">
<form method="post" action="regval.php" id=frm1>
<table>
<tr><td>Username <td><input name='txtuser' id="txtuser" class="required"
  onblur='chkuser(this.value)'><td> <span id="sp1"></span>
<tr><td> Password <td><input type="password" name='txtpwd' class="required"
  minlength="20" id="pwd">
<tr><td> Re-Password <td><input type="password" name='txtpwd' equal-to="#pwd">
<tr><td> Email <td> <input type="name" name='txtemail' class="required email">
<tr><td> Age <td> <input name='txtage' class="required number">
<tr><td> <input type="submit" name="sub" value="Register">
</table>
</form>

```

```

<script>
    function chkuser(val)
    {
        $("#sp1").html("<img src=spinner-grey.gif width=20>")
        $.post("validate.php", {qs:val}, function(xy)
        {
            if (xy==0)
            {
                $("#sp1").css ("color", "green");
                $("#sp1").html ("Username is Available")
            }
            else
            {
                $("#sp1").css ("color", "red");
                $("#txtuser").val(" ")
                $("#sp1").html ("username is Not Available")
            }
        })
    }
</script>

```

error.htm?

please enable javascript <a href="reg.php">Go to Reg</a>

validate.php

```

<?php
$qs = $_REQUEST['qs'];
mysql_connect ('localhost', "root", " ");
mysql_select_db ("test");
$data=mysql_query ("select * from tbl_user where uname='$qs' ");
sleep(3);
echo mysql_num_rows($data);
?>

```

## A SERVER-SIDE REGISTRATION

18.OCT.2019 (97)

Save: Regval.php

```

<?php
include "conn.php";
$uname=$_POST['txtnum'];
$pwd=$_POST['txtpwd'];
$pwd="abc". $pwd. "$%#"; } important
$pwd=crc32(md5($pwd)); } part
---connect();
if(mysql_query("insert into tbl-user values ('$uname', '$pwd')"))
{
echo "Account is Created";
}
else
{
if(mysql_errno()==1062)
echo "Username is already Registered";
else
echo mysql_error();
}
}

```

Reg.php

19.OCT.2019

```

<?php
if(isset($_POST['sub']))
{
$uname=$_POST['txtnum'];
$pwd=$_POST['txtpwd'];
$email=$_POST['txtemail'];
$pwd="abc". $pwd. "&#";
$pwd=sha1(md5($pwd));
mysql_connect("localhost", "root", " ");
mysql_select_db ("test");
if(mysql_query("insert into tbl_user values ('$uname', '$pwd', '$email', 0)"))
{
$randno=rand(1,1000);
$randno=$uname.$randno;
mysql_query("insert into tbl_activation values ('$uname', '$randno')");
$str=<<<abc
Registration completed successfully to active ur account click on

```

(98)

```
<a href='activate.php?rn=$random'> ACTIVATION LINK </a>
abc;
mail($email, "Activation Link", $str, "From: admin@mysite.com\r\nContent-type:
echo "Registration Completed";
}
?>
```

```
<form method="post" action=" " >
Username: <input type="text" name='txtnum'> <br>
password: <input type="text" name="txtpwd"> <br>
Email: <input type="text" name="txtemail"> <br>
<input type="submit" name="sub" value="Register">
</form>
```

Activate.php

```
<%php
$q = $_REQUEST['rn'];
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data = mysql_query("select uname from tbl_activation where acode ='$q'"),
$rec = mysql_fetch_row($data);
$uname = $rec[0];
mysql_query("Update tbl_user set active=1 where uname = '$uname' ");
echo "Activation completed successfully";
%>
```

Login.php

```
<script src="jquery-1.3.js"></script>
<script src="cookie.js"></script>
<script>
function readcookie(uname)
{
$("#txtn").val($.cookie(uname))
}
function fun()
{
if (document.getElementById('chk').checked)
{
un = $("#txtn").val()
pwd = $("#txtpwd").val()
```

```

    $Cookie(un, pw, {expires:10})
}

</script>
? <?php
? if(isset($_POST['sub']))
? {
?     $uname = $_POST['txtuser'];
?     $pwd = $_POST['txtpwd'];
?     $pwd = "abc". $pwd. "$";
?     $pwd = sha1(md5($pwd))
?
?     mysql_connect("localhost", "root", " ");
?
?     mysql_select_db("test");
?
?     $sqlstr = "select * from tbl_user where uname = '$uname' and pwd = '$pwd'";
?     $dataq = mysql_query($sqlstr);
?
?     if(mysql_num_rows($dataq) == 1)
?     {
?         $dataq1 = mysql_query("select * from tbl_user where uname = '$uname' and active = 1");
?         if(mysql_num_rows($dataq1) == 0)
?             echo "Activate Account";
?         else
?             echo "Welcome To User", Success
?     }
?     else
?         echo "Invalid User";
? }
?

<form method="post" action=" " >
    Username: <input name="txtuser" id="txt1" onblur="readcookie(this.value)"><br>
    Password: <input name="txtpwd" id="txt2" type="password"><br>
    <input type="checkbox" id="Chk1"> Remember Me <br>
    <input type="submit" name="sub" value="Login" onclick="fun1()"/></form>

```

```

<script>
    function funupdate(uname)
    {
        location='update.php?' + uname
    }
</script>





```

### Update.php

```

<?php
session_start();
$qs=$_SERVER['QUERY_STRING'];
$_SESSION['oun']=$qs;
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
$data = mysql_query("select * from tbl_user
                    where uname='".$qs."'");
$rec = mysql_fetch_row($data);
$str <<<abc
<form method="post" action="save.php">
abc;
for ($i=0; $i<count($rec); $i++)
{
    $fname=mysql_field_name($data,$i);
    $fname=ucfirst($fname);
    $str .= "$fname:<input type='text' name=$fname VALUE=$rec[$i]<br>";
}

```

20.OCT.2012 (100)

```

$str .= "<input type='submit' value='Save'>";
echo $str;
}

```

### Save.php

```

<?php
session_start();
$str="update tbl_user set";
$count=count($_POST);
$i=0;
foreach($_POST as $k=>$v)
{
    $i++;
    $str .= $k."=". "'$v'";
    if($i!=$count)
        $str .= ",";
}
$uname=$_SESSION['oun'];
$str .= "where uname='".$uname."'";
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
mysql_query($str);
header("location: get.php");
}

```

Up

Active: <input type="text" name="txtactive" value="\$rec[4]"> <br>  
<input type="Submit" name="sub" value="Save">  
</form>  
abc;  
echo \$str;  
?>

### Save.php

<?php  
session\_start();  
\$uname=\$\_POST['txtuser'],  
\$pwd=\$\_POST['txtpwd'],  
\$email=\$\_POST['txtemail'],  
\$age=\$\_POST['txtage'],  
\$active=\$\_POST['txtactive'],  
\$ou=\$\_SESSION['oun'],  
\$sqlstmt="update tbl-user set uname='\$uname',  
pwd='\$pwd', email='\$email', age='\$age',  
active=\$active where uname='\$ou';  
mysql\_connect("localhost", "root", "");  
mysql\_select\_db("test");  
mysql\_query(\$sqlstmt);  
header("location:get.php");  
?>

### Update.php

<?php  
session\_start();  
\$qs=\$\_SERVER['QUERY\_STRING'];  
\$\_SESSION['oun']=\$qs;  
mysql\_connect("localhost", "root", "");  
mysql\_select\_db("test"),  
\$data=mysql\_query("select \* from tbl-user  
where uname='\$qs'"),  
\$rec=mysql\_fetch\_row(\$data),  
\$str<<< abc  
<form method="post" action="Save.php">  
<Username:> <input type="text" name="txtuser"  
value="\$rec[0]"> <br>  
password:> <input type="text" name="txtpwd"  
value="\$rec[1]"> <br>  
Email:> <input type="text" name="txtemail"  
value="\$rec[2]"> <br>  
Age:> <input type="text" name="txtage"  
value="\$rec[3]"> <br>

## \* mysql\_close()

By using this f<sup>n</sup>, we can close the opened database conn<sup>n</sup>.

```
<?php
mysql_connect("localhost", "root", "");
mysql_close();
mysql_select_db("test");
?>
```

## \* Joins:

By using joins we can get records from multiple tables based on cond's.

Joins are mainly divided into 3 types.

i) Inner Join 2) Outer Join 3) Cross or ~~Cartesian~~ Cartesian Join.

## 1) Inner Join:

Inner Join is again divided into 3 types.

a) Equi-join, b) Non Equi Join c) Self Join.

### a) Equi Join:

If u want the records from a particular Table, based on equal cond<sup>n</sup>, comes under Equi-Join.

Ex: i) select emp.ename, dept.dname from emp, dept where emp.deptno = dept.deptno

OR

ii) select emp.ename, dept.dname from emp inner join dept on emp.deptno = dept.deptno

OR

iii) select e.ename, d.dname from emp e, dept d where e.deptno = d.deptno

Using innerjoin (Speed more Execut<sup>n</sup>)

↳ Using Alias name

### b) Non-Equi Join:

If u want to get the records from multiple Tables without Equal Cond<sup>n</sup>, we can go for Non-Equi-Join.

Ex: Select e.name, s.grade from emp e, salgrade s where e.sal between s.lowsal and s.hightsal

OR

Select e.ename, s.grade from emp e inner join salgrade s on e.sal between s.loowsal and s.hightsal.

### 3) Self Join:

If u want to get records from single table using cond's, we can

go for self join.

Ex: select e.ename as Employee, m.ename as Manager from emp e, emp m where e.mgr = m.empid

Select distinct m.ename as Manager from emp e, emp m where e.mgr=m.empid (103)

OR  
Select m.ename as Manager from emp e, where e.mgr=m.empid.

\* Select e.ename, tbl-city.cname, dept.dname.

Select e.ename, c cname, d dname from emp e, tbl-city c ,dept d  
where e.deptno = d.deptno and e.cityid = c.cid.

OR

~~Select e.ename, c cname, d dname from emp e join dept d on  
e.deptno = d.deptno join tbl-city c on e.cityid = c.cid~~

## II) Outer Join:

To get the matched Records from one Table matching and Unmatched Records from another Table, It is divided into 3 types

### a) Left Outer Join:

To get matching and Unmatching records from LHS Table, Matching Records from RHS.

Ex: Select e.ename, d.dname from emp e left outer join dept d on e.deptno = d.deptno.

### b) Right Outer Join:

To get the matching Records from LHS, Matching & Unmatching Records from RHS.

Ex: Select e.ename, d.dname from emp e right outer join dept d on e.deptno = d.deptno

### c) Fuzz Outer Join:

It is a combination of Left Outer Join & Right Outer Join using an Union keyword, We can combine these 2 Joins.

Ex: Select e.ename, d.dname from emp e left outer join dept d on e.deptno = d.deptno union select e.ename, d.dname from emp e right outer join dept d on e.deptno = d.deptno.

III>

### Cross OR Cartesian Join:

25.OCT.2019 (104)

This join we're using to get the Records from multiple tables without any cond?

Ex: Select \* from emp, dept

### \* View:

View is a Virtual Table of Physical Table. Views we're using to call the common columns no. of times from a Table.

Views we're using to call the columns from a Table. Instead of using multiple calls of specified columns from a Table we can go for a view.

View doesn't occupy memory. It contains Select statement for every call of view, gets the Records from a Table.

#### Types of Views:

##### 1) Simple View:

If we create any view by getting the records from single table comes under simple view.

Ex: Create view vw\_view1 as select empid, ename, sal from emp

Create view vw\_view1 as select empid as 'Employee ID', ename as 'Employee

Name', sal from emp

##### 2) Complex View:

If we create any view by getting the records from multiple tables comes under complex views.

Ex:

Create view vw\_view3 as select e.ename, e.sal, e.cityid, d.dname from emp e, dept d where e.deptno = d.deptno

### \* Order By:

By using this ~~statement~~ class we can change the Order of table records. We can get the records in Ascending or Descending order by using this class.

Ex: Select \* from dept order by deptno asc

Select \* from dept order by deptno desc

X: By using php:

```
<table border='2'>
<?php
if(isset($_REQUEST['qs'])) {
    $ord=$_REQUEST['qs'];
} else
    $ord="desc";
if($ord=="asc")
    $ord="desc";
else
    $ord="asc";
```

```
mysql_connect("localhost", "root", "");
mysql_select_db ("test");
$data = mysql_query ("select * from dept order by
deptno $ord");
echo "<th><a href='Ord.php?qs=$ord'>DeptNo</a>";
echo "<th>DeptName";
while ($rec=mysql_fetch_row($data))
{
    echo "<tr><td>$rec[0]</td><td>$rec[1]</td>";
}
</table>
```

## \* ~~mysql~~-connect:

(105)

It is same as mysql library to connect with mysql database thru php program. But it is providing more functionalities. It requires connection id as first argument for every sql stat. Execut.

Ex:

```
<?php
$con = mysqli_connect("localhost", "root", "");
mysqli_select_db($con, "test");
$data = mysqli_query($con, "select * from dept");
while ($rec = mysqli_fetch_row($data))
{
    echo $rec[0];
    echo $rec[1];
}
?>
```

## \* Sql injection:

26. Oct. 2012

sql injection is the concept of passing the sql statements to the s/p controls. by the user. User can pass this sql statements to hack the information from a Database.

User can execute their own sql statements to get the inform<sup>n</sup> from db of server, Add the username and passwords, ~~Destroy~~ the DB inform<sup>n</sup>, etc.

User will use s/p controls, querystring, Etc. to Hack the inform<sup>n</sup> from DB.

~~User will use some special characters to pass their sql statement thru the s/p controls.~~

To provide security to the Applicat<sup>n</sup>, we need to follow these steps

Step1: Don't Accept lengthy stat. from user, thru the s/p controls .

Step2: Use JavaScript Validation to check whether user is passing invalid characters thru the s/p controls .

Step3: Used addslashes, put the backslashes in front of single & double quotes. We can also use mysql\_real\_escape\_string, to put the sql slashes in front of single and double quotes.

Step4: Use magic\_quotes\_gpc Configuration setting. It is used to put the backslashes in front of single and double quotes. The value should be "ON".

Configuration setting: magic\_quotes\_gpc=ON

```

<?php
if (isset($_POST['sub']))
{
    $uname=$_POST['txtuname'];
    $pwd=$_POST['txtpwd'];
    mysql_connect("localhost", "root", "");
    mysql_select_db("test");
    $uname=mysql_real_escape_string($uname);
    $pwd=mysql_real_escape_string($pwd);
    $sqlstt="select * from tb_user where
        uname='".$uname"' and pwd='".$pwd."'";
    echo $sqlstt."  
";
    $data=mysql_query($sqlstt);
    if(mysql_num_rows($data)==1)
    {
        echo "Valid";
    }
    else
        echo "Invalid";
}
<form method="post" action=" ">
    Username:<input type='text' name='txtuname'>
    <br/>
    password:<input type='password' name='txtpwd'>
    <br/>
    <input type='submit' name='sub' value='Login'>
</form>

```

### SAVE AS:

### O/P:

### \* mssql:

(106)

By using this library we can  
create a conn' Betw' php applicatn  
and sql server database.

This library is providing no.  
of fn's.

### 1) mssql\_connect():

This fn used to  
create a conn' Betw' php applicatn  
and sql server database.

### 2) mssql\_select\_db():

To select a DB  
From a sql server.

### 3) mssql\_query():

To execute an sql  
statement in mssql server.

```

<?php
if (mssql_connect("Welcome-pc", "sa",
    "123"))
    echo "Connected";
else
    echo "Not Connected";
mssql_select_db("master");
$data=mssql_query("select * from
    tb_user");
while ($rec=mssql_fetch_row($data))
{
    echo $rec[0];
    echo $rec[1];
}

```

### 4) mssql\_init():

By using this fn, we can  
initialize sql statements.

### 5) mssql\_execute():

To Execute the  
initialized sql statements

some functions

## \* Store procedure:

27. Oct. 2013 (07)

It is a db inmemory object used to executes multiple statements.  
Store procedures we r using to increase the performance of Application.  
Store procedures execute only for 1st call. From 2nd call onwards, get the info from previously executed code.

To Create the store procedure, The Syntax is

### Syntax:

```
Create procedure <name> as begin  
sql stat.  
sql stat.  
end
```

To call the store procedure from mysql we should use mysqli library.

Ex:

```
delimiter //  
Create procedure sp_get()  
Begin  
select * from emp;  
END //
```

Ex:

```
<?php  
$con=mysqli_connect("localhost","root","");  
mysqli_select_db($con,"test");  
$data=mysqli_query($con,"call sp_get()");  
echo mysqli_error();  
while($rec=mysqli_fetch_row($data))  
{  
echo $rec[0];  
echo $rec[1];  
echo "<br>";  
}  
?>
```

prog: prog. To see Records Next, previous  
on clicking Next and previous  
session\_start();  
Button.

30-Oct-2017 (108)

```
<?php
mysql_connect("localhost", "root", " ");
mysql_select_db("information_schema");
if(isset($_REQUEST['ind']))
    $sind = $_REQUEST['ind'];
else
{
    $ind = 0;
    $data = mysql_query("select * from character_sets");
    $_SESSION['tr'] = mysql_num_rows($data);
}
echo "Totrec". $_SESSION['tr'];
$totrec = $_SESSION['tr'];
$data = mysql_query("select * from character_sets
    limit $ind, $totrec");
while ($rec = mysql_fetch_row($data))
{
    echo "<tr><td>$rec[0]</td>$rec[1]</td>$rec[2]</td>";
}
echo "</table>";
$ind = $ind + $totrec;
if($ind >= 0)
    echo "<a href='PreNxtRec.php?ind=$ind'>PRE</a>";
    $ind = $ind + $totrec;
if($ind < $_SESSION['tr'])
    echo "<a href='PreNxtRec.php?ind=$ind'>NEXT</a>";
}


```

SAVE AS: PreNxtRec.php

OOPS:

# \* PHP Registration form with check Availability, validation

109

```
<noscript>
<meta http-equiv="refresh" content="0; 'error.html' ">
</noscript>
<script src="jquery-1.3.js"></script>
<script src="jquery.validate.js"></script>
<script>
function chkuser(un)
{
document.getElementById('sp1').innerHTML=<img src=spinner_grey.gif width=20>
$.post("check.php", {un:un}, function (data)
{
if (data==0)
{
document.getElementById('sp1').style.color="green"
document.getElementById('sp1').innerHTML="Username is Available"
}
else
{
document.getElementById('sp1').style.color="red"
document.getElementById('sp1').innerHTML="Username is Not Available"
}
})
}

function fun1()
{
$("#frm1").validate()
}

</script>
<body onload="fun1()">
<form method="post" action="reg.php" id="frm1">
<table>
<tr> <td> Username <td> <input name='txtnuser' class="required" onblur="chkuser(this.value)"> <td> <span id="sp1"></span>
<tr> <td> Password <td> <input type="password" id="pwd" name='txtpwd' class="required" minlength="5">
<tr> <td> RetypePwd <td> <input type="password" name='txtrpwd' equalTo="#pwd">
<tr> <td> Email <td> <input name='txtemail' class="email required">
<tr> <td> <input type="submit" name="sub" value="Register">
</table>
</form>
```

```
reg.php
<?php
$user = $_POST['txtuser'];
$pwd = $_POST['txtpwd'];
$pwd = sha1(md5($pwd));
$email = $_POST['txtemail'];
mysql_connect ("localhost", "root", " ");
mysql_select_db ("test");
if (mysql_query("insert into tb1_user values ('$user', '$pwd', '$email', 0)"))
{
    echo "Registration is completed";
}
else
    echo "Username is already Registered";
?>
check.php
<?php
$qs = $_REQUEST['tv'];
mysql_connect ("localhost", "root", " ");
mysql_select_db ("test");
$data = mysql_query ("select * from tb1_user where uid='$qs'");
echo mysql_num_rows ($data);
?>
```

# JOOMLA

## Installation:

Copy JOOMLA Zip S/W file and paste  
in htdocs and also extract their,  
Rename extracted folder and open at  
browser by typing URL address  
<http://localhost/projectfoldername>

default Username and password  
→ admin

SOON

XAMPP 1.5.7.

COPY Joomla S/W & paste in htdocs & extract their Joomla

03. Oct. 2019  
189

26. Oct. 2019

\* Joomla : Joomla

i) Joomla is an open source content Management syst. software

implemented by "Mambo group" (Initially Joomla called Mambo Technology, after it becomes as Joomla).

ii) Joomla is providing no. of predefined modules. By using these Modules We can customize the content of Website based on our Requirements.

iii) Joomla Administrator module is providing all manager to customize website content.

iv) first we need to install Joomla project in our syst. Then we can add & Remove the modules.

v) Joomla is available as Open Source licence with diff. version like Joomla

1.x, Joomla 1.5.7, and Joomla 1.5.8, and 2.x etc.

First, we need to install Joomla project in our syst. it requires no. of installations?

\* Installation steps of Joomla : steps, After installation, it provides one ~~step~~ using this we can manage the content. Joomla website by RAR

i) Extract Joomla S/W and place in the htdocs locat<sup>o</sup> of XAMPP FOLDER.

ii) Open the Website in Browser by http://localhost/pre

name of Extracted File in htdocs

Some supporting tables to run the project, we have to specify database name, username & password at the time of installation.

(English - United States)

Step 1: choose language:

choose the language what u want to use at the time of Joomla ~~installed~~.

(English - United States)

Step 2: PreInstallat<sup>n</sup> check:

In this step, we can see the Recommended configurat<sup>n</sup> setting value, we need to change them.

Joomla will still operate even if your settings do not match.

(No need to change, Click on Next)

Step 3: License:

Displays the license inform<sup>n</sup> of Joomla. Click on Next ←

Step 4: Database Configuration:

Joomla installs some supporting tables.

Joomla is providing some database tables, to run the website we

Need to specify database server name, userid & passwords.

hostname: localhost  
Username: root  
password: \_\_\_\_\_  
Database name: any

Step 5: FTP Configuration:

Enter FTP servername, Username & password, if we r working

LINUX

with ~~Windows~~ OS no need to specifies FTP configurat<sup>n</sup> settings, if we r working

Step 6: Main Configuration: with Windows OS.

By using configurat<sup>n</sup>, we can enter the sitename, ~~admin~~ email id,

admin username & admin password. click on install sample data which install

Some ~~sample~~ data i.e. included in the Joomla installat<sup>n</sup> package.

Step 7: Finish:

This is Final step, we need to delete the installat<sup>n</sup> folder to run the

Joomla website. After installat<sup>n</sup> to open url address: http://localhost/projectfoldername

To open administrator module URL address is http://localhost/projectfoldername/administrator  
Go C:/xampp/htdocs/myproj & delete installat<sup>n</sup> / ~~for~~ project folder

Joomla is providing diff. types of Manager, to manage the content of websites like, front page Manager, Article Manager, Etc. (190)

1) Front Page Manager → By using this option, we can manage Frontpage Articles. To set enable and disable the Front page articles using this option.

Article and uses the appropriate structure than our requirement.

## 2) Template Manager:

By Using this option, we can manage the Templates of Joomla website. This option is available in the Extensions Menu.

To add New Template Click on Install / Uninstall option of Extension Menu. By default 3 Types of Templates are available. We can also

install new Templates. We can install new Templates in Joomla project using Install / Uninstall option of Extension Menu.

3) Module Manager: Can install new Templates in Joomla project using Install / Uninstall option of Extension Menu.

By using this option, we can enable / disable the modules of Joomla Website. Modules are nothing But Menus, polls, Login form etc.

Module Manager is available in Extensions menu.

Joomla user groups are divided into 2 types

~~Administrator~~

~~Guest User~~

Backend users work with the Joomla  
Backend users can work with the Joomla  
Mobile.

## \* Plugin Manager:

By using Plugin Manager, we can enable or disable the Plugins. An installed Plugins are displayed on this window.

## \* index.php

It is a file available in Template folder contains the design script of Front page. We can change the Header and Footer information by using this file.

By using this file we can change the title text of Joomla website.  
It is available in project folder.

(Open Myfolder → index.php → Open with Editplus →

Template  
is installed

successfully

We can select  
the Template  
From Template  
Manager Window

We can select that  
Template as a  
default Template.

All installed templates  
are storing in Templates  
folder of Joomla project  
Folder. We can also copy &  
paste the Template directly  
in this location.



## \* Article Manager:

By using Article Manager, we can create new articles & we can ~~read~~ edit the existing articles. JOOMLA is providing TINYMCE editor to edit the article contents. This Editor is providing more options like bold, italic, font styles, etc. to edit article contents. By Default, we cannot execute php script thru the articles. Execute the php script we need to install "sourcererplugin".

## \* Installation of Sourcererplugin:

We can install the Sourcererplugin using install/uninstall option of Extensions Menu. Once, The Sourcerer is installed, we can find the source button or Article Manager TINYMCE Editor by using this we can execute html, javascript and php script. We should place this script within the ~~script~~ source declaration syntax.

{source}...{/source}

Go to control panel → Article Manager → Add new Article Manager → New option →

Title: MyArt

Frontpage

Days

Mios:

Section:

Extensions → Install/Uninstall → Sourcerer

Ex: {source}

```
<?php
$x=100;
$y=200;
$z=$x+$y;
echo "value is $z";
?>
{/source}
```

## akindic plugin:

By using this plugin, We can display the content of website in different Indian languages. It is an open source plugin. We can directly install this plugin in joomla Website. Path: Install

copy this plugin & paste in plugins folder of project folder / plugin/editors/ tinymce/jscripts/tinymce folder.

edit file tinymce.php Add 2 lines in tinymce.php file those are :

```
$plugins[]='akindicplugin';
$buttons3[]='akindicplugin'; } gmp
```

These lines we should add followed by a statement i.e. \$buttons2[]='forecolor' once, this plugin is added successfully, we can find out an icon on 'tinymce' editor with a text 'xit'. By using this ~~xit~~ icon we can select different Indian languages

## \* Article Parameters:

(193)

2 Types of parameters are available those are article parameters and Advanced parameters.

By using Article parameters we can change the access level of Article, article created date and time, article published date and time, etc.

By using Advanced parameters we can Hide the "pdf" icon, print icon, Mail icon etc. of an Articles.

## \* Metadata Information:

By using this option, we can provide the Meta keywords

and description to the Articles.

correct of menu items

## \* Menu Manager:

By using Menu Manager, We can create new Menus. Menus are used to maintain the menu items. Menu items we can use to open the internal articles and External Webpages.

### Module Manager:

05.OCT.2012

If u want to enable or disable the menus, we can go for Module Manager. Module Manager is available in Extensions Menu.

### Steps to Create Menus And Menu Items:

i) Open Menu Manager, Enter Unique Name, Title, Module Title per new Menu.

### Steps To Open Internal Links / External Links in Menu items:

ii) Open Menu when u want to create Menu item & select a New option from Menu item Manager of that menu.

iii) Select the link type (Internal link /External link) what you want to use with this menu item. If it is an Article, Click on internal link. Select Article Layout from Articles.

iv) Enter Menu item title and then select Article from Basic parameters option.

v) If u want to open external webpage, Click on External link Then Enter Link title and link page.

### Menu item Edit Options:

1) Display in: By using this option, we can change the menu of menu item.

2) Parent item:

By using this option, we can set a parent item to the child item.

3) Published:

To display & hide the menu item.

4) Order:

To change the order of menu item.

5) Access Level:

To specify the accessfull scope to the link. we have 3 default access specifiers  
> Public > Registered > Special.

## 6) on click, open in:

(194)

To specify the window where u want to open the link page.

We can open the parent window with Browser Navigator, New Window without/with Browser Navigator.

## \* Login Form:

By using Login Form, We can Register the New User Accounts as well as we can Login to the Website using Existing User Accounts. When New User Account has been created, it sends Activation Email. By using Activation Link user can Activate his Account.

## \* User Manager:

See all user accounts what are registered in a website

By using this option of Admin Module, We can enable and disable the User Accounts. Administrator can also create the New User accounts & He can Edit the Existing User accounts. `Myproject/com-user/register/default.php`

## \* Polls:

By using this option (Poll Manager), We can create the new polls and we can edit existing polls. This option is available in Content Menu. Polls we're using to get the Opinions from users by displaying a question.

### Steps To Create New Polls:

- 1) Open poll Manager from content option, Then select New Button. Enter <sup>Poll</sup> poll title & options. Logtime is the time interval betw 2 polls & another poll.
- 2) Enable the poll by clicking on published option.
- 3) To display a poll as Today poll, Go to the Module Manager. Select Poll option and Then select poll from Modules parameters dropdown list.

## \* Search:

(sections, categories, articles etc)

By using search module, we can search the contents of a Website. We can enable and disable the search module, We can go for module manager.

## \* Media Manager:

By using media manager we can create a new folder in the images folder as well as we can upload the images into the website.

## \* Concept To Create New Control in Registration Form:

Step1: By default, JOOMLA Registration Form is providing the predefined controls if u want to add New controls, we need to add a file i.e. `default.php` file. This file is available in project folder/components/  
~~com\_user~~ Edit com\_user/view/register/tmpl folder.

By default contains the design script of Registration form, we need to add New controls at this locatn.

Step2: To store the data of column in Database Table, we need to create a ~~new~~ column in "jos\_users" table. The columnname & the controlname should be same.

Step 8:

Add a variable in user.php file. The variable name should be same as control name.

This file is available in project folder/libraries/Joomla/database/table folder.

```
Var $params = null;  
Var $Address = null;
```

### \* Language Manager:

By using this option, we can change the Language option of Administrator Module. We can change the Left and Right position of module by using position property of that module. To change the position, open the module manager, select Section Manager module & then select position of that module.

### \* Section Manager:

By using this option, we can create the sections in Joomla project.

It is used to divide a project into different sections.

### \* Category Manager:

By using this option, we can create the categories within the sections.

### \* Global Configurations:

By using this option, we can change the configuration settings of a system, site and server.

#### A) Site Settings:

##### a) site offline:

To display the website in OFFLINE state

##### b) OFFLINE MSG:

To change the OFFLINE Text. (We can specify the msg what we want to display when the site is OFFLINE)

##### c) Site Name:

To change the Name of <sup>Web</sup> site

##### d) Default WYSIWYG Editor:

To change the Editor ~~Type~~ of Article Manager.

By default, it displays "Editor-TinyMCE" editor

##### e) List Length:

To increase / decrease the total no. of list items per page.

#### B) Metadata Settings:

By using this option, we can provide the Metakeywords &

Description to the Article.

#### \* SEO Settings:

#### \* Search Engine Friendly URLs:

By using this option, we can enable or disable

search engine friendly URL addresses. By default, it is 'Disabled'.

#### \* User Setting:

Allow & Stop user Registration, new user activation?

#### \* System Setting:

By using this setting, we can change Log folder location, user setting, session setting.

Allow User Registration: To allow and stop user registration.  
To allow (Stop New User Account creation)

### New User Registration Time:

To specify Registration file new user.

All user comes under Registered user. To change the user type of registered user

### New User Account Activation:

By using this option we can allow & stop the

activation link for new users.

### \* Session Settings:

To change the session lifetime and session Handler.

### \* Media Settings:

By using this option, we can change the Media folder settings.

We can specify max file size to upload, we can specify the extensions to the path.

### \* Server Settings:

Specified file. local time settings  
(don't change) To change the ftp settings, Database settings And mail settings.

i) \* FTP Settings: can we change these settings, when we r working with LINUX OS.  
(We don't work with FTP settings)

### \* Locale Settings:

To change the Local time settings.

### \* Database Settings:

To change the DB, servername, userid, password etc. & Also we can change the prefix word.

### \* Mail Settings:

Name  
mail To change the from name, from address etc.

### \* User Types: (User Groups)

User Types are mainly divided into 2 types.

1) Front-end User Type

2) Back-end User Type

#### 1) Front-End User Type:

It can access the Website

#### 2) Back-End User Type:

It can manage the admin module of website.

Back-End User Type again divided into 3 types.

##### a) Super Administrator:

This user can work with the all modules of Joomla project

##### b) Administrator:

Admin can work with very few features options. Administrators cannot work with Language Manager, Global Configuration, Template Manager etc.

c) Manager: Manager has very few roles. He can't work with Language Manager, Menu Manager, Global Configuration, Tools, Etc.

(197)

\* Configuration.php: It is available in project folder of website. By using this file, we can change the configuration settings. We can also use Configuration settings option Manager. But finally, the values will store at this location.

## \* Tools

i) Write Message: It is an option available in Tools used to write the mails to the backend users of Joomla project.

ii) Read Message: To Read the Messages.

Only Super Administrator can Read and Write the Messages, what he got from another user.

iii) Mass Mail:

By using this option, super administrator can send the mails to the groups of Joomla Website.

## \* JFactory:

It is a class in Joomla package used to get the info of Logged In User. It is providing getUser() method to get the details of Logged in User. This method is providing username, password, etc. properties. email properties are available in users table. getUser() is a method used to maintain these properties.

{source}

```
<?php  
$user = JFactory::getUser();  
$username = $user->username;  
$em = $user->email;  
echo $username;  
echo $em;
```

This should type in Notepad and placed in Article Manager

## \* Page Title:

By using this option, we can hide & change the title of front page. It is available in Home menu item of Main menu. We should change the options of parameters.

## \* Artisteer:

It is a licensed Tool used to create the Templates of Joomla, Drupal etc. These Tool will work with the support of .NET Framework.

## \* Plugin Manager:

By using Plugin Manager we can enable or disable the Joomla Plugins. It is available in "Extensions" Menu.

## \*Steps to Add New Controls in Registration Form:

30.OCT.2009 (198)

- i) User Registration form is available in default.php page which is located in project folder | components | components | com\_user | views | register | tmpl.
- ii) We should add controls in this page.
- iii) Add a column in database jos\_users table, The column name should be same as controlname.
- iv) Add a variable in user.php file By assigning null value, The variable name should be same as control name. This file is available in project folder /libraries/joomla/database/Table folder.

## \* Prog. of Categorizedwise List

→

```
<select multiple size="10">
<optgroup label="PHP" style="color: Red">
<option> Appress </option>
<option> Black Book </option>
</optgroup>
</select>
```

## \* Prog. To play video files.

→

