5... More with Forms

Objectives...

- To learn how check box and radio button works.
- To cover how listboxes are used with php and operations on it.
- To study how to validate our data and restrict it.
- To learn how to send emails in php.

5.1 INTRODUCTION

- PHP easily supports form elements and any form elements is easily available in PHP script. To get data through form we need $_GET and $_POST super global variables.
- In this chapter we will see how form processing takes place with checkbox, radio buttons and select. You will be able to learn how to collect input from a web form, how to validate form data. Later we will discuss about how to send emails through php.

5.2 DEALING WITH CHECKBOXES AND RADIOBUTTONS

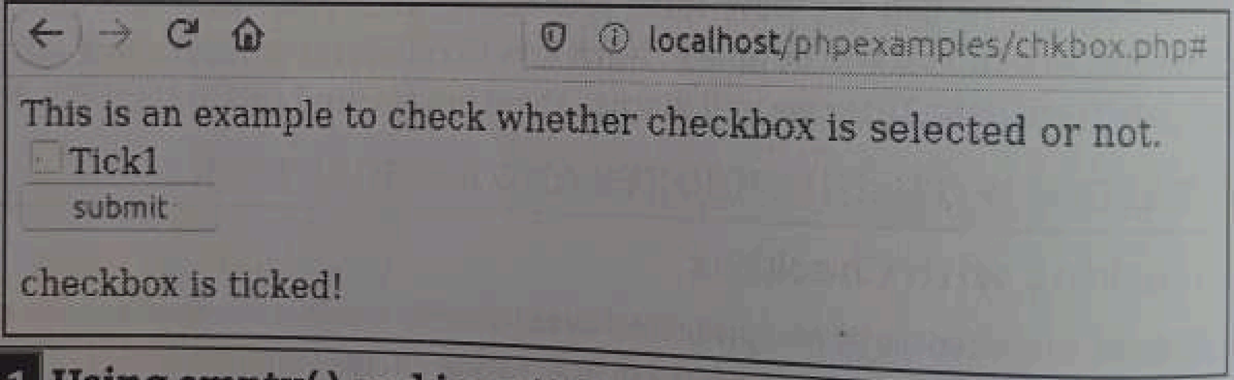
5.2.1 Working with Checkbox

- Checkboxes are used to give options to user. User can select more than one option with checkboxes. Radio buttons restricts user to one choice amongst others.
- There are two types of checkboxes based on the number of options you require.
  - Checkbox
  - Checkbox group
- Clicking on checkbox will change it to ON to OFF or vice versa. When you want to make a single choice, radio buttons or select option is better.
- We can take checkboxes using form tag of html using input element.
- For example: `<input type="checkbox" name="gender" value="male">Male`
- Attribute type name will be common for all checkboxes.
- We will have a look at example how checkbox element is checked or not using $_POST[].

Program 5.1: Program to demonstrate use of checkbox.

```
<html>
<form method="post" action="#">
This is an example to check whether checkbox is selected or not. <br>
<input type="checkbox" name="check1" value="test">Tick1<br>
<input type="submit" value="submit">
</form>
</html>
<?php
$selected=false;
if ($_POST["check1"]) {
$selected=true;
}
if($selected){
echo "checkbox is ticked!";
}
?>
```

Output:



```
This is an example to check whether checkbox is selected or not.
checkbox is ticked!
```

5.2.1.1 Using empty() and isset() Function with Checkbox

1. **empty() function:**

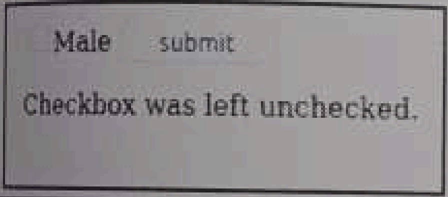
- It is used to check whether the variable is empty or not.
- The PHP empty( ) function works on both existent and non-existent variables. Thus, no warning or error message is spawned even if the variable doesn't exist.
- If the variable is non-existent, or if the variable exists but is empty, then empty( ) returns true. Otherwise, if the variable exists and it contains data, empty( ) returns false.

Program 5.2: Program using checkbox with empty() function.

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

```
<form action="#" method="post">
<input type="checkbox" name="gender" value="male">Male
<input type="submit" value="submit">
</form>
</body>
</html>
<?php
if( empty($_POST["gender"]) )
{ echo "Checkbox was left unchecked."; }
else { echo "Checkbox was checked."; }
?>
```

Output:



```
Male   submit

Checkbox was left unchecked.
```

- The above program test for checkbox whether gender field is checked or not. If it is checked if condition returns true otherwise returns false.

2. **isset( ) function:**

- The isset ( ) function is used to check whether a variable is set or not. If a variable is already unset with unset( ) function, it will no longer be set.
- The isset( ) function return false if testing variable contains a NULL value.

Syntax:

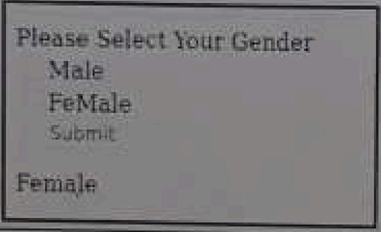
```
isset( value1,value2)
it returns a boolean value true if the variable is not null, otherwise
false
```

Program 5.3: program to demonstrate use of isset( ) function

```
<html>
<body>
<form action="#" method="post">
Please Select Your Gender<br>
<input type="checkbox" name="gender1" value="Male">Male</input></br>
<input type="checkbox" name="gender2" value="Female">FeMale</input></br>
<input type="submit" name="submit" value="Submit"/>
</form>
```

```
</body>
</html>
<?php
if (isset($_POST['gender1'])){
echo $_POST['gender1'];
}
if (isset($_POST['gender2'])){
echo $_POST['gender2'];
}
?>
```

Output:



```
Please Select Your Gender
   Male
   FeMale
   Submit

Female
```

- In above code isset( ) function will test for checkbox gender1 and gender2 if it is set appropriate value will be echoed.

5.2.1.2 Working with Multiple Checkboxes

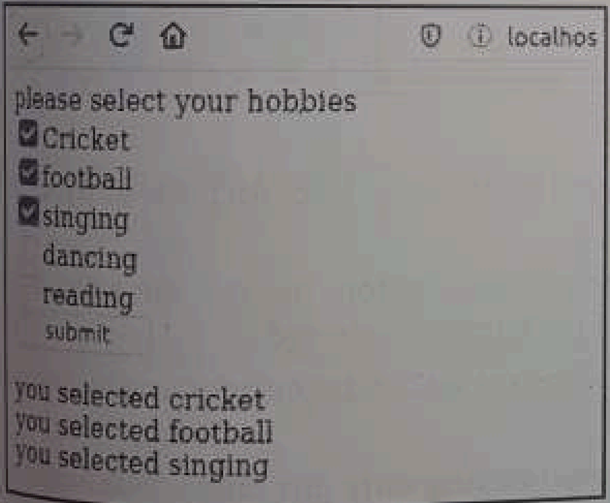
- There are often situations where a group of related checkboxes are needed on a form. The advantage of check box group is that the user can select more than one options.
- To get value of multiple check checkboxes, attribute name ="checkbox" tag must be initialize with an array.

Program 5.4: Program to use multiple checkbox to select hobbies.

```
<html>
<body>
<form method="post" action="#">
please select your hobbies <br>
<input type="checkbox" name="check_b[]" value="cricket">Cricket<br>
<input type="checkbox" name="check_b[]" value="football">football<br>
<input type="checkbox" name="check_b[]" value="singing">singing<br>
<input type="checkbox" name="check_b[]" value="dancing">dancing<br>
<input type="checkbox" name="check_b[]" value="reading">reading<br>
<input type="submit" name="submit" value="submit">
</form>
```

```
</body>
</html>
<?php
$checked = array();
if(isset($_POST['submit']))
{
if(isset($_POST['check_b']))
{
foreach($_POST['check_b'] as $ch)
{
echo "you selected $ch <br>";
}
}
}
?>
```

Output:



```
localhos
please select your hobbies
☑ Cricket
☑ football
☑ singing
  dancing
  reading
  submit

you selected cricket
you selected football
you selected singing
```

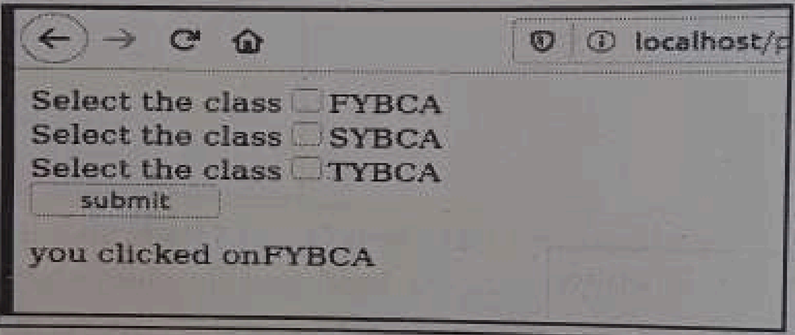
* In above example whatever hobbies are selected that are displayed on web page.