```
<embed src="e:/2.avi" autoplay="false" playcount="2" width="200" height="200">
```

DRUPAL

Document Root :

d:

DRUPAL

DRUPAL

(280)

DRUPAL

Sitename: localhost

## DRUPAL

DRUPAL is a content management syst. providing no. of predefined modules used to manage the content of website. DRUPAL is open source s/w implemented by "Dries" as a message board.

Versions like 5.x, 6.x, 7.x, etc.

DRUPAL is available with different ENVIRONMENTS. First we need to install this project then we can manage the content of website. By default no. of ~~predefined~~ modules are available in drupal. By using those modules we can manage Installation : (DRUPAL 7.1.5) ~~Installation~~ by

### 1) choose profile:

By using this option, we can select the installation type.

2 Types of installations are available

- a) Standard &
- b) Minimal

### 2) choose Language:

Select the language what u want to use.

### 3) Verify Requirements:

This option checks the configuration settings what it requires

If all configuration settings ~~match~~ matches, it will redirect to setup db options.

### 4) DB Setup:

Enter the DB Name, Username, Password, To install the Drupal supporting Tables.

### 5) Configure site:

To Enter the site info like sitename, emailid, admin username, password and email address.

To open the Drupal Website, The URL address is <http://localhost/projectfolderName>.

## \* Content:

Content is used to display Text on Browser. By default, 2 types of contents are available. Polls, Blogs, Articles etc. These contents are available in 1) Articles & 2) Basic Page Adminstrator menu. To create new content the option is "Create Content". In drupal, By default 2 types of articles are available those are page and story find By using "Add content", We can Add/View content using ~~the~~ content we can find the Existing content.

## \* Options of Content:

### 1) Text Format:

By using this option, we can specify the type of ~~script~~ format what u want to execute. The available formats are 3 types. Those are

- 1) Filter HTML
- 2) Full HTML
- 3) Plain Text

1) Filter HTML supports very few <HTML> Tags.

2) Full HTML supports all <HTML> Tags.

3) Plain Text doesn't support any <HTML> Tags.

By Default, php script doesn't execute ~~in drupal~~. To execute php script we need to install a plugin called php code. By using modules option, we can enable the php code.

Select Modules and check php code then click on Save configuration. Once, the php code is installed, we can find out ~~php~~ <sup>Code</sup> option in format types.

By using this, we can enable the php code in articles.

## \* Menu Settings:

To Create Menus for Articles,

## \* Revision Information:

To Create the New Revisions for DRUPAL contents.

Revisions are used to maintain the Update what we done on ~~DRUPAL~~ DRUPAL content.

## \* Comment Settings:

To Enable and Disable the comments.

## \* Authoring Information:

By using this option, we can change the Author name, and Article author date and time inform?

## \* Publishing Options:

To publish and Unpublished Article, promote the Article on Frontpage etc. We r using this option. We can also display article as sticky article or Frontpage.

## \* Structure:

By using this option, We can change the structure of Application.

It is providing 4 options.

1) Blocks

2) Content Types

3) Menus etc.

4) Themes

## Blocks:

By using this option, we can change the regions of Website. Different Types of Regions are available Those r header, sidebarfirst, sidebarsecond, footer etc.

Menus:

By using this option we can create Menus and Menu items.  
Menu items we're using to open the internal links as well as External links.

Themes:

By using this option, we can change the DRUPAL Website Themes.  
we can also upload new themes. And customize the Existing themes.

By using configure option, we can change the colors of Themes.

Diff. Types of predefined color sets are available. We can select any one of them  
and also we can customize our own colors.

We can also enable or disable Logo, Sitename, SiteLogon etc.

~~User Management:~~Content Type:

To create new Content type and to ~~edit~~ Edit the Existing  
Content Type. By default, DRUPAL 6.8 we have 2 Types of Content Types.  
Those are page and story.

~~Post Settings:~~

To specify no. of posts on main page and length of  
Trimmed posts

User Management:

By using this option, we can customize User accounts  
It also we can create the new user Roles, we can change the User permissions  
etc.

Roles:

By using this option we can add New user Roles.

Permissions:

To allow and Deny the permissions, to access the DRUPAL Modules By User

Profile:

By using this option, we can add new controls in the User's Registration form  
it is available in User Management Menu item. first, we need to install the  
profiles module using modules menu. By using Modules option, we can install  
the profiles module. Once, it is install successfully, we can find out profiles  
menu item in User Management.

~~Steps to~~

## \* Steps To Add New control in Registration Form:

(204)

1) Click on profiles from User Management Then select control type (single line Text Field, Multiline Textfield etc). And Enter Title, Category, Name with the New control. Then select Disable in User Registrat<sup>9</sup> Form Option of Then click on Save Configurat<sup>9</sup>.

## \* User Settings:

By using this option, we can allow and stop the public Registrat<sup>9</sup>s, email activat<sup>9</sup>s. We can also change the email settings.

Blog: Environment/  
Local? Where no. of  
people communicate

By using Blogs option, we can allow the User to create

## Book Module:

18.OCT.2012

By using Book Module, we can create the Books and Book pages in DRUPAL website. Once, we added the Book pages, user cannot see the books what we added. To show the books to the user, we need to Enable Book Navigation Option. It is allowing the User to see the Books & Book Pages.

By Default, User can't create Books, To allow the User to create Books need to provide permission to User.

## Polls:

(father of computer: Charles Babage)

By using this module, we can capture polls from user & diff. Topics in the form of multiple choice Questions.

## Search:

By using this module, we can enable the search options.

## Reports:

By using this option, we can see the recent log entries, access denied errors, page Not Found Errors Etc. Joomla using Configuration.php

\* Once, The project is uploaded, To connect with the DB we have to Change the DB Servername, Username and password in settings.php file. and ~~default.~~ default.settings.php file. These files are available in project folder/sites/default folder.

Name: Nilesh S. Satpute

HTML

Mob No: 08007849849 / 0897774876

Address: LIG QTR. NO.66,

Vinkar colony,

Manewada,

Nagpur-2;  
(44002;

HTML  
By: gubba Raju

HTML5: HTML + CSS + JS + JQ + AJAX + NOS  
nithm25css3@gmail.com  
(Seminar material)

only use: Opera → ②  
Google Chrome → ①  
Safari → ③

nithm2/javascript@gmail.com

http://nithm25css3.com  
http://nithm25css3@gmail.com

A black and white photograph showing a dense cluster of small, dark, worm-like organisms, likely nematodes, crawling over a light-colored, textured surface. The organisms appear to be moving in various directions, creating a sense of motion. The lighting highlights the contrast between the dark organisms and the lighter background.

## Introduction To World Wide Web (W3C):

What is W3C?

Internet's ACT II Normalized  
v1.0

→ It stands for (World Wide Web Consortium) which is an International Consortium of companies. It was founded in 1994 by Tim Berners Lee.

### W3C Activities:

- i) W3C creates and maintains WWW Standards.
- ii) W3C is working to standardize the Web.
- iii) W3C standards are called W3C Recommendations.
- iv) W3C is organized in a number of Organizations etc.

### W3C Numbers:

There are the following well known nos. in W3C.

- i) IBM
- ii) Microsoft
- iii) America Online
- iv) Apple
- v) Adobe
- vi) Macromedia
- vii) Sun Microsystems etc.

### \* What is Internet?

→ Internet stands for International N/W. It is the combination of 2 Resources

- 1) Web Resources
- 2) N/W Resources.

### 1) Web Resources:

It is collection of Electronic pages or E-pages, developed & implemented by Tim Berners Lee, he is the father of Web.

### 2) N/W Resources:

Collection of HW & SW Resources. Implemented & developed by "Klein Rock" is the father of N/W.

### \* Internet Def's:

- 1) Collection of Web Resources & N/W Resources.
- 2) It is a global N/W of computing Resources.
- 3) A N/W of N/Ws based on the TCP/IP comm.
- 4) A community of people who use & develop those N/W.

Connectivity  
Btw 2 comp      Connectivity  
Btw 2 N/W.

\* Hotmail is the 1st email developed by Indian "Sameer Bhatia" \*

Email:

Electronic mail syst is a fast, Easy and inexpensive way to communicate with other Internet users.

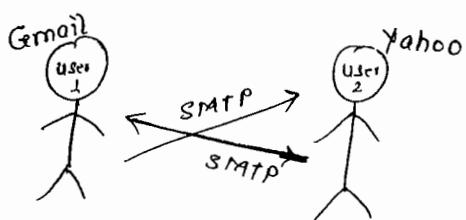
Email syst. working based on the following 2 services

→ SMTP

Simple mail Transfer protocol, This server takes care of delivering emails from one server to another server.

## 2) MIME :

MIME (Multipurpose Internet Mail Extensions); It is a mechanism to send content such as files, Images, Movies, Graphics, Audio & Videos etc. via the Internet mail syst.



## TELNET:

It allows a user to log into a Remote controller.

FTRP:

It allows a user to x'ferr files one comp. to another comp. thr. Internet conn?

24W.W.:

It is a hypertext interface to internet inform? Resources organized by W3C.

http:

This is a protocol used to exchange Hypertext Documents that makes the worldwide possible.

URL:

It is the fundamental NW identifier for any Resource connected to Web.

Ex: http://www.nareshit.com

Protocol Web Resources Secondary Level Domain (SLD) Top Level Domain (TLD)

Website:

This is a locat<sup>D</sup> on the Web where, people can find.

Website contains collect<sup>n</sup> of Webpage with the following Advantages.

- 1) Business is open 24x7 ~~is increased fast~~.
  - 2) Advertising Opportunities.
  - 3) Tremendous cost saving.

(3)

- 4) creates a Brand image
- 5) Customer Satisfact?
- 6) Increased customer Base, etc.

Q: Define Webserver.

→ Every websites sites on this comp/server, it is connected to the Internet.

Q: Browser :

→ Browser is a client-side SW. OR

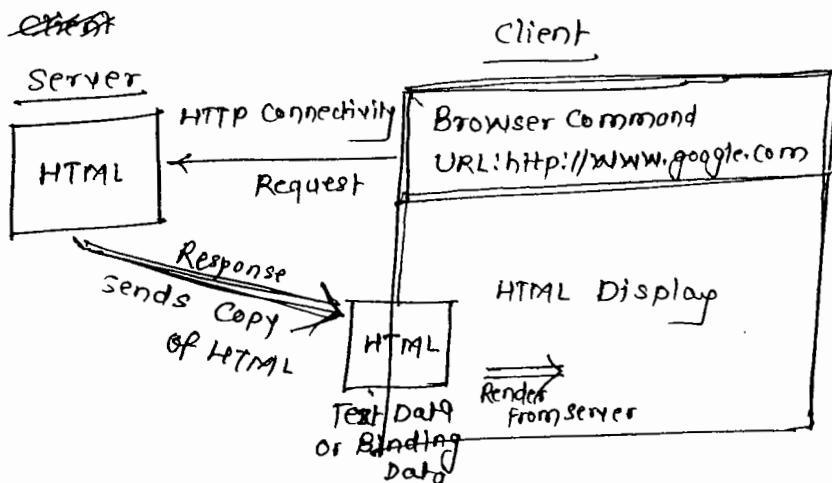
Browser taking request from the user (client), Giving the Response from the Server.

Popular Browsers are : Google chrome  
 Mozilla Firefox  
 Internet Explorer  
 Opera  
 Safari

HTML : specially identifying Data

(HyperText Markup Language), using this lang. we are creating Webpage for any Website. HTML popularly known as SGML (Standard Generalized Markup Lang.)

\* Architecture of Web Working Principal:



05. SEPT. 2019

\* Types of Websites:

Generally, Websites are divided into following 2 Types.

- 1) Static Website
- 2) Dynamic Website

Static Websites are implemented only in client-side scripting like Javascript, HTML, CSS etc.

Dynamic Websites are the combination of client-side & server side Scripting like Javascript, HTML, PHP.

## Types of Webpages:

Webpage is which is able to view on the Browser. These pages are divided into following 2 types.

- 1) Static Webpages.
- 2) Dynamic Webpages.

Static Webpages, user unable to interact Directly.

Ex: HTML, CSS.

Dynamic Webpages, user can directly interact with Webpages.

Ex: HTML, CSS, JavaScript.

These pages we generally we r developing on <sup>→ Dynamic HTML</sup> DHTML (HTML + AnyScriptt e.g. CSS)

## \* Diff. Betw Scripting and Languages:

### Scripting

i) Loosely typed prog. lang.

### \* Types of Scripts:

scripts are divided into following 2 Types.

#### 1) Client-side Script ⇒

A script which executes in the Webbrowser.

Ex: HTML, CSS, JavaScript, jQuery, etc,

Every browser contains interpreters to execute client-side programming.

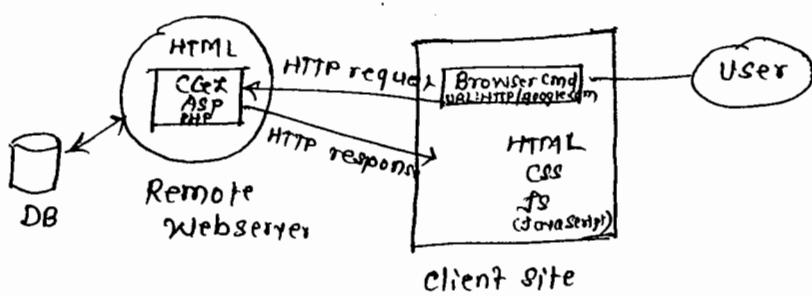
#### 2) Server-side Script:

This Script always executes in the Webserver.

Ex: PHP, ASP, CGI etc.

↳ (common Gateway Interface)

## \* Architecture of Scripts Execut?



### Note:

If U open any site like gmail, rediffmail login form,  
 If U enter Username & did not enter password,  
 Then it ask about password  
 It's actual duty of JavaScript.

## \* Introduction To Hypertext / HTML \*

(5)

- ⇒ Hypertext means supertext or special Identity or Easily linked Resources.  
Markup Language means Group of Special Tags.

## \* Versions of HTML:

As per W3C recommendation, The following versions are released into Market.

### 1) HTML 1.0 (1989 - 1994):

It is the first version, It supports Inline images, style presentation's etc.

### 2) HTML 2.0 (1995):

It supports form controls backgrounds, Tables, etc.

### 3) HTML 3.0 (1997):

It supports advanced tables & complex mathematic eq's.

### 4) HTML 4.0 (1999):

This version is supporting advanced style sheets & Multimedia elements.

### 5) HTML 5.0 (2009 - 10):

It is Web platform, It is combination of Technologies.

## \* Features of HTML:

⇒ HTML is not a Lang. or not a Scripting, It is Hypertext Resources, It has the following list of features:

1) It is a Global lang. understood by all Browsers.

2) It is extracted from SGML.

3) It is not case sensitive.

4) We can create static Webpages

5) It contains collection of commands called Tags.

6) HTML is Error-free lang, If U pass any invalid Tag, browser ignores that.

## \* Introduction to HTML Tags:

The text placed betw  $\langle$  Left angular brace ( $<$ )

& Right angular brace ( $>$ ) is called as Tags.

Syntax:  $\langle \dots \rangle \dots \langle \text{HTML} \rangle$

Eg:  $\langle \text{HTML} \rangle \dots \langle \text{HTML} \rangle$

Tags are classified into 2 Types:

1) Container Tags

2) Empty Tags

## 1) Container Tags:

(6)

The Tags which is having opening and closing Tag are called Container Tags.

Ex: `<HTML> ... </HTML>`

`<BODY> ... </BODY>`

### \* Note: \*

\* The closing Tags always starts with a forward (/) slash.

## 2) Empty Tags:

The Tags which contains only opening But no closing is known as Empty Tags.

Ex: `<br/>`  
`<hr/>`  
`<img/>`

## \* Structure of HTML Document:

`<html>` → Opening  
`<head>`  
`<title>`  
My Web Page  
`</title>`  
---  
---  
`</head>`  
`<body>`      } → Content  
---  
`</body>`  
`</html>` → Closing

Every HTML document contains mainly the following 3 parts:

- 1) Opening Tag.
- 2) Contents.
- 3) Closing Tag.

### → Above code Explanation:

- 1) In the above script, If U delete `<html>` tag Nothing reflected to O/p.
- 2) This `html` tag indicates the document contents `<html>` property.
- 3) In the above script, If U remove `<head>` tag, Nothing reflected to O/p.
- 4) If U remove `<title>` tag, it is treating as a body.
- 5) If U remove `<body>` tag, it treats as normal string, Normal string displayed on the browser.

## \* Developing HTML files:

### Diff. Betw .htm and .html.

? Modern OS are supporting, file extension is 3 or more than 3 characters. Whereas, Old OS are supporting, file extension maxm 3 characters length.  
To execute your Webpage Globally on all OS always the best extenstion is `.htm`.

(Note: chromeexperiments.com)

Generally, Every Webpage contains the following 4 critical Elements.

- 1) HTML
- 2) Head
- 3) Title
- 4) Body

### \* Preparing HTML Scripts in Windows Environment.

Step 1: Open Notepad, prepare the required script (Hypertext)

Ctrl + → Zoom in

Step 2: Save the Textfile with .htm or html extension.

Ctrl - → Zoom out

Step 3: If u need to edit right click on the icon, Select Open with Notepad.

Step 4: Do required modifications, save it and run it.

Ex: <html>  
 <head>  
 <title>  
 My First Webpage  
 </title>  
 </head>  
 <body>  
 Welcome to HTML World....  
 Hi friends! O/P: Welcome to HTML World...  
 </body>  
</html>

Solve: FirstWebpage.htm \* Implementing html on Linux Environments:

Step 1: Open gedit or vi editor

Step 2: Change the mode as insert by hitting small 'i'

Step 3: Prepare the required Script, save with the extension of .htm or .html.

Step 4: Hi friends!

If u want to modify, Right click open with gedit to modify & save it.

Step 5: Double click on the file, view the required O/P.

Note: In Linux OS,  
 Default Browser is  
"Mozilla" or "Conqueror"

§ In Windows OS, Default Browser is "Internet Explorer."

### \* Working with html comments or <comment> tag:

Comments are non executable stat. or ignore stat. with the help of these comments we can declare customized stat. as per user understandability. If the comments are not available, we are unable to declare customized stat., if declare Xlator unable to understand.

In Every prof. lang. or scripting lang. comments are common Notes. In HTML, we're using the following Note as comment.

<!---->

It is a single & multiline comment symbol.

Ex: <html>  
 <head>  
 <title>  
 HTML Comments  
 </title>  
 </head>

<body>  
 <! project Name: Web Development>  
 <! Module Name: Contact Us>  
 Welcome to html World...  
 </body>  
</html>

Solve: comment1.htm

O/P: Welcome to html

World...



strikethrough striking effect : `<s> . . . </s>`  
 underline : `<u> . . . </u>`  
 superscript : `<sup> . . . </sup>`  
 subscript : `<sub> . . . </sub>`  
 blockquote : `<blockquote> . . . </blockquote>`  
 small : `<small> . . . </small>`  
 big : `<big> . . . </big>`  
 teletype : `<tt> . . . </tt>`  
 del : `<del> . . . </del>`  
 quotation : `<q> . . . </q>`  
 center : `<center> . . . </center>`

### Save: "Formatting Tags & HTML"

Ex:

```

<html>
<head>
<title>
  Formatting Tags
</title>
</head>
<body>

```

Op:

~~It is a bold format~~

Op:

It is a bold format  
 It is also a bold format  
 It is italic format  
 It is also italic format  
~~It is removed content~~  
It is underline format  
 It is something special  
 It is power of (2)<sup>3</sup>  
 It is base of (2)<sup>3</sup>  
 It is a small format  
 It is a big format  
 It is a teletype content

`<b> It is a bold format </b> <br/>`  
`<strong> It is also bold format </strong> <br/>`  
`<i> It is italic format </i> <br/>`  
`<em> It is also italic format </em> <br/>`  
`<s> It is removed content </s> <br/>`  
`<u> It is underline format </u> <br/>`  
`<blockquote> It is something special </blockquote>`  
`It is power of (2) <sup> 3 </sup> <br/>`  
`It is base of (2) <sub> 3 </sub> <br/>`  
`<small> It is a small format </small> <br/>`  
`<big> It is a big format </big> <br/>`  
`<tt> It is a teletype content </tt> <br/>`

<del> It is also removed content </del><br/>  
<q> It is double quotes </q><br/>  
<center> It is a webpage </center> ~~It is~~  
</body>  
</html>

~~It is also removed content~~

It is double quotes

It is a webpage

## \* Working with Tag Attributes and Parameters:

Attributes, These attributes having the following Generally, html tags contains features:

- i) html tags can contain one or more attributes.
- ii) Attributes are the additional strength of the tag.
- iii) Attributes always provide in the start tag.
- iv) Attributes consist Name, value, pairs separated by equal signs.
- v) Attribute values are in single or double quoted etc.

Syntax: <tag attribute="parameter">

Ex: <body bgcolor="lightgreen">  
     ↑      ↑      ↑  
     Tag   Attribute   Parameter/Value

## \* Uses of <Body> Tag:

It contains all the contents of html document such as text, hyperlinks, images, special characters, lists, tables, frames, forms etc.

In Every html document, body is the most powerful sect.

Syntax:

<body> . . . </body>

Ex: <html>

  <head>  
    <title>  
      Body Tag  
    </title>  
  </head>  
  <body>

    It is a Webpage Body section

  </body>

</html>

It has following list of attributes and parameters.

1) bgecolor:

Attribute parameters  
text: Color name or hexadecimal no.

background: image path

bgecolor: Color name or hexadecimal no.

using this attribute, we can change the background color of the page (1)

```
<html>
<head>
<title>
    Body Tag
</title>
</head>
<body bgcolor = "lightgreen">
    It is Webpage body section
</body>
</html>
```

SAVE AS: BGColor.htm

O/P: It is a Webpage Body section

(Background color turns to lightgreen)

### background:

By using this attribute, we can apply background image for webpage

```
<html>
<head>
<title>
    Body Tag
</title>
</head>
<body background = "html.png">
    It is a Webpage body section
</body>
</html>
```

SAVE AS: BackGround.htm

O/P: It is a Webpage body section  
(Background is Image)

### text:

This attribute is used to change the color of the text in the page

Ex: <html>

```
<head>
<title>
    Body Tag
</title>
</head>
```

SAVE AS: BodyTextColor.htm

O/P: It is a Webpage Body section

(Text color is Red)

```
<body text = "red">
```

It is a Webpage Body section

```
</body>
```

```
</html>
```

```
<html>  
<head>  
  <title>  
    Body Tag  
  </title>  
</head>
```

### \* Loading images from diff paths.

⇒ If u declare like this, ~~background= "manjog.jpg"~~ background= "manjog.jpg" in some place. Or in same folder.

~~background~~

background= "C:/Users/public/pictures/sample pictures/~~Hydrangea~~"

The above locat<sup>n</sup>, some browsers able to identify like IE, Google Chrome etc.

Bcoz, the html Resources, in one locat<sup>n</sup>, image Resources in other locat<sup>n</sup>, That time the following universal path recognized by all browsers.

background="file:///C:/Users/public/pictures/sample pictures/Hydrangea.jpg"

08. SEPT. 2012

### A. Paragraph tag:

Using this tag we can divide into paragraph. It is a container tag.

Syntax: <p> --- </p>

#### Attributes

#### Parameters

left, Right, centre, Justified.

Ex: <html>

```
  <head>  
    <title> Paragraph Tag </title>  
  </head>
```

<body>

<p> HTML is a Markup Language. It is from W3C Corporation,  
CSS is a styles sheets to provide more visibility to webpage  
HTML5 is a Webpage <sup>platform</sup> from W3C </p>

</body>

</html>

Syntax: <p> ---  
--- </p>

### Using Attributes:

```
<html>  
  <head>  
    <title> Paragraph Tag </title>  
  </head>
```

<body>

<p align="left"> HTML is a Markup Language </p> CSS is a style sheet  
<p align="center"> It is from W3C Corporation </p> CSS is a style sheet  
<p align="Right"> CSS is a stylesheets </p>  
<p align="justify"> HTML is a Markup language. It is from W3C Corporation  
justified CSS is a style sheet </p>

Note:  
print same like  
as written in  
p tag.

SAVE AS: ParagraphAttributes.htm

HTML is a Markup Language

It is from W3C Corporation

HTML is a Markup Language & It is from W3C Corporation. CSS is a style sheet

a style sheet

<p align="center"> It is from W3C Corporation </p>

<p align="Right"> CSS is a stylesheets </p>

<p align="justify"> HTML is a Markup language. It is from W3C Corporation

justified CSS is a style sheet </p>

## \* Font tag:

(13)  
It is used to format the text such as changing the text size, color & style. It is a container tag.

Syntax: `<font> . . . </font>`

Attributes      Parameters

color      Any color name or hexadecimal no.

size      1 to 7. // Default size = 3.

face      Arial, etc.

```
<html>
  <head>
    <title> Font Tags </title>
  </head>
  <body>
    <font color="Red"> html is a Markup Language. </font>
    <font size=6> html is a Markup language. </font>
    <font face="Arial"> html is a markup language </font>
  </body>
</html>
```

SAVE AS: FontTag.htm

O/P:

HTML is a Markup Language. HTML is a

In a Red color      Markup Language. HTML is a Markup

language. In a Font size = 6

language

In a

Arial font

Face

## \* Headings in html:

html contains the following six heading types. all headings are container tags.

Syntax:

`<h1> . . . </h1>`

`<h2> . . . </h2>`

`<h3> . . . </h3>`

`<h4> . . . </h4>`

`<h5> . . . </h5>`

`<h6> . . . </h6>`

Ex: `<html>`

~~<head>~~ <head>

<title> Headings in HTML </title>

</head>

<body>

`<h1>` It is heading <h1> → It is a big

`<h2>` It is heading <h2> heading Tag.

`<h3>` It is heading <h3>

`<h4>` It is heading <h4>

`<h5>` It is heading <h5>

`<h6>` It is heading <h6>

</body>

SAVE AS: HeadingTag.htm

</html> O/P: It is a Heading

Attributes      Parameters

align      left, Right, center

Ex: `<html>`

```

<head>
  <title> Headings in HTML </title>
</head>
<body>
  <h1 align=Left> one </h1>
  <h2 align=Right> Two </h2>
  <h3 align=Center> Three </h3>
  <h4 align=Left> Four </h4>
  <h5 align=Right> Five </h5>
  <h6 align=Center> Six </h6>
</body>
</html>

```

14  
SAVE AS: HeadingAttributes.html

O/P: It is a Heading  
It is a Heading

Note: Font tag is not available in html file.

### HR Tag:

It is used to draw a line across the Webpage. It is an Empty Tag.

Syntax: `<hr>`

It has the following list of Attributes.

<u>Attributes</u>	<u>Parameters</u>
color	Any color name/ hexadecimal
size	(Pixel) Pix
width	% or Pix
align	left, Right, center
noshade	noshade

Note: The Default width of Horizontal <sup>rule</sup> is "100%"  
 The Default alignment of Horizontal is "Center"  
 Noshade Attribute will be applied only,  
 When we are not specifying the color of  
 horizontal Rules.

Ex: `<html>`

```

<head>
  <title> HR Tags </title>
</head>
<body>

```

`<hr color="maroon" size=3 width=100 align="left">`

`<h1> one </h1>`

`<hr color="#ff00ff" size=5 width=200 align="right">`

`<h2> Two </h2>`

`<h3> Noshade = noshade </h3>`

`<h3> Three </h3>`

`<br>`

`..</body>`

Ex:

`<html>`

`<head>`

`<title> HR Tags </title>`

`</head>`

`<body>`

`<h1> one </h1>`

`<hr>`

`<h2> Two </h2>`

`<hr>`

`<h3> Three </h3>`

`<hr>`

`<h3> Four </h3>`

`<hr>`

`<h3> Five </h3>`

`<hr>`

`<h3> Six </h3>`

`<hr>`

`<h3> Seven </h3>`

`<hr>`

`<h3> Eight </h3>`

`<hr>`

`<h3> Nine </h3>`

`<hr>`

`<h3> Ten </h3>`

`<hr>`

`<h3> Eleven </h3>`

`<hr>`

`<h3> Twelve </h3>`

`<hr>`

`<h3> Thirteen </h3>`

`<hr>`

`<h3> Fourteen </h3>`

`<hr>`

`<h3> Fifteen </h3>`

`<hr>`

`<h3> Sixteen </h3>`

`<hr>`

`<h3> Seventeen </h3>`

`<hr>`

`<h3> Eighteen </h3>`

`<hr>`

`<h3> Nineteen </h3>`

`<hr>`

`<h3> Twenty </h3>`

`<hr>`

`<h3> Twentyone </h3>`

`<hr>`

`<h3> Twentytwo </h3>`

`<hr>`

`<h3> Twentythree </h3>`

`<hr>`

`<h3> Twentyfour </h3>`

`<hr>`

`<h3> Twentyfive </h3>`

`<hr>`

`<h3> Twentysix </h3>`

`<hr>`

`<h3> Twentyseven </h3>`

`<hr>`

`<h3> Twentyeight </h3>`

`<hr>`

`<h3> Twentynine </h3>`

`<hr>`

`<h3> Thirtynine </h3>`

`<hr>`

`<h3> Thirtytwo </h3>`

`<hr>`

`<h3> Thirtysix </h3>`

&lt;p

## \* Marquee Tag:

Using this tag we can create a scrolling Text or scrolling image from Left to Right, Right to Left, Top to Bottom, Bottom to Top.  
It is a container Tag.

Syntax: <marquee> . . . </marquee>

### Attributes      Parameters

Ex: <html>

1) Behaviour "slide" ⇒ start & stop as

soon as Text touches

the Margin. (By default 1 time)

<head>

<title> Marquee Tag </title>

</head>

<body>

<marquee> Scrolling </marquee>

</body>

</html>

### SAVE AS:

MarqueeTag.htm

O/P: (See on screen)

2) bcolor

"color-code"/  
Name ⇒ specifies the color  
name as background color.

3) Direction

"left" ⇒ Left to Right

"Right" ⇒ Right to Left

"UP" ⇒ Bottom to Top

"Down" ⇒ Top to Bottom

4) Width

"size, px" ⇒ specifies width in marquee.

5) height

"size, px" ⇒ specifies height in marquee.

### Ex: Behaviour Attributes in Marquee Tag:

```
<html>
  <head>
    <title> Marquee Tag </title>
  </head>
  <body>
```

SAVE AS: MarqueeBehavior.htm

O/P: See on Screen

```
    <marquee behaviour = "scroll"> Scrolling </marquee>
    <marquee behaviour = "slide"> Slide </marquee>
    <marquee behaviour = "alternate"> Alternate </marquee>
```

</body>

</html>

SAVE AS: MarqueeBehaviorBgcColorWidthDir.htm

O/P: See on Screen

Ex 2: <html>

<head>

<title> Marquee Tag </title>

</head>

<body>

<marquee behaviour = "scroll"> Scrolling </marquee>

<marquee behaviour = "slide"> Slide </marquee>

<marquee behaviour = "alternate" bcolor = lightgreen width = 150 height = 350 direction = "Down">

Alternate </marquee>

Continued

## Remaining attributes of Marquee

6) Loop: "2" continues in limited times.  
"infinite" loop continues in infinite.

10-SEPT-2013

(16)

7) scrollamount, "3" specifies speed to scroll on the text.

Ex: <html>

```

<head>
<title> Marquee Tag </title>
</head>
<body>
<marquee behavior="scroll" scrollamount=1> scrolling </marquee>
<marquee behavior="slide" loop=3 scrollamount=2> slide </marquee>
<marquee behavior="alternate" scrollamount=3> Alternate </marquee>
</body>
</html>

```

SAVE AS:

MarqueeBehaviorScrollAmt.htm

(see o/p on screen)

Ex: <html>

```

<head>
<title> Marquee Tag </title>
</head>
<body>
<marquee behavior="scroll" scrollamount=1> scrolling </marquee>
<marquee behavior="slide" loop=3 scrollamount=2> slide </marquee>
<marquee behavior="alternate" onmousedown="this.stop(); onmouseup="this.start(); scrollamount=3"> Alternate </marquee>
</body>
</html>

```

SAVE AS:

MarqueeBehaviorScrollAmt2.htm

(see o/p on screen)

## \* PRE Tags:

Pre stands for 'preformatted text', if we specify the text in <pre> tag, the browser consider line breaks and spaces specified in the Text Editor. It is a container tag.

Syntax: <pre> --- </pre>

Note: O/P will be same as we written on the screen.

Ex: <html>

```

<head>
<title> PRE Tags </title>
</head>
<body>
<pre> HTML Markup Language

```

It is from W3C

Thank U...

</pre>  
</body>

Naresh i Technology

## The image Tag : (No closing tag)

17

By using this tag, we can insert images in the Webpage.

These images are classified into the following 2 types:

- 1) Internal Images.
- 2) External Images.

### Internal Images:

These images will be automatically loaded to Webpage.

### External Images:

These images <sup>to load</sup>, User should perform a kind of action.  
It is an empty tag.

Syntax: <img>

#### Attributes

1) src

#### Parameters

image path

2) border

pix

3) height

pix or %

4) width

pix or %

5) align

left, right, top, middle, bottom

6) alt or title

any text

Ex1 <html>

<head>

<title> IMG Tag </title>

<head>

<body>



</body>

</html>

Note: Whenever you put a cursor on img then the name will come.

1) In the image tag, the Left { Right alignment's <sup>will be</sup> applied to the image

Ex2 <html>

<head>

<title> Img tag </title>

2) The Top, Middle and Bottom alignments applied to the text along with image.

<body>

applied to the text along with image.



</html>

Ex3 <html>

<head>

<title> Img Tag </title>

<head>

<body>

<p> An Image

  
with align="bottom" </p>

<p> An Image

 with align="top" </p>

<p> An Image

 with align="middle" </p>

</body>

</html>

## External Images:

```
<body> <html>
      <head>
        <title>
        </head>
      <body>
        
      </body>
    </html>
```

\* Note: We can not copy ~~path~~ ~~image~~ URL address. Some browser supports to see URL address & some not. Internet Explorer support most of URL address.

Note: Banking Websites & Finance Websites don't allow to display on Webserver even if copy URL address.

Note: Ctrl + R  
For Refresh.

## \* Working with HTML Links:

Links are used to navigate easily from one page to another page or one page to another site, etc. These links are divided into the following 2 types:

- 1) Internal Links.
- 2) External Links.

{ <sup>o.o</sup> Means HyperText Super }

### 1) Internal Links:

Linking within the page or within the website is called as Internal links.

### 2) External Links:

Link to external files like documents, other website etc. is called as External link. To provide linking or to create the linking we r using Anchor Tag. It is a container tag.

Syntax: <a> --- </a>

#### Attributes

	<u>Parameters</u>
href	url (uniform Resource Locator)
name	any name
target	_blank, _parent, any name.

Ex: <html>

```
<head>
<title>Working with links</title>
</head>
<body>
<a href="http://www.unix.org">My Tutorial</a>
</body>
</html>
```

### Internal Links:

These links allows to link another sect<sup>n</sup> on the same page, u can scroll the page up or down to view desired locat<sup>n</sup>.

Q: Write a script to make a link within the page.

```
→ <html>
  <head>
    <title> Internal Links </title>
  </head>
  <body>
```

**19**

```
<p>
<a href="#C7"> see also chapter 7. </a>
</p>
<h2> Chapter 1 </h2>
<p> Introduction To HTML </p>
<h2> Chapter 2 </h2>
<p> HTML document structure </p>
<h2> Chapter 3 </h2>
<p> What is Tag? </p>
<h2> Chapter 4 </h2>
<p> How to create Order List And
    Unordered List </p>
<h2> Chapter 5 </h2>
<p> How to provide Meta keywords </p> Chapter 4
<h2> Chapter 6 </h2>
<p> What is Attribute? </p>
<h2><a name="C7"> Chapter 7 </h2></a>
<p> What is Event And How to use? </p> How to provide Meta keywords
</body>
</html>
```

### Text Links:

Text links allows the programmer to create a text that acts as a link. whenever, User click on that, it will refer to another page.

### Target page:

It allows a link to be created, when u clicked on, it's opening a new window or in the same window the webpage launching.