Program 5.5: Program to demonstrate multiple checkbox to create different classes.

```
<html>
<body>
<form action="#" method="post">
Select the class<input type="checkbox" name="class"
value="FYBCA">FYBCA</input><br>
Select the class<input type="checkbox" name="class"
value="SYBCA">SYBCA</input><br>
```

```
Select the class<input type="checkbox" name="class"
value="TYBCA">TYBCA</input><br>
<input type="submit" name="submit" value="submit"/>
</form>
</body>
</html>
<?php
if (isset($_POST['class'])){
echo "you clicked on".$_POST['class'];
}
?>
```

Output:



```
Select the class ☐FYBCA
Select the class ☐SYBCA
Select the class ☐TYBCA
submit
you clicked onFYBCA
```

5.2.2 Working with Radio Buttons

- The radio buttons are for single choice from multiple options. All radio buttons in the group have the same name attribute.
- Only one button can be selected per group. Radio button, use the value attribute to store the value that is sent to the server if the button is selected.
- The value attribute is mandatory for checkboxes and radio buttons, and optional for other field types.
- A Radio Button is a way to restrict users to having only one choice. Examples are : Male/Female, Yes/No, or we can use it for multiple quizzes.
- We can use a radio button in php with input element and attribute type radio.
- For example,

```
<input type="radio" name="city" value="pune"> pune <br>
<input type="radio" name="city" value="mumbai"> mumbai<br>
```

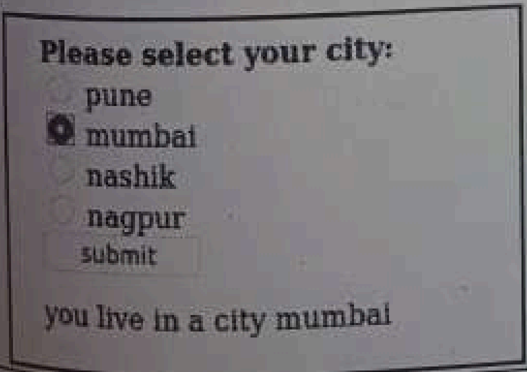
- To get the value of a radio button with PHP code, you access the NAME attribute of the HTML form elements. In the HTML above, the NAME of the Radio buttons is the same - "city". The first Radio Button has a value of "pune" and the second Radio Button has a value of mumbai.

- Now lets see one more example to display the values which are selected with radio buttons.

Program 5.6: Program to Demonstrate use of Radio Button.

```
<html>
<body>
<form action="#" method="post">
<b>Please select your city:</b> <br>
<input type="radio" name="city" value="pune"> pune <br>
<input type="radio" name="city" value="mumbai"> mumbai<br>
<input type="radio" name="city" value="nashik"> nashik <br>
<input type="radio" name="city" value="nagpur"> nagpur <br>
<input type="submit" value="submit">
</form>
</body>
</html>
<?php
$city = $_POST['city'];
if( ( $city != null ) )
{
echo "your live in a city $city ";
}
?>
```

Output:



Please select your city:

○ pune
● mumbai
○ nashik
○ nagpur

```
you live in a city mumbai
```

5.3 RETRIEVING VALUES FROM LISTS

- Select box, also known as a "drop-down" or "pull-down" box. A select box contains one or more "options". Each option has a "value", just like other inputs, and also a string of text between the option tags.

```
<p>
<select name="gender">
<option value ="">select </option>
<option value="Male">Male</option>
<option value="Female">Female</option>
</select>
```

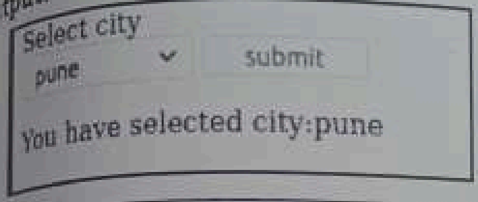
- The selected value can be read with $_post[] array just like a text input and it can b validated also.

```
<?php
if(isset($_POST['gender']))
{
$var=$_POST['gender'];
echo "$var";
}
```

Program 5.7: Program to retrieve values of select element with PHP.

```
<html>
<body>
<form action="#" method="post">
Select city <br>
<select name="city">
<option value="pune">pune</option>
<option value="mumbai">mumbai</option>
<option value="kolhapur">kolhapur</option>
<option value="nagpur">nagpur</option>
<option value="aurangabad">aurangabad</option>
</select>
<input type="submit" name="submit" value="submit">
</form>
</body>
</html>
<?php
if(isset($_POST['submit'])){
$result = $_POST['city'];
echo "You have selected city:" .$result;
}
?>
```

Output:



```
Select city
pune        submit

You have selected city:pune
```

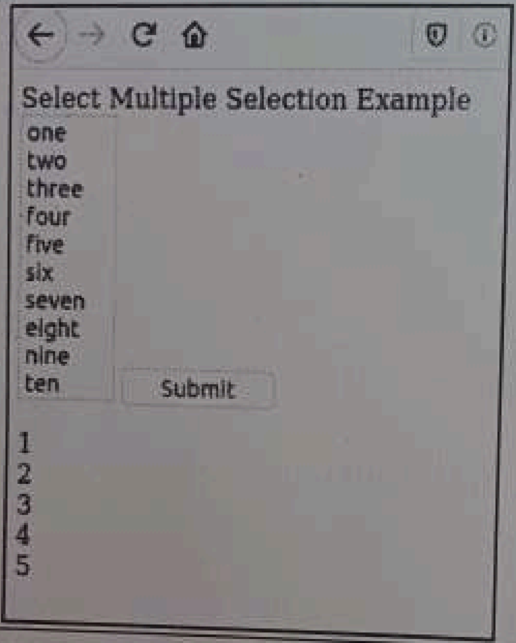
- To get value of multiple select option from select tag, name attribute in HTML <select> tag should be initialize with an array [ ].

Program 5.8: Program with list element having multiple selection.

```
<html>
<head>
<title>select tag example</title>
</head>
<body>
<form id="form1" name="form1" method="post" action="#">
Select Multiple Selection Example <br>
<select name="select[]" size="10" multiple="multiple" tabindex="2">
<option value="1">one</option>
<option value="2">two</option>
<option value="3">three</option>
<option value="4">four</option>
<option value="5">five</option>
<option value="6">six</option>
<option value="7">seven</option>
<option value="8">eight</option>
<option value="9">nine</option>
<option value="10">ten</option>
</select>
<input type="submit" name="Submit" value="Submit">
</form>
</body>
</html>
<?php
if ($_POST) {
```

```
foreach($_POST['select'] as $value) {
print $value. "<br>";
}
}
?>
```

Output:



```
Select Multiple Selection Example
one
two
three
four
five
six
seven
eight
nine
ten
1
2
3
4
5
```

- In above example we have taken select[ ] with size 10 and we have printed all the elements of select tag using foreach loop. [ ] this tells the PHP engine to treat the multiple values as array elements.

5.4 VALIDATING AND RESTRICTING DATA

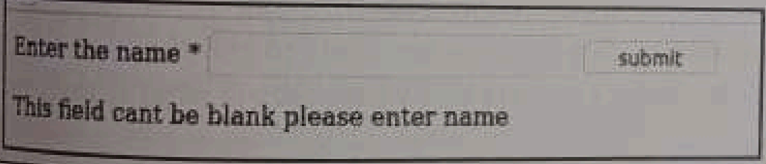
- Data validation is the most important part of web applications, you have to accept and store your data properly without any damage.
- Mostly we use html form to capture the information through various form elements.
- The form is created using HTML and validation and processing of form contents is done with PHP.
- We will apply validations on HTML form elements textfields, radio buttons, checkbox, multiple select option so that user will be able to submit correct information.
- Validations refer to action of checking accuracy of something. When user enters information through HTML form we need to validate this information through code like checking correct usernames, phone numbers, pan card etc.
- We will apply validations on strings, numbers, email ID and date.

- As already we have used empty() function we will apply the same for string validations. If the user leaves the required field empty, it will show error message.

Program 5.9: Program for validation with input field.

```
<html>
<body>
<form name="form1" method="post" action="#">
Enter the name * <input type="text" name="text1"></input>
<input type="submit" value="submit"></input>
</form>
</body>
</html>
<?php
if (empty($_POST["text1"])) {
echo "This field cant be blank please enter name.";
} else {
$name = $_POST["text1"];
echo $name;
}
```

Output:



```
This field cant be blank please enter name
```

5.4.1 Validating String

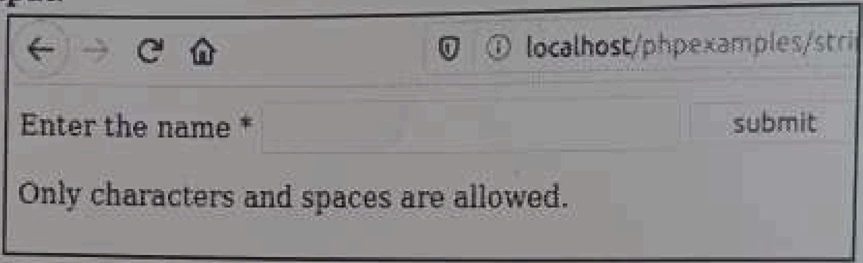
- We will accept input from user and allows to accept only alphabets and whitespace otherwise error message will be thrown.

Program 5.10: Program for string validation in PHP.

```
<html>
<body>
<form name="form1" method="post" action="#">
Enter the name * <input type="text" name="text1"></input>
<input type="submit" value="submit"></input>
</form>
</body>
</html>
<?php
```

```
$name = $_POST ["text1"];
if (!preg_match ("/^[a-zA-z]*$/", $name) ) {
echo "Only characters and spaces are allowed.";
} else {
echo $name;
}
?>
```

Output:



Enter the name * submit

Only characters and spaces are allowed.