```
<html>
<head>
    <title> Internal Links </title>
</head>
<body>
    <a href="http://www.nareshit.com" target="blank"> Go To Naresh IT </a>
</body>
</html>
```

Note: **target = "Parent"**  $\Rightarrow$  Default Nature, it opens on the same page.  
**target = "blank"**  $\Rightarrow$  It's opening on another Web page.

### SAVE AS:

LinkWithinPage.htm

O/P:

See Also Chapter 7  $\leftarrow$

Chapter 1

Introduction To HTML

Chapter 2

HTML Document Structure

Chapter 3

What is Tag?

How To Create Order List And Unordered List

Chapter 5

How To provide Meta keywords

Chapter 6

What is Attribute?

Chapter 7  $\leftarrow$  only chapter 7 display in diff. color.

What is Event And How to use?

## \* Picture links:

A picture link allows the programmer to create a picture that act as a link, when user clicked on the image, it is redirecting to concerned webpage or website.

Ex: <html>  
    <head>  
        <title> Links Picture Links </title>  
    </head>  
    <body>  
        <a href="http://www.w3c.org">  
              
        </a>  
        <a href="http://www.nareshit.com">  
              
        </a>  
    </body>  
</html>

## \* Local links:

It is very similar to normal link Tag, A local link, links from a Webpage to a directory on the user's computer.

Ex: <html>  
    <head>  
        <title> Local Links </title>  
    </head>  
    <body>  
        <a href="file:///E:/HTML Class Materials/HTML Materials/images/Hmer.jpg">  
            Click Here to open Local Image  
        </a>  
    </body>

## \* HTML mailto links:

We can also make a link to link email addressees from href attribute is set with mailto followed by email address. When u click on email link it opens email application, using that u can send mails & receiving mails.

Ex: <html>  
    <head>  
        <title> Mail Links </title>  
    </head>  
    <body>  
        <a href="mailto:hyd\_nit@gmail.com"> Mail Link </a>  
    </body>

## html subdirectory link:

Main Level: http://www.domain.com  
subdirectories

First Level: http://www.domain.com/subdir1/  
subdirectory

2nd Level: http://www.domain.com/subdir1/subdir2/  
subdirectory

(4)

## <table> tag:

12. SEPT. 2019

### Working with table tag:

Tables are used to represent a data in a Tabular format. The best way to split a page into diff. sect's, To Represent Tables we should use <table> tag, it is a container tag.

### main tags:

~~<thead> <tbody> <tfoot>~~

1) <table>: It is a container tag. Everything we are doing inside this tag.

Syntax: `<table> . . . </table>`

~~Ex:~~

2) <table rows: <tr>  
<tr> The horizontal line represents Rows in the Table, To represent these table rows, We are using <tr> tag. It is a container tag.

Syntax:

`<tr> . . . </tr>`

Ex:

```
<table>
  <tr> . . .
  .
  .
  </tr>
</table>
```

### 3) <td>:

(Table data / Table cells) Each Row consist of no. of cells, Each cell defined by a tag it is known as <td>, It is a container tag. The intersect of Rows & Columns are called as Cells.

Syntax: `<td> . . . </td>`

Ex: `<table>`

`<tr>`

`<td>`

`. . .`

`...`

`</td>`

`</tr>`

`</table>`

Ex: <html>  
 <head>  
 <title>  
 Working With Table Tag  
 </title>  
 </head>  
 <body>  
 <table border=1>  
 <tr>  
 <td> Std No. </td>  
 <td> Std Name </td>  
 </tr>  
 <tr>  
 <td> 101 </td>  
 <td> Smith </td>  
 </tr>  
 <tr>  
 <td> 102 </td>  
 <td> Ravi </td>  
 </tr>  
 <tr>  
 <td> 103 </td>  
 <td> Scott </td>  
 </tr>  
 </table>  
 </body>  
</html>

O/P: Save As: TableTagWithTr&Td.htm

StdNo.	StdName
101	Smith
102	Ravi
103	Scott

### \*Table heading:

<th>:

To represent Table heading we are using <th> tag, It is a Container tag.

Syntax:

Ex: <th> . . . </th> O/P:  
 <tr>  
 <th> StdNo </th>  
 ,<th> Smith </th>

### \* Table Tag Attributes & Parameters

<u>Attributes</u>	<u>Parameters</u>
i) border	pixels
ii) bordercolor	Any color
iii) bgcolor	Any color
iv) background	Image path
v) height	pixels or %
vi) width	pixels or %
vii) align	Left, Right, Center
viii) valign	Top, Middle, Bottom
ix) rules	rows, <del>cols, all, none</del>
x) cellspacing	pixels
xi) cellpadding	pixels
xii) rowspan	number
xiii) colspan	number

Example: Table tag with border, bordercolor, bgcolor, background. (First Four)

⇒ <html>  
 <head>

<title> Working With Table Tag With Attributes </title>

</head>

<body>

<table border=1 bordercolor=red

bgcolor="lightgreen" background="#FF0000" pg>

<tr>

<th> Std No. </th>

<th> Std Name </th>

</tr>

<tr>

<td> 101 </td> TableTagWithBorderFor

<td> Smith </td> BorderBackground

</tr>

O/P:

Std No.	Std Name
101	Smith
102	Ravi

O/P:

Std No.	Std Name
101	Smith
102	Ravi

O/P: Table Background is Google chrome Border is Red Slight Green color

</table> </body> ...

## Ex: Table Level And Cell Level Attributes.

(23)

```

<html>
  <head>
    <title> Working with Table and Cell Attributes </title>
  </head>
  <body>
    <table border=2>
      <tr>
        <th bgcolor='lightgreen'> StdNo. </th> only this
        <th> StdName </th>
      </tr>
      <tr>
        <td> 101 </td>
        <td background="chrome.png"> Smith </td>
      </tr>
      <tr>
        <td> 102 </td>
        <td> Ravi </td>
      </tr>
    </table>
  </body>
</html>

```

Save As: TableThBgColor&TdBgColor.htm

Off:

StdNo.	StdName
101	Smith
102	Ravi

part shows  
in  
right green  
color.

if this  
box  
The  
chrome  
2 con will  
show

## height And width:

specifying the width and height on your table can make your pages much easier to control.

## Difference Betw Pixels and Percentage:

There are 2 ways of marking the width & height of your table. 1) By specifying certain no. of pixels

Other Way 2) Marking using %.

### Pixels:

are fixed size units that are used in screen media, pixel will make a certain length & stay there, one pixel is equal to one dot on your comp. screen.

### Percent: (%)

This unit is much close to "em" unit. It is scalable for all kinds of devices, like PC's, mobiles etc. An "em" is a unit of measurement in the field of Typography.

Note: By default Pixel.

Ex:

```
<table border=2 height=300 width=300  
<tr>
```

background='water.gif'  
="c:/Users/user/Pictures/Chrome.png"/>

```
<th> Empid </th>  
<th> EmpName </th>  
<th> Location </th>
```

```
</tr>
```

```
<tr>
```

```
<td> 111 </td>
```

```
<td> Nil </td>
```

```
<td> Nagpur </td>
```

```
</tr>
```

```
<tr>
```

```
<td> 222 </td>
```

```
<td> Pin2 </td>
```

Hyderabad

```
<td> Nagpur </td>
```

```
</tr>
```

```
<tr>
```

```
<td> 333 </td>
```

```
<td> Pawan </td>
```

```
<td> Chandrapur </td>
```

```
</tr>
```

O/P:

Empid	EmpName	Location
111	Nil	Nagpur
222	Pin2	Hyderabad
333	Pawan	Chandrapur

Note: we can increase height by setting  
height = 300, 400, 500 also

We can increase width by setting  
width = 300, 400, 200 etc

and Google chrome icon shows as 9  
in a table.

\* By using rules attribute

all rows, cols, for table

Say As: TableUsingRulesAttribute.htm

<table border=2 rules=none> Note: Here,

We set, attribute  
rules = none

so o/p ⇒

Empid	EmpName	Location
111	Nil	Nagpur
222	Pin2	Hyderabad
333	Pawan	Chandrapur

If we set, attribute

rules = cols then

o/p ⇒

Empid	EmpName	Location
111	Nil	Nagpur
222	Pin2	Hyderabad
333	Pawan	Chandrapur

If we set, attribute

rules = rows then

o/p ⇒

Empid	EmpName	Location
111	Nil	Nagpur
222	Pin2	Hyderabad
333	Pawan	Chandrapur

## \* empty cells creation:

```
Ex: <table border=2>
      <tr>
        <th> EmpId </th>
        <th> EmpName </th>
        <th> Location </th>
      </tr>
      <tr>
        <td> 101 </td>
        <td> Nil </td>
        <td> Nagpur </td>
      </tr>
      <tr>
        <td> 102 </td>
        <td> Pawan </td>
        <td> Chandrapur </td>
      </tr>
      <tr>
        <td> 103 </td>
        <td> &nbsp;</td>
        <td> Hyderabad </td>
      </tr>
      <tr>
        <td> 104 </td>
        <td> Pin2 </td>
        <td> &nbsp;</td>
      </tr>
```

</table> SAVE AS:

~~Opp:~~

EmptyCellsCreation.htm

EmpId	EmpName	Location
101	Nil	Nagpur
102	Pawan	Chandrapur
103		Hyderabad
104	Pin2	

## \* Cellspacing

13-SEPT-2014

(25)

This attribute is used to change the spacing b/w the cells.

## \* Cellpadding

This attribute is used to change the space b/w the contents to the inner adjust of a cell.

### 1) Cell Padding:

Using this attribute we can ctrl cell padding with the help of pixels.

#### Syntax:

<Table cellpadding="x">

**Ex:** <table border=2 cellpadding=15>

```
<tr>
  <th> EmpId </th>
  <th> EmpName </th>
  <th> Location </th>
```

If u not put "f" in front of tr

"f" in front of td

of nbsp <tr>

then, ofp will be

nbsp instead of blank

space <tr>

<td> 101 </td>

<td> Nil </td>

<td> Nagpur </td>

<td> 102 </td>

<td> Pawan </td>

<td> Chandrapur </td>

<tr>

<tr>

<td> 103 </td>

<td> Pin2 </td>

<td> Hyderabad </td>

</tr>

</table>

Opp:

SAVE AS:

TableCellPadding.htm

EmpId	EmpName	Location
101	Nil	Nagpur
102	Pawan	Chandrapur
103	Pin2	Hyderabad

## Cells Spacing:

By using this Attribute  
U can ctrl the dist. b/w the cells with  
the help of pixels.

### Syntax:

```
<TABLE CELLSPECING="x">
```

Ex:

```
<table border=2 cellspecing=15>
<tr>
<th> EmpId </th>
<th> EmpName </th>
<th> Location </th>
</tr>
<tr>
<td> 101 </td>
<td> Nilesh </td>
<td> Nagpur </td>
</tr>
<tr>
<td> 102 </td>
<td> Pawan </td>
<td> Chandrapur </td>
</tr>
</table>
```

O/P:

EmpId	EmpName	Location
101	Nilesh	Nagpur
102	Pawan	Chandrapur

## Using Both CELLSPECING & CELLPADDING



Ex: <table border="1" CELLPADDING="7"  
CELLSPECING="7">

SAVE AS: TableCellsSpacing&CellPadding.htm

O/P:

EmpId	EmpName	Location
101	Nilesh	Nagpur
102	Pawan	Chandrapur

## Table Headers Vertically

26

```
<table border=2>
<tr>
<th> EmpNo </th>
<td> 101 </td>
</tr>
<tr>
<th> EmpName </th>
<td> Nilesh </td>
</tr>
<tr>
<th> EMP Location </th>
<td> Chandrapur </td>
</tr>
```

</table>

SAVE AS:

O/P:

EmpNo	101
EmpName	Nilesh
EmpLocation	Chandrapur

## Table Cells And Images:

```
<TABLE BORDER="7" CELLPADDING="10" CELLSPECING="10" BGCOLOR="#0000FF">
<CAPTION ALIGN="BOTTOM">Butterfly </CAPTION>
<TR>
<TD>
<TABLE BORDER="7" CELLPADDING="10" CELLSPECING="10" BGCOLOR="#FF0000">
<TR>
<TD> <IMG SRC="fish1.gif" TITLE="butterfly"> </TD>
</TR>
</TABLE> <u>SAVE AS:</u> <u>Butterfly.htm</u>
</TD>
</TR>
</TABLE> <u>O/P:</u> See on screen
```

## of colspan and rowspan:

These 2

- features allows u to extend columns & rows across multiple other columns & rows.

### 1) Colspan:

It extends cells in a horizontal

- (Left & Right) Row.

#### Syntax:

```
<TD COLSPAN="X">
```

Ex: <table border=1>

```

<tr>
  <th>EmpId</th>
  <th>EmpName</th>
</tr>
<tr>
  <td colspan=2 align=center>
    Naresh Technology
  </td>
</tr>
<tr>
  <td>101 </td>
  <td>Naresh solution </td>
</tr>

```

</table> SAVE AS: Table<sup>Col</sup>Span.htm </table>

O/P:

EmpId	EmpName
Naresh Technology	
101	Naresh Solutions

There are 3 ways of Aligning for Both of these. 27

A Horizontally we can Align Left, Right and Center.

B Vertically we can Align Top, Bottom and Middle.

~~Horizontal alignment~~

Note: Horizontally, we can use align Attribute.

Vertically, we can use valign Attribute.

Ex:

```

<table border=2>
<tr>
  <td rowspan=2 valign="bottom">
    cell 1 </td>
  <td> cell 2 </td>
</tr>
<tr>
  <td> cell 3 </td>
</tr>

```

SAVE AS: TableRowSpan.htm

O/P:

	cell 2
cell 1	cell 3

### 2) Rowspan:

It extends cells in a

Vertical Row up & down

#### Syntax:

```
<TD ROWSPAN="X">
```

#### Table Alignment:

There are 2 ways.

We can align Text in a cell.

1) Horizontally

2) Vertically

## \* FRAMESET \*

### frameset:

By using this Tag we can divided the Webpage as Multiple frames. In Each frame we can display another Website. Frameset is a container Tag.

Syntax: <frameset> . . . </frameset>

#### Attributes

1) rows

2) cols

3) border

4) bordercolor

#### Parameters

pixels, %

pixels, %

pixels

any color name.

### Frame:

It is used to ~~call~~ external webpage. It contains src property to specify the path of external webpage. Using Frames we can place & view multiple files in a single window. It is an Empty Tag.

### Syntax:

<frame>

#### Attribute

src

name

scrolling

#### Parameters

file path

any name

yes, no, default

### Ex:

```
><frameset rows="50%, 50%">
  <frame src="http://www.licindia.com">
  <frame src="http://www.google.co.in">
</frameset>
```

### \* By using rows and cols:

### Ex:

```
<frameset rows="50%, 50%" cols="60%">
  <frame src="http://www.licindia.com">
  <frame src="http://www.myindia.com">
  <frame src="http://www.unix.egg">
</frameset>
```

### Op:

28  
14. SEPT. 2018

### Advantages:

- 1) Frame provides Technical sofisticated appearance to the Website.
- 2) It is a facility to improve the useability of the Website.
- 3) Frames generally include Navigat? links, headers or footers.

### DisAdvantages:

- 1) It is difficult to display entire page
- 2) The Webdeveloper must be track of more html document linked within frame.

#### \* Note:

We don't have body parts in the frame.

### Scrollbars:

It is an Attribute to implement scrollbars on your required frame.

It has the following parameters

- 1) YES : Turns the scroll bar ON.
- 2) NO : Turns the scroll bar OFF.
- 3) AUTO : The webpage detect if needed.

### Ex:

```
<frameset rows="50%, 50%">
  <frame src="http://www.licindia.com"
        SCROLLING="YES">
  <frame src="http://www.myindia.com"
        SCROLLING="NO">
</frameset>
```

### Op:

\* frame errors: A Body Tag imp. For <noframes>  
 The majority of browsers commonly support frames. There are some old browser versions that do not. In that case you are able to specify an error msg or warning msg. "Why frame failed", If any msg you need to specify, you can insert in between <NOFRAMES> Tag. This tag should be placed in between <body> tags.

Syntax: <NOFRAMES>.....</NOFRAMES>

Ex:

```

<html>
  <head>
    <frameset rows="50%, 50%">
      frame src="http://www.licindia.com"
      frame src="http://www.myindia.com"
    </frameset>
    </head>
    <body> For NoFrames
          <body> Tag
          <NoFrames> important
    </NoFrames> OOps you're using too lower version
    browser, Update and Try...
    </NoFrames>
    </body>
  </html>

```

Op:

\* frame border: It is separate diff. frame on a Webpage. We are able to modify the appearance of the border. To change the width of the border, insert  
 → BORDER = "#"

Ex: <frameset rows="50% 50%" border=20  
 bordercolor=red>  
 <frame src="http://www.licindia.com">  
 <frame src="http://www.myindia.com">  
</frameset>

\* Working with <iframe> tag: It is defined

as inline frame. It is used to open another document. It is a container tag supported by all browsers.

Syntax: <iframe>.....</iframe>

Attributes	Parameters
1) src	path or URL
2) scrolling	auto
3) align	left, center, right
4) height	pix or %
5) width	pix or %

Ex:

```

<iframe src="htm1.png" width="200" height="120" scrolling="auto" align="left">
</iframe>
<iframe src="http://www.nareshit.com" width="300" height="200" scrolling="auto" align="left">
</iframe>

```

(SVG: Scalable Vector graphics)  
 (RIA: Rich Internet Application)

## \* FORMS \*

Forms used to display dynamic websites that allows u to interact directly with webpage. Forms r the inform' tool for the Webmaster to receive inform' from the End-user.

A form will take I/p from the end user depends on Us required option.

Forms are used to create User interact Webpages. forms are comes under DHTML.

↳ (HTML + Any client-side script)  
(JavaScript)

To create forms we use `<form>` tag. It is a container tag.

Syntax: `<form> ... </form>`

Attributes:

<u>Attributes</u>	<u>Parameters</u>
1) name	any name
2) method	get, post
3) action	url (Uniform Resource Locator)

↳ Action:

By using this attribute, we can specify the pagename to which page u want to submit control values.

2) Method:

To specify the type of method we r using submit form values. The possible methods r get and post.

Form tags:

Form tag contains the following list of subtags.

<u>Tag</u>	<u>Description</u>
------------	--------------------

<code>&lt;form&gt;</code>	Defines a form for user i/p.
<code>&lt;input&gt;</code>	Defines an I/p field data.
<code>&lt;button&gt;</code>	Defines a push button
<code>&lt;textarea&gt;</code>	Defines a text-area (a multi-line <del>text</del> <sup>I/p box</sup> )
<code>&lt;label&gt;</code>	Defines a label to the description
<code>&lt;fieldset&gt;</code>	Defines a border to the I/p data.
<code>&lt;legend&gt;</code>	Defines a caption name write into Field-set.
<code>&lt;select&gt;</code>	Defines a drop-down select list box.
<code>&lt;option&gt;</code>	Defines an option value in the drop down box.

15.09.2013,

(30)

Forms Form fields are classified into following 2 types:

- 1) Input fields
- 2) Select fields

1) Input Fields:

<u>FieldName</u>	<u>Keyword</u>	<u>Syntax</u>
Textbox	text	<code>&lt;input type="text"&gt;</code>
password box	password	<code>&lt;input type="password"&gt;</code>
checkbox	checkbox	<code>&lt;input type="checkbox"&gt;</code>
radio button	radio	<code>&lt;input type="radio"&gt;</code>
Submit button	Submit	<code>&lt;input type="submit"&gt;</code>
reset button	reset	<code>&lt;input type="reset"&gt;</code>
textarea	textarea	<code>&lt;textarea&gt; ... &lt;/textarea&gt;</code>

Ex: `<h2>Login Form </h2>`

`<form>`

Email id: `<input type="text">`

`<br/>`

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

`<br/>`

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

`</form>`

SAVE AS: FormLogin.htm

Op: Login Form

Email id:

Password:

Attributes

Parameters

1) name	anyname
2) value	anyvalue
3) size	pix
4) maxlength	number
5) rows	number
6) cols	number
7) readonly	true, false
8) disabled	disabled
9) checked	checked
10) multiple	true, false

## \* Login Form With Attributes:

```

<h2> Login Form </h2>
<form>
Email Id: <input type="text" name="Nil"
           value=name@ mail.com maxLength=6>
<br/>
password: <input type="password" name="pass"
           value=password size=10>
<br/>
<input type="Submit" value="Login" />
</form>

```

SAVE AS: FormLoginWithAttributes  
htm

Op: Login form

Email id :

Password:

## \* Example Using All Sip fields:

```

<h3> All Form Sip fields </h3>
<form>
FirstName: <input type="text" Name="FirstName" value="Pawan"/> <br/>
Password: <input type="password" Name="password" value="Nilesh"/> <br/>
checkbox: <input type="checkbox" Name="checkbox" value="check"/> <br/>
Radiobutton: <input type="Radio" Name="Radio" value="Male"/> <br/>
Reset: <input type="Reset" Name="Reset" value="clickMe"/> <br/>
TextArea: <input type="textArea" Name="Textarea" value="html2"/> <br/>
<input type="textArea" Name="Textarea" value="html2" checked="checked" />
<input type="Submit" value="submit"/>
</form>

```

SAVE AS: All Form Sip fields

First Name :

Password :

checkbox :

Radiobutton:

Reset:

Text Area:

## All Form Slip Fields.

### 1) TextArea:

It contains following Attributes.

<u>Attributes</u>	<u>Value</u>	<u>Description</u>
-------------------	--------------	--------------------

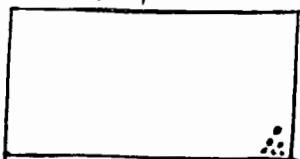
- 1) cols "Number" Define the specify no. of character visible in one line of Text Area.
- 2) rows "Number" Define the specify no. of lines visible in Text Area.
- 3) name "Message" specify unique name for Slip Element.

Ex:

```
<h4> TextArea </h4>
<Form>
<TextArea rows="4" cols="20">
</TextArea>
</Form>
```

SAVE AS: formTextArea.htm

Op:



That Area  
is 4x2  
dimension

### 2) Select:

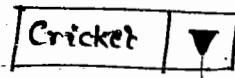
This Tag provides select particular item from drag down.  
A single element ~~is contained~~ contains name attribute with option tag.

Ex:

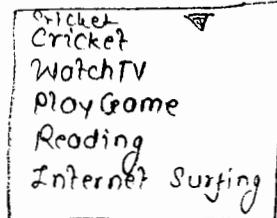
```
<Form>
  select your Hobby : <br/>
  <Select Name = "hobby">
    <option value = "Cricket" Selected> Cricket
    <option value = "Watch TV" Selected> Watch TV
    <option value = "Play Game" Selected> Play Game
    <option value = "Reading" Selected> Reading
    <option value = "Int" Selected> Internet Surfing </select>
  </Form>
```

SAVE AS: formSelect.htm

Op: Select your Hobby



After clicking  
This Button it will  
shows



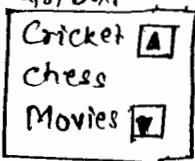
### List Box:

```
<h3> ListBox : </h3>
<Select Name = "Visible" Size = "3" multiple = "multiple">
  <option value = "1"> Cricket </option>
  <option value = "2"> chess </option>
  <option value = "3"> Movies </option>
  <option value = "4"> songs </option>
  <option value = "0"> All </option>
</Select>
```

If U put 5  
then it will display  
5 items in an Op  
Box, like here we  
put 3, so it  
displays 3  
items

SAVE AS: formListBox.htm

Op: ListBox



### Fieldset:

It defines a group of form elements as being logically related. browsers draws a box around the set of fields to indicate that they r related.

Ex: `<fieldset>`

```
<input type="checkbox" name="Cricket" value="Cricket"/> Cricket
<input type="checkbox" name="WatchTV" value="WatchTV"/> WatchTV
<input type="checkbox" name="playgame" value="play"/> Play
</fieldset>
```

SAVE AS: `formfieldset.htm`

O/P:

Cricket  WatchTV  Play

### \*LEGEND:

It is used with give a title for each set of fields. It contains the following 3 Attributes.

#### Attributes

#### Parameters

SAVE AS: `formfieldsetLegend.htm`

Align left, right, center

O/P:

Personal Style

`<FIELDSET>`

Name:

```
<LEGEND Align=Center> Personal style </LEGEND>
name: <input name="name"><br/>
email: <input name="email">
</fieldset>
```

Email:

### \* Forms with <table> Tag:

17. SEPT. 2012

```
<form> <html>
<table> <head>
<tr> <title> Working with Table </title>
<tr> <td> User Name: </td>
<td> <input type="text" /> </td> <tr>
<td> password: </td>
<td> <input type="password" /> </td>
</tr>
<tr> <td> &nbsp; </td>
<td> <input type="button" value="Login" /> </td>
</tr>
</table>
```

SAVE AS: `FormWithTable.htm`

Light Blue color

User Name:

Password :

<input type="text"/>
<input type="text"/>
<input type="button" value="Login"/>

## \* Advanced form Tags:

form is a container used to hold ifp controls. If u want to xfer the ctrl values from one page to another page (Browser To server), when User click on the Submit button.

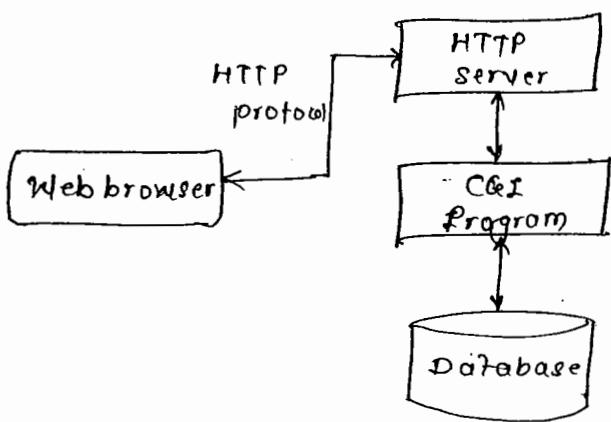
HTTP protocols contains 2 partitions head and body. The data xferd thr. body is always secure Bcoz, internally encoding.

The data xferd thr. head is Raw data (No security)

## \* Introduction to http:

HyperText xfer protocol is an Application level protocol. Using this protocol we can xfer the Resources like html files, image files, query results etc.

### ! Block diag. of http:



## \* HTTP Methods:

Method is Another Attribute used to specify the way we want to send our data to server page

### 1) GET:

In this method, We don't have security to our data & only limited data can be send to the server page. This is default method of the form.

Ex:

```
<form action = "nit.html" method = "get">
```

Ex:

```
<form action = "nit.html" method = "get">
```

Enter Name: <input name = "user">

<br/>

password: <input type = "password" name = "pass" >

<br/>

<input type = "Submit" value = "signIn" >

</form>

2) POST:

In this method, we have security for our data & we can send bulk of data to the server page.

Ex: <form action="nit.html" method="post">

<form action="nit.html" method="post">  
 Enter Name: <input name="user">  
<br/>  
 Password: <input type="password" name="pass">  
<br/>  
<input type="submit" value='SignIn'>  
</form>

Action :

This attribute is used to specify the URL of the server page ~~to~~ to which we want to send our form data.

Ex: <form name="myform" action="user.aspx">

★ DIFF. BETW GET AND POSTGET↳ Max<sup>n</sup> websites

use post method

POST

- |                                    |                               |
|------------------------------------|-------------------------------|
| 1) Data is visible on URL Address. | 1) Not visible POST informat. |
| 2) Unsecured                       | 2) Highly Secured.            |
| 3) Excellent Performance           | 3) Good Performance.          |
| 4) Xfers ltd. amt. of data         | 4) Xfers huge amt. of data    |
| 5) Unable to upload files.         | 5) Able to Upload files.      |

Q&P:

## Form Example

## User Information:

First Name:

Last Name:

User Details:

Username:

password:

Male/Female Details:

O Male

O Female

Choose Your Country:

India

 ★ Sample project on Form Tag:

```

<font size=4 face=Arial>
<p align="center"><font size=6 face=cursive
  style=background-color:
  yellow>Form Example
</font></p>

<form name="input" action="#" method="post">
<fieldset>
<legend>User Information:</legend>
First Name: <input type="text" name="user1"/>
Last Name: <input type="text" name="lastname1"/>
password: <input type="password" name="password1"/>
<input type="checkbox" name="gender1" value="M" checked="checked"/> Male
<input type="checkbox" name="gender1" value="F"/> Female
<br/> Choose Your Country:
<input type="radio" name="country1" value="India"/> India
<input type="radio" name="country1" value="USA"/> USA
<br/> Submit      Reset
</fieldset>

```

```
<fieldset>
<legend> User details: </legend> username: <input type="text" name="User2"/>
<br/>
password:
<input type="password" name="password2"/>
</fieldset>

<fieldset>
<legend> Male/Female Details: </legend> <input type="radio" name="sex" value="male"/>
Male <br/>
<input type="radio" name="sex" value="female" female </fieldset>
<fieldset>
<legend> choose Ur country: </legend> <br/> <select Name="select">
<option value="India"> India </option>
<option value="United state"> United states </option>
<option value="Australia"> Australia </option>
<option value="New Zelend"> New Zelend </option>
<option value="England"> England </option>
</select>
<br/>
</fieldset>
<br/>
<input name="submit" type="submit" value="submit"/>
<input name="Reset" type="Reset" value="Reset"/>
</form>
</font>
```

## \* Lists \*

In HTML, we are having 3 Types of List.

- 1) ORDERED List
- 2) Unordered List
- 3) Definition List

### 1) Ordered List: / Numbered List

It is also called as Numbered List. It is used to give nos. to the list items.  
It is a container tag.

Syntax: <ol> . . . </ol>

To specify the list items, we use <li> Tag.  
It is also a container tag

Syntax: <li> . . . </li>

Ex: <html>

```
<head>
  <title> Lists in HTML </title>
</head>
<body>
  <ol type=A>
    <li> HTML </li>
    <li> CSS </li>
    <li> JavaScript </li>
    <li> JQuery </li>
    <li> Ajax </li>
    <li> HTML5 </li>
  </ol>
</body>
</html>
```

Q.P.:

- 1. HTML
- 2. CSS
- 3. JavaScript
- 4. JQuery
- 5. Ajax
- 6. HTML5

### \* Attributes

### Parameters

- 1) type i, li, a, A, 1
- 2) start Any Number  
↳ Applicable only for Numbers

```
<html>
<head>
  <title>
    </title>
  </head>
  <ol type=A>
```

18. SEPT. 2013

(37)

### 2) Unordered List: / Bulleted List:

It is also called as Bulleted List. It is used to give Bullets to the List items. It is a container tag.

Syntax:

<ul> . . . </ul>

To specify the list items we use <li> Tag. It is also a container tag.

Syntax:

<li> . . . </li>

Ex.:

```
<ul>
  <li> HTML </li>
  <li> CSS </li>
  <li> JavaScript </li>
</ul>
```

<h2> Unordered List </h2>

```
<ul>
  <li> HTML </li>
  <li> CSS </li>
  <li> JavaScript </li>
</ul>
```

### Attributes

type

### Parameters

disc, circle, square  
↓  
(By default)

Ex: <ul> Unordered List </ul>

```
<ul type=circle>
  <li> HTML </li>
  <li> CSS </li>
  <li> JavaScript </li>
</ul>
```

<h2> Unordered List </h2>

```
<ul type=square>
  <li> HTML </li>
  <li> CSS </li>
  <li> JavaScript </li>
</ul>
```

### 3) Definition List:

It is also called as Descriptive List. It is used to give def<sup>n</sup> to Def<sup>n</sup> terms. It is a container Tag.

Syntax:

```
<dl>...</dl>
```

To specify def<sup>n</sup> data, we use <dd> tag. It is a container Tag.

Syntax:

```
<dd>...</dd>
```

To specify def<sup>n</sup> Term, We use -

<dt> Tag. It is also a container Tag

Syntax:

```
<dt>...</dt>
```

Ex:

```
<dl>
  <dt>.NET</dt>
  <dd>It is a kind of framework from Microsoft. It is programmer friendly Technology.</dd>
  <dt>HTML</dt>
  <dd>It is Hypertext Resources, It is from W3C Organization. Latest version is HTML5.</dd>
  <dt>JQuery</dt>
  <dd>It is a collection of JavaScript libraries from jquery Corporation.</dd>
</dl>
```

SAVE AS:

Output:

\* **Address Tag:** shows address in italic

(38)  
<address> Tag is used to indicating an address. An Address is usually renders in italic. It is a container Tag.

Syntax:

```
<address>...</address>
```

Ex:

```
Manewada, Vinkar colony,<br/>
```

```
Quarter NO. 66, <br/>
```

```
Nagpur <br/>
```

```
440027 <br/>
```

```
India.
```

</address> SAVE AS:

O/P:

\* **blink Tag:** supports only Firefox

Using this tag, Makes text blink repeatedly. It is a container tag

Syntax:

```
<blink>...</blink>
```

Ex: <blink>

Nilesh

```
</blink>
```

(Note: It doesn't support Internet Explorer & chrome Browsers)

\* **code Tag:** Font displays like dos prompt face

It allows the user to specify code or a command that generates a different font to signify the code

Syntax:

```
<code>...</code>
```

Ex:  
`<code>`

It is Dos kind of font face

`</code>`

\* `<ins>`:

stands for insert. By using `<ins>` we can insert on html document.

Syntax:

`<ins> ... </ins>`

Ex:

Collection of JavaScript libraries are

`<ins>jQuery</ins>`

O/P:

\* `<NOBR>`:

By using this tag, we can continue the line without break.

NOBR stands for No Break.

Syntax:

`<nobr>...</nobr>`

Ex:

`<nobr>` collect<sup>n</sup> of Javascript Libraries are collect<sup>n</sup> of javascript are Javascript Libraries are `</nobr>`

## \* `<META>` Tag: \*

Introduction:

Meta means inform<sup>n</sup>. Metadata means data abt data. OR Data Referring some other kind of data.

`<meta>` tags are used to store inform<sup>n</sup> relevant to browsers & search engines.

`<meta>` Tag provides metadata abt html document. This data will not be displayed on the page. Meta elements are typically used to specify page descript<sup>n</sup>, keywords, author of the document etc.

<u>Attributes</u>	<u>value</u>	<u>Description</u>
1) <u>content</u>	"text"	Specifies content of webpage
2) <u>http-equiv</u>	i) "Content-type" ii) "Content-style-type" iii) "expires" iv) "refresh"	Specify http header information. Specify the time of expires. Specify the time of refresh
3) <u>name</u>	v) "set-cookie" i) author ii) description iii) keywords	Specify the limit of cookie. Specify the name of author. Specify the website content. Specify the website content keyword.

`<meta>` tag contains the following 2 metavalue

1) keywords

2) Descript<sup>n</sup>.

Ex:

- Go to any search Engine like Google, yahoo, etc. Enter Any keyword (Cricket, holidays, Tours, packages etc)
- It displays list of Website Results.
  - Go to one particular Website.
  - View source find keywords.
  - Those keywords are Meta keywords.

### \* Rules for Metatype:

- Used meta tags with care.
- Do not include words that are not present on your page.
- Do not repeat words etc.
- <Meta> tag always inside the head Element.  
It can be used by browsers, Search Engines or other Web services.

### Keywords:

This metatag was intended to be keywords with special relevance for the page. Bcz, of Misuse, Many search Engines skip ur website, Sometimes, These are treating as spam pages.

Ex:

```
<meta name="keywords" content="html, webdesign, webdevelopment, javascript, css, jquery, html5, css3.. .>
```

### Description:

Most search engines will display the description when they list results from a search. If u do not include this tag, then the search engine will simply list the 1st words on the page.

Ex:

```
<metaname="description" content="It is good Website for S/W materials & S/W">
```

### \* Autoload pages:

You can use refresh meta tag to automatically load a page after a few seconds.

Syntax: <meta http-equiv="refresh" Content=5; url="AnuradhaLecture">

Ex:

Ex:

20. SEPT. 2023

40

```
<html>
<head>
<title>Working with Meta Tag </title>
<center> Meta </center>
Meta Tag
<meta http-equiv="refresh"
Content="5; url=http://www.nareshit.com">
</head>
</html>
```

### RSAC:

Recreational SW Advisory Council  
on the Internet. It is Renamed as  
ICRA (Internet Content Rating Association)

To check rating on Internet Explorer  
Go to Tools menu, select Internet Option,  
Click on Content Tab, Enable content  
Advisor.

### Introduction to DHTML:

It is a term for

collection of Technologies used together to create interactive & animated website.

By using a combination of Strategic markup lang., a client-side scripting lang.

Ex:

```
<html>
<head>
<title>
DHTML
</title>
</head>
<body>
<script>
```

```
document.write("<h1> Welcome to HTML </h1>");
document.write("<i> It is static format </i>");
```

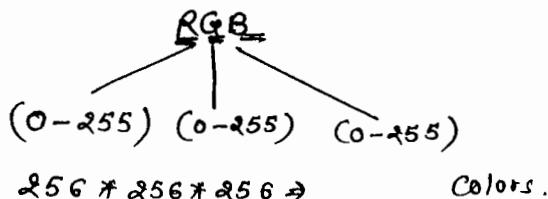
</script> Save As: Dhtml1.htm

<h1> Welcome to HTML
It is static format

## \* Features of DHTML:

- i) DHTML allows animate Text & Images.
- ii) We can move Elements Independently.
- iii) Auto Refresh.
- iv) Implementing of Rollover buttons & DropDown Menus.

## \* HTML Colors:



As per W3C, The following 16 color names are Recommended, They are as following:

- 1) Black 2) Gray 3) Silver 4) White
- 5) Yellow 6) Lime 7) Aquamarine 8) Fuchsia
- 9) Red 10) Green 11) Blue 12) Purple
- 13) Maroon 14) Olive 15) Navy 16) Teal

## \* Color values:

html colors are defined using Hexadecimal Notat's. for the Combinat' of Red, Green & Blue color values. The lowest value is the light source, The highest value is 255 (FF). These Values are specified as 3 pairs of 2-digit nos, starting with a # sign.

## \* HTML Colors Hexa Codes: (16 million diff. colors)

is a 6-digit representation of a color. 1st to represent Red value, Next to represent Green value, Last to represent Blue value.

- #FF0000 → Red
- #00FF00 → Green
- #0000FF → Blue
- #000000 → Black
- #FFFFFF → White

## \* HTML Styles:

### HTML span Tag:

It is a basic html tag. It is used to apply inline style on specific character.

By using this Tag we can display formatted Text on the Browser.

#### Syntax:

<span>...</span>

#### Ex:

<html>

<title> Span Tag </title>  
</head>

<body>

<span style="font-size:40; color:red">

It is a span Tag </span>

</body>

</html> SAVE AS: SpanTag.htm

Output: It is a spanTag.  
Font size 40 in red color

### HTML div Tag:

div stands for division

It can hold the content of other html elements. By using this Tag we can design the Webpage as a Multiple division.

#### Syntax:

<div>...</div>

#### Attributes

#### Properties

- 1) align ⇒ It accepts values like left, Right, center
- 2) style ⇒ It provides the inline style.
- 3) id ⇒ Using this attribute we can modify the elements in JavaScript lang
- 4) class ⇒ Attribute of html <div> tag accepts the name css class to apply the styles.

Ex:

```

<html>
  <head>
    <title>
      DIV Tag </title>
    </head>
    <body>
      <div>
        <p> It is a para </p>
        <b> It is a Bold </b>
        <i> It is an Italic </i>
      </div>
    </body>
</html>

```

\* <div> Tag with style attribute:

→ <div>

Content placed inside HTML div Tag. <br/>

U cannot see the Div Tag. <br/>

Boundaries without using CSS styles. <br/>

</div>

<br/>

<div style="border: solid 3px #FF0000">

Content placed inside HTML DIV Tag. <br/>

CSS style border property. <br/>

It is used to visualize Boundaries of this Div

Tag. <br/>

</div>

SAVE AS: DIVTag1.htm

O/P:

Content placed inside HTML DIV Tag

U cannot see the Div Tag.

Boundaries without using CSS styles.

Content placed inside HTML DIV Tag.

CSS style border property.

It is used to visualize Boundaries of this DIV Tag.

→ Border in Red color

Solid 3px → Means Border Thick size

3

Ex:

```

<html>
  <head>
    <title> DIV Tag </title>
  </head>
  <body>
    <div align="left">
      <p> It is a paragraph </p>
      <b> It is a Bold </b> <br/>
      <i> It is an Italic </i> <br/>
    </div>
    <div align="center">
      <p> It is a paragraph </p>
      <b> It is a Bold </b> <br/>
      <i> It is an Italic </i> <br/>
    </div>
    <div align="right">
      <p> It is a paragraph </p>
      <b> It is a Bold </b> <br/>
      <i> It is an Italic </i> <br/>
    </div>
  </body>
</html>

```

O/P:

It is a paragraph

It is a Bold

It is an Italic

It is a paragraph

It is a Bold

It is an Italic

It is a paragraph

It is a Bold

Ex: `<div style="background-color:orange; width:100%; height:100%>` 1. SEPT. 2014

`<span style="font-size:30"> It is Div1 </span>` ↗ for full length.  
`</div>` (Mandatory)

`<div style="background-color:white; width:100%; height:100%">`

`<span style="font-size:10"> It is Div2 </span>`

`<center></center>`

`<div style="background-color:green; width:100%; height:100%">`

`<span style="font-size:30"> It is Div3 </span>` SAVE AS: Tiranga.htm  
`</div>`

Note: See also

Tiranga.htm

<u>O/P:</u>	
It is div1	→ orange
It is div2	→ White
It is div3	→ Green

Ex:

`<div style="background-color:orange; width:100%; height:100%">`

`<span style="font-size:30"> It is Div1 </span>`

`</div>`

`<div style="background-color:red; width:150; height:400; float:left">`

`<span style="color:lightgreen"> It is Div2 </span>`

`</div>`

`<div style="background-color:pink; width:400; height:400; float:left">`

`<center></center>`

`</div>`

`<div style="background-color:green; width:180; height:400; float:left">`

It is Div4

`</div>`

SAVE AS: DivSmiley.htm

O/P:

Ex: (TLDs)  $\Rightarrow$  Top Level Domains

(4c)

.in : India

.gov : Government Agencies

.edu : Educational Institutions

.org : Organizations (nonprofit)

.mil : military

.com : commercial Business

.net : N/W organizat's.

.ca : Canada

### \* Avoid following:

- $\Rightarrow$  While U r choosing a domain name, remember following pts.
  - i) If possible do not use Numerical characters.
  - ii) If possible do not use mobile phone txt msg abbreviat's.
  - iii) If possible do not use '2' for 'to', '~~4~~' for 'for', 'u' for 'you'.
  - iv) If possible do not give ur domain name with unnecessary words like 'the', ~~the~~ 'LTD' and 'company' etc.

### \* Do It:

While U r choosing a domain name, Try to follow the following pts.

- i) Keep a domain name short.
- ii) Make sure it is easy to spell.
- iii) Make sure it is easy to pronounce.
- iv) Use simple & plain English.
- v) Use Real words from Dictionary.
- vi) Use search engine keywords.
- vii) Choose right domain extensions, like .com, .in etc.

## Q: How To Register a Domain Name?

$\Rightarrow$  To create a unique address in World Wide Web (WWW),  
visit the following sites:

- 1) WWW.godaddy.com
- 2) WWW.byethost.com
- 3) WWW.noads.biz

### Step 1:

Go to Any WebSite (byethost.com)

Step 2: Click on free hosting Note: (Always use Mozilla Firefox browser)

### Step 3:

click on sign up for free hosting link.

~~It~~ displays sign up form for free Hosting.

# ★ Website ★

SEPT. 2023

(45)

## Miscellaneous Tags:

### 1) <bdo> :

(Bidirectional Overwrite) By using this tag we can override current text dir'

It is a container tag.

#### Syntax:

```
<bdo> . . . </bdo>
```

#### Ex:

```
<html>
```

```
  <body>
```

<bdo dir="rtl"> Here is some Text that should be written from right to left </bdo>

```
  </body>
```

```
</html>
```

### <param> :

It is used to define parameters for plugins, embedded with an object element. It is an Empty Tag.

#### Syntax:

```
<param>
```

#### Ex: <body>

```
  <object data="Windows XP Shutdown.Wav">
```

```
    <param name="autoplay" value="true" />
```

```
  </object>
```

```
</body>
```

#### O/P:

We can hear the sound of Shutdown of Windows XP.

## Introduction To Web Hosting :

It is a service, a Type of Internet hosting service that allows individuals or organizations to make their own Website accessible via the World Wide Web.

### \* Domain Name:

Domain Names are used to identify one or more IP addresses.

Domain Name consist of following 2 parts.

Ex: www.nareshit.com (TLD)

Here, .com represents Top Level domain, nareshit represents Secondary level domain (SLD). SLD name we can provide upto 128 characters length.

Every Domain Name has a suffix that indicates which Top Level domain it belongs to.

Ex: (.TLDs)  $\Rightarrow$  Top Level Domains.

(4c)

in : India

gov : Government Agencies

edu : Educational Institutions

org : Organizations (nonprofit)

mil : military

com : commercial Business

net : N/W organizat's.

ca : Canada

### \* Avoid following:

$\Rightarrow$  While U r choosing a domain name, remember following pts.

i) If possible do not use Numerical characters.

ii) If possible do not use mobile phone txt msg abbreviat's.

iii) If possible do not use '2' for 'to', '~~4~~' for 'for', 'u' for 'you'.

iv) If possible do not give Ur domain name with unnecessary words like 'the', ~~LTD~~ 'LTD' and 'company' etc.

### \* Do It:

While U r choosing a domain name, Try to follow the following pts.

i) Keep a domain name short.

ii) Make sure it is easy to spell.

iii) Make sure it is easy to pronounce.

iv) Use simple & plain English.

v) Use Real words from dictionary.

vi) Use search engine keywords.

vii) Choose right domain extensions, like .com, .in etc.

## Q: How To Register a Domain Name?

$\Rightarrow$  To create a unique address in World Wide Web (WWW), visit the following sites:

- 1) WWW.godaddy.com
- 2) WWW.byethost.com
- 3) WWW.noads.biz

### Step 1:

Go to Any Weblink (byethost.com)

Step 2: Click on free hosting Note: (Always use Mozilla Firefox browser)

### Step 3:

Click on SignUp for free hosting link.

~~It~~ It displays signup form for free Hosting.

6) <Dir>:

It is used to list directory titles. It is a container tag.

Syntax:

<dir> ... </dir>

Ex:

```
<dir>
  <li> HTML5 </li>
  <li> CSS </li>
</dir>
```

7) <font> Tag:

It is used to format a text.

Syntax:

<font> ... </font>

Ex:

```
<body>
  <p><font size="3" color="red" face="Tahoma"> Welcome to Nagpur </font></p>
</body>
```

8) <frameset> Tag:

By using this Tag, we can divide multiple frames.

9) <frame> Tag:

It is used to call external webpages.

Ex:

```
<frameset rows="50%,50%">
  <frame src="http://www.myweb.com">
  <frame src="http://www.eftours.com">
</frameset>
```

10) <noframes> Tag:

To display frame related errors.

Syntax:

<noframes> ... </noframes>

11) <s> :

<strike>: } Using These tag we can specify deleted content in a HTML document

Ex:

<s> It is a removed content </s>  
<br/>

<strike> It is a removed content </strike>

12) <t>:

By using this tag, we can display Teletype formatted Text.

Ex: <t> It is a teletype formatted text </t>

13) <pre> /<xmp>:

It displays preformatted Text.

Ex:

```
<pre> It is preformatted text </pre>
<xmp> It is preformatted text </xmp>
```



# CASCADING STYLE SHEETS

**css**

CASCADING  
STYLE  
SHEETS

## \* CSS \*

13. Sept. 2013

(53)

- CSS stands for Cascading Style Sheet. used to describe the way how to provide styles to the html element. It is client-side scripting language.
- \* Style object is providing no. of properties. By using CSS, we can apply these properties to the html elements in diff. ways.
- \* The properties of style object or color, font, hyper, size, cursor etc.

### \* Types of style sheets:

#### 1) inline style:

- If we apply the styles within the <html> tag with the help of style attribute, comes under inline styles, inline styles we cannot access from another html element.

Ex: <input type="text" style="color: blue; font-size: 30">

#### 2) internal style sheet:

- If we declare styles within the style tag with the help of selectors comes under Internal styling.

Internal styles we can apply on any element within the webpage.

- Selector is a group of style properties. diff types of selectors are available like class selector, id selector etc.

Ex: <style type="text/css" language="CSS">

• abc{

    color: green;  
    cursor: pointer; // help / wait  
    text-decoration: underline;  
}

</style>

<div class="abc">

    Hello

</div>

<input class="abc" type="button" value="Click" />

#### 3) External style sheet:

If we declare styles in external webpage comes under these concept. External styles we can link from any webpage.

By using <link> tag we can call the external styles from html page.

<Link> tag contains some properties.

i) rel:

By using this property we can specify the relationship b/w the html page & external style sheets.

## External CSS

By using this property, we can specify the path of external ~~style sheets~~ 54 ~~style sheets~~ ~~style sheets~~.

Mystyle.css

```

• styles {
    color: blue;
    font-family: Arial;
}
• styles 2 {
    color: green;
    font-weight: bold;
}

```

html  
<link href="mystyle.css" rel="stylesheet">  
<p class="style2">  
This is paragraph  
</p>  
<div class="style2">  
This is Division  
</div>

## Types of Selectors:

### i) Type selector:

If we create a selector, with html element name as 'selector name' comes under this category. If u want to apply similar properties to the all elements of a type ~~comes under~~ we can go for type selector.

Ex: <style>

```

div {
    color: blue;
}
</style>
<div>Hi</div>
<div>Hello</div>
<p>This is Paragraph</p>

```

Ex: <style>

body {

```

background-repeat: no-repeat;
font-size: 30px;
margin: 20px;
text-align: justify;
background-attachment: fixed;
}

```

</style>

```

<body background="13.jpg">
content
</body>

```

### ii) id selector:

If we create any selector with hash (#) character comes under id selector. id selector we can apply on html elements using id attribute. We cannot apply id selector to multiple elements becoz, we r calling the id selector from html elements, id attribute we should not apply same id to the multiple elements.

Ex: <style>

```

#sel1 {
    color: red;
    font-size: 30px;
}
</style>

```

### iii) class selector:

It is same as id selector But we can apply these properties to multiple elements.

Class selector names start with . character and we can call this selector from elements using class attribute.

Ex:

```
<style>
  .cls1 {
    border: 5px dotted green;
    width: 100px;
    height: 100px;
  }
</style>
<span class="cls1">
  Hello
</span>
<div class="cls1">
  
```

### v) descendant selector:

14. SEPR. 2019  
55

This selector contains

minimum 2 Elements. The 2nd Element has been descended to the first Element. If u want to apply properties to an Element based on another element we can go for descendant selector.

Ex: <style>

```
div a{
  text-decoration: none;
  font-size: 30px;
  color: red;
}
```

</style>

```
<a href="abc.php">Link1</a>
<div>
<a href="dfsd.php">Inner Div</a>
</div>
```

### vi) adjacent sibling selector:

If u want to apply properties to the elements based on the adjacent elements, we can go for this selector.

This selector is not supported by all browsers.

<style>

```
li + li + li {
  color: red;
}
```

</style>

<ul>

<li> PHP </li>

<li> ASP </li>

<li> JSP </li>

</ul>

## What is attribute selector:

Sometimes, we use pseudo elements:

Need to provide ~~same~~ styles to the elements based on element properties values that time we can go for Attribute Selectors.

Ex:

```
<style>
a[href='abcd.php'] {
    color: red;
    font-size: 30px;
}
</style>
<a href='abc.php'>Link1</a>
<a href='abcd.php'>Link2</a>
```

## ~~Selectors based on attribute~~

56

Sometimes, ~~we~~ we need to provide the styles to the 1<sup>st</sup> character of 1<sup>st</sup> line of an element that time we can go for pseudo elements.

Ex:

```
<style>
p:first-letter {
    color: white;
    background-color: green;
    font-size: 50px;
}
</style>
```

<p>Welcome To PHP  
<br>This is second  
</p>

## \* Pseudo classes \*

Sometimes, we are applying styles to the elements based on the actions what we are performing on the elements we can go for pseudo ~~dynamic~~ classes or dynamic, we have different types of pseudo classes.

```
<style>
a:link {
    color: green;
    font-size: 30px;
}
a:hover {
    color: red;
    text-decoration: none;
    font-size: 50px;
}
a:active {
    color: yellow;
}
a:visited {
    color: black;
}
</style>
```

## \* group selector:

To apply the properties to multiple elements based on id, class name, element tag, etc. We can go for group selector. It is group of different selector types.

Ex:

```
<style>
div, .cls1, #p1 {
    color: green;
    font-weight: bold;
}
</style>
```

<div> Hi </div>

<p id="p1">

This is paragraph

</p>

<span class="cls1">

This is span

</span>

```
<Style>
  .span{
    color: red;
    background-color: yellow;
  }
</Style>
```

Welcome to ~~www~~  
 Welcome Implemented  
PHP in 1995 Rasmus Lerdorf in 1995

\* 15. SEPT. 2013 \*

### \* Visibility:

By using this property, we can visible or hide the elements on webpage.  
 It contains 2 properties visible and hidden.

```


```

### \* position:

By using this property we can change the position of a ctrl. It contains 2 possible values absolute or relative.  
 By using position absolute we can display the ctrl in absolute top and left posis.

The Default position property of every control is relative.

```


```

### \* Z-index:

By using this property we can change the z-position of control

```
<Style>
  #img2{
    z-index: -1;
  }
</Style>
```

```


```

### \* Overflow:

By using this property we can hide OR we can provide scroll bar to the overflow content.

```
<Style>
  #div2{
    background-color: lightblue;
    width: 100;
    height: 100;
    overflow: scroll;
  }
</Style>
```

```
#div1{
  background-color: lightyellow;
  width: 100;
  height: 100;
  overflow: hidden;
}
</Style>
```

```
<div id="div1"> Content </div>
<div id="div2"> Content </div>
```

Ex: `<style>`

```

    li {
        position: relative;
    }
</style>
<ul>
    <li>
        
    </li>
</ul>

```

filters

style object is providing filter property with no. of values, we can apply this filter on html elements. filter does not support all browsers.

~~~~ The available filter properties are fliph(), flipv() etc.

1) flipv():

By using this we can change the vertical posit?

``

2) fliph():

By using this property, we <sup>can</sup> change the horizontal dir<sup>n</sup> of element.

Ex:

3) xray():

To get the xray style of an element.

4) invert():

To apply invert colors on html element.

5) blur():

By using this property we can apply the blur <sup>property</sup> on the elements.

6) gray():

By using this property we can change the color photo into black & white  
But not Black & white into color.

Ex: ``

Note:

By using this we can display a ctrl in wave format.

```

```

alpha:

By using this property we can apply opacity on the elements.

Ex:

```

```

18 SEP 2019

```
<style>
body {
    margin: 100 100 100 100;
}
ul li {
    list-style: none;
    position: relative;
    float: left;
}
img {
    width: 100;
    height: 75;
}
ul li a #imgbig {
    display: none;
}
ul li a:hover #imgbig {
    display: block;
    width: 200;
    height: 150;
    position: absolute;
    left: -50;
    top: -40;
    z-index: 1;
    border: 4px solid green;
    border-color: silver green blue yellow;
}
</style>
<ul>
<li>
```

```
<a href="#">

</a>
</li>
```

```

<script>
function fun1()
{
    alert("hi")
    alert("hello")
}
</script>
<input type="button" value="click" onclick="fun1()"/>

```

Ex: 2) <script>

```

function fun1()
{
    x = document.getElementById('t1').value alert(x)
    y = document.getElementById('txt2').style.width alert(y)
    document.getElementById('txt2').style.color = "blue"
    document.getElementById('but1').value = "Go"
    document.getElementById('but2').style.visibility = 'hidden'
}
</script>
<input type="text" id="t1">
<br>
<input type="text" id="txt2" style="width: 100%">
<input type="button" value="click" onclick="fun1()" id="but1">

```

Ex: <style>

```

#p1 {
    color: white;
    font-size: 30px;
    font-weight: bold;
    text-align: right;
    cursor: pointer;
}
#divcontainer {
    position: absolute;
}

```

```

#divhead {
    width: 250px;
    height: 30px;
    background-color: black;
}

```

```

<script>
function funclose()
{
    document.getElementById('divcontainer').style.visibility = 'hidden'
}
function fun1()
{
    tpos = document.body.scrollTop
    document.getElementById('divcontainer').style.top = tpos
}
</script>
<body onscroll="fun1()">
<div align="center">
    

```

```

</div>
<div id="divcontainer">
<div id="divhead">
<p id="p1" onclick="funclose()">X </p>
</div>

</div>
content

```

20-SEPT-2019

Ex: Example for Advertisement ~~Advertisement~~

```

<style>
#innerdiv{
    width: 610;
    position: absolute;
    left: 400;
}
#container{
    position: relative;
    overflow: hidden;
    width: 400;
    height: 150;
    border: 5px double white;
}
</style>
<script>
lpos = 400;
function func1()
{
    lpos = lpos - 2
    if (lpos <= -600)
        lpos = 400
    document.getElementById('innerdiv').style.left = lpos
    setTimeout("func1()", 0);
}
</script>
<body onload="func1()" bgcolor="lightblue">
<div id="Container">
<div id="innerdiv">



</div>
</div>

```

```

Ex: <style>
# container{
    background-color: yellow;
    width: 150;
    height: 200;
    padding-left: 50;
}

# container # innerdiv{
    background-color: green;
    width: 150;
    height: 200;
}

# container # innerdiv ul li{
    padding-left: 0;
    width: 100%;
}

# container # innerdiv ul li{
    list-style: none;
    padding-left: 0;
    padding-bottom: 10;
    border-top: 1px solid silver;
    border-bottom: 1px solid silver;
}

# container # innerdiv ul li a{
    color: white;
    text-decoration: none;
    font-weight: bold;
}

# container # innerdiv ul li a:hover{
    color: red;
    font-family: arial;
    font-size: 20;
}

<style>
<div id="container">
<div id="innerdiv">
<ul>
<li><a href="#">home</a></li>
<li><a href="#">about us</a></li>
</ul></div></div>

```

# \* Introduction To CSS (Cascading style sheets) \*

(68)

## HTML special characters:

HTML contains following list of special characters.

| <u>Character</u> | <u>Decimal Entity</u> | <u>Name Entity</u> | <u>Description</u>          |
|------------------|-----------------------|--------------------|-----------------------------|
| ©                | §#169;                | &copy;             | Copyright symbol            |
| ™                | §#8482;               | &trade;            | Trademark symbol            |
| ®                | §#174;                | &reg;              | Registered symbol           |
| '                | §#8216;               | &lsquo;            | Left/opening single quote   |
| '                | §#8217;               | &rsquo;            | Right/closing single quote  |
| “                | §#8220;               | &lquo;             | Left / opening Double quote |
| ”                | §#8221;               | &rquo;             | Right/ closing Double quote |
| ←                | &larr;                |                    | Leftward Arrow              |
| ↑                | &uarr;                |                    | Upward Arrow                |
| →                | &rarr;                |                    | Rightward Arrow             |
| ↓                | &darr;                |                    | Downward Arrow              |
| ♠                | &spades;              |                    | spade suit                  |
| ♣                | &clubs;               |                    | club suit                   |
| ♥                | &hearts;              |                    | Heart suit                  |
| ♦                | &diams;               |                    | Diamond suit                |
| ¥                | &yen;<br>&euro;       |                    |                             |

Ex: <html>

<head> — → &ndash;  
— → &mdash;

<title>HTML special characters</title>

Output:

Save As:

Specialcharacters.htm

©

TM

®

‘

“

”

←

↑

→

↓

♠

♣

♥

♦

```

</head>
<body>
<br/>§#169;
<br/>§#8482;
<br/>§#174;
<br/>§#8216;
<br/>§#8217;
<br/>§#8220;
<br/>§#8221;
<br/>&larr;;
<br/>&uarr;;
<br/>&rarr;;
<br/>&darr;;
<br/>&spades;
<br/>&clubs;
<br/>&hearts;
<br/>&diams;
</body></html>

```

## \* Cascading Style Sheets: (64) 22 of 1000

### Cascading:

Multiply styles can overlap in order to specify a range of style for whole website down to unique element. Cascading inheriting features in a chain sys.

### Style:

CSS deals specially with a presentation domain of designing a webpage (color, font, layout etc.)

### sheet:

It is a CSS file separated from HTML & linked to HTML file.

## \* Types of Attributes:

Attributes r divided into following types.

### 1) Element Specific Attribute:

means tag

Ex: <body> → bgcolor, background, text

<img> → src, width, height, alt, align...

### 2) Global Attributes:

These attributes are common for all Elements.

1) class 2) id 3) lang 4) insert 5) spellcheck 6) style.

### 3) Event Handler Content Attribute:

are related to Javascript.

1) onchange 2) oninput 3) onprogress 4) onclick 5) oninvalid etc.

25. SEPT. 2019

## Q: What is CSS?

⇒ CSS stands for Cascading Style Sheets. Using these sheets we can manage websites page layout easily. CSS has satisfying the following list of pts.

- styles normally stored in a style sheets.
- Cascading means inheriting the features.
- CSS we can embed with HTML, javascript, jquery etc.
- extension of CSS file is ".CSS"

## \* css Versions History:

CSS has following list of versions.

### 1) CSS 1.0:

It is the earlier version from W3C, W3C recommended in 1996

It supports diff. properties typography like fonts, alignments, spacing, margin etc.

### 2) CSS 2.0 (1998):

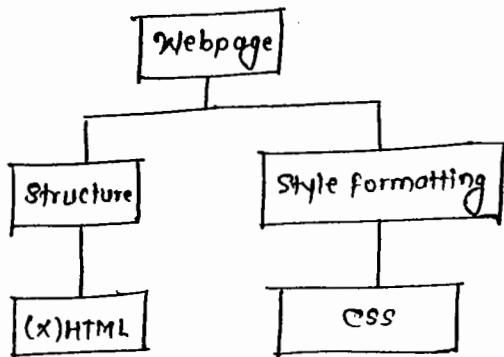
This version contains advanced features like powerful selectors, CSS Box models, diff. posits etc.

### iii) CSS 3.0 (2008):

This version is divided into diff. modules. These modules contains all advanced features. This version supporting all modern browsers.

Q: Why to use styles?

→ Styles are basic formatting tool more flexible & easy to implement.



#### XHTML

- i) Should follow the same order in which U open the tag, like  
`<b><p><i>...</b></p></i>`

- ii) There is ~~not~~ <sup>Compulsory</sup> close those tags which is opened.

- iii) Don't use Capital letters in Tag.

#### HTML

- i) No need to follow the same order in which U open the tags.

- ii) There is no need to close all tags which is already opened.

- iii) We can use anywhere capital letters.

### CSS Advantages:

#### 1) Accessibility:

We can access the required styles on any Webpage,

#### 2) Code Rendering:

We can call ess layouts from any Webpage to other pages.

These styles improve website performance.

#### 3) Flexibility:

We can ~~not~~ update style sheets as per the end user needs.

#### 4) Designing:

Style contains Huge collect<sup>n</sup> of properties to design Webpages more attractive.

#### 5) Easy to Manage:

#### 6) Global Change:

#### 7) Save a lot of Time:

#### 8) Inline External style sheets.

#### 9) Multiple device compatibility.

#### 10) Global web standards.

## \* Structure of CSS:

```
<html>
  <head>
    <style type="text/css">
      {
      }
    </style>
  </head>
  <body>
  </body>
</html>
```

Ex:

```
<html>
  <head>
    <title> CSS Structure </title>
    <style type="text/css">
      p ← Element selector/Tag selector
      text-align: right;
      font-size: 40px;
      color: red;
      text-decoration: underline;
    </style>
  </head>
```

(66)

```
  <body background="goodmorning.gif" style="background-repeat: no-repeat"
        bgcolor=lightgreen>
    <p> It is CSS Example </p>
  </body>
</html>
```

## \* CSS Syntax:

Generally, CSS syntax contains following 3 parts:

- 1) Selector
- 2) Property
- 3) Value.

### 1) Selector:

Selector is normally `<html>` tag or element that u wish to define.

### 2) Property:

It is the attribute u wish to change.

### 3) Select:

Syntax: Every property has a value.

Selector      Declaration

→ `h1` {color: blue; font-size: 12px; }  
         ↑      ↑      ↑      ↑  
        Property   Value   Property   Value

Ex:

```
<style>
  h1 ← Element/Tag selector
  {
    color: blue;
    text-align: center;
  } Declaration property: Value
</style>
```

`<h1>` It is a Tag/Element selector  
`<h2>` It is not a Tag selector

It is not in CSS selector.

SAVE AS:

CSS Tag Selector.htm

Opp:

It is a Tag/Element Selector → blue Color

It is not a Tag selector.

## \* CSS Comments:

Comments are used to explain ur code, & may help u when u edit the source code at a later date. Comments r ignored by browsers.

CSS comments begins with "/" and ends with "\*/", like this:

```
/ * This is a comment */
```

Ex:

```
p
{
    text-align: center;
    /* This is another comment */
    color: black;
    font-family: arial;
}
```

i) Comments are used to explain our code  
ii) Comments are non executable stat. or ignore statements.

iii) Every Technology contains comments,  
iv) Using these comments we r declaring customized stat. to make the prog. understandable.  
v) These comments may help u to edit the source code in future.

Comments r divided into  
1) Single-line comments  
2) Multi-line comments  
3) Document comments

In CSS, Comments are ignored by browsers. Here Every comments begins with /\* and ends with \*/

```
/ * This is a comment */
```

Q: Write a Script using style and background.

→

```
<body background="chrome.png" style="background-repeat: no-repeat;
background-attachment: fixed; margin: 100; text-align: justify">
Required Information
Required Information
Required Information
</body>
```

## \* Types of Style Sheets:

26. SEPT. 2019

### 1) Inline:

If we specify styles inside the tag in the body part, these styles are applicable only for the particular line.

Ex: <span style="color: red; font-size: 50"> It is a Span Tag </span>

<p style="color: red"> Hi I am in Red Color </p>

<span style="color: blue"> Hi I am Sky in Blue Color </span>

<b p style="color: green"> Hi I am in Bold </b> </p>

<span style="color: lightblue"> Hi I am Sky in Blue Color </span>

SAVE AS:

InlineSpan.htm

OR

InlineCSSSpan.htm

Opp: It is a Span Tag ← Red Color with Font 50

Hi, I am in Red Color ← Red Color

Hi, I am Sky in Blue Color Hi, I am in Bold ← Half Text Blue & Half Text in Green Color with Bold font

Hi, I am Sky in Blue Color ← Light Blue Color

## 2) Internal or Embedded style sheets:

(68)

If we specify the styles in our html file, These styles are called embedded styles. These styles cannot be used in other files. These styles are applicable for the entire page.

Syntax:

```
<html>
  <head>
    <style type="text/css">
      </style>
  </head>
  <body>
    </body>
</html>
```

Ex:

```
<html>
  <head>
    <title> Internal style sheets </title>
    <style>
      h1
      {
        color: red;
        background-color: lightgreen;
        text-align: center;
        text-decoration: overline;
      }
    </style>
  </head>
</html>
```

SAYE AS:

InternalCSS.html

O/P: It is an Internal Style Sheet

It is not in CSS selector  $\rightarrow$  with h1 It is Internal Style sheet <h1>.

It is Not Related to CSS styles  $\rightarrow$  with h2 <h2> It is Not in CSS selector </h2>

h3 <h3> It is Not related to CSS styles </h3>

</body>

</html>

Note:

3) External style sheets: External style sheet doesn't contain <style> tag

If we declare the styles outside our html file, Then they are called as External styles. These styles can be reuseable, they can be used for more than one file.

Every External file we can save with the extension of .css, The changes if you make in External file it reflect to all html files which are using those styles.

Syntax:

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

To prepare External style sheets, we should follow the following 2 steps.

Step 1) prepare CSS file & save with .css extension.

```
h1
{
  color: red;
  background-color: lightgreen;
  text-align: center;
  text-decoration: overline;
}
```

Save As: "one.css"

Save As: One.css

Step 2:

Prepare html file and call external style sheet using src attribute.

Ex: <html> Save with .htm

SAVE AS: stylesheet.htm

<head>

<link rel="stylesheet" href="one.css">

</head>

<body>

<h1> It is External style </h1>

<h2> It is not External style </h2>

</body>

</html>

O/P :

It is an External Style → Red color with h1

light green

It is not an External Style → h2

To view External styles, open html file on any Browser

{ Note: <b> tag doesn't contain align attribute  
Whereas, <div> tag contains align attribute. }

Ex: 2:

Step 1: preparing css file.

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

div

```
{
    text-align: center,
    color: green;
    font-size: 40px;
}
```

SAVE AS:

Pw0.css

Step 2: prepare html file.

<html>

<head>

<link rel="stylesheet" href="one.css">

</head>

<body>

<b> It is External style </b>

<div> It is Div External style </div>

</body>

</html>

SAVE AS: stylesheet1.htm

O/P :

It is an External style → Red color with default font

It is Div an External style

↳ Green color with 40 font in center

## Working with CSS SELECTORS :

(70)

```
<style>
xyz ← Type Selector
{
color: red;
font-size: 60px;
}
</style>
```

```
<body>
<xyz> It is style </xyz>
</body>
```

While we're using selector names we can use other than html tags.

### Selectors:

A selector is a chain of one or more simple selectors, Each selector there are properties inside curly {} brackets.

In CSS, Selectors are divided into the following

#### Types:

1) Tag Selectors / Element Selectors.

2) Type Selectors

3) Id Selectors

4) Class Selectors.

5) Group Selectors.

6) Universal Selectors.

Etc.

### 1) Tag Selectors / Element Selectors:

These Selectors are from html, Every <html> tag

contains related Attributes & parameters. These Attributes we can implement as "properties".

Ex:

```
<html>
  <head>
    <style type="text/css">
      body
      {
        background-color: #00ddcc
      }
    </style>
  </head>
  <body>
    <h1> My CSS Webpage ! </h1>
    <p> Hello World ! </p>
  </body>
</html>
```

SAVE AS: TagSelector.htm

O/P: My CSS Webpage  
Hello World  
(Background is light blue)

### 2) Type Selectors:

These Selectors we can use as tag Selectors or other than tag.

Some style properties are unable to implement on these Selectors.

Ex:

```
<style>
```

Nilesh

```
{
  color: red;
  font-size: 30px;
}
```

```
</style>
```

```
<body>
```

<nilesh> It is Nilesh </nilesh>

```
</body>
```

SAVE AS: TypeSelector.htm

It is Nilesh

### 3) ID Selectors:

It is used to specify a style for a single unit element. The id selector uses the id attribute of the html element. It is defined as "#".

Ex:1

```
<html>
  <head>
    <style>
      #div
      {
        text-decoration: underline;
        color: red;
        text-align: center;
        font-family: tahoma;
        font-size: 30;
      }
    </style>
  </head>
  <body>
    <div id='div'> It is a ID selector </div>
    <div> It is not a ID selector </div>
  </body>
</html>
```

Ex:2

```
<html>
  <head>
    <style>
      #div
      {
        text-decoration: underline;
        color: red;
        text-align: center;
        font-family: tahoma;
        font-size: 30;
      }
    </style>
  </head>
  <body>
    <#div>
      {
        text-decoration: overline;
        background-color: lightgreen;
        color: red;
      }
  </body>
</html>
```

SAVE AS: IdSelector.htm

O/P:

It is an Id Selector

It is not an Id Selector

If We not put # before div Then O/P

Will be It is an Id Selector

It is not an Id Selector

SAVE AS:

IdSelector1.htm

O/P:

It is an Id Selector

It is another Id selector with more styles

### 4) Class Selectors:

It is used to specify a style for a group of elements, It allows us to set a particular style for many html elements with the same class. It is defined with ". "

Ex: <html>

```
= <head>
  <style>
    .div1
    {
      text-decoration: underline;
      background-color: lightgreen;
      color: red;
      font-family: tahoma;
      font-size: 20;
    }
  </style>
</head>
<body>
  <div> It is not Class Selector </div>
  <h4 class='div1'> It is class selector with more styles </h4>
</body>
</html>
```

SAVE AS: classSelectors.htm

O/P:

It is not a class selector, red text overlined

It is class selector with more style

Ex:2

```
<html>
<head>
  <style type="text/css">
    p#bold
    {
      font-weight:bold;
      color:red;
    }
  </style>
</head>
<body>
  <p id="bold">Rendered with  
Bold & Red.</p>
</body>
</html>
```

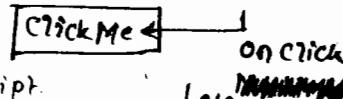
O/P: Rendered With Bold & Red.  
(In Red Color)  
Bold Text

Ex:1 By using Javascript

```
<style>
</style>
<script>
  function MyFunc()
  {
    document.getElementById('div1').innerHTML="Nareshi  
Tech";
  }
</script>
<body>
  <div id='div1'> Microsoft </div>
  <div id='div2'> SunMicros </div>
  <button onclick="MyFunc()"> Click Me</button>
</body>
```

SAVE AS: JavaScript5.htm

O/P: Microsoft  
SunMicros

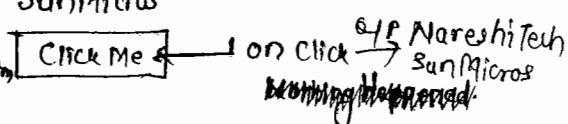
Ex:2 By using JavaScript.

```
<style>
</style>
#div1
{
  color:red;
}
<script>
  function MyFunc()
  {
    document.getElementById('div1').innerHTML="NareshTech";
  }
</script>
<body>
  <div id='div1'> Microsoft </div>
  <div id='div2'> SunMicros </div>
  <button onclick="MyFunc()"> Click Me</button>
</body>
```

SAVE AS: ~~Java~~

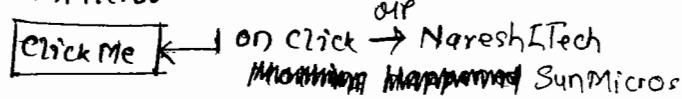
JavaScriptByIdSelector.htm

O/P: Microsoft ← In Red color  
SunMicros



SAVE AS: JavaScriptByClassSelector.htm

O/P: Microsoft > Text in Red color  
SunMicros >



## 5) Group Selectors:

We can Group Selectors using a comma (,) Separator.