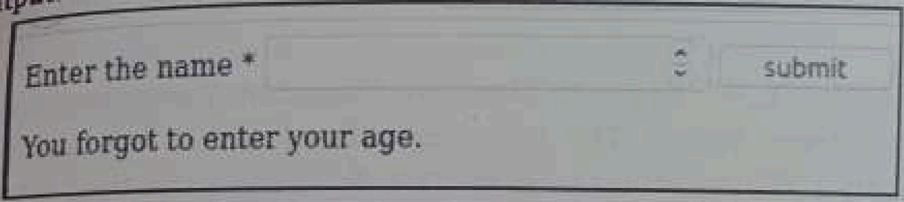
5.4.2 Validating Numbers

- When we enter numbers into web forms, it is to validate them to enter numbers only and to check the specific length of that numbers like we provide mobile numbers which should be upto 10 digits.
- Pin codes should be of 6 digits, house numbers, cash amounts. Pan card, Aadhar card etc. For these fields we want only numbers entered into the form field. If any characters in the sequence are non-numeric, then we want to tell the user that the sequence of characters which they entered in are not valid. This is what is referred to as number validation.

Program 5.11: Program for number validation with is_numeric( ) function.

```
<html>
<body>
<form name="form1" method="post" action="#">
Enter the number * <input type="number" name="age"></input>
<input type="submit" value="submit"></input>
</form>
<?php
if ((isset($_POST['age'])) && (is_numeric($_POST['age']))) {
$val=$_POST['age'];
echo "your age is". $val;
} else {
echo '<p>You forgot to enter your age.</p>';
}
```

Output:



```
Enter the name *                     submit
You forgot to enter your age.
```

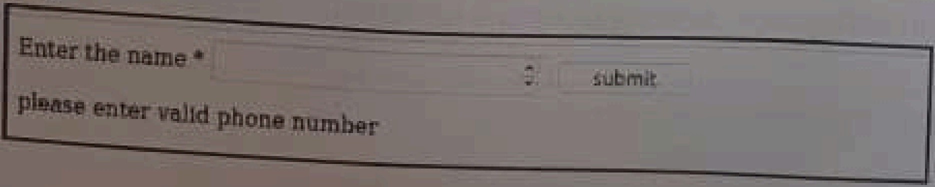
5.4.3 Validating Length of Inputted Number

- We will accept phone number from user and will check the length of the number we can use strlen function. If the length is not equals to 10 error message will displayed.

Program 5.12: Program for length validation of number.

```
<html>
<body>
<form name="form1" method="post" action="#">
Enter the name * <input type="number" name="phone"></input>
<input type="submit" value="submit"></input>
</form>
</body>
</html>
<?php
//function declaration
function count_digit($number) {
return strlen((string) $number);
}
$num = $_POST['phone'];
$length=count_digit($num); //calling a function
if($length<10 || $length>10)
echo "please enter valid phone number";
else
echo $length;
?>
```

Output:



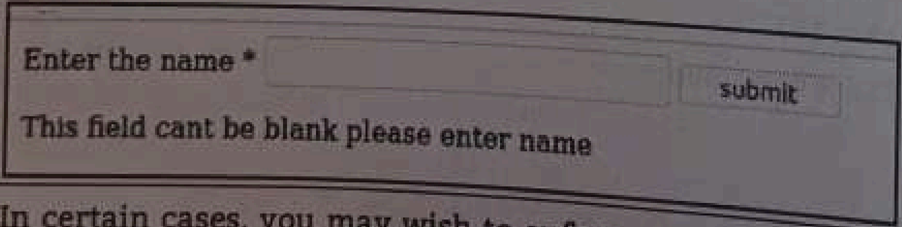
```
Enter the name *                     submit
please enter valid phone number
```

- Here we have used a function count_digit which will count the length of string and it will be returned.
- Now we use FILTER_VALIDATE_INT flag with filter_var() function. This method can be used to validate any integer values. The advantage of using this function is, it converts string numbers ("25") to actual integers (25). So that we can treat integer inputs as integer without any hesitation after sending the input through this function.

Program 5.13: Program to demonstrate use of FILTER_VALIDATE_INT.

```
<html>
<body>
<form name="form1" method="post" action="#">
Enter the number* <input type="number" name="phone"></input>
<input type="submit" value="submit"></input>
</form>
</body>
</html>
<?php
if (!empty($_POST['phone'])) {
$number = $_POST['phone'];
$number = filter_var($number, FILTER_VALIDATE_INT);
if ($number === false) {
exit('Invalid Integer');
}
}
?>
```

Output:


Enter the name * [submit]
This field cant be blank please enter name

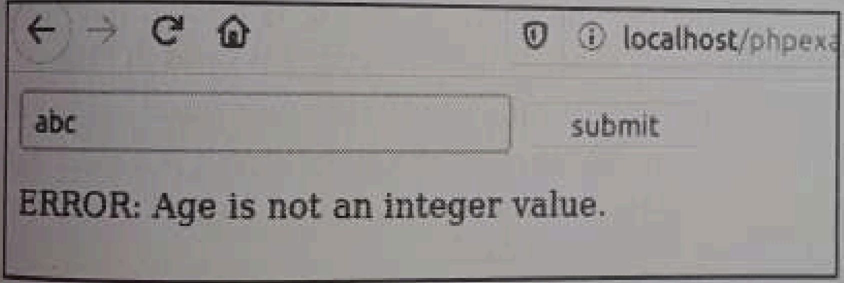
- In certain cases, you may wish to enforce a range constraint on the numeric values that your application will accept. Testing is simple using PHP's comparison operators, as shown here:

Program 5.14: Program to validate age of person.

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

```
<form method="post" action="">
<input type="text" name="age" placeholder="Enter your age">
<input type="submit" name="submit" value="submit">
</form>
</body>
</html>
<?php
$valid = array();
if((strval($_POST['age'])==strval((int)$_POST['age'])) && ($_POST['age'] >=
1 && $_POST['age'] <= 100)) {
$valid['age'] = trim($_POST['age']);
echo "age is valid";
} else {
die ('ERROR: Age is not an integer value.');
}
?>
```

Output:



```
ERROR: Age is not an integer value.
```

5.4.4 Validating email

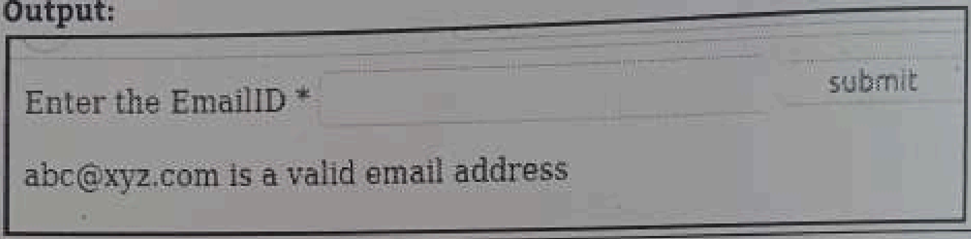
- Email Id can be validated with standard filters like FILTER_VALIDATE_EMAIL - it is used to check whether email id is valid or not.

Program 5.15: Program to validate email using FILTER_VALIDATE_EMAIL.

```
<html>
<body>
<form name="form1" method="post" action="#">
Enter the EmailID * <input type="text" name="email"></input>
<input type="submit" value="submit"></input>
</form>
</body>
</html>
<?php
$email = $_POST['email'];
```

```
if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
echo("$email is a valid email address");
}
else {
echo("$email is not a valid email address");
}
?>
```

Output:



```
Enter the EmailID *                                   submit

abc@xyz.com is a valid email address
```

- FILETER_SANITIZE_EMAIL flag is also used to pass filter_val( ) function for vaidating email it removes extra characters from email and validate it.

5.4.5 Validating Dates

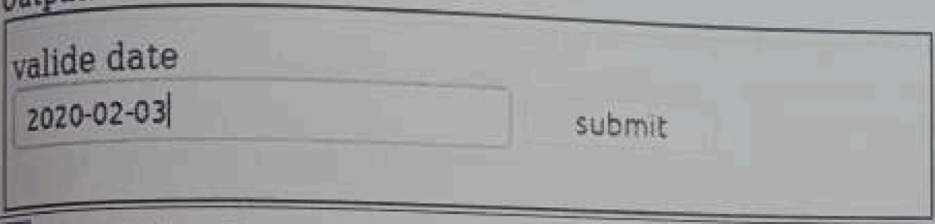
- Date validation helps to check whether the provided string is a valid date format. Using the DateTime class you can easily check if the date string is valid in PHP.
- Validating date is very useful for server-side validation of the date input using PHP.
- In PHP the validateDate( ) function checks whether the given string is a valid date. It uses PHP DateTime class to validate date based on the specified format. This function returns TRUE if date string is valid, otherwise FALSE.

Program 5.16: Program for date validation.

```
<?php
function validateDate($date, $format = 'Y-m-d'){
$d = DateTime::createFromFormat($format, $date);
return $d && $d->format($format) === $date;
}
$test_date=$_POST['bdate'];
if(validateDate("$test_date"))
{
echo" valide date";
}
else{
echo "invalid date";
}
?>
```

```
<html>
<body>
<form method="post" action="">
<input type="text" name="bdate" placeholder="Enter YYYY-MM-DD">
<input type="submit" name="submit" value="submit">
</body>
</html>
```

Output:



```
valide date
2020-02-03        submit
```

5.4.6 Matching Patterns

- If you want to have more complex validations, php supports regular expressions, a powerful tool for matching and validating string patterns.
- Regular expression syntax includes the use of special characters. The characters that are given special meaning within a regular expression, are: . * ? + [ ] ( ) { } ^ $ |.
- You will need to backslash these characters whenever you want to use them literally. For example, if you want to match ".", you'd have to write \.. All other characters automatically assume their literal meanings.
- The simplest expression will find exact character inside strings. In these expressions breackets have special meanings [ ] they are used for range of characters.