(73)

Ex1:

```
h1{font-family:sans-serif}  
h2{font-family:sans-serif}  
h3{font-family:sans-serif}
```

It is Equivalent

To

```
h1,h2,h3{font-family:sans-serif}
```

Ex2: <style>

```
P,div,h1,b  
{  
    font-size:20px;  
}
```

```
</style>  
<body>  
    <P>It is a group selector</P>  
</body>
```

Save As: GroupSelector.htm

O/P: It is a Group Selector.

∴ In HTML  
{: Minimum font size:3}

## 6) Universal Selectors: (\*)

It indicates an Asterik (\*) mark, it matches any element type, it is applicable for the entire page.

Ex: \*

```
{  
    margin:0;  
    padding:0;  
}
```

Ex: 3

<html>

<head>

<style>

```
#div1,.cls1
```

{

text-decoration:underline;  
color:red;

}

</style>

<body>

<div id='div1'>Hello</div>

<div class="cls1">Hii</div>

</body>

</html>

O/P:

Hello  
Hii

## \* CSS Units

28. SEPT. 2012

74

CSS units contains many properties-specific units. There r following list of units existed in CSS.

1) em:

em means the height of the elements font. It is a type of measurement, it is equal to point size.

Note: 1.3 em is not a valid length value, But 1.3em is a valid.

Ex: <html>

```
<head>
  <style type="text/css">
    p
    {
      letter-spacing: 3em;
    }
  </style>
<head>
<body>
  <h1> CSS units! </h1>
  <p> This Text is em unit trouble to Read </p>
</body>
</html>
```

2) ex:

This value defines a measurement relative to a font's x-height.

Ex: <html>

```
<head>
  <style type="text/css">
    p
    {
      line-height: 1.3ex;
    }
  </style>
<head>
<body>
  <h1> CSS units! </h1>
  <p> This Text is em unit trouble to Read </p>
</body>
</html>
```

3) px:

It defines a measurement in screen pixels.

Ex: <html>

```
<head>
  <style type="text/css">
    p
    {
      font-size: 10px;
    }
  </style>
```

```
<body>
  <h1> CSS units! </h1>
  <p> This Text is px unit easy to Read </p>
</body>
</html>
```

4) cm:

It defines a measurement in centimeters.

Ex: <html>  
<head>

```
<style type="text/css">
```

P

{

Font-size: 2cm;

}

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1> CSS Units! </h1>
```

```
<p> This Text is cm unit  
Trouble to read </p>
```

```
</body>
```

```
</html>
```

6) mm:

It determines measurement in millimeters. (75)

Ex:

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

P

{

Word-spacing: 15mm;

}

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1> CSS units! </h1>
```

```
<p> This Text is mm unit trouble to Read </p>
```

```
</body>
```

```
</html>
```

5) in:

It defines a measurement in inches. 1 inch = 2.54 cm.

Ex: <html>

```
<head>
```

```
<style type="text/css">
```

P

{

Word-spacing: .15in;

}

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1> CSS units! </h1>
```

```
<p> This Text is in unit  
Simple to Read </p>
```

```
</body>
```

```
</html>
```

7) pt:

It defines a measurement in points.

1 pt =  $\frac{1}{72}$  inches.

Ex: <html>

```
<head>
```

```
<style type="text/css">
```

P

{

Font-size: 15pt;

}

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1> CSS units! </h1>
```

```
<p> This Text is pt unit easy to Read </p>
```

```
</body>
```

```
</html>
```

### 8) pc:

defines a measurement in picas.  
1 pica = 12 pts.

Ex:

```
<html>
  <head>
    <style type="text/css">
      p {
        font-size: 25pc;
      }
    </style>
  </head>
  <body>
    <h1> CSS Units! </h1>
    <p> This Text is in picas </p>
  </body>
</html>
```

### ★ Color Units And Setting Backgrounds

A color\_value is a keyword or a numerical RGB specification.

#### i) Backgrounds in CSS:

##### 1) Background-color:

By using this property, we can set background color of an element.

Ex:

```
<html>
  <head>
    <style type="text/css">
      p {
        background-color: lightgreen;
      }
    </style>
  </head>
  <body>
    <p> These are Background properties </p>
  </body>
</html>
```

### 9) %:

defines a measurement as percentage relative to another value.

Ex:

```
<html>
  <head>
    <style type="text/css">
      p {
        font-size: 16pt;
        line-height: 125%;
      }
    </style>
  </head>
  <body>
    <h1> CSS Units! </h1>
    <p> This Text is in Percentage </p>
  </body>
</html>
```

#### ii) background-image:

This property is used to set the background image of an element.

Ex:

```
<html>
  <head>
    <style type="text/css">
      p {
        background-image: url('firefox.jpg');
      }
    </style>
  </head>
  <body>
    <p> These are Background Properties </p>
  </body>
</html>
```

### iii) background-repeat:

This property is used to control the repeat of image in the background.

Ex: <html>

```
<head> type="text/css">
p {
background-image: url('firefox.jpg');
background-repeat: no-repeat;
}
</style>
</head>
<body>
<p>These are background images </p>
</body>
</html>
```

### iv) background-position:

This property is used to control the position of an image in the background.

Ex: <html>

```
<head>
<style type="text/css">
body ← Tag selector
{
background-image: url('water.gif');
background-repeat: no-repeat;
background-attachment: fixed;
background-position: center;
}
</style>
</head>
</body> </html>
```

### v) background-attachment:

This property is used to control the scrolling of an image in the background. 77

## \* Advanced Background Properties:

- 1) background-clip
- 2) background-origin
- 3) background-size

## \* Fonts in CSS:

css contains following list of font properties.

### 1) font-family:

This property is used to change the face of your font.

Ex: <html>

```
<head>
<style type="text/css">
p {
font-family: tahoma;
}
</style>
</head>
<body>
<p> The Font-Face is Tahoma </p>
</body>
</html>
```

### 2) font-style:

This property is used to make a font italic.

Ex: <html>

```
<head>
<style type="text/css">
p {
font-family: tahoma;
font-style: italics;
}
</style>
</head>
```

```
<body>
  <p> The font face is Tahoma </p>
  <p> The font style is Italic </p>
</body>
</html>
```

### 3) font-variant:

This property is used to create a small caps effect.

Ex:

```
<html>
  <head>
    <style type="text/css">
      p {
        font-variant: small-caps;
        font-weight: bold;
        font-style: italic;
      }
    </style>
  </head>
  <body>
    <p> The font variant is small Caps </p>
    <p> The font variant is Normal </p>
  </body>
</html>
```

### 4) Font-Weight:

This property is used to increase or decrease how Bold or light a font appears.

Ex: <html>

```
<head>
  <style type="text/css">
    p {
      font-weight: bold;
    }
```

78

```
h1 {
  font-weight: bold;
}
<style>
</head> <p>
<body> font-weight is Bold </p>
<h1> font-weight is Bolder </h1>
</body>
</html>
```

### 5) font-size:

This property is used to increase or decrease the size of Ur font.

Ex: <html>

```
<head>
  <style type="text/css">
    p {
      font-size: small;
    }
```

```
h1 {
  font-size: large;
}
```

```
h3 {
  font-size: 10px;
}
```

</style>

</head>

<body>

```
<p> The font-size is small </p>
```

```
<h1> The font-size is large </h1>
```

```
<h3> The font-size is large </h3>
```

</body>

</html>

## ~~Advanced CSS~~

- 1) @font-face
- 2) font-size-adjust
- 3) font-stretch

## CSS Text Properties

css contains following list of Text properties.

### 1) color:

This property is used to set the color of a text.

Ex:

```
<html>
<head>
  <style type="text/css">
    p
    {
      color: red;
    }
  </style>
</head>
<body>
  <p>Text color is Red </p>
</body>
</html>
```

### 2) direction:

This property is used to set the text direction.

Ex:

```
<html>
<head>
  <style type="text/css">
    p
    {
      color: red;
      direction: rtl;
    }
  </style>
</head>
<body>
  <p>Text color is red </p>
</body>
</html>
```

### 3) letter-spacing:

This property is used to add / subtract the space betw the letters. (79)

Ex:

```
<html>
<head>
  <style type="text/css">
    p
    {
      letter-spacing: 10px;
    }
    h1
    {
      word-spacing: 20px;
    }
  </style>
</head>
<body>
  <p>It is letter spacing </p>
  <h1>It is word spacing </h1>
</body>
</html>
```

### 4) text-indent:

This property is used to indent the text of a paragraph.

Ex:

```
<html>
<head>
  <style type="text/css">
    p
    {
      text-indent: 30px;
    }
  </style>
</head>
<body>
  <p>It is indent property of Text </p>
</body>
</html>
```

## 5) text-align:

This property is used to align the Text on the document.

Ex: <html>

```

<head>
  <style type="text/css">
    p
    {
      text-align: center;
    }
  </style>
</head>
<body>
  <p> It is a Text align </p>
</body>
</html>

```

## 6) text-decoration:

This property is used to underline, overline, strike through.

Ex:

```

<html>
  <head>
    <style type="text/css">
      p
      {
        text-decoration: strike-through;
      }
    </style>
  </head>
  <body>
    <p> It is removed content </p>
  </body>
</html>

```

## 7) text-transform:

This property is used to Capitalized txt or convert the txt to Uppercase or Lowercase characters.

Ex:

```

<html>
  <head>
    <style type="text/css">
      p
      {
        text-transform: capitalize;
      }
      div
      {
        text-transform: uppercase;
      }
      h2
      {
        text-transform: lowercase
      }
    </style>
  </head>
  <body>

```

<div> SAMPLES:  
</div>  
<h2>

<p> every word starting with capital letter </p>  
<div> & it is in Uppercase </div> <h2> & it is in Lowercase </h2>  
</body>

## 8) white-space:

This property is used to control the flow & formatting of the text.

Ex: <html>

```

<head>
  <style>
    p
    {
      white-space: nowrap;
    }
  </style>

```

↳ nowrap means

(8)

```

</head>
<body>
  <p> It is sample text line. It is a
    sample text line. It is a sample
    text line. It is a sample text
    line. It is sample text line. </p>
</body>

```

Ex:

### i) hide:

hide property is used to set the height of an image. This property can have a value in length, pixels or in %.

Ex:

## ★ CSS Borders:

Border property allows u to specify how the border of the box representing an element should look.

There are 3 properties of a Border, U can change,

### 1) Border-color:

Using this property, we can change the color of a Border.

### 2) Border-style:

Using this property, we can specify border styles like solid, dashedline, double line etc.

### 3) Border-width:

Specifies width of a Border.

## ★ CSS Images ★

### CSS images:

CSS place a good role

to control image display, u can set following image properties using CSS.

#### i) border

The border property is used to set the width of an image border.

#### ii) height:

To set height of an image.

#### iii) width:

Set the width of an image.

Ex:

```

<head>
  <style>
    p {
      border-color: red;
      border-style: solid;
    }
  </style>
</head>
<body>
  <p> solid Border </p>
</body>

```

## 2) Border - Styles:

This property allows following.

- i) none
- ii) solid
- iii) dotted
- iv) dashed
- v) double
- vi) groove
- vii) ridge
- viii) inset
- ix) outset
- x) hidden;

Ex: <head>

```

  <style>
    p {
      border-color: red;
      border-style: solid;
      border-width: 10px;
    }
  </style>

```

```

  <h1>
  {
    border-color: red;
    border-style: dotted;
    border-width:
  }

```

```

  <div>
  {
    border-color: red;
    border-style: dashed;
    border-width: 10px;
  }

```

```

</style>
</head>

```

82 <body>

```

  <p> solid Border </p>
  <div> Dashed Border </div>
  <h2> dotted Border </h2>
</body>

```

## \* CSS Cursors:

In CSS, cursor property allows us to specify the type of the cursor should display to the user.

Ex: <head>

```

  <style>
    p {
      cursor: crosshair;
    }
    div {
      cursor: pointer;
    }
  </style>
</head>
<body>
  <p> Cross Hair </p>
  <div> Pointer </div>
</body>

```

</html>

Properties of cursor

```

<div style="cursor: auto">Auto </div>
<div style="cursor: crosshair">crosshair</div>
<div style="cursor: default">default </div>
<div style="cursor: pointer">pointer </div>
<div style="cursor: move">move </div>
<div style="cursor: e-resize">e-resize </div>
<div style="cursor: ne-resize">ne-resize </div>
<div style="cursor: nw-resize">nw-resize </div>
<div style="cursor: n-resize">n-resize </div>
<div style="cursor: se-resize">se-resize </div>
<div style="cursor: sw-resize">sw-resize </div>
<div style="cursor: s-resize">s-resize </div>
<div style="cursor: w-resize">w-resize </div>
<div style="cursor: text">text </div>

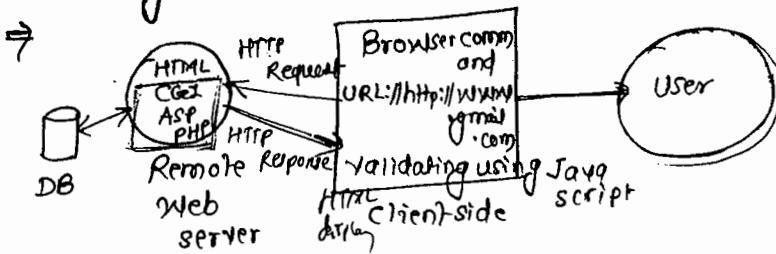
```

JAVASCRPTS

JAVASCRPTS

# JAVASCRIPT

## Working Principles of Client-side And Server-side Scripting.



### \* Other Client-side Scriptings:

- 1) BASIC; from this lang. implemented VBScript.
- 2) PERL; it is client-side scripting not supported by all Browsers.

Netscape Navigator → LiveScript → JavaScript  
By Brendan Eich in 1995

# \* JAVASCRIPT

15. SEPT. 2023 (87)

## \* Difference Betw Scripting & Languages.

### Scripting

1) Light weight programming.

Ex: document.write("NIT"); ← Javascript

2) echo 'NIT'; ← shell

3) MsgBox ("NIT") ← VBScript

4) print("Welcome"); ← PERL

2) Easy to Understand.

3) Easy to Implement.

4) No header files required.

5) No special libraries.

6) No special compilers.

7) Interpreter based.

8) <sup>Global</sup> loosely Typed programming (~~No variable declaration~~,  
No need to follow comma, semicolon etc.)

9) <sup>Client Side</sup> Validat<sup>n</sup> purpose

10) Poor Graphics implementation

Ex: Javascript, VBScript, perlScript etc.

### \* Introduction to JavaScript:

It is First Web Scripting lang. It is client side scripting lang developed by

"Brendan Eich" in Netscape Corporation at 1995. It is supporting ~~most~~ all

major modern browsers like google chrome, Mozilla Firefox, Opera, Safari, IE.

Netscape Corporation came out with a client side Scripting i.e. known as LiveScript. It is the first Web Scripting lang.

Microsoft Corporation's JavaScript version is JScript, it was 1st introduced with IE version 3. (It is initially LiveScript, Later Renamed as JavaScript)

### \* Nature of JavaScript: It is object based programming language.

It is based on Object Oriented programming concept.

Its Syntax is quite similar to C, C++ and Java. It is easy & simple to implement.

### Q: What is JavaScript?

→ JavaScript is satisfying the following list of pts.

i) It is basically LiveScript.

ii) It is pure Client-side programming tool.

iii) It can integrate with HTML tags.

iv) It is lightweight Scripting

v) It is interpreted programming.

vi) It is designers programming tool. vii) It can react to events. viii) It can read & write HTML

### Language

1) Heavy weight programming.

Ex: include <stdio.h>  
main()

{  
=

2) Complex to Understand.

3) Trouble to Implement.

4) Header files are Mandatory.

5) Required Special Libraries.

6) Every lang. they have their own compilers.

7) <sup>Compiler</sup> Compiler Based.

8) Strictly Typed programming

(variable declaration required semicolon, comma, quoted mandatory)

9) Verification & validation Both.

10) Rich Graphics implementation.

11) Ex: C, C++, Java, COBOL, FORTRAN.

vii) It can integrate with any server-side programming.

viii) Open source and cross platform

ix) It creates & Read with Events

ix) JavaScript can integrate with HTML & CSS called DHTML.

### \* JavaScript Syntax:

It consists of group of (stat) that are placed within the following tags. `<script> ... </script>`

Ex: `<script language="javascript" type="text/javascript">`

`</script>`

Ex:

`<script language="javascript" type="text/javascript">` Javascript(DHTML).htm  
document.write("Welcome To Client-side Scripting"); O/P: Welcome To Client-side  
document.write(`\n`)  
document.write("It is Live Script")  
document.write(`\n`)  
document.write("from Netscape Corporation, first web scripting")  
`</script>`

Note: Here, "}" is very important if we didn't put Inverted

Inverted  
commas

Comma, Then O/P will be  
blank.

### \* JavaScript Structure:

Generally JavaScript follows the following structure

```
<html>
  <head>
    <title> Example </title>
    <script language="JavaScript" >
      !-
      --->
    </script>
  </head>
  <body>
    ...
  </body>
</html>
```

### \* NOTE:

It is a preferred way to keep  
`<script>` tag within `<head>` tag. \*

Ex:

SAVE AS: Javascript(DHTML).htm  
<html> O/P: Welcome to Client-side Scripting htm  
 <head> It is a live script → Bold (H1)  
 <title> Working with JavaScript (DHTML) script from Netscape Corporation, first web  
 </title>  
 <script language="JavaScript" type="text/javascript">  
 document.write("<h1> Welcome to client  
side scripting </h1>");  
 document.write("<b>It is Live Script </b>")  
 document.write("<br/>")  
 document.write("<s> From Netscape Corporation  
First Web Scripting </s>");  
 </script> Observation:  
 </head> Note:  
 <body bgcolor=lightgreen> Background color  
 </body> is light green  
</html> Semicolon Not Necessary after  
Parameter

## \*Script:

17. SEPT. 2012

(89)

Generally Script or <Script> tag contains following 2 Essential Attributes.

- 1) Language
- 2) Type.

### 1) Language:

This Attribute specifies what Scripting lang. we are using.

### 2) Type:

This Attribute indicates & recommend to interpreter what value should be set.

Ex: <script language="JavaScript" type="text/JavaScript">

    JavaScript code

    JavaScript code

    JavaScript code

</script>

## JavaScript Comments:

Comments are non Executable stat. or ignore stat. Using this Comment Notat's we can declare customized stat. as per the End user requirement. In Javascript, Comment Notat's are divided into following 2 Types:

- 1) Single Line Comments.

- 2) Multi Line Comments.

### 1) Single Line Comments:

These comments are Restricted to within the line.

The Notat' is // (Double forward slash).

Ex: /\* document.write("Hello"); // This will write "Hello"

    document.write ("<h1> This is a header </h1>"),  
    document.write ("<p> This is a paragraph </p>"),  
\*/

} Multiple line  
Comments.

### 2) Multiline comments:

These comments are Related to Multiple java stat. are

multiple lines.

Notation: /\* \*/

Ex:

## \* Debugging JavaScript Errors:

Generally Javascript is a disable component in a browsers. We should enable Javascript Option manually.

### 1) Enable javascript in Internet Explorer:

- i) Click on Tools Menu
- ii) Choose Internet Options from Menu
- iii) Click the security tab on the Internet Options pop up.
- iv) Select custom level button to access your security setting.

- v) scroll down to Scripting section,  
 vi) Check Enable Button for Active Scripting.  
 vii) select Finish & confirm the Msg.

### \* <script> Tag Description:

It is a Tag which indicates script interpreter required to execute lines between <script> tags.

<script>

≡

</script> These Tags.

If U didn't mention <script> tag, Browser interpreters are printing as Normal strings. These strings displayed on the Webpage.

```
<Script type="text/javascript">
  document.write("It is 'Live' Script");
  document.write("<br>"),
  document.write("It is "JavaScript" from Netscape");
</script>
```

SAVE AS: JavaScript2.htm

Output

It is 'Live'script'

It is "JavaScript" from  
Netscape.

### \* Working with single & double quotes in Javascript:

JavaScript supports single & double quotes to produce the same results. The ~~above~~ Example below will make U a clear picture on single & double quotes.

```
<Script type="text/JavaScript">
  document.write("It is 'Live' Script");
  document.write("<br>"),
  document.write("It is "JavaScript" from Netscape"),
  document.write("JavaScript is NOT Java"),
</script>
```

SAVE AS: JavaScript28.htm

Output

It is 'Live'script'

It is "JavaScript" From  
Netscape JavaScript is  
"NOT" Java.

### \* Working with JavaScript Special Characters:

In Javascript, U can add special characters to ~~inserted~~ screen by using / (back slash) sign.

#### Insert a special character :

back slash is used to insert apostrophes, new lines, quotes & other special characters into a text screen.

<script>

Var ~~txt~~ txt = "We r the so-called "Good" from the North.";

document.write(txt);

SAVE AS: JavaScript4.htm

</script>

Output : Blank

In the above Example, we must place ~~\~~ (backslash) ~~before each~~ before each double quote.

~~<Script>~~

```
var txt = "We r the so-called \ Good" from the North.",  
document.write (txt);
```

~~</script>~~

SAVE AS:

JavascriptUsingBackslash.htm

Output:

We r so-called "Good" from the  
North

| <u>Code</u> | <u>OP</u>       |
|-------------|-----------------|
| '           | single quote    |
| "           | double quote    |
| \           | back slash      |
| \n          | new line        |
| \r          | Carriage Return |
| \t          | tab             |
| \b          | backspace       |
| \f          | form feed       |

### \* Whitespaces & Line Breaks:

→ JavaScript ignores spaces, tabs, & newlines that appear in JavaScript.

Using this Whitespaces & LineBreaks make ur script more Readable.

#### 1) WhiteSpace:

The following Examples are equivalent

```
Var name="nit";  
Var name="nit", "nit";
```

#### JavaScript is Case Sensitive:

JavaScript is basically case-sensitive programming. A fn named "myfunction" is not equal to "myFunction" & a variable named "myVar" is not the same as "myvar".

#### Break up a Code Line:

We can break up a code line within a txt string

With a backslash

```
<Script>  
document.write ("Hello") ("Hello  
World!");  
</script>
```

Correct code

SAVE AS: JavascriptUsingBackslash.htm

OP: Hello World!

```
<Script>  
document.write \ ( ("Hello World!");  
</script>
```

Incorrect code

## \* Difference Bet' document.write & window.document.write

→ There is No diff. bet' the above stat. Here, The Window Object is highest level object, it contain other objects & their methods.

Document is Obj content inside the Window object Write is a method of document object.

In that situat', Window object is not Necessary.

Ex: `<script>`

```
document.write("HelloWorld!");
document.write("<br/>");
window.document.write("Naresh i Tech");
</script>
```

Ex: `<script>`

```
document.write("Hello World!");
document.writeln("Naresh i Tech");
</script>
```

Ex: `<head>`

```
<script>
function Show_Message()
{
    alert("Welcome To Pop-up Boxes");
}
</script>
<head>
<body>
<button onclick="Show_Message()">
```

## \* Semicolons or optional:

18. SEPT. 2012

It is always optional.

In Javascript, Every stat. should ends with semicolon

Ex: ~~<head>~~ `<script>`

```
document.write("Welcome To Java Script");
document.write("<br/>");
document.write("Client side Scripting");
</script>
```

Note:

Closing every stat. with a semicolon always good scripting practice.

## \* Script with `<html>` tags:

```

<script>
document.write("<font color = '#FF0000' size=7 face=tahoma>JavaScript+HTML=")
document.write("DHTML</font>'");
document.write("<br/>");
document.write("<b>It is bold</b>");
```

</script>

## \* Javascript placed html file:

```

<html>
  <head>
    <title>JavaScript </title>
    <script> <script> Tag can be
    </script> used anywhere.
  </head>
  <body>
    <script> Restrict that <script> tag used only under
    </script> <body> tag or <head>
                  or any other tag.
```

In html document we can include javascript part anywhere. But there are following most preferred ways.

- 1) Script in `<head>...</head>` section
- 2) Script in `<body>...</body>` section
- 3) Script in `<body>...</body> & <head>...</head>`
- 4) Script in External file & then include in `<head>...</head>` section.

## \* JavaScript code:

It is a sequence of javascript stat. Each stat. is executed by browser in the sequence they are Return.

```

<script type = "text/javascript">
  document.write("<h1>This is Heading </h1>"),
  document.write("<p>This is a Paragraph </p>"),
</script>
```

The above Ex. is JS code. It executes Sequentially.

## \* JavaScript Blocks:

JavaScript stat. can be grouped together in Blocks.

The purpose of block is to make the seq. of stat. execute together.

```

<script type = "text/javascript">
{
  document.write("<h1>This is a Heading </h1>"),
  document.write("<p>It is a first Block </p>"),
}
{
  document.write("<h1>It is a second Block </h1>"),
}
</script>
```

The above ex. is JS Block. It executes Blockwise,

## \* JavaScript POPUP Boxes:

94

JavaScript has 3 kinds of POPUP Boxes.

- 1) Alert Box
- 2) Confirm Box
- 3) Prompt Box

### 1) Alert Box:

Used to display Alert Popup, when user click on "ok" to proceed to next step.

Syntax: alert ("Message");

```
<html>
<head>
    <title> POP UP Boxes </title>
    <script>
        alert("Welcome To MyWebpage");
        alert("Bye...");
```

</script>

```
</head>
<body>
    <script>
        alert("Welcome To Body of the Page");
        alert("Bye...");
```

</script>

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

### \* How to Write Text on Multiple lines in an alert box?

→ We should not use `<br>` Tag here, `alert()` is a method of window object that can't interpret HTML tags.

If U break text into multiple lines we should use new line escape character. It consist of (1) Backslash symbol & an Alphabet.

The following are commonly used escape characters.

`\n`: Inserts a New line.

`\t`: Inserts a Tab.

`\r`: Carriage return.

`\b`: Backspace.

`\f`: Form feed.

`'`: Single quote.

`"`: Double quote.

`\`: Backslash.

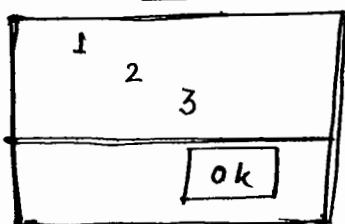
Ex:

```
<script>
  alert ("JavaScript\nis\na\nclient-side\nprogramming\nlanguage");
</script>
```

O/P:

95  
JavaScript  
is  
a  
client-side  
programming  
language

Ex:      O/P:



```
<script>
  alert("1\n|t2\n|t3");
</script>
```

Ex: <head>

```
<head>
  <script>
    function show-message()
    {
      alert('It is alert popup');
      alert("Bye...");
    }
  </script>
</head>
<body>
  <input type="submit" onclick="show-message()" value="ShowMe" />
</body>
```

### Alert with functions :

## 2) Confirm Box:

It is often used, if u want to verify or accept something.

When a Confirm Box ~~executes~~ <sup>popup</sup>, User should get Ok or Cancel to proceed.

If User clicks ok, it returns True, If User clicks Cancel, It returns False.

### Syntax:

```
confirm ("Message");
```

Ex: <head>

```
<script>
  confirm ("Click Ok or Cancel");
</script>
</head>
```

Ex: 2

```
<head>
<script>
  xyz = confirm ("Click Ok or Cancel");
  if (xyz == True)
  {
    alert("User Selected Ok Button");
  }
  else
  {
    alert("User Selected Cancel Button");
  }
</script>
</head>
```

Ex:

```

<head>
<script>
    Function show_confirm()
    {
        xyz = confirm ("Click Ok or Cancel");
        if (xyz == true)
        {
            alert ("User Selected Ok Button");
        }
        else
        {
            alert ("User Selected Cancel Button");
        }
    }
</script>
</head>
<body>
<button onclick="show_confirm()">
    Show Message </button>

```

### 3) Prompt Box:

A prompt box is used to get a value before entering a page. When a prompt box pops up, a user should click either Ok or Cancel.

If User selected Ok, it Returns the Get Value.

If User selected Cancel, it Returns Null.

#### Syntax:

```
prompt("Somewhat", "DefaultValue");
```

Ex: ~~<script>~~

```

<head>
<script>
    xyz = prompt ("Enter Any Number", "9");
    alert ("Entered Number is: " + xyz);
</script>
</head>

```

~~Ex:~~  
~~<head>~~  
~~<script>~~  
~~xyz~~

### External JavaScript:

JavaScript can be placed in External JavaScript files. Extension is .js.

Note: External script can't contains <Script> tag.

To use external script, point to the .js file we're using "src" attribute.

To create external JavaScript file, follow the following steps.

Step 1: Create JavaScript file & save with .js extension.

```
document.write ("<h1>Welcome to JS  
External programming!!</h1>");  
document.write ("<h2>Bye..!!</h2>");
```

Step 2: prepare html file & call the JavaScript.

```

<html>
<head>
<script type="text/javascript" src="myscript.js"></script>
</head>
<body>
</body>
</html>

```

Save file with .html extension or .htm extension.

Step 1

Ex: 2) Prepare JavaScript

```

function show_alert()
{
    alert ("Hi");
}

```

```

function show_confirm()
{
    confirm ("Hello");
}

```

Step 2:  
prepare HTML file.

(97)

```
<html>
  <head>
    <script type="text/javascript" src="myscript.js"></script>
  </head>
  <body>
    <button onclick="show_alert()"> Show </button>
    <button onclick="show_confirm()"> Show_Confirm </button>
  </body>
</html>
```

Date : 20. SEPT. 2012

### \* Working with javascript Variables:

#### Rules:

- i) Every variable should starts with an Alphabet or underscore.
- ii) Variable should not contains unnecessary special characters.
- iii) Variable should not contains embedded periods. (dots)
- iv) Variables are case sensitive.
- v) Count, Count and COUNT are 3 diff. variables.
- vi) Variable should be reasonable length.
- vii) Variables declared with 'var' keyword.

Ex: Var x; variant : collect of Datatype  
Var Carname;

These are the Empty variables. If u want, u can assign values to these variables.  
Var x = 5;  
Var name = "NIT";

Q: Write a Script, Display sum of 2 nos.

```
> <html> <head> <title>variables </title>
  <script type="text/javascript">
    var a=10;
    var b=20;
    var c=a+b
    document.write("The value of x is "+x);
    document.write("Sum of 2 nos. is "+c);
    document.write("<br>"),
    document.write("The value of y is "+y);
    document.write("<br>"),
    document.write("The value of z is "+z);
    document.write("<br>"),
  </script>
  </head>
  <body>
  </body>
</html>
```

**Q:** Write a script to read 2 values, to display sum and difference.

```
<html>
<head>
<title> Variables </title>
<script type="text/javascript">
var x = prompt("Enter any value: ", "10");
var y = prompt("Enter any value: ", "20");
var z = x + y;
var p = x - y;
document.write("The value of z is " + z);
document.write("<br>");
document.write("The value of p is " + p);
</script>
</head>
<body>
</body>
</html>
```

### \* String Variables:

In Javascript, String variables should be enclosed in either single or double quoted marks.

Ex: greeting = "Hello there"

Warning = 'Be Careful'

U may include a single quote within a double quoted string. OR

A double quote within a single quoted string.

Ex:

greeting = "Hello there" is a greeting  
warning = 'Be Careful' is a warning

### \* Numeric Variables:

Creating a Numeric

Variable is as simple as assigning a Value.

Ex: count = 42

temperature = 98.4

### \* <noscript> tag:

Used to provide alternate

Content for users, When Ur browser unable to support javascript, intesituat no script will be executed.

#### Syntax:

```
<noscript> . . . </noscript>
```

#### Ex: <html>

```
<head>
```

```
<title> NOSCRPT TAG </title>
```

```
<script>
```

```
function show_Alert()
```

```
{
```

```
    alert("Working with NOSCRPT");
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<noscript>
```

OOPS U r using too lower versions  
Is unable to support...

```
</noscript>
```

```
<input type="submit" onclick="show_Alert()" value="NOSCRPT"/>
```

```
</body>
```

```
</html>
```

#### Note:

If Browser supports, Script part will be executed, otherwise Noscript part will be Executed.

## \* JAVASCRIPT Operators:

In Javascript, Operators are divided into following types.

### 1) Arithmetic Operators:

Using these operators we can perform arithmetic operation for mathematical operation.

Following table describes operator description & Example.

| Operator | Description                     | Example |
|----------|---------------------------------|---------|
| +        | Addition                        | j+12    |
| -        | Subtraction                     | j-22    |
| *        | Multiplication                  | j*7     |
| /        | Division                        | j/3.14  |
| %        | Modulus<br>(division remainder) | j%6     |
| ++       | increment                       | ++j     |
| --       | Decrement                       | --j     |

### 2) Assignment/Comparison:

Comparison Operators are generally used to compare values. The following table describes operator description & example.

| Operator | Description                            | Example |
|----------|----------------------------------------|---------|
| ==       | is equal to                            | j==42   |
| !=       | Is not equal to                        | j!=17   |
| >        | Is greater than                        | j>0     |
| <        | Is less than                           | j<100   |
| >=       | Is greater than or Equal to            | j>=23   |
| <=       | Is less than or Equal to               | j<=23   |
| ==       | Is equal to (and of the same type)     | j==56   |
| !=       | Is not equal to (and of the same type) | j!=1    |

## 3) Logical Operators: (29)

To compare logical Expressions.

| operator | Description | Example      |
|----------|-------------|--------------|
| &&       | And         | j==2 && k==2 |
|          | Or          | j<100    j>0 |
| !        | NOT         | !(j==k)      |

## \* String Concatenation: (+)

The + Operator used on strings. To add 2 string variables this operator required.

Ex:  
str1 = "Nil";  
str2 = "Pin2";  
str3 = str1 + str2;

Ex: 2 In a prog. 2 prompt boxes existed if u r reading values thru prompt. By default it is treating as "strings."

Ex: Nil & 28  
28 Nil Nil

```
<html>
<body>
<script type="text/javascript">
var x;
x=8+8;
document.write(x);
document.write("<br>");
x="8"+"8";
document.write(x);
document.write("<br>");
x="8"+8;
document.write(x);
document.write("<br>");
</script>
</body>
</html>
```

## \* Working with conditional Control statements:

JavaScript contains following conditional stat.

- 1) If statement
- 2) If... else statement
- 3) If...else If...<sup>else</sup> statement
- 4) Switch statement

### 1) If Statement:

Using this statement, we can execute a part of the code, if a specified cond' is True.

#### Syntax:

```
if (condition)
{
    Code to be executed if cond' true
    Code to be executed if cond' true
}
```

#### Ex:

```
<html>
<head>
    <title> If condition </title>
    <script>
        var x = prompt("Enter any no: ", '10');
        if (x == 10)
        {
            alert ("Entered Number is Equal:");
        }
    </script>
</head>
<body>
</body>
</html>
```

### 2) If...else:

Using this stat, we can execute a part of code if the given cond' is True, we can execute another part of code ~~also~~ if the given cond' is false.

Syntax:

```
if(Condition)
{
    Code to be executed if cond' is True
    Code to be executed if cond' is True
}
else
{
    Code to be executed if cond' is False
    Code to be executed if cond' is False
}
```

#### Ex:

```
<html>
<head>
    <title> If condition </title>
    <script>
        var x = prompt("Enter any no: ", '10');
        if (x > 10)
        {
            alert ("Entered no. is Greater");
        }
        else
        {
            alert ("Entered no. is Less");
        }
    </script>
</head>
<body>
</body>
</html>
```

### 3) if... else if... else:

By using we can

Select one of select block of code to be executed.

Syntax:

```
if (condition 1)
{
    code to be executed if condn1 is True
}
else if (condition 2)
{
    code to be executed if condn2 is True
}
else
{
    code to be executed if neither condn1 nor condn2
}
```

Ex:

```
<html>
<head>
<title>if... else if statement </title>
<script type="text/javascript">
var b = prompt("Enter any number", "100");
if(b>100)
{
    alert("Number is Greater");
}
else if(b<100)
{
    alert("Number is less");
}
else
{
    alert("Number is equal");
}</script>
</head>
<body>
</body>
</html>
```

Ex:

```
<html>
<head>
<title>if... else if statement </title>
<script type="text/javascript">
Var book = prompt("Enter Ur favourite
book", "HTML, CSS, JS, HTML5");
if(book == "HTML")
{
    alert("U r selected HTML");
}
else if(book == "CSS")
{
    alert("U r selected CSS");
}
else if(book == "JS")
{
    alert("U r selected JS");
}
else if(book == "HTML5")
{
    alert("U r selected HTML5");
}
else
{
    alert("Invalid Book");
    alert("Thank U... ");
}</script>
</head>
<body>
</body>
</html>
```

## Switch statements

```

<html>
<head>
<title> if...else if statement Button</title>
<script type="text/javascript">
function fav-book()
{
    var book=prompt("Enter ur favourite book
                    (html,css,j's,HTML5)");
    if (book=='html')
    {
        alert("U r selected HTML");
    }
    else if (book=='css')
    {
        alert("U r selected css");
    }
    else if (book=='j's')
    {
        alert("U r selected javascript");
    }
    else if (book=='HTML5')
    {
        alert("U r selected HTML5");
    }
    else
    {
        alert("Invalid Book");
        alert("Thank U");
    }
}
</script>
<head>
<body>
<button onclick="fav-book()"> fav-book </button>
</body>
</html>

```

## 4) Switch statements:

(102)

By using switch case statement select a specific block code to be executed.

Switch case stat. are alternative ways of executing st. It is an Enhanced version of if...else stat.

### Syntax:

```

switch(n)
{
    case1:
        execute code block1
        break;
    case2:
        execute code block2
        break;
    default:
        code to be executed if n is different
        from case1
}

```

### Ex:

```

<html>
<head>
<title> Switch case statements </title>
<script type="text/javascript">
var alpha=prompt("Enter any Alphabet
                  :","A");
switch(alpha)
{
    Case 'A':
        alert("It is a Book");
        break;
    Case 'B':
        alert("It is a Book");
        break;
    Case 'C':
        alert("It is a Book");
        break;
    default:
        alert("Invalid Book");
}
</script>

```

Ex: <html>

<head>

<title> Switch With Date </title>

<head>

<script type="text/javascript">

function MyDay()

{

var x;

Var d=new Date().getDay();

Switch (d)

{

Case 0:

x="It is Sunday";

break;

Case 1:

x="It is Monday";

break;

Case 2:

x="It is Tuesday";

break;

case 3:

x="It is Wednesday";

break;

case 4:

x="It is Thursday";

break;

case 5:

x="It is Friday";

break;

case 6:

x="It is Saturday";

break;

}

document.getElementById("demo").innerHTML=x;

}

</script>

</head>

<body>

<p> Click the button to display what day it is

Today</p>

<button onClick="MyDay()"> Today </button>

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

</body>

</html>

QUESTION

SOLVED

<switch> Statement with <title></title>

</head>

## \* JavaScript Keywords / Reserved Words:

JavaScript contains following 19 set of keywords or Reserved words.

abstract, boolean, break, byte, case, catch,  
char, class, const, continue, debugger, default,  
delete, do, double, else, enum, extends, false,  
final, finally, float, for, function, goto, if,  
implements, import, instanceof, int,  
interface, long, native, new, null, package,  
private, protected, public, return, short,  
static

## \* JavaScript Looping Control Statements:

In Java, there are 2 kinds of loops.

- 1) for loop
- 2) while loop

### 1) for loop:

We can execute block of stat. repeatedly until the given cond' is false. If false, it terminates the loop. It contains following 3 parameters.

- i) An initialization Expression
- ii) A condition Expression
- iii) A Modification Expression,

The above parameters are separated by semicolon

for (expr1; expr2; expr3)

#### Syntax:

for (initializat'; test cond'; iterat' stat)

{  
Stat(s) to be executed if test cond' is true  
Stat(s) to be executed if test cond' is true  
Stat(s) to be executed if test cond' is true}

Q: A Script To display 1 To 10 nos.

104

⇒ <html>  
    <title> Working with for </title>  
    <head> ~~<body>~~  
    <body>  
        <Script type="text/javascript">  
            for (i=0; i<=6; i++)  
            {  
                document.write("The no. is " + i),  
                document.write("<br>"),  
            }  
        </script>  
    </body>  
</html>

Q: A Script to display HTML Headings styles.

Ex:

<html>  
    <head>  
        <title> For loop </title>  
    </head>  
    <body>  
        <Script type="text/javascript">  
            for (i=0; i<=6, i++)  
            {  
                document.write("<h" + i + "> This is Heading " + i);  
                document.write("<h" + i + ">");  
            }  
        </script>  
    </body>  
</html>

Output:

## 2) While Loop:

It executes blocks of stat. Repeatedly until the given cond is False, If False, it Terminates the loop.

Syntax:

```
while (variable<=endvalue)
{
    code to be executed
    code to be executed
}
```

Ex:

```

<html>
  <head>
    <title> While Loop </title>
  </head>
  <body>
    <script type="text/javascript">
      var i=0;
      while (i<=5)
      {
        document.write("The value is:" + i)
        document.write("<br/>");
        i++;
      }
    </script>
  </body>
</html>

```

3) do - while loop:

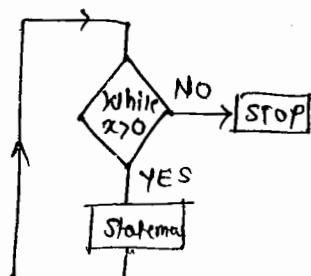
Executes blocks of stat repeatedly until times.

Syntax:

```

do
{
  Code to be Executed
  Code to be Executed
}
while (variable<=end value);

```

Ex:

```

<html>
  <head>
    <title> do - while loop </title>
  </head>
  <body>
    <script type="text/javascript">
      var i=0;
      do
      {
        document.write("The value is:" + i),
        document.write("<br/>");
        i++;
      }
      while (i<=5)
    </script>
  </body>
</html>

```

## \* Working with break And Continue :

Date: 22. SEP. 2023

106

### 1) break :

Using this stat. we can break execute of the prog. under a specific cond?

Ex: <script type="text/javascript">

```
i=0;
do
{
if (i==2)
{
break;
}
document.write("The no. is " + i);
document.write("<br/>");
i++;
}
while (i<=5);
</script>
```

### 2) Continue :

Using this stat. we can execute group of stat. repeatedly, within the given cond? matching the stat. not executed.

Ex: <html>

```
<body>
<script type="text/javascript">
var i=0;
for (i=0; i<=10; i++)
{
if (i==4)
{
continue;
}
document.write("The number is " + i);
document.write("<br/>"),
}
</script>
</body>
</html>
```

### Output:

The number is 0  
The number is 1  
The number is 2  
The number is 3  
The number is 4  
The number is 5  
The number is 6  
The number is 7  
The number is 8  
The number is 9  
The number is 10

## \*function \*

- A f<sup>n</sup> is a block of code that will be executed only by an occurrence of an event at that time f<sup>n</sup> is called.

## \* Working with JavaScript Function:

- func<sup>n</sup> is a group of reusable code which can be called anywhere in Ur prog. This eliminates the need of writing same code Again And Again.

This will help programmers to write modular code.

U can divide Ur big prog. in a no. of small & manageable f<sup>n</sup>s.

It is called Code Reusability.

functions can be defined Both in the head and body sections.

### Syntax:

```
function functionname(var1,var2,...,varx)
{
    Specify some code
    Specify some code
    Specify some code
}
```

```
Ex: <html>
    <body>
        <script type="text/javascript">
            function HTML_Headings()
            {
                for(i=1; i<=6; i++)
                {
                    document.write("<h" + i + "> The Heading is:" + i);
                    document.write("</h" + i + ">");
                }
            }
        </script>
    </head>
    <body>
        <button onclick="HTML_Headings()">HTML_Headings </button>
    </body>
</html>
```

### Output:

```
The Heading is : 1
The Heading is : 2
The Heading is : 3
The Heading is : 4
The Heading is : 5
The Heading is : 6
```

Ex: Write a Script To display msg from TextBox.

\* 108

```
> <head>
<script>
function DisplayMessage(message)
{
    alert("Ur message is:" + message);
}
</script>
</head>
<body>
<form>
Enter Message: <input type='text' name='message'>
                <input type='submit' onclick="DisplayMessage(form.message.value)" value="Display Message"/>
</form>
</body>
```

### \* Function Calling :

In JavaScript fn is divided into the following 2 parts:

- 1) Called function
- 2) Calling function

Ex: <script language="JavaScript">  
function showAddress()  
{  
 document.write("Subba Raju");  
 document.write("<br>");  
 document.write("Sr. Quality Analyst"),  
 document.write("<br>");  
 document.write("Naresh i Technology"),  
 document.write("<br>");  
 document.write("Hyderabad");  
}

Called

</script>  
<script language="JavaScript"> showAddress();  
</script>

Calling

Output:

Q: Write a Script to Return Values from a function.

109

```
> <head>
  <script>
    function myFunc()
    {
      var a=10;
      alert("My value is: "+a);
    }
  </script>
</head>
<body>
  <p> Click the button to display the value... </p>
  <input type="submit" onclick="myFunc()" value="Display_Value"/>
</body>
```

\* Working with Return Stat:

It is used to specify the value i.e. return from fn. A JavaScript fn can have an optional return stat. This is required if u want to return a value from the fn.

```
<head>
<script>
  function myFunc()
  {
    return ("Welcome to Javascript");
  }
</script>
<script>
  document.write(myFunc());
</script>
```

Q: Write a Script to Return sum of 2 nos.

```
> <script>
  function add(x,y)
  {
    return x+y;
  }
</script>
<script>
  document.write("The sum is: " + add(2,3));
</script>
```

## \* Calling a f<sup>n</sup> with ~~Argumemt~~:

When U call a f<sup>n</sup>, U can pass some values

to it, These values r called Arguments or parameters. These Arguments can be used inside the f<sup>n</sup>. U can send as many arguments as U like separated by commas.

(11)

Syntax:

```
function myFunction(var1, var2) Ex:  
{  
  JS statements  
  JS statements  
  JS statements  
}
```

```
<head>  
<script>  
function Emp(name, job)  
{  
  alert("You are " + name + ", Job is: " + job);  
}  
</script>  
<head>  
<body>  
<button onclick="Emp('ksRajy', 'S Engineer')"  
      > passingArgs </button>  
</body>
```

Ex:

```
=> <head>  
<script>  
function Emp_Details(name, job)  
{  
  alert("You are " + name + ", Job is " + job);  
}  
</script>  
<head>  
<body>  
<button onclick="Emp_Details('ksRajy', 'S Engineer')"  
      ksRajy </button>  
<button onclick="Emp_Details('Ravi', 'Sr. Engineer')"  
      Ravi </button>
```

## Lifetime of Javascript Variables:

31050 24. SEPT. 2013 (III)

### 1) Local Javascript Variables:

A variable declared within Javascript f<sup>n</sup> becomes local that we can access within the f<sup>n</sup>. U can have local variables with the same name in diff. f<sup>n</sup>s. Local Variables are recognized by the f<sup>n</sup> in which they are declared. Local variables r deleted as soon as the f<sup>n</sup> Execut<sup>n</sup> is completed.

Ex: <script>

```

year=2011;
function Global_Local()
{
    month=2;
}
Global_Local();
document.write("year is:" + year +
                ", Month is:" + month);
</script>

```

<script>

```

var z=10;  $\Rightarrow$  Global Variable
function fun1()
{
    var x=10;  $\Rightarrow$  Local Variable
}
function fun2()
{
    var y=20;  $\Rightarrow$  Local Variable
}
function fun3()
{
    var z=30;  $\Rightarrow$  Local Variable
}
</script>

```

### 2) Global Javascript Variables:

Variables declared outside the f<sup>n</sup> becomes Global

These variables can access in all scripts & f<sup>n</sup>s. Global variables are deleted when U close the page.

Ex: <script>

```

year=2012;
function Local_Variables()
{
    month=2;
}
function Global_Local_Variables()
{
    month=4;
}
Local_Variables();
Global_Local_Variables();
document.write("year is:" + year + ", month is:" + month);
Global_Local_Variables();

```

## \* Background Color :

```

<head>
  <script>
    function Bg_color1()
    {
      document.bgColor="red";
    }
    document.function Bg-color2()
    {
      document.bgColor="green";
    }
    function Bg-color3()
    {
      document.bgColor="blue";
    }
  </script>
</head>
<body>
  <input type="submit" onclick="Bg_color1()" value='color-Red' />
  <input type="submit" onclick="Bg_color2()" value='color-Green' />
  <input type="submit" onclick="Bg_color3()" value='color-Blue' />
</body>

```

In Javascript, bgColor is a property in document object. By using this property, we can change background color of Webpage. The Ex. describes, bgColor property.

## document.body.bgColor :

{  
fmp : Without <body> Tag, the O/P will not Generate}

This property executes only in body part of the Webpage.

```

<html>          Method 1
  <body>
    <script> document.body.bgColor="teal"; </script>
    <p> The background color was changed by a Script </p>
  </body>
</html>

```

```

<html>          Method 2
  <body>
    <input type="submit" onclick="document.body.bgColor='teal'" value='Click Me' />
    <p> The background color was changed by a script </p>
  </body>
</html>

```

Q: Write a script to change webpage background color with the help of dropdown.

(13)

```
→ <body>
<form name="bgcolorform"> Try it Now: Select onChange="if (this.selectedIndex==0)
document.bgColor=this.options[this.selectedIndex].option value='choose'> set
background color option value="FF0000"> Dark Red no if-
option value="00FF00"> Dark Green
option value="0000FF"> Dark Blue
option value="000000"> Black
option value="FFFFFF"> White

</select>
</form>
</body>
```

## \* Real-time Application on JavaScript Application

```
<html>
<head>
<title> JavaScript Tutorial </title>
<script language="JavaScript">
function getFullName()
{
    var firstName, lastName, fullName;
    firstName = document.creditApplication.txtFirstName.value;
    lastName = document.creditApplication.txtLastName.value;
    fullName = firstName + " " + lastName; document.creditApplication.txtFullName
    .value = fullName;
}
</script>
</head>
<body>
<h2> RealTime Application </h2>
font face="Verdana, Tahoma, Arial" size="3">
form Name="Credit Application">
<table border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="100"> First Name: </td>
<td><input type="text" name="txtFirstName" size="14"></td> </tr>
<tr>
<td width="100"> Full Name </td>
<td><input type="text" name="txtFullName" size="30"></td> </tr>
<tr>
<td width="100"> <br> <br> </td>
```

RealTime Application

First Name:

Last Name:

Full Name:

Show FullName

```
<input type="button" value="Show Full Name" onclick="getFullName()>
</td>
</tr>
</table>
</form>
```

### Built-in functions:

JavaScript contains several built-in f<sup>n</sup> like ~~etc.~~

- 1) alert
- 2) confirm
- 3) prompt etc.

### JavaScript print f<sup>n</sup>:

This send the contents of Webpage to the user's printer.  
It is equivalent to fillemenu print option.

This print method we should click in a event. When user performing the event, the method will be executed.

```
<body>
  <form>
    <input type="submit" value="printThisPage" onclick="window.print()"/>
  </form>
</body>
```

javascript - window.location: Using this property, we can redirect current Webpage. Whenever U redirect, U should mention a new location.

```
<body>
  <script> window.location = "http://www.nareshit.com";</script>
</body>
```

25. SEPT. 2012

## \* JavaScript Events:

By using javascript, we have the ability to create dynamic Webpages. Events or Actions that can be detected by javascript. Every element on a Webpage has certain Events which can trigger in Javascript.

Ex:

We r using onClick event on a button, that indicated when user click on button it returns the f<sup>n</sup> returned value. Events r normally used in combination with f<sup>n</sup>.

Ex: Clicking a button

A page is finished loading.

An image is finished loading.

Moving mouse cursor over an element.

Entering on S/I/P field.

Submitting a form.

## \* javascript Events:

| <u>Event</u>   | <u>Description</u>                                                  |
|----------------|---------------------------------------------------------------------|
| i) abort       | occurs when the user cancels loading of an image.                   |
| ii) click      | occurs when the user clicks on a link or form element change occurs |
| iii) error     | occurs when an error happens during loading of a document or image  |
| iv) focus      | occurs when the focus is given to a form element or a window        |
| v) load        | occurs when a page is loaded into Navigator.                        |
| vi) mouseout   | occurs when the user moves the ptr off                              |
| vii) mouseover | occurs when the user moves the ptr over                             |
| viii) reset    | occurs when the user clears form using reset button.                |
| ix) select     | occurs when the user selects a form elements field.                 |
| x) submit      | occurs when a form is submitted.                                    |
| xi) unload     | occurs when a user leaves a page.                                   |

Ex:

### onClick:

This event occurs, when user clicks on an Element.

#### Syntax:

1) In HTML,

```
<element onclick = "SomeJavaScriptCode">
```

2) In JavaScript

```
Object.onClick = "someJavaScriptCode"
```

\* The following html elements, not supported onClick Event:

<base>, <bdo>, <br>, <head>, <html>, <iframe>, <meta>,  
<param>, <script>, <style> and <title>.

\* onClick Supported JavaScript Object:

Document, Window.

Ex: <Script>

```
function Myfunc()
```

```
{
```

```
    alert ("It is called Function");
```

```
}
```

*Element*

*Event*

*JsCode*

*ClickMe*

*</body>*

Ex:

```

<head>
  <script>
    function Display_Datec()
    {
      document.getElementById('demo').innerHTML = Datec;
    }
  </script>
</head>
<body>
  <button onclick="Display_Datec()"> Display_Date </button>
  <p id='demo'> </p>
</body>

```

### \* onload:

This event occurs, when an object has been loaded. It is most often used within the body element to execute a script once a webpage has completely loaded.

#### Syntax:

- 1) In HTML,  $\Rightarrow$  <element onload="SomeJavaScriptCode">
- 2) In JavaScript,  $\Rightarrow$  object.onload = "someJavaScriptCode"

#### Supported HTML Tags:

<body>, <frame>, <frameset>, <iframe>, <img>, <link>, <script>, <style>

etc.

#### Supported JavaScript Objects:

Document, Window, XMLHttpRequest, XMLHttpRequest

#### Ex:

```

<head>
  <script>
    function Page_Load()
    {
      alert ("Page Loaded Successfully");
    }
  </script>
</head>
<body onload="Page_Load()">
  <p> To load the page Refresh or Reopen </p>
</body>

```

Ex:1

```

<head>
<script>
  Function Display_Date()
  {
    document.getElementById("demo").innerHTML = Date();
  }
</script>
</head>
<body>
  <p> Double click on following button... </p>
  <button onclick="Display_Date()"> Display </button>
  <p id='demo'> </p>
</body>

```

Ex:2

```

<head>
<script>
  function Copy_Text()
  {
    document.getElementById("field2").value = document.getElementById("field1").value;
  }
</script>
</head>
<body>
  Field1: <input type="text" id="field1" value="HelloWorld!"> → U can put any name
  Field2: <input type="text" id="field2" value="NareshTech"/> <br/> <br/>
  <button onclick="Copy_Text()> Copy Text </button> → On Double Click
  Cursor move on field 2 and copy first field content.

```

### \* onmouseover And onmouseout:

onmouseover event occurs when you bring Ur mouse ptr over an element onmouseout event occurs when you takes Ur mouse out of an element.

Ex:1

```

<a href="http://www.nareshTech.com/" onmouseover="document.bgColor='#FFFF00'" 
  onmouseout="document.bgColor='#FFFFFF'">
  Move Ur Mouse Over Me!
</a>

```

Ex2:

```

<script>
  Function M_Over()
  {
    alert("Mouse Over");
  }
  Function M_Out()
  {
    alert("MouseOut");
  }
</script>

```

```

<body>
<button onmouseover="M_Over()" onmouseout="M_Out()>
  Bring Mouse pointer </button>
</body>

```

Ex:

```
<marquee onmouseover="this.stop()" onmouseout="this.start()>
    Bring Pointer </marquee>
```

### ~~Marquee Tag with ondblclick~~

Q.P.:

```
<marquee behaviour="scroll" direction="left">
    Scrollamount="6"
```

```
<id="marquee1"> DblClick </marquee>
```

**Slower****Medium****Faster**

```
<input type="button" value="slower" ondblclick="document.getElementById('marquee1').  
setAttribute('scrollamount', 1, 0);">
```

## \* What are Event Handlers ?

→ Event Handlers are Scripts, which programmer using to handle events.  
In JavaScript, There are the following list of Event Handlers.

| <u>Object</u>                                         | <u>Event Handlers Available</u>      |
|-------------------------------------------------------|--------------------------------------|
| i) Selection List                                     | onBlur, onChange, onFocus.           |
| ii) Text Element                                      | onBlur, onChange, onFocus, onSelect  |
| iii) Textarea Element                                 | onBlur, onChange, onFocus, onSelect. |
| iv) Button Element                                    | onClick                              |
| v) CheckBox                                           | onClick                              |
| vi) Radio Button                                      | onClick                              |
| vii) Hypertext Link                                   | onClick, onMouseOver, onMouseOut     |
| viii) Clickable Image <del>maps</del> <sup>Area</sup> | onMouseOver, onMouseOut              |
| ix) Reset Button                                      | onClick                              |
| x) Submit Button                                      | onClick                              |
| xi) Document                                          | onLoad, onUnload, onError            |
| xii) Window                                           | onLoad, onUnload, onBlur, onFocus    |
| xiii) framesets                                       | onBlur, onFocus                      |
| xiv) form                                             | onSubmit, onReset                    |
| xv) Image                                             | onLoad, onError, onAbort             |

## Form Events :

The following events are of form level.

### 1) onblur:

This event occurs, when we're leaving the object. It is very useful in Mainly used for validation form validation.

Note: onblur event is opposite of onfocus.

#### Syntax: In HTML:

```
<element onblur="someJavaScriptCode">
```

In JavaScript:  
Object.onblur = "SomeJavaScript Code";

#### Not Supported HTML Elements:

<base>, <bdo>, <br>, <head>, <html>, <iframe>, <meta>, <param>, <script>,  
<style> & <time>

#### Supported JavaScript Objects:

Document, Window.

Ex: <html> <head>

<script>

```
function Upper_Case()
{
    document.getElementById('fname');
    x.value = x.value.toUpperCase();
}
```

```
function Lower_Case()
{
    document.getElementById('Lname');
    x.value = x.value.toLowerCase();
}
```

```
</script> </head>
```

```
<body>
```

First Name: <input type="text" onblur="Upper\_Case()" id='fname' />

Last Name: <input type="text" onblur="Lower\_Case()" id='Lname' />

```
</body>
```

```
</html>
```

## 2) onfocus:

This Event occurs, when an element gets focused.  
onFocus is most often used with ~~script~~,  tag,  tag &  tag.

122

### Syntax:

1) HTML, <element onfocus="SomeJavaScriptCode">

### 2) JavaScript,

object.onfocus = "SomeJavaScriptCode"

### Not Supported Elements:

<base>, <bdo>, <br>, <head>, <html>, <iframe>, <meta>,  
 <param>, <script>, <style> and <title>

### Supported HTML Elements:

document, window.

### Ex:

```

<head>
<script>
Function SetStyle(x)
{
    document.getElementById(x).style.background='green';
}
</script>
</head>
<body>
    First Name: <input id='fname' onfocus="SetStyle(this.id)">
    <br/>
    Last Name: <input id='lname' onfocus="SetStyle(this.id)">
</body>
</html>

```

## 3) onResize: This Event occurs, when the size of an Element has changed.

### Ex: <script>

```

Function ShowMsg()
{
    alert ("Window Resized");
}
</script>
<body onresize ="ShowMsg()">
    <p>Restore or Maximize the Browser.</p>
</body>

```

### Syntax:

#### 1) HTML:

<element onresize="someJavaScriptCode">

#### 2) JavaScript:

object.onresize = "SomeJavaScriptCode"

## Working With JavaScript Errors And Exceptions

There are 3 Types of Errors in programming.

- 1) Syntax Errors
- 2) Runtime Errors
- 3) Logical Errors

### 1) Syntax Errors:

These are called parsing Errors, occurred at compiletime for traditional programming lang, At Interpret time for javascript.

Ex: `<script type="text/javascript">`  
    `<!--`  
    `window.print();`  
    `-->`  
`</script>`

In above Ex, Syntax Error occurs, In JavaScript a block contains a Syntax Error. It is affected to complete Block.

### 2) Runtime Errors:

These r called Exceptions occurred during Execut.

Ex: `<script type="text/javascript">`  
    `<!--`  
    `window.printme();`  
    `-->`  
`</script>`

### 3) Logical Errors:

These r the most difficult types of Errors to track down, These Errors r not the result of Syntax or Runtime Error. It is the mistake in the logic that drives your script. These errors unable to catch Box, it depends on Ur Business Requirements.

## JavaScript try...catch...finally statements:

That stat. allows U to test

a block of code for Errors. In Latest JavaScript version, added Exception Handling Capabilities.

### Catching Errors:

When browsing Webpages on Internet, sometimes JavaScript alert box displays asking "Do U wish to Continue?" or "Do U wish to debug?" These msgs r Very useful for Developers, not End users.

try...catch Statement: It allows us to test a block of code for errors. That try block contains the code to be run, catch block contains code to be executed, if an error occurs.

Syntax:

```
<script>
try {
    Code to run [break;]
}
catch(e)
{
    Code to run if an exception occurs [break;]
}
</script>
```

Ex:

```
<html>
<head>
<script>
function MyFun()
{
    var a=10;
    alert("The value is: " + a);
}
</script>
</head>
```

```
<button onclick="MyFun()"> click </button>
</body>
</html>
```

O/p:

Ex:

```
<script>
alert("Welcome To Try");
alert("Thank U..");
</script>
```

In the above script, There is runtime error, We should handle with try...catch block like as follows:

Ex:

```
<script>
try {
    alert("Welcome To Try");
}
catch(e)
{
    alert(e.description)
}
alert("Thank U..");
</script>
```

O/p :

Ex:

```
<html>
<head>
<script>
function MyFun()
{
    var a=10;
    try {
        alert("The value is: " + a);
    }
    catch(e)
    {
        alert(e.description)
    }
}
</script>
</head>
```

body

```
<button onclick="MyFun()"> click </button>
</body>
</html>
```

O/p:

finally: By using this stat. we can display block of statements which r need to execute if it is not related to catch block.

Ex: <script>  
function Myfun1()  
{  
var a=10;  
try  
{  
alert("The value is: "+a);  
}  
catch(e)  
{  
alert(e.description);  
}  
finally  
{  
alert("Always Executing");  
}  
}

### Ex: Throw statement

⇒ <script>  
var x=prompt("Enter the no. between 5 and 10");  
try  
{  
if(x>10)  
{  
throw "Err1";  
}  
else if(x<5)  
{  
throw "Err2";  
}  
else if(isNaN(x))  
{  
throw "Err3";  
}  
}

### Catch (err)

{  
if(err=="Err1")  
{  
document.write("Error! The value is too  
High.");  
}  
if(err=="Err2")  
{  
document.write("Error! The value is too  
Low.");  
}  
if(err=="Err3")  
{  
document.write("Error! The value is not a  
number");  
}

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

28. SEPT. 2022

### Throw statement:

This stat. allows u to create an exception. If u use this stat. together with try-catch stat., u can control the flow of the prog.

Syntax: throw exception

Note: Throw is written in lowercase letters. If u write Uppercase Javascript generates an error.

In above Example, a value is higher than 10 or less than 5 or not a number. We're going to throw an Error. That Error is caught by the catch argument & displays proper error msg.

### Window.onerror:

It is a feature in JavaScript, The Error Event is fired whenever a mistake available on window object. It is an Event handler whenever an Exception occurred.

Ex: `<script>`

```
window.onerror=function()
{
    alert("Error Handled");
}
<script>
<body>
<button onclick="Myfunc()">
    ClickMe </button>
</body>
```

The onError event handler provides 3 pieces of info:

1) Error Msg.

2) URL

3) Line Number.

### 1) Error Message :

The same msg that the browser would display for the given error.

### 2) URL :

The file in which the error occurred.

### 3) Line Number :

The line no. in the given URL caused the Error.

Ex: `<script type="text/javascript">` 120  
`window.onerror=function(msg,url,line)
{
 alert("Message: "+msg);
 alert("Url: "+url);
 alert("Line Number: "+line);
}`  
`</script>`  
`<head>`  
`<body>`  
`<p> Click the following to see the Results </p>`  
`<form>`  
`<input type="button" value="Click Me" onclick="Myfunc()">`  
`</form>`

## ★ Working With JavaScript Objects

1) ~~String Object~~

2) ~~Date Object~~

3) ~~Math Object~~

In the JavaScript objects are the following Types.

1) Array Objects

2) Boolean Object

3) Date Object

4) Math Object

5) String Object

6) Number Object

7) RegExp Object

8) Global properties & Functions.

### Browser objects:

- 1) Window Object
- 2) Navigator Object
- 3) Screen Object
- 4) History Object
- 5) Location Object

## Core DOM objects:

- 1) DOM Node
- 2) DOM NodeList
- 3) DOM NamedNodeMap
- 4) DOM Document
- 5) DOM Element
- 6) DOM Attr

## HTML DOM objects:

- 1) Document Object
- 2) Event Object
- 3) HTMLElement Object
- 4) Anchor Object
- 5) Area Object
- 6) Base Object
- 7) Body Object
- 8) Button Object
- 9) Form Object
- 10) Image Object
- 11) Input Button Object
- 12) Input Checkbox Object
- 13) Input File Object
- Etc.

## Methods:

Methods are actions that can be performed on object.

### Methods

- 1) Resize
- Close
- Open
- Maximize
- Minimize
- Restore etc.

### Ex:

<script>

```
Var txt = "Naresh i Technologies";
document.write(txt.toUpperCase());
document.write("<br/>");
document.write(txt.toLowerCase());
```

</script> SAVE AS: JavaScript-Methods.htm

NARESH TECHNOLOGIES Oct. 2012  
Naresh Technologies

## 1) Javascript Array Object:

Array object is used to store multiple values in a single variable.

### \* creating Javascript Array:

Creating an Array slightly diff. from creating a normal variable bcoz, JavaScript has variables & properties associated with arrays.

### Creating an Array:

In Javascript, we can create an array in following 3 ways.

#### Way 1: Regular

```
Var myNames = new Array();
myNames[0] = "Ravi";
myNames[1] = "Nilesh";
myNames[2] = "Pintu";
```

#### Way 2: Condensed

```
Var myNames = new Array("Ravi", "Nilesh",
" Pintu");
```

#### Way 3: Literal

```
Var myNames = ["Ravi", "Nilesh", "Pintu"];
```

### Ex: <script>

```
Var txt = "Naresh i Technology";
document.write(txt.length);
</script>
```

### Properties:

Properties r the values associated with an object.

Properties : Width

Height

Name

Length

We can create objects as a container for related variables.

Variables of an object are called properties.

To define a JavaScript Object we use the new Object() keyword.

Once, we created a new Object, we can assign properties to it. These properties behaves exactly like variables.

Ex:

```
<html>
<head>
<title> Person Object </title>
<script>
    person = new Object();
    person.name = "Nilesh";
    person.age = 25;
    person.sex = 'Male';
    person.height = 5.5
document.write(person.name + " is "
    + person.age + " years old"
    + person.sex + " and "
    + person.height + " foot tall.");
</script>
</head> SAVE AS: JavascriptObject.htm
<html> O/P : Nilesh is 25 years old Male
and 5.5 foot tall,
```

There is a keyword associated with an object which is WITH. WITH creates a kind of halfway Reference. We can drop the repeated references to 'person'. The person has already made with keyword.

```
<html>
<head>
<title> Person Object </title>
<script>
```

```
    person = new Object();
    with (person)
    {
        name = 'Ram',
        age = 30,
        sex = 'Male',
        height = 6
    }
    document.write(name + " is " + age +
        " years old, " + sex + ", and "
        + height + " foot tall.");
</script>
</head>
</html> SAVE AS: JavascriptObjectWith.htm
```

O/P : Ram is 30 years old Male and 6 foot tall

\* Remember following pts. while Using Arrays.

- i) Array is a special type of variable.
- ii) Values are stored into an Array by using the Arrayname.  
Ex: myArray[2] = "Hello World";
- iii) Values in an Array are accessed by the ArrayName & locat<sup>n</sup> of the value.

Ex: myArray[2];

iv) JavaScript has Built-in-functions for Arrays.

\* Access An Array:

U refers to an element in an Array by referring to the index number.

Ex: var name=myNames[0];

The above stat. modifies the 1<sup>st</sup> element

myNames[0] = "Pankaj";



(39)

```

<script>
arr = new Array
('fish.jpg', 'smiley.jpg', 'nature.jpg', 'Nature3.jpg', 'Nature5.jpg');
('Nature3.jpg', 'Nature5.jpg'),
i=0;
function fun1()
{
i++;
if (i==5)
{
i--;
alert("No more images")
}
else
{
document.getElementById('img1').src ="img/" + arr[i];
}
</script>

<br>
<input type="button" value="NEXT" onclick="fun1()"/>

```

Q: Write a Script to display Reverse Array Elements.

02.OCT.2014

```

<head>
<script>
var names = ["Nile", "Pin2", "Pawan"];
function myfunc()
{
names.reverse();
var x = document.getElementById('demo');
x.innerHTML = names;
}
</script>
</head>
<body>
<button onclick="myfunc()">
<p id='demo'> Click on following button to get Reverse array elements...!</p>
<button onclick="myfunc()"> Reverse - A Element </button>
</body>

```

Ques:

Using this method, remove last element of an Array

```
<head>
<Script>
var names = ["zyx", "abc", "ooo"];
function myFunc()
{
    names.sort();
    var x = document.getElementById('demo');
    x.innerHTML = names;
}
</Script>
</head>
<body>
<p id='demo'> Click on following Buttons to Remove Array Elements </p>
<button onclick="myFunc()"> Pop-Elements </button>
</button>

```

### \* Javascript Boolean Object:

Boolean object represents boolean values either true or false. The Boolean object is used to convert a non boolean value to a boolean value.

Syntax: For creating a boolean object

```
var myBoolean = new Boolean();
```

Ex: <script>

```
var b1 = new Boolean(0);
var b2 = new Boolean(1);
document.write("0 is Boolean" + b1 + "<br/>");
document.write("1 is Boolean" + b2 + "<br/>");
```

```
</script>
```

If boolean object has no initial value or the passed value any one of the following

0

-0

null

""

false

undefined

NaN (Not a Number)

Opp:

0 is Boolean false

1 is Boolean True.

## \* Boolean Object properties:

### Properties      Description

- i) constructor It returns the fn that created the boolean objects prototype.
- ii) prototype It allows u to add properties and methods to a boolean object.

### Methods:

#### Method      Description

- i) toString() Converts a boolean value to a string & Returns the result.
- ii) valueOf() It returns the primitive value of a boolean object.

Q: Write a script using toString() object method.

→ <html>

  <script>

```
  function myFunction()
  {
    var myVar = new Boolean(1);
    var x = document.getElementById("demo");
    x.innerHTML = myVar.toString();
  }
```

</script>

  <body>

    <p id="demo" > Click the button to Display the button to get string </p>
    <button onclick="myFunction()" > Display </button>

  </body>

</html>

## \* Working with Javascript Date Object:

Date object is a datatype built-in

into the javascript. date objects are created with the new date. Once, a date object is created, no. of methods allows u to operate on it.

~~Most~~ Most method simply allows u to get and set the year, month, seconds, date, hour etc

Q: Write a script to display current date.

→ <head>

  <script>

```
  function Display()
  {
    document.getElementById('demo').innerHTML = Date();
  }
```

</script>

</head>

  <body>

    <p id="demo" > Click on this button to display date time...!! </p>

    <button onclick="Display()" > Display Date </button>

</body>

The date object we can instantiate in the following 4 ways.

139

Way 1: Var x = new Date();

Way 2: Var x = new Date(milliseconds);

Way 3: Var x = new Date(dateString);

Way 4: Var x = new Date(year, month, day, milliseconds);

Date object contains following 7 Arguments.

- 1) year
- 2) Month
- 3) Day
- 4) Hour
- 5) Minute
- 6) Seconds
- 7) Milliseconds.

These 7 arguments are mainly contains date object.

Ex:

```
<script>
    var dt=Date();
    document.write("Date And Time is: "+dt);
</script>
```

### Get the javascript time:

The Date Object has been created, That variable holds current Time And Date. To get the information we need to display using the following fns.

1) getTime(): displays no. of Milliseconds.

2) getSeconds(): Displays no. of Seconds.

3) getMinutes(): Displays no. of Minutes.

4) getHours(): Displays no. of Hours.

5) getDay(): Displays day of week. (0-6) 0=sunday 6=saturday.