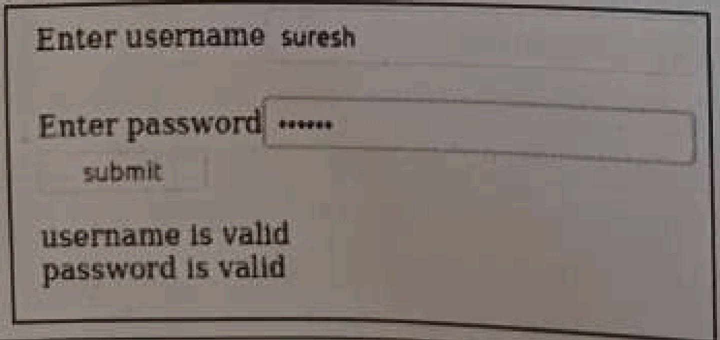
Metacharacter	Description
^	Beginning of string
$	End of string
.	Any character except newline character
\s	A single whitespace character
\S	A single non white space character
\d	A digit between 0 to 9
\w	An alphanumeric or numeric character or underscore
[A-Z]	It will match from uppercase A to uppercase Z
[a-z]	It will match from lowercase a to lowercase z
[0-9]	It will match for 0 to 9 decimal digit
[a-Z]	It will match from lowercase a to uppercase Z

- We will use these regular expressions for validating username and password.
- The username must be three and eight characters long containing alphanumeric characters.
- The password should be five to eight characters long and it should contain atleast number.

Program 5.17: Program for validating username and password.

```
<html>
<body>
<form name="form1" method="post" action="">
<span>Enter username<input type="text" name="username">
</input></span><br><br>
<span>Enter password<input type="text" name="password"></input></span><br>
<span> <input type="submit" value="submit"></input></span>
</form>
</body>
</html>
<?php
if (isset($_POST["username"]) && preg_match('/^([a-zA-
Z]){3,8}$/',$_POST["username"])) {
echo "username is valid";
} else {
die ('<br>username is invalid. <br>');
}
// checking password
if (isset($_POST["password"]) &&
preg_match('/^(?=.*\d).{5,8}$/',$_POST["password"])) {
echo "password is valid";
} else {
die ('password is invalid.<br>');
}
?>
```

Output:

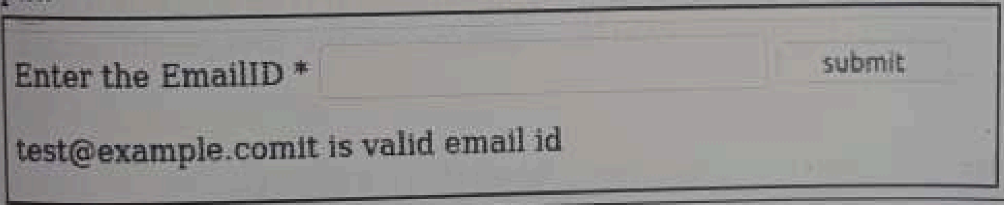


```
Enter username suresh

Enter password ••••••
submit

username is valid
password is valid
```

Program 5.18: Validating email id through regex expressions.

```
<html>
<body>
<form name="form1" method="post" action="#">
Enter the EmailID * <input type="text" name="email"></input>
<input type="submit" value="submit"></input>
</form>
</body>
</html>
<?php
$email = $_POST['email'];
$regex  =  '/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-
z]{2,3})$/';
if (preg_match($regex, $email)) {
echo $email . "it is valid email id";
} else {
echo $email . " invalid email, Please try again.";
}
```

Output:



```
Enter the EmailID *                              submit

test@example.comit is valid email id
```

5.4.7 PCRE Expressions: (Perl Compatible Regular Expressions)

1) **preg_match( ) function :**

- In PHP, you can use the preg_match( ) function to test whether a regular expression matches a specific string. Note that this function stops after the first match, so this is best suited for testing a regular expression more than extracting data.

Method:

- preg_match(string $pattern, string $input, array matches, int $flags, int $offset)
- It searches for a match to the given regular expression in pattern.

Parameters:

- $pattern – The pattern to search as a string.
- $subject – It is input string.
- Matches – It is optional.if matches are provided then it is filled with the result of search.

- Flags – It is optional. A set of options that change how the match array is structured.
- Offset – It is optional. And default value is 0. It indicates how far string is to be searched.

Program 5.19: Program to demonstrate use of preg_match() function.

```
<?php
if (preg_match("/php/i", "PHP is the web scripting language.")) {
// i indicates case sensitive search
echo " match found.";
} else {
echo " match not found.";
}
?>
```

Output:

```
match found.
```

2) preg_match_all():

- The preg_match_all function searches for all the matches in a string and outputs them in a multi-dimensional array ($matches) ordered according to $flags. When no $flag value is passed, it orders results so that $matches[0] is an array of full pattern matches, $matches[1] is an array of strings matched by the first parenthesized sub-pattern, and so on.