6) getMonth(): Number of Month (0-11)

7) getDate(): Displays day of Month (0-30)

8) getFullYear(): The four digit year.

Ex: Write a Script to Get Current Month, Date, Year.

<Script>

```
var currenttime = new Date();
var month = currenttime.getMonth() + 1
var day = currenttime.getDate()
var year = currenttime.getFullYear()
document.write(month + "/" + day + "/" + year)
```

</script>

Ex: Write a Script To Display Get Full Year. (H.W)

<Script>

```
var currenttime = new Date();
function DisplayYear()
{
    var d = new Date();
    var x = document.getElementById('demo');
    x.innerHTML = d.getFullYear();
}
```

</script>

<body>

<p id='demo'> ~~click the button to get full year...!!~~ </p>

<button onclick="DisplayYear()"> Full Year </button>

</body>

## \* Set Dates:

Date Object supports setDate, we can easily manipulate the date by using the methods available in the date object

### 1) setDate:

Using this method, we can set the Day of Month to the Date object.

Syntax:

Date.setDate(day)

Ex:

~~var d = new Date();~~

Write a script to display the setDate.

&lt;html>

<script>

function myFunction()

{

var d = new Date();

d.setDate(15);

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

x.innerHTML = d;

</script>

```

<body>
  <p id="demo"> Click the button to display date after changing day of month </p>
  <button onclick="myFunction()"> display </button>
</body>

```

## \* JavaScript String Object:

03.Oct.2019

~~Object~~ The string object is used to manipulate a stored piece of text. String objects are created with new String().

Syntax: var txt=new String("string"); OR  
var txt="string";

\* String Object Properties: String object has a few properties.

### i) length:

This property returns the length of a string in characters.

Syntax: string.length

Ex:

```

<script>
  var str="NareshiTech";
  document.write(str.length);
</script>

```

### ~~Methods~~ \* String Object Methods:

String object contains several methods like ...

toUppercase(), toLowercase(), charAt(), ...

Ex:

```

<script>
  var str="NareshiTech";
  document.write(str.charAt(3));
  document.write("<br/>");
  document.write(str.toUpperCase());
  document.write("<br/>");
  document.write(str.toLowerCase());
</script>

```

~~<body>~~

~~<p id="demo"> Click the button to display the string~~

```

<head>
<script>

function Uppercase()
{
    var str="Naresh i Technologies";
    document.getElementById('demo').innerHTML=str.toUpperCase();
}

function Lowercase()
{
    var str="Naresh i Technologies";
    document.getElementById('demo1').innerHTML=str.toLowerCase();
}

function char_Index()
{
    var str="Naresh i Technologies";
    document.getElementById('demo2').innerHTML=str.charAt(5);
}

</script>
</head>
<body>
    Click To Uppercase <p id='demo'>
    <button onclick="Uppercase()> Uppercase </button>
    Click To change To Lowercase <p id='demo1'>
    <button onclick="Lowercase()> Lowercase </button>
    Click To display Index char<p id='demo2'>
    <button onclick="char_Index()> Display_Char</button>
</body>

```

### JavaScript String Match():

This method searches a string for a match. If no match found, it returns null.

#### Syntax:

`string.match(regexp)`

Ex: `<head>`

`<script>`

`var str = "JavaScript";`

`document.write(str.match("script"));`

`</script>`

`</head>`

O/P: Null



### JavaScript String Styles:

JavaScript contains all formatted string styles like HTML.

`<script>`

`var txt="Naresh i Tech";`

`document.write("<p>Big: "+txt.big()+"</p>");`

`document.write("<p>Small: "+txt.small()+"</p>");`

`document.write("<p>Bold: "+txt.bold()+"</p>");`

`document.write("<p>Italic: "+txt.italic()+"</p>");`

document.write("<p> Fixed: " + txt.fixed() + "</p>");  
document.write("<p> Strike: " + txt.strike() + "</p>");  
document.write("<p> fontcolor: " + txt.fontcolor("green") + "</p>");  
document.write("<p> fontsize: " + txt.fontSize(6) + "</p>");  
</script>

### String Replace:

By using this Method, we can Replace the String.

#### Ex:

```
<script>
var txt = "visit to Nagpur";
document.write(txt.replace("Nagpur", "Chandrapur"));
</script>
```

## \* JavaScript Math Object:

It allows u to perform mathematical Tasks. The Math object contains several mathematical constants & Methods.

### \* Math object properties.

#### 1) JavaScript PI property:

Ex: write a script To display PI value.

```
⇒ <head>
<script>
function PIValue()
{
    document.getElementById("demo").innerHTML = Math.PI;
}
</script>
</head>
<body bgcolor="#lightblue">
<p id='demo'> Click to display PI Approximate value </p>
<button onclick = "PIValue()">Display - PI </button>
</body>
```

#### 2) abs

it Returns absolute value of x.

Math.abs(-1)

Q: Write a script to display set of Absolute values when U Click the Button.

→ <html>

```
<Script>
function Myfunction()
{
    var a = Math.abs(7.25);
    var b = Math.abs(-7.25);
    var c = Math.abs(null);
    var d = Math.abs("Hello");
    var e = Math.abs(2+3);
    var x = document.getElementById("demo");
    x.innerHTML = a + "<br/>" + b + "<br/>" + c + "<br/>" + d + "<br/>" + e;
}
</Script>
<body>
<p id="demo"> Click the Button to see the Absolute value of different numbers </p>
<button onclick="myfunction()">Absolute_value</button>
</body>
</html>.
```

round():

```
<Script>
document.write(Math.round(0.60) + "<br/>");
document.write(Math.round(-4.40) + "<br/>");
</Script>
```

## A Javascript Number Object:

In Javascript, The Number object is an object wrapper for primitive numeric values. Number Objects are created with new Number().

Syntax: var num = new Number(value);

Number Object properties:

| <u>Property</u> | <u>Description</u>                                  |
|-----------------|-----------------------------------------------------|
| MAX_VALUE       | It Returns the largest no. possible in Javascript.  |
| MIN_VALUE       | It Returns the smallest no. possible in Javascript. |
| NAN             | Represents "Not-A-Number" value.                    |
| prototype       | Allows U to add properties & Methods to an object.  |

Q: Write a script to display Max & Min value in a Javascript.

→ <Script>

```
function Max_VALUE()
{
    document.getElementById('demo').innerHTML = Number.MAX_VALUE;
```

```

function Min_Value()
{
    document.getElementById("demo1").innerHTML=Number.Min_Value;
}
</script>
</head>
<body>
<p id="demo"> Click to display Max value in JavaScript </p>
<button onclick="Max_VALUE()"> Max_Value </button>
<p id="demo"> Click to display Min value in JavaScript </p>
<button onclick = "Min_VALUE()"> Min_Value </button>
</body>

```

04. Oct. 2012

## \* JavaScript Regular Expression Object

Q: What is RegExp?

→ It is an object that describes a pattern of characters. When U search in a Text U can use a pattern to describe what U r searching for.

A simple pattern can be a one single character, a more complicated pattern can consist of more characters.

Regular Expressions are used to perform powerful pattern matching of Search and replace fn on Text.

Syntax:

1) var patt=new RegExp(pattern,modifiers);  
OR

2) var patt=/pattern/modifiers;

pattern specifies the pattern of an Expression.

Modifier specifies if search should be Global or Case sensitive or Multiline matcher etc.

### Brackets:

Brackets are used to find a Range of Characters.

The following Table describes Expression And Description.

| Expression       | Description                                         |
|------------------|-----------------------------------------------------|
| [abc]            | Find Any character Betw the Brackets.               |
| [^abc]           | Find Any character Not Betw the Brackets .          |
| [0-9]            | Find Any digit 0-9.                                 |
| [A-Z]            | Find Any character from Uppercase A to Uppercase Z  |
| [adgk]           | Find Any character in the given set.                |
| [A-z]            | Find Any character From Uppercase A to Lowercase Z  |
| [a-z]            | Find any character from Lowercase a to Lowercase Z. |
| [!adgk]          | Find any character outside the given set.           |
| [red/blue/green] | Find any of the Alternative specified.              |

## Quantifiers :

The following Table describes Quantifiers & Descript<sup>n</sup>.

| <u>Quantifier</u> | <u>Description</u>                                               |
|-------------------|------------------------------------------------------------------|
| $n^+$             | Matches any string that contains atleast one 'n'.                |
| $n^*$             | Matches any string that contains '0' or more occurrences of 'n'. |
| $n^?$             | Matches any string that contains '0' or '1' occurrences of 'n'.  |
| $n\$$             | Matches any string with 'n' at the end of the string.            |
| $^n$              | Matches any string that begins with 'n'.                         |

## Meta Character :

It is simply an alphabetical character preceded by a backslash (\).

The following Table describes character & Descript<sup>n</sup>.

| <u>Character</u>      | <u>Description</u>                              |
|-----------------------|-------------------------------------------------|
| $\cdot$ (period)      | Matches a single character                      |
| $\backslash s$        | A whitespace character                          |
| $\backslash S$        | Not a whitespace character.                     |
| $\backslash d$        | A Digit Bet <sup>n</sup> 0-9.                   |
| $\backslash D$        | A Non-digit                                     |
| $\backslash w$        | A Word character (a-z, A-Z, 0-9, _)             |
| $\backslash W$        | A Nonword character                             |
| $\backslash b$        | A Literal Backspace.                            |
| $[\text{aeiou}]$      | Matches a single character in the gvn set.      |
| $[\^{} \text{aeiou}]$ | Matches a single character outside the gvn set. |

## RegEx Object Properties :

This object contains the following properties.

| <u>Property</u> | <u>Description</u>                    |
|-----------------|---------------------------------------|
| Global          | Specifies if the "g" modifier is set. |
| ignoreCase      | Specifies if the "i" modifier is set. |
| Multiline       | Specifies if the "m" modifier is set. |

## JavaScript ignoreCase property:

This property specifies whether 'i' modifier is set.

This property Returns True if 'i' modifier is set. Otherwise, It returns False.

### Syntax:

RegExObject.ignoreCase

Ex: ~~Example~~

Q: Write a Script using i modifier, & check whether 'i' modifier is set or not.

```
> <html>
  <body>
    <script>
      var str = "Goto Naresit!";
      var patt1 = /NiT/i;
      if (patt1.ignoreCase)
      {
        document.write("i modifier is set!");
      }
      else
      {
        document.write("i modifier is Not set!");
      }
    </script>
  </body>
</html>
```

## \* RegEx Methods

### Methods      Description

compile()      Compile a Regular Expression

test()      Test for a match in a string, it returns True or False.

~~exec()~~      Test for a match in a string, it returns the first Match.

Ex: Write a Script using Test() Method.

> <script>

```
  var str = "Hello World!";
  var patt = /Hello/g;
  var result = patt.test(str);
  document.write("Returned Value: " + result);
</script>
```

## \* Working With Browser Objects:

This Object contains info<sup>n</sup> abt Browser. It has the following properties.

### Property      Description

appCodeName      It Returns the code name of the Browser.

appName      It Returns the Name of the Browser.

appVersion      It Returns the version info<sup>n</sup> of the Browser.

cookieEnabled      It determines whether cookie are enabled in the Browser.

Ex:

```
<script>
  document.write("Navigator info: " + navigator.appVersion);
</script>
```

**Q: Write a script to display Browser Name.**

→ <script>

```
document.write("Browser Name: " + navigator appName),  
</script>
```

## \* Navigator Object Methods:

| <u>Method</u> | <u>Description</u> |
|---------------|--------------------|
|---------------|--------------------|

javaEnabled()      Specifies if Browser javascript enable or not.

Ex:

```
<script>  
document.write("Browser Name: " + navigator.javaEnabled()),  
</script>
```

## ~~Screen~~ Screen Object:

This object contains info abt the visitor's screen.

```
<script>  
document.write("The screen width is: " + screen.width),  
</script>
```

| <u>Properties</u> | <u>Description</u> |
|-------------------|--------------------|
|-------------------|--------------------|

availHeight      Returns the Height of the screen. (Excluding the Windows Taskbar)  
availWidth      Returns the width of the screen. (Excluding the Windows Taskbar)  
height      Returns the total height of the screen.  
width      Returns the total width of the screen.

## \* History Object:

This object contains the URL visited by the user. This object is part of window object. We can access the window.history property.

| <u>Property</u> | <u>Description</u> |
|-----------------|--------------------|
|-----------------|--------------------|

length      Returns the no. of URLs in the history list.

## History object Methods:

| <u>Method</u> | <u>Description</u> |
|---------------|--------------------|
|---------------|--------------------|

back()      It loads the previous url in the history list.

forward()      It loads the Next url in the history list

go()      It loads the specific url in the history list.

## \* Length property:

By using this property, we can return URL's in the History list

(Note: Internet Explorer & Opera starts at '0', firefox, chrome & Safari starts at '1')

Syntax: history.length

Ex: `<body>`

```
<script type="text/javascript">
    document.write("Number of URL are:" + history.length);
</script>
</body>
```

Ex:

```
{ Function HBack()
{ document.write ("Number of URL
}
```

Ex:

```
<html>
<head>
<script>
    function Back_History()
    {
        window.history.back()
    }
    function Forward_History()
    {
        window.history.forward()
    }
</script>
</head>
```

Ex: Write a script to check  
Textbox Empty or Not.

```
<html>
<head>
<script>
    Function notEmpty()
    {
        var myTextField=document.getElementById('myText');
        if(myTextField.value!=" ")
            alert("User Entered Text");
        else
            alert("User Not Entered Text");
    }
</script>
</head>
<body>
    Enter Any Text: <input type='text' id='myText'>
    <button onclick="notEmpty()">Check Form</button>
</body>
</html>
```

05.OCT.2019

`<body>`

```
<button onclick="Back_History()">His_Back</button>
<button onclick="forward_History()">His_Forward</button>
</body>
```

Note: If History is available Then it works only,  
otherwise, It's Not Work"

SAVE AS:

OPP:

~~check form~~

~~check form~~

SAVE AS:

OPP:

Enter Any Text:

Ex:

```

<Script type="text/javascript">
    function calc()
    {
        alert(document.getElementById("Value1").value + document.getElementById("Value2").value)
    }
</Script>
<body>
    <form>
        <input type="text" id="value1" size="3" value="20"/> * Expected
        + <input type="text" id="value2" size="3" value="100"/> * Actual
        <input type="button" value="Submit" onclick="calc()"/>
    </form>
</body>

```

Expected

O/P:  +

1244

Submit

### \* Window Object:

(JSON: Javascript Object Notes)

Window Object represents an Open window in a Browser. If a Document contain frames the Browser creates one Window object for the html document. And one Additional Window object for each frame.

Note: There is No public standard that applies to the Window object, All Browser support it.

### \* Window Opener property:

It returns a reference to the window that created the window. When opening a window with window.open method,

Syntax: window.opener

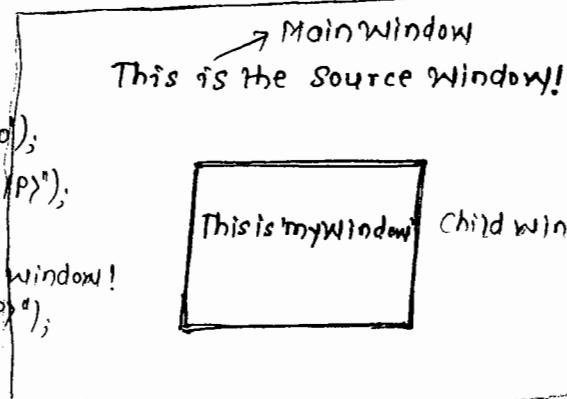
Ex:

```

<head>
<Script>
    function openWin()
    {
        myWindow=window.open(", ", 'width=200,height=100');
        myWindow.document.write ('<p>This is myWindow</p>');
        myWindow.focus();
        myWindow.opener.document.write ("<P>This is source window!
        <IP>");
    }
</Script>
</head>
<body>
    <input type="button" value="Open" onclick="openWin()"/>
</body>

```

O/P:  Open 'myWindow' ←



## Location Object:

This object contains info abt the current URL. The Location object is the part of Window object & it is accessed thr. the Window.location property.

### Location object properties:

| <u>Property</u> | <u>Description</u> |
|-----------------|--------------------|
|-----------------|--------------------|

|          |                                                    |
|----------|----------------------------------------------------|
| hash     | Returns the Anchor portion of URL                  |
| host     | Returns the hostname & port of URL.                |
| hostname | Returns the hostname of a URL.                     |
| href     | Returns the Entire URL.                            |
| pathname | Returns the pathname of URL.                       |
| port     | Returns the port number <sup>the</sup> server URL. |
| protocol | Returns the protocol of a URL.                     |
| search   | Returns the query portion of a URL.                |

Q: Write a Script To Display URL.

→ <script>  
document.write(location.href);  
</script>

~~Ex:~~  
~~<script>~~  
~~function~~

## \* Location Object Methods:

| <u>Methods</u> | <u>Description</u> |
|----------------|--------------------|
|----------------|--------------------|

|           |                                               |
|-----------|-----------------------------------------------|
| assign()  | Loads a New document                          |
| reload()  | Reload the current document.                  |
| replace() | Replaces the current document with a New one. |

Ex: <head>  
      <script>  
          function Replace\_Location()  
          {  
            window.location.replace("http://www.nareshit.com")  
          }  
      </script>  
      </head>  
      <body>  
        <button onclick="Replace\_Location() ~~onmouseover~~ Next\_Location</button>

## \* Javascript Browser Object:

03/08/2013

146

In Javascript, Navigator object contains all inf. abt visitor's browser name, version, cookie enabled etc. Using Navigator object, we can do following list of jobs.

BrowserCodeName : Mozilla

BrowserName : Netscape

BrowserVersion : 5.0 (Windows)

CookiesEnabled : true

Platform : Win32

User-agent header etc.

Ex: <body>

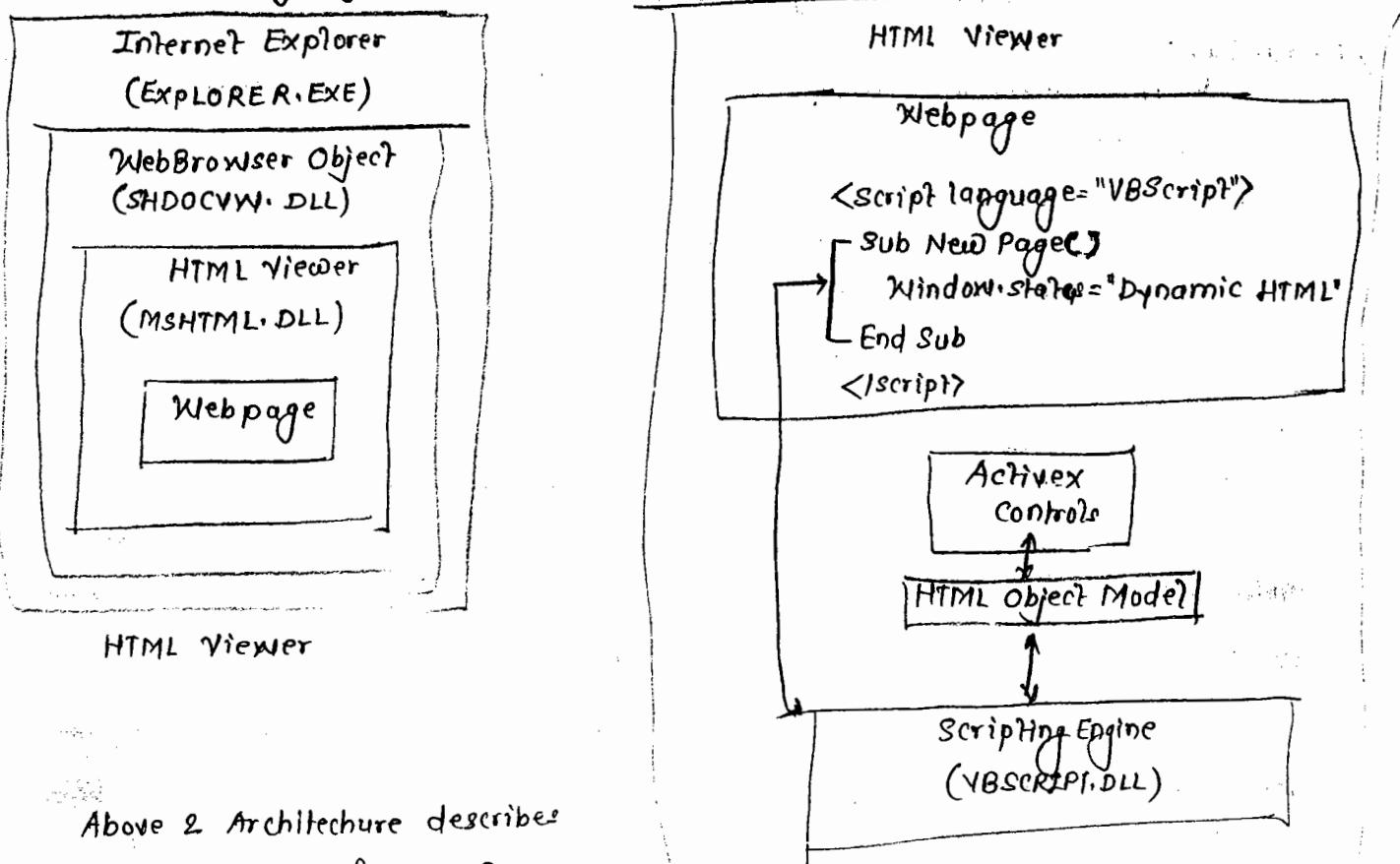
```
<div id="example"></div>
<script>
txt = "<p> Browser CodeName: " + navigator.appCodeName + "</p>";
txt += "<p> BrowserName: " + navigator.appName + "</p>";
txt += "<p> Browser Version: " + navigator.appVersion + "</p>";
txt += "<p> Browser Cookies Enabled: " + navigator.cookieEnabled + "</p>";
txt += "<p> Platform: " + navigator.platform + "</p>";
txt += "<p> User-agent header: " + navigator.userAgent + "</p>";
document.getElementById("example").innerHTML = txt;
</script>
</body>
```

## Q: How JavaScript Works?

(142)

- ⇒ Browser means collection of libraries, objects, controls, Engines, Interpreters etc.
- Whenever, We r executing any Hypertext Resources or Style Resources or Script Resources that time browser calling internally that interpreter to execute.

Following dig. describes internal layers of a Browser.



Above 2 Architecture describes Working principle of Every Browser.

Executing scripts.

\* Write a script to change when u perform onclick event.

08.Oct.2012

```
⇒ <html>
  <body>
    <script>
      function changeImage()
      {
        element = document.getElementById('myimage')
        if(element.src.match("bulbon"))
        {
          element.src = "pic_bulboff.gif";
        }
        else
        {
          element.src = "pic_bulbon.gif";
        }
      }
    </script>
```

`<img id="myimage" onclick="changemage()" border="1" width="100px" height="180">`

`<p> Click To Turn ON/OFF the light !! </p>`

## \* Working with DOM objects

↳ Document Object Model

### 1) Head Tag:

It contains general info like meta info, script info, style info, etc. The `<head>` tag inside elements shouldn't appear in body part in web browser.

The `<meta>` tag used to improve the ranking in search engine, The `<style>` tag used to apply more styles on webpage. `<script>` tag used to develop dynamic webpages. head section contains following list of tags.

| <u>Tag</u>                    | <u>Descriptions</u>                                          |
|-------------------------------|--------------------------------------------------------------|
| <code>&lt;!DOCTYPE&gt;</code> | Defines a document type, write before a start tag            |
| <code>&lt;html&gt;</code>     | To indicate html contents                                    |
| <code>&lt;head&gt;</code>     | Defines info abt webpage docs.                               |
| <code>&lt;meta&gt;</code>     | Defines meta info used to improve rankings in search engines |
| <code>&lt;title&gt;</code>    | Defines the document title.                                  |
| <code>&lt;link&gt;</code>     | Defines the link using the external style sheet.             |
| <code>&lt;base&gt;</code>     | It is used to specify URL relative to webpage URL links.     |
| <code>&lt;style&gt;</code>    | To apply style sheets.                                       |
| <code>&lt;script&gt;</code>   | To validate webpage on client-side etc.                      |

## Introduction to DOM:

DOM stands for "Document Object Model". It is W3C std. The DOM defines a standard for accessing HTML & XML documents.

W3C DOM is a platform & language neutral interface that allows programs & scripts to dynamically access & update the content, structure & style of a document.

W3C DOM std. is separated into 3 different parts:

- 1) Core DOM: It is std. Model for any structured document.
- 2) XML DOM: It is std. Model for XML documents.
- 3) HTML DOM: It is std. Model for HTML Documents.

## 1) Core DOM:

The Node object represents a Node in HTML document. The document is:

- i) The Document
- ii) An Element
- iii) An Attribute
- iv) Text
- v) A Comment.

## 2) Node Object properties:

Current Webpage

The following table describes a few Node object properties.

| <u>Property</u> | <u>Descriptions</u>                                |
|-----------------|----------------------------------------------------|
| 1) attributes   | It Returns a collection of a Nodes attributes.     |
| 2) baseURI      | It Returns an absolute base URI of a Node.         |
| 3) childNodes   | It Returns a Node_list of a childNodes for a Node. |
| 4) firstChild   | It Returns the firstChild of a Node.               |
| 5) lastChild    | It Returns the lastChild of a Node.                |

### baseURI property:

It returns the URI of the page, that specified node can be a document, element, or Attribute or Any other node type.

Note: This property supports all other major browsers Except Internet Explorer.

#### Syntax:

node.baseURI

Q: Write a script to display the URI of current page.

```
> <html>
  <script>
    function myFunction()
    {
      var x=document.getElementById("demo");
      x.innerHTML=document.body.baseURI;
    }
  </script>
<body>
  <p id="demo">
    Click the Button To Display the BaseURI </p>
  <button onclick="myFunction()">
    Display_URI </button>
  </body>
</html>
```

## XML DOM

XML DOM defines the object and properties of all XML elements, Methods etc.

XML Stands for Extensible Markup language. It is designed to transport & store data. It is easy & simple to learn. To learn XML U should have HTML & javascript fundamentals.

Q: What is XML?

- i) It is a Markup language like HTML.
- ii) It is designed to carry data, Not to display data.
- iii) XML Tags are not predefined, U must define ur own tag.
- iv) XML is designed to be self Descriptive.
- v) XML is a W3C Recommendation.

Q: Diff. Betw HTML And XML.

→ HTML

- i) W3C Recommendation (Standard)
- ii) predefined Tags.
- iii) To design to display data.
- iv) Every documents starts with <html> Tag
- v) No Case Sensitive.
- vi) We need to follow Tag Hierarchy.
- vii) Every open tag No need to close

XML

- i) W3C Recommendation (Standard)
- ii) User defined Tags.
- iii) Transport & store data.
- iv) Every documents starts with <ROOT> Tag.
- v) 100% Case sensitive.
- vi) We must follow Tag Hierarchy.
- vii) Every opened tag must be closed.

### \* XML Syntax Rules:

These Rules r simple and Logical. All XML elements must

have a closing Tag.

Ex: <P> This is a paragraph } <br> It is illegal in XML

<P> This is a paragraph</P> } This is a current format in XML  
<br/>

Ex: <Message> This is incorrect </message>  
<message> This is correct </message>

In XML, elements must be properly Nested.

In HTML, we can see improperly Nested Elements.

Ex: <b><i> This Text is Bold & Italic </b></i>

In XML, All Elements must be proper.

Ex: <b><i> This is Bold & Italic </i></b>

\* XML documents must have root Element. It is parent of all other Elements.

151

Ex: 

```
<root>
    <child>
        <subchild>...</subchild>
    </child>
</root>
```

\* XML Attribute values must be Quoted.

Ex: ~~```
<note date=12/11/2007>
    <to> Ravi </to>
    <from> Raghav </from>
</note>
```~~

Incorrect

09.OCT.2013

### \* Comments in XML:

These comments are similar to HTML. Comment Notation are

`<!-- This is a comment -->`

Font color: Black  
One paragraph  
One space  
Font size: 3  
One line  
Font Family: Times New Roman

} By default in HTML

\* whitespace is preserved in XML.

\* HTML truncates multiple whitespace characters to one single whitespace.

\* XML not truncated.

\* What is HTML DOM?

### HTML DOM

→ A std. object model for HTML, A std. programming interface for HTML & W3C std.  
HTML DOM defines the objects & properties of all HTML elements & the methods to access them. HTML DOM is a std. for how to get, change, add or delete HTML elements.

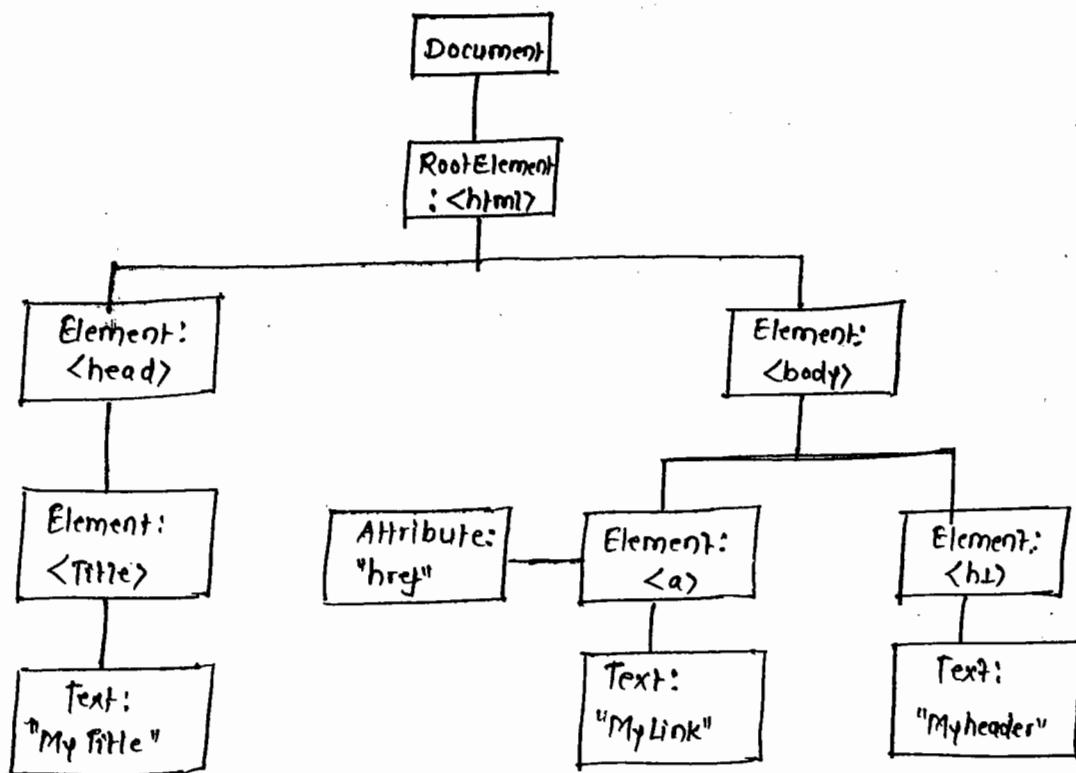
### DOM Nodes:

According to W3C, HTML DOM std, Everything in an HTML document is a Node.

- i) One Entire document is a document Node.
- ii) Every HTML element is an Element Node.
- iii) The Text inside HTML Elements are Text Nodes.
- iv) Every HTML attribute is an Attribute Node.
- v) Comments are Comment Nodes.

## \* HTML DOM Tree Structure

152



In the above DOM Tree structure, Top Node is called the Root. Every Node has exactly one parent, a Node can have any no. of children. Siblings are Nodes with the same parent.

Ex: <html>

```
<head>
  <title> DOM Tutorial </title>
</head>
<body>
  <h1> DOM Lesson One </h1>
  <p> Hello World! </p>
</body>
</html>
```

\* Recognize following pts:

- i) <html> Node has No parent (Root Node)
- ii) <html> Node has 2 Child Nodes (Head & Body)
- iii) <head> Node has 1 child Node (Title)
- iv) <title> has 1 child Node (Text Node)
- v) <h1> and <p> nodes are siblings.
- vi) <head> element is first child of html.
- vii) <body> element is Last child of html.
- viii) <h1> element is First child of Body.
- ix) <p> element is Last child of the Body.

\* html Dom properties:

Properties are values of Nodes that U can get or set.

1) innerHTML property:

It is useful for getting or replacing the contents of html elements.

Ex: <script>

```
function Max_VALUE()
{
  document.getElementById('demo').innerHTML = Number.MAX_VALUE;
}

function Min_VALUE()
{
  document.getElementById('demo1').innerHTML = Number.MIN_VALUE;
}
```

## HTML DOM Methods:

Methods & actions we can perform on Nodes. It contains

following methods.

- i) getElementById
- ii) appendChild
- iii) removeChild Etc.

Q: Write a Script By using getElementById method

→ <script>

<body>

<p id="intro"> Hello World! </p>

This Example demonstrates the <b>getElementById</b> Method.

</body>

</html>

Q: Write a Script To Disable a Button.

→ <html>

<script>

function disableElement()

{

document.getElementById("firstbtn").disabled = true;

}

</script>

<body>

<form> Buttons: <input type="button" id="firstbtn" value="Ok">

<input type="button" id="secondbtn" value="Ok">

</form>

<button on\_click="disableElement()"> Disable Button </button>

</body>

Output:

Buttons:

Disable Button

Q: Write a Script to get the value of a Button.

→ <body>

<input type="button" id="button1" value="Cancel!">

<p> Text on the Button is :

<Script>

document.write(document.getElementById("button1").value);

</Script>

</p>

</body>

# ★ JAVASCRIPT VALIDATIONS (JSvalidations) ★

154

Validation : On level  
as per User Requirement  
Verification : programmer  
doing at low level like  
debugging etc.

- ii) If a text field is all alphanumeric characters.
- iii) If a text has the current no. of characters.
- iv) If the selected item from drop down valid or not.
- v) Entered email id valid or not.
- vi) User entered fields length valid or Not.
- vii) Entered valid date and format etc.

Q: Write a script to check the text field empty or Not. If Empty display entered

⇒ <script>

```
function func()
{
    if (txt == " ")
    {
        alert ("Empty");
    }
    else
    {
        alert ("document.getElementById("txt").value");
    }
}
```

OR

```
<html>
<head>
<script>
    function notEmpty()
    {
        var myTextfield=document.getElementById("my text");
        if (myTextfield.value!="")
        {
            alert ("You Entered: "+myTextfield.value);
        }
        else
        {
            alert ("Would U please enter some text?");
        }
    }
</script>
</head>
```

O/P:  
Enter Any Text :  From checker

```

<body>
    Enter Any Text <input type="text" id="myText" />
    <input type="button" onclick="notEmpty()" value='form checker' />
</body>

```

~~</html>~~

~~Q~~ Write a Script to validate Textfields, When User Entered display Silver border,  
When User moving without Entering display Red Border.

→ <head>

<title>Validating HTML forms using JS </title>

<script>

```

function funremove(cid)
{
    if(document.getElementById(cid).value == "Enter Text")
        document.getElementById(cid).value = ""
}

```

```
function funchklen(len, cid)
```

```
{
    if(len == 0)
```

```
    document.getElementById(cid).value = "Enter Text"
```

```
    document.getElementById(cid).style.borderColor = "red"
}
```

```
else
```

```
{document.getElementById(cid).style.borderColor = "silver"
}
```

</script>

<head>

<body>

<form name="UserValidation">

Username: <input type="text" id="txt1" onblur="funchklen(this.value.length,'txt1')"

value="Enter Text" onclick="funremove('txt1')"> <br>

Password: <input type="text" id="txt2" onblur="funchklen(this.value.length,'txt2')"

onclick="funremove('txt2')"> <br>

</form>

</body>

Username:

Password :

## Validating Text fields With a msg when User leaving the object.

like      Username:  please fill Field

          Password:

```
<head>
<title>
```

```
<script>
function funremove(cid,spanid)
{
    if(document.getElementById(cid).value == "Enter Text")
    {
        document.getElementById(cid).value = ""
    }
}
```

```
function funchklen(len,cid,spanid)
{
    if(len==0)
    {
        document.getElementById(cid).value = "Enter text"
        document.getElementById(cid).style.borderColor="red"
        document.getElementById(spanid).innerHTML="Please Fill Field"
    }
    else
    {
        document.getElementById(cid).style.borderColor="silver"
        document.getElementById(spanid).innerHTML=" "
    }
}
```

```
</script>
```

Username: <input type="text" id="txt1" onblur="funchklen(this.value.length,'txt2','sp1')"  
 value="Enter Text" onclick="funremove('txt1')"/> span id="sp1" > <br>

password: <input type="text" id="txt2" onblur="funchklen(this.value.length,'txt2')"  
 onclick="funremove('txt2')"/>

</form>

</body>

nareshTech@gmail.com → Top level domain

## \* Validating Email id:

Generally Electronic mail id having following 5 parts.

1) Combination of letters, numbers, periods, Hyphens, signs And Underscores

Ex: nareshi-Tech.info4u

2) '@' symbol

3) Combination of letters, numbers, Hyphens, periods etc. Ex: Way2sms,

4) period

5) Top level domain. *(User optional & Domain)*

Ex: nareshTech @ gmail.com → Top level domain

User  
Optional      Secondary  
domain      Seperator Betw  
(SCD)      Secondary f ~~for~~ Top  
level domain

googleexperiment.com

Email id:  Invalid Email id

Ex: <script>

function chkemail(val,sid)

{

exp=/^([a-zA-Z0-9\_.-])+@([a-zA-Z0-9\_-]+)\.([a-zA-Z]+)\.([a-zA-Z]{2})\$/;

if (!exp.test(val))

{

document.getElementById(sid).innerHTML = "Invalid Email id"

document.getElementById(sid).style.color = "Red" *gir=* nithmljavascrip~~t~~@gmail.com

}

else

document.getElementById(sid).innerHTML = "

}

Book: simplyjavascript

3rd Edition or 4th Edition

JSON ⇒ Javascript Object Notat<sup>ion</sup>

By John Resig

j<sup>query</sup>.com

Subject: Evening  
6-7

</script>

Email id:<input type="text" id="txtemail" value="example@domain.com" onblur="chkemail(this.value,'sp2')"><span id="sp2"></span>

Client side platform: html + CSS + JS + JQ + XML + JSON + HTML5 + CSS3 +

AJAX

$\partial_L^k$

~~Oops~~  
~~CONCEPTS~~



## OOPS

PHP 5.0 is supporting Object Oriented programming concept like Inheritance, Abstract class, Interfaces etc. Ooops in programming lang. to increase the performance of Application, security and Reuseability.

By using OOPS Concept we can develop the applicat<sup>n</sup> with specific design pattern like 3-Tier Architecture, N-Tier Architecture, MVC Architecture etc

In OOPS Concept, the primary Template is a class, used to maintain the Methods and variables.

### Class:

Class is a collect<sup>n</sup> of properties and methods. According to the OOPS Concept, Everything <sup>we</sup> should place within the class.

### Methods:

A Function declarat<sup>n</sup> within the class is called as Method. ~~function~~

### Property:

Variables within the class, we can call as property.

By Default, we cannot call the class member directly from the script.

If User want to call class member, first we need to create an object to that class

Without a class we write a prog. So that, C++ is not a Fully Object Oriented programming.

By Using ~~class~~, a 'class' keyword we can create the class.

By Using ~~new~~, 'new' keyword, we can ~~not~~ allocate the new memory location to load the class contents.

By default, Every class internally contains the default constructor.

In class name, this constructor is used to load the class content into newly created memory location.

### Field:

A variable within the class, we called as field.

### Object:

Object is an instance of a class, we can call the class properties with the help of an object.

## Programs How To Create an Object.

```
<?php
class clsl
{
    function func1()
    {
        echo "From class";
    }

    function func2()
    {
        echo "From ext fun";
    }

    func1();
    $obj=new clsl();
    $obj->func1();
}
```

### New:

By using this ~~is~~ keyword we can allocate new memory allocated.

### Constructor:

constructor is a special type of method which contains classname as methodname.

Every class internally contains one constructor. We can call it as default constructor.

This default constructor is used to load the class contents into the RAM memory location.

We can also create the constructor explicitly using classname or \_\_ (underscore underscore) keyword. We can call this constructor as explicit constructor.

Explicit Constructors we are using to initialize the variables, to open the DATABASE connections when class is loading.

102

Constructor with \_\_construct keyword is called as Explicit constructor.  
(By default it executed)

```
<?php
class clsl
{
    function __construct()
    {
        echo "From __cons";
    }

    function clsl()
    {
        echo "cons with class";
    }

    $obj=new clsl();
```

Destructor  $\Rightarrow$  Not used in web application

Destructor is also a special type of fn executes when the class object is destroying. We can explicitly create the destructor by using \_\_destruct keyword.

By default, in PHP when the script execution is completed, the destructor will execute otherwise, by using unset fn we can destroy the object of class at the middle of script Execut.

Destructor we are using to uninitialized variables, to close the db connect etc.

```
function __destruct()
function __destruct()
{
    echo "Dest is executed";
}
```

To declare the variables within the class we should use any access specifier or var datatype.

```
<?php
class clsinfo
{
    var $sno=100;
}

$obj=new clsinfo();
echo $sno;
```

## \* Inheritance:

It is a concept of accessing the features of one class from another class. If we inherit the class features into another class, we can access both class properties the object of class to which class we're inheriting the properties. We can extend the features of a class by using 'extends' keyword.

## \* Types of Inheritance:

### 1) Single Inheritance:

In this concept, we have only 2 classes. One is Base Class  
Another is Derived Class.

Ex: <?php

```
class cls1 {
    function func1()
    {
        echo "from class1";
    }
}
```

Class cls2 extends cls1

```
{.
    function func2()
    {
        echo "from class2";
    }
}
```

\$obj = new cls2();

\$obj->func1();

}

X PHP Not supported

### 2) Multiple Inheritance:

In this concept, A child class can have 'n' number of parent classes. PHP doesn't support this concept. (Bcoz, of Ambiguity)

### 3) Multilevel Inheritance:

In this concept, a class behaves as both Base class & Derived class.

### 4) Hierarchical Inheritance:

In this concept, A Base class can have no. of derived classes

Doesn't support  
by PHP

## X) Hybrid Inheritance:

It is a combination of Single, Multiple, Multilevel &

Hierarchical Inheritances. PHP doesn't support this concept.

164

## \* Some Helper Functions in PHP to get the Information About Class And Object.

### 1) get\_class:

By using this, we can get the classname of an object.

Ex: <?php

```
class cls1 {  
}  
  
$obj = new cls1();  
echo get_class($obj);  
?>
```

### 2) get\_class-vars:

To get the all variables of a class as Array Elements.

Ex: <?php

```
class cls1 {  
    var $x=100;  
    var $y=200;  
}
```

```
print_r(get_class_vars("cls1"));  
?>
```

### 3) get\_class-methods:

To get the all methods of a class as an Array.

Ex: <?php

```
class cls1 {  
    function fun1() {}  
    function fun2() {}  
}  
  
print_r(get_class_methods("cls1"));  
?>
```

### 4) get\_declare\_classes:

To get the all declare classes in current script along with predefined classes.

Ex: <?php

```
class cls1 {  
}
```

```
print_r(get_declared_classes());  
?>
```

### 5) get\_object-vars:

To get the all variables of an object as an Array.

Ex: <?php

```
class cls1 {  
    var $x=100;  
    var $y=200;  
}  
  
$obj = new cls1();  
print_r(get_object_vars($obj));  
?>
```

### 6) isclass::exists:

To check whether the specified class exists or Not.

<?php

```
class cls1 {  
}
```

```
echo isclass_exists("cls1");  
?>
```

## 7) is\_subclass\_of :

By using this f<sup>n</sup>, we can check whether the 1<sup>st</sup> class is subclass of 2<sup>nd</sup> class or not.

Ex: <?php

```
class cls1 {
}
class cls2 extends cls1 {
}
echo is_subclass_of("cls2", "cls1");
?>
```

## 8) method\_exists:

By using this f<sup>n</sup>, we

can check whether the \$fp method is existed or not.

Ex: <?php

```
class cls1 {
    function fun1() {
    }
}
echo method_exists("cls1", "fun1");
?>
```

## 9) interface\_exist:

To check the \$fp interface is existed or Not. (By using this f<sup>n</sup>,

~~<?php~~ We can check whether the class \$cls specified interface is existed in current script or Not)

## 10) property\_exists:

To check whether the \$fp

properties are Existed or Not.

Ex: <?php

```
class cls1 {
}
echo property_exists("cls1", "x");
?>
```

## \* this:

By using ~~this~~, we can call the properties methods of a class with that class methods.

Ex: <?php

```
error_reporting(E_ALL);
class cls1 {
    var $x = 100;
    function func() {
        $x = 1234;
        echo $x;
        echo $this->x;
    }
}
$obj = new cls1();
$obj->func();
?>
```

## \* Scope Resolution Operator (::)

By using this operator, we can access the class properties with ~~out~~ the help of ~~an~~ an object. Sometimes, we need to call

parent:: the class properties without help of ~~an~~ object. That time we can go for scope Resol~~ope~~ operator

By using this keyword, we can access the properties of parent class from derived class.

Ex: <?php

```
class cls1 {
    function fun1() {
        echo "From Parent Class";
    }
}
class cls2 extends cls1 {
    function fun2() {
        echo "From Derived Class";
    }
    function fun3() {
        $this->fun1();
        parent::fun1();
    }
}
```

```

$obj = new cls1();
$obj->fun1();
?

```

### static:

It is a keyword we can directly access, the static class properties with the help of class name. (Without an Object)

Ex: <?php

```

class cls1
{
    static function fun1()
    {
        echo "hi";
    }
}
cls1::fun1();
?

```

### Access Specifiers:

By using access specifiers

We can specify the scope of accessibility to the class contents.

By default, We have 3 types of access

Specifiers in php. Those are

- 1) public
- 2) private
- 3) protected.

#### 1) public :

Allows the accessibility from parent class, derived class, and outside the class.

#### 2) protected:

We can access from Base class & derived class.

#### 3) private:

These property we can access within the class only where we declared. We cannot access from derived class & outside the class.

08.OCT.2012

### return:

By using this keyword, we can return a value from a F?

Ex:

```

<?php
class cls1
{
    function fun1()
    {
        echo "From method";
        return 100;
    }
}
$obj = new cls1();
$x = $obj->fun1();
echo $x;
?

```

|           | Within class | Derived class | Outside class |
|-----------|--------------|---------------|---------------|
| public    | ✓            | ✓             | ✓             |
| protected | ✓            | ✓             | ✗             |
| private   | ✓            | ✗             | ✗             |

Ex: <?php

```

class clsBase
{

```

```

    public $pub = "public var";
    protected $pro = "protected var";
    private $pri = "private var";
    function funBase()
    {
        echo $this->pub;
        echo $this->pro;
        echo $this->pri;
    }
}

```

```

class clsDer extends clsBase
{
    function funDer()
    {
        echo $this->pub;
        echo $this->pro;
        echo $this->pri;
    }
}

$obj = new clsBase();
// $obj->funBase();
// echo $obj->pub;

// $obj1 = new clsDer();
// $obj1->funDer();
// $obj1->funBase();
}

```

167

```

function fun2()
{
    $obj = new cls();
    $obj->fun1();
    $obj->funx();
}

```

### \* Interfaces:

Interfaces r same as abstract classes But interfaces doesn't allow method defn. Interfaces can't maintain Non-abstract methods. By using 'interface' keyword, we can create the interfaces in Application.

09. Oct. 2018

### \* Differences Betw Abstract classes & Interface → Abstract class      Interface

- i) Abstract class comes under partial Abstract?
- ii) Abstract classes can maintain abstract methods & Non abstract Methods.
- iii) In Abstract classes, We can't create the Variables
- iv) In Abstract classes, We can use any access specifier.
- v) By using 'extends' keyword we can access the abstract class features from derived class.
- vi) By using 'implements' keyword we can get interface from derived class.
- vii) By using interfaces multiple inheritance is not possible.

### \* Abstract Classes:

By using Abstract

classes, we can create the Abstract methods as well as Non Abstract Methods within the class.

Abstract classes r using to provide a structure to the class contents.

Abstract methods are method

declarat?. They don't have

class implementat?.

In Abstract class, can't be instantiate means we can't create object for abstract class. By using 'abstract' keyword, we can

get the abstract classes.

### (\* PHP doesn't support Overloading)

```

<?php
abstract class clsabs
{
    abstract function fun1();
    abstract function fun2();
    function funx()
    {
        echo 'From concrete Method!';
    }
}

class clsB extends clsabs
{
    function fun1()
    {
        echo 'Implementation from der class';
    }
}

```

```

Ex: <?php
interface iface1
{
    public function fun1();
}

interface iface2
{
    public function fun2();
}

class clsl implements iface1,iface2
{
    function fun1()
    {
        echo "Implementation from derived class";
    }

    function fun2()
    {
        $obj = new clsl();
        $obj->fun1();
    }
}

```

```

echo clsl::$x;
$obj = new clsl();
echo $obj->y;
clsl::$x=500;
echo clsl::$x;
$obj1 = new clsl();
echo $obj1->y;
$obj1->y=444;
echo $obj->y;
?

```

### \* MVC Architecture:

The main aim of MVC Architecture is separate the Business logic & Application data from User interface.

MVC stands for "Model View And Controller". Different types of architectures are available those r 3-tier architecture, n-tier architecture, mvc architecture etc.

The main Advantage of Architecture is Reuseability, security & increasing the performance of Application.

### \* Class Variable:

If we declare any variable within the class, without static keyword comes under static variable.

### \* Instance Variable:

Variable Declaration without static keyword comes under instance variable.

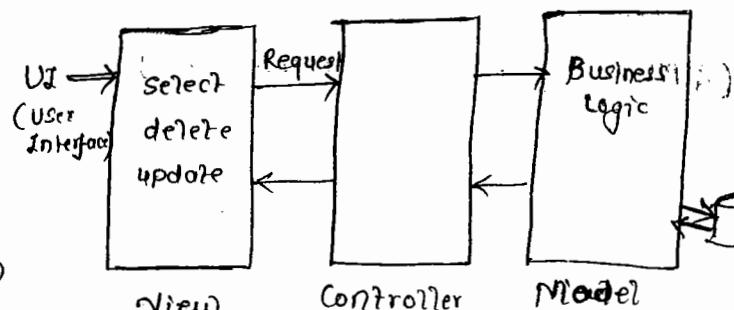
instance variable we can create number of copies by creating no. of objects. But static variable we can't create no. of objects.

Ex: <?php

```

class clsl {
    static public $x=100;
    var $y=200;
}

```



### Model:

All the Model object knows all abt data that need to be displayed. The Model represents application data & business rules that govern to an update of data. Model is not aware abt presentation data & how the data will be display to the browser.

## 2) View:

View represents the presentation of application. View object refers to the model remains same if there is any modification in the Business Logic.

In other words, we can say that it is the responsibility of view to maintain consistency in its presentation if the model changes.

## 3) Controller:

Whenever, User sends a request for something, it always goes through controller. A controller is responsible for intercepting the request from view & passes to the model for appropriate action. After the action has been taken on the data, the controller is responsible for directly the appropriate view to the user. In Graphical User Interfaces, controller & view works very closely together.

### Step 1:

15. Oct. 2019

### SAVE AS: controller.php

```
<?php
include "model.php";
if (isset($_REQUEST['mt']))
{
    $tit = $_REQUEST['mt'];
    $arr = clsmodel::getinfo($tit);
    include "view2.php";
}
else
{
    $titles = clsmodel::gettites();
    include "view.php";
}
```

### Step 2: SAVE AS: view.php

```
<table border='1'>
<?php
for ($i=0 ; $i < count($titles) ; $i++)
{
    echo "<tr><td><a href='controller.php?mt=$titles[$i]'">$titles[$i]</td>";
}
```

### Step 3: SAVE AS: model.php

```
<?php
include "conn.php";
class clsmodel
{
    static function gettitles()
    {
        __connect();
        $data = mysql_query("select mttitle
from hbl-movies");
        $i = 0;
        $arr = array();
        while ($rec = mysql_fetch_row($data))
        {
            $arr[$i] = $rec[0];
            $i++;
        }
        return $arr;
    }
    static function getinfo($t)
    {
        __connect();
        $data = mysql_query("select * from
hbl-movies where mttitle = '$t'");
        $rec = mysql_fetch_row($data);
        return $rec;
    }
}
```

### Step 4: SAVE AS: conn.php

```
<?php
function __connect()
{
    mysql_connect("localhost", "root", "");
    mysql_select_db("test");
}
```

Step 5: Save As: View2.php

```
<?php
$str=<<<abc>
<h2 style='color: red'>Titles: $rv[0]
<h2 style='color: green'> Director: $rv[1]
<h2 style='color: blue'> Casting: $rv[2]
<h2 style='color: black'> Report: $rv[3]
abc;
echo $str;
?>
```

★

★ DOM Document:

By using this class we can create XML file. Also we can read the elements of XML file. XML is used to "Ferr the Data Bet" the Heterogeneous platforms. It stands for "Extensible Markup Language".

★ Difference Bet" the HTML And XML:HTML

- i) HTML is used for User Interface      ii) XML is used for X'Ferring the data.
- Design.
- 3) HTML is ~~alwaus~~ a Case Insensitive.      iv) XML is a Case Sensitive.
- 5) HTML doesn't required <RootElement>      vi) XML requires <RootElement>.
- 4) No need to close opened <Html> Tags.      vii) Every Tag U must closed.
- 5) HTML contains predefined Tags.      viii) XML contains User defined Tags.

RootElement :

Ex: <RootElement>  
           <child> XYZ </child>  
           <child>  
             <child>  
               ABC  
             </child>  
           </child>  
   </RootElement>

## Functions

16.OCT.2018 (17)

Functions available in Dom document class.

### Create Element():

By using this f<sup>n</sup>, we

Can create new Element.

### append\_child():

To append an Element as child element.

### Create Text Node

To create the Text Content to an Element.

### GET ELEMENTS BY TAGNAME():

Get the Elements to specify the Tag.

### ModeValue():

By using this property We can get the Content of an Element.

### load():

To load the XML document to the current applicat<sup>n</sup>.

### Savexml():

To save the XML element.

### Ex:

```

$doc=new domdocument();
$root=$doc->createelement("myfile");
$doc->appendchild($root);
$sn=$doc->createelement("sno");
$root->appendchild($sn);
$txt=$doc->createtextnode("100");
$sn->appendchild($txt);
$add=$doc->createElement("Address");
$root->appendchild($add);
$city=$doc->createElement("city");
$add->appendchild($city);
$txt=$doc->createtextnode("Hyd");
$city->appendchild($txt);
$cont=$doc->savexml();
file_put_contents("$tname.xml",$cont)
echo "File is Created";
}

```

```

echo "XML file is Created";
}

```

### Ex:

```

<?php
if(isset($_POST['sub']))
{
$tname=$_POST['txtab'];
mysql_connect("localhost","root","");
mysql_select_db("test");
$data=mysql_query("select * from $tname");
$doc=new domdocument();
$root=$doc->createelement($tname),
$doc->appendchild($root),
$fc=mysql_num_fields($data),
while($rec=mysql_fetch_row($data))
{
$row=$doc->createelement("Row"),
$root->appendchild ($row),
for($i=0; $i<$fc; $i++)
{
$fname=mysql_field_name($data,$i),
$child=$doc->createelement($fname),
$row->appendchild ($child),
$txt=$doc->createtextnode($rec[$i]),
$child->appendchild ($txt),
}
}
$cont=$doc->savexml(),
file_put_contents("$tname.xml",$cont)
echo "File is Created";
}

```

```

<Form method="post" action=" ">
Table Name:<input type="text" name="txtab">
<br>
<input type="submit" value="Click" name="sub">
</Form>

```

Ex:  
1.php

```
$doc = new domdocument();
$doc->load("tbl-parent.xml");
$rows = $doc->getelementsbytagname("Row");
foreach ($rows as $row)
{
    $pid = $row->getelementsbytagname("pid");
    $txt = $pid->item(0)->nodevalue;
    echo $txt;
    $pname = $row->getelementsbytagname("pname");
    $txt2 = $pname->item(0)->nodevalue;
    echo $txt2;
    echo "<br>";
}
}
```

## \* Exception Handling ⇒ 17.Oct.2013

Exception is a base class in Object Oriented programming library used to handle the runtime errors.

Errors are mainly divided into 3 types.

- 1) Compilation Error
- 2) Runtime Error (Exception)
- 3) Logical Error.

We can rectify compilation Error &

Logical Error at the time of application implementation  
But Runtime Errors we cannot Rectify.

Runtime

Errors will occur based on the input values given by the user at the time of application execution.  
To get Runtime Error, program execution will stop without any information.

To prevent the application execution from Undetermined, Terminated, we're handling Exception.

Exception class is providing 2 constructors

- 1) A constructor with Single Argument
- 2) A constructor with double Argument (two)

exception

172

## \* Methods of Exception class

### 1) getMessage():

By using this method, we can get the error msg.

### 2) getLine():

To get the Error line number.

### 3) getCode():

To get the Error code.

### 4) getFile():

To get the filename where the Except? is occurred.

## \* try():

By using try block we can maintain the executable statements.

## \* Catch():

If there is any Except? occurred in the try block, the stat of catch block will execute.

## throw:

173

By using this keyword, we can

Throw the Error Msg into the Except<sup>n</sup> class.

Ex:

```

<?php
if (isset($_POST['sub']))
{
    $fname = $_POST['txtfile'];
    try
    {
        $fp = fopen($fname, "r");
        if (!$fp)
        {
            throw new Exception("file is not Available");
        }
        else
            echo "file is Opened";
    }
    catch (Exception $e)
    {
        $str = "Error occurred (i.e) ". $e->getMessage() . " in line no " . $e->getLine() . " in prog";
        $e->getFile();
        echo "<span style='color:red'> $str </span>";
    }
    try
    {
        $db = mysql_connect("localhost", "root", " ");
        if (!$db)
        {
            throw new Exception(mysql_error());
        }
        else
            echo "DB is connected";
    }
    catch (Exception $e)
    {
        $str = $e->getMessage();
        echo "<span style='color:red'> $str </span>";
    }
}
<form method="post" action="">
    filename:<input name="txtfile"> <br> <input type="submit" name="sub" value="Get" >

```

Ex: <?php

```

include "myexception.php";
try {
    $con = mysql_connect("localhost", "root", ""),
    if (!$con)
        throw new myexception("1001"),
    else
        echo "Connected";
}
catch (myexception $e)
{
    echo $e->getmymessage();
}

```

save: myexception.php

```

<?php
class myexception extends Exception
{
    function myexception ($ecode)
    {
        $this->ec = $ecode;
    }
    function getmymessage ()
    {
        $fp = fopen ("error.txt", "r");
        while ($line = fgets ($fp))
        {
            list ($em, $ec) = explode (":", $line),
            $ec = trim ($ec);
            if ($this->ec == $ec)
                return $em;
        }
    }
}

```

save: error.php

Database not connected : 1001

File is not Opened : 1002

FTP Error : 1003

### \* Object copy :

It is a concept of copying one object into another object means we can refer a single address location with multiple reference variables.

```

<?php
class clsinfo
{
    var $sno=100;
}
$obj=new clsinfo();
echo $obj->sno;
$obj1=$obj;
echo $obj1->sno;
$obj1->x=500;
echo $obj->x;
?

```

### \* Object cloning :

It is a concept of creating a duplicate object with the existing object. By using clone() fn, we can create the duplicate object of an existing object.

```

<?php
class clsinfo
{
    var $sno=100;
}
$obj=new clsinfo();
echo $obj->sno;
$obj1=clone ($obj);

```

18 OCT 2012

AJAX

AJAX

13C

AJAX

## AJAX

18. Oct. 2024 177

- i) AJAX Stands for "Asynchronous Javascript And XML".
- ii) AJAX We r using to ~~sub~~ send the Request to the server without Submitting the Webpage.
- iii) By Using AJAX we can increase the performance of Application instead of Submitting the Entire webpage to the server. We have sending the requested data decreases the N/W Traffic.
- iv) without submitting the Webpage, we can Execute server side script ~~immediately~~ that's Why User can see the Webpage continuously & also he can perform ~~one~~ other task mean while.
- v) Instead of submitting the Webpage, we can sending 2nd. data & decreases the Bandwidth consumption of Webserver.

### \* Asynchronous:

- By using Asynchronous process, We can send multiple Request to the server with irrespective of previous Responses.

### \* JavaScript:

- By Using JavaScript, Document Object Model (DOM), We can create AJAX Object And we can send that object to the server.

### \* XML:

- Data X'fers Bet<sup>n</sup> Browser And Server in the form of XML.

### \* STEPS TO CREATE AJAX OBJECT AND SEND THAT OBJECT TO THE SERVER

- i) Find out the Browser Type and create object to that Browser class.
- ii) By using "navigator.appname" we can find the Browser Type If The Browser is Internet Explorer we can create object for ActiveX object class otherwise, we can create object for XML http request class.
- iii) Load the object with some values like method(get/post), Serverside Script name Synchronized / Asynchronized process.
- iv) Send that object to the server to execute Server-side script.

### \* Object Methods

#### 1) Open():

- By using this method, We can open the object by specifying Serverside scriptname, method, Synchronized / Asynchronized process.

#### 2) Send():

- Send the object to the server.

#### 3) Abort():

### 3) abort():

To stop the Execut<sup>n</sup> of AJAX object.

19.Oct.2012

(178)

### \* Object Properties:

#### 1) Ready state:

By using this property we can get the state of AJAX object. It contains 5 different states.

- i) If state is "0", object initialized.
- ii) If state is "1", Object Loading.
- iii) If state is "2", ~~Wanted~~ Object loaded.
- iv) If state is "3" Interactive.
- v) If state is "4" completed successfully.

#### 2) Response state:

By using this property, we can get the Text content what is available in AJAX object.

#### 3) onreadystatechange:

This Event

will fire for every state change of AJAX object.

#### Ex: Prog. To create An AJAX Object.

```
<script>
function fun1()
{
if(navigator.appName == "Microsoft Internet Explorer")
obj = new ActiveXObject("Msxml2.XMLHTTP");
else
obj = new XMLHttpRequest();
alert(obj)
}
</script>
<input type="button" value="Create" onclick="fun1()">
```

SAVE AS: CreateAjaxObject.htm

O/P:

Create

A [Object]

Object XMLHttpRequest

```
<script>
function fun1()
```

```
{ if(navigator.appName == "Microsoft
Internet Explorer")
obj = new ActiveXObject("Msxml2.XML
HTTP")}
```

```
else
```

```
obj = new XMLHttpRequest();
obj.open('get', 'Hello.php', true)
obj.onreadystatechange = fun2
obj.send()
```

```
}
```

```
function fun2()
```

```
{
```

```
if(obj.readyState == 4)
```

```
{
```

```
alert(obj.responseText)
```

```
}
```

```
}
```

```
</script>
```

```
<input type="button" value="Click"
onclick="fun2()">
```

SAVE AS: Ajax1.htm

O/P:

Click

The screenshot shows a browser window with a single alert dialog box. The dialog box has a title bar with the word 'Alert'. Inside the dialog box, the text 'Hello' is displayed. In the bottom right corner of the dialog box, there is an 'OK' button.

```

<script>
    Function fun1(pname)
    {
        if(navigator.appName=="Microsoft Internet Explorer")
        Obj=new ActiveXObject("Msxml2.XML HTTP")
        else
        Obj=new XMLHttpRequest();
        Obj.open("get", "pname", true)
        Obj.onreadystatechange=fun2
        Obj.send()
    }
    function fun2()
    {
        if(Obj.readyState==4)
        {
            alert(Obj.responseText)
        }
    }
</script>
<input type="button" value="page1" onclick="fun1('page1.php')">
<input type="button" value="page2" onclick="fun1('page2.php')">

```

### \* Program of changing the Cricket Score

Ex:

Within 2seconds.

PART: I  
SAVE AS: CricketScoreRead.htm

```

<script>
setInterval("fun1()", 2000)           ↑2sec
function fun1()
{
    if(window.ActiveXObject)
    {
        Obj=new ActiveXObject("Msxml2.XML HTTP"),
    }
    else
        Obj=new XMLHttpRequest();
        Obj.open("post", "C:\wamp\www\CricketScoreRead.php", true);
        Obj.send();
        Obj.onreadystatechange=function()
        {
            if(Obj.readyState==4)
            {
                document.getElementById('div1').innerHTML=Obj.responseText
            }
        }
    }
</script>
<body onload="fun1()">
<div id="div1" style="font-size: 30"></div>
</body>

```

**PART-II** **SAVE AS:** CricketScoreRead.php

(180)

```

<?php
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data = mysql_query ("select * from tbl-score");
$rec = mysql_fetch_row ($data);
$str <<<abc
<h2 style='color: red'> Runs : $rec[0]<br>
<h2 style='color: green'> Wickets : $rec[1]<br>
<h2 style='color: blue'> Overs : $rec[2]
abc;
echo $str;
?

```

Note: In this prog, we also create a database of name

tbl-score with 4 columns i.e. id, Runs, Overs, Wickets.

Ex:

The important thing is, if we change in database like  
Runs, Wickets or Over, it automatically changed in

Program of Checking Username is Available OR Not.

Browser.

get.php

```

<?php
$qs = $_SERVER['QUERY_STRING'];
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data = mysql_query ("select * from tbl-user where uname='$qs'");
echo $data;
mysql_num_rows($data);
?

```

ajax.html

```

<script>
function fun1(val)
{
if(val.length!=0)
{
document.getElementById('sp1').innerHTML = "<img src=loading.gif width=20>";
obj = new XMLHttpRequest();
obj.open("POST", "get.php?" + val, true);
obj.send();
obj.onreadystatechange=fun2
}
}

function fun2()
{
if(obj.readyState==4)
{
if(obj.responseText==1)
{
document.getElementById('sp1').style.color="red"
}
}
}

```

```

        document.getElementById('sp1').innerHTML = "Username is Not Available" (81)
    }
}
else
{
    document.getElementById('sp1').style.color = "green"
    document.getElementById('sp1').innerHTML = "Username is Available"
}
}
}

</script>

```

Username : <input id='txtuname' onblur='fun1(this.value)'>

<Span id="sp1"></span>

\* A program to create a dropdown list of Countries with states, with the help of Mysql connectivity.

⇒

```

<Script>
function fun1(val)
{
    obj=new XMLHttpRequest();
    obj.open("POST", "getstat.php?" + val, true)
    obj.send()
    obj.onreadystatechange=function()
    {
        if(obj.readyState==4)
        {
            document.getElementById('sp1').innerHTML=obj.responseText
        }
    }
}

```

</Script>

Countries : <select id="drpcountry" onchange="fun1(this.value)">

<?php

mysql\_connect("localhost", "root", "");

mysql\_select\_db("test");

\$data = mysql\_query ("Select \* from tb1\_country");

echo

"<option> ---Select--- </option>";

while (\$rec=mysql\_fetch\_row(\$data))

{ echo "<option value=\$rec[0]>\$rec[1]</option>";}

}

</select>

<Span id="sp1"></span>

```

getstr.php
<?php
$qs = $_SERVER['QUERY_STRING'];
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$data = mysql_query("select sname from db_state where cid=$qs");
echo "States: <Select>";
while ($rec = mysql_fetch_row($data))
{
    echo "<option>$rec[0]</option>";
}
echo "</Select>";
?>

```

## Program: How to Access Gmail Account Username with DB 22.OCT.2012

```

<script>
    function funover(t)
    {
        t.style.backgroundColor = "lightblue"
    }
    function funout(t)
    {
        t.style.backgroundColor = "white"
        t.style.color = "black"
    }
</script>
<style>
    li
    {
        list-style: none;
        padding-top: 10px;
        padding-left: 0px;
        font-size: 25px;
        cursor: pointer;
    }
    #div1
    {
        border: 1px solid silver;
        padding-left: 0px;
    }
</style>
<script>
    function funinsert(un)
    {
        document.getElementById('txt1').value = un
        document.getElementById('div1').style.visibility = "hidden"
    }
    function fun2()
    {
        if (obj.readyState == 4)
        {
            document.getElementById('div1').innerHTML =
                obj.responseText
        }
    }
</script>
<table cellpadding="0" cellspacing="0">
    <tr><td> Username </td>
    <td> <input type="text" id="txt1" style="font-size: 30px" onkeyup="fun1(this.value)">
        <tr><td><td><div id="div1">Hi</div></td></td></tr>
</table>

```

```

function fun1(val)
{
    if (val.length != 0)
        document.getElementById('div1').style.visibility =
            "visible"
    else
        document.getElementById('div1').style.visibility =
            "hidden"
    if (navigator.appName == "Microsoft Internet Explorer")
        obj = new ActiveXObject("Msxml2.XMLHTTP");
    else
        obj = new XMLHttpRequest();
    obj.open("post", "get.php?" + val, true);
    obj.send();
    obj.onreadystatechange = fun2
}

```

```

function fun2()
{
    if (obj.readyState == 4)
    {
        document.getElementById('div1').innerHTML =
            obj.responseText
    }
}

```

```

<table cellpadding="0" cellspacing="0">
    <tr><td><td><div id="div1">Hi</div></td></td></tr>
</table>

```

```

get.php
<?php
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$qS = $_SERVER['QUERY_STRING'];
$data = mysql_query("Select * from emp where ename like '$qS%'");
while ($rec = mysql_fetch_row($data)) {
    echo "<i onclick='funinsert('$rec[1]')"
    . "onmouseover='funover(this)'"
    . "onmouseout='funout(this)'" . $rec[1] . "</i>";
}
</pre>

* <script>
    function fundel(id)
    {
        obj1 = new XMLHttpRequest()
        obj1.open("post", "del.php?" + id, true)
        obj1.send(null)
        obj1.onreadystatechange = function()
        {
            if (obj1.readyState == 4)
                getrec();
        }
    }

    function getrec()
    {
        obj = new XMLHttpRequest()
        obj.open("post", "get.php", true)
        obj.send()
        obj.onreadystatechange = function()
        {
            if (obj.readyState == 4)
            {
                document.getElementById('div1').innerHTML = obj.responseText
            }
        }
    }
</script>
<body onload="getrec()">
<div id="div1">
</div>
</body>

```

```

get.php
<?php
mysql_connect("localhost", "root", " ");
mysql_select_db("test");

while ($rec=mysql_fetch_row($data))
{
    echo "<tr><td> $rec[0] </td> $rec[1] <td> $rec[2] <td><img src='b_drop.png'
        onclick=fundel($rec[0])>";
}
?>
</table>

```

```

del.php
<?php
$qs=$_SERVER['QUERY_STRING'];
mysql_connect("localhost", "root", " ");
mysql_select_db("test");
mysql_query("delete from emp where empid=$qs");
?>

```

## \* Chat Application :

25.OCT.2012

```

<script src="jquery-1.3.js"></script>
<script>
    function funimage(t)
    {
        str= $('#spbig').html(str)
    }
    function funenter()
    {
        uname=$('#txtuser').val() document.getElementById('divName').style.visibility="hidden"
        document.getElementById('tab1').style.visibility="visible";
    }
    setInterval('read()', 2000)
    function read()
    {
        $.post("read.php", {}, function(data)
        {
            $('#div1').html(data)
        })
    }
</script>

```

```
<script>
function funSend()
{
    imgstr=$("#spbig").html();
    tvol=$("#textmsg").val();
    tvol=<b>'+uname+</b>
    <i> says:</i>" +tvol;
    tvol=tvol+imgstr+" "+$("#spbig").html();
    $.post("write.php", {qs:tvol}, function()
    {
        $("#textmsg").val(" ");
    });
}
```

<script>

|  |                                                                                                                                      |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------|--|
|  | <br><span style="font-size: 2em;">(neutral)</span>  |  |
|  | <br><span style="font-size: 1.5em;">(smile)</span>  |  |
|  | <br><span style="font-size: 1.5em;">(tongue)</span> |  |

<br>

<span id="spbjq">

<Ispean>

```
<tr><td><input type="text" id="text">
          <td><input type="text" id="text">
```

</div>

1313

Table

```
Username: <input id='txtuser'> <br> <input type="button" value="Enter" onclick="funente()  
</div>
```

body>

11

read.php

```
<?php
```

echo

三

Kritik

19

38

卷之三

३२१

**str =**

FP = F

2x2.

Write  
from