  Syntax:

  ```
  preg_match_all($pattern, $subject, &$matches, $flags);
  ```

Program 5.20: Program to demonstrate use of preg_match_all function.

```
<?php
preg_match_all(
"|<[^>]+>(.*)</[^>]+>|U","<b>this is first example: </b><b>this is second
example</b>",
$matches
);
var_dump($matches);
?>
```

Output:

```
array(2) { [0]=> array(2) { [0]=> string(30) "this is first example: " [1]=> string(29) "this
is second example" } [1]=> array(2) { [0]=> string(23) "this is first example: " [1]=>
string(22) "this is second example" } }
```

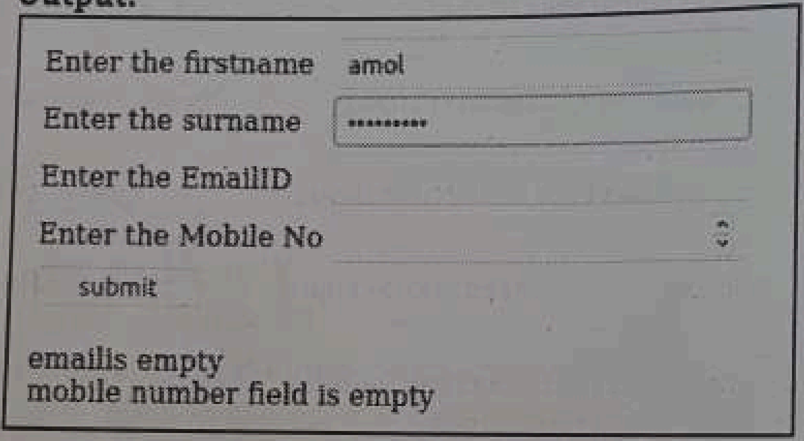
Now we will design a student registration form with student name, surname its mobile and email id. And perform empty validations on it. If any of the field in left blank user will get error message about field saying it can not be left blank.

Program 5.21: Program for student registration with fields firstname, surname, email, mobile.

```
<html>
<body>
<form method="post" action="studreg.php">
<table>
<tr><td>Enter the firstname</td><td><input type="text" name="firstname"></td></tr>
<tr><td>Enter the surname</td><td><input type="password" name="surname"></td></tr>
<tr><td>Enter the EmailID</td><td><input type="text" name="email"></td></tr>
<tr><td>Enter the Mobile No</td><td><input type="number" name="mobile"></td></tr>
<tr><td><input type="submit" name="submit" value="submit"></td></tr>
</table>
</body>
</html>
<?php
if(isset($_POST['submit']))
{
$firstname=$_POST['firstname'];
$surname=$_POST['surname'];
$email=$_POST['email'];
$mobile=$_POST['mobile'];
}
if(empty($firstname))
{
echo "<br>firstname is empty";
}
if(empty($surname))
{
echo "<br>surname is empty";
}
```

```
if(empty($email))
{
echo"<br>emailis empty";
}
if(empty($mobile))
{
echo "<br>mobile number field is empty";
}
?>
```

Output:



```
Enter the firstname    amol
Enter the surname      .........
Enter the EmailID
Enter the Mobile No
 submit

emailis empty
mobile number field is empty
```

5.5 SENDING EMAIL

- PHP comes with a default function mail( ) that allows you to send mail directly from a PHP script. Its very simple to use just you have to use mail( ) function and pass parameters to it. There are some parameters which you have to pass there can be five parameters but first three are required.

Syntax:

```
mail($to, $subject, $message, [$additional headers, [$additional
parameters]]
```

Parameters:

1. $to – receiver of the mail.
   Like, abc@sample.com, user@example.com
2. $subject – subject of the email to be sent.
3. $message – message to be sent with mail. The length should not be larger than 70 characters.
4. $additional headers – strings or array to be inserted at the end of email header. These are used to add extra headers like cc, bcc.

While sending mail headers must be contained.

```
$headers = implode("\r\n", [
'From: John smith <test@example.com>',
'Reply-To: test@example.com',
'X-Mailer: PHP/' . PHP_VERSION
]);
```

- The optional fifth parameters is used to pass additional flags as command line options to program configured to be used when sending mail, for example this can be used to set the envelope sender address with -f sendmail option.

```
$additional parameters = 'fno-reply@example.com'
```

- mail( ) function is reliable(), it will return true value when mail() is successfully sent otherwise it will return false.

```
$result=($to, $subject, $message, $headers, $fifth);
```

- Sometimes mail( ) function returns true value but it does not guarantee that mail is sent successfully there are other parameters also affect like some mails can be blocked by server policy.
- TRUE is just an indication that your mail has successfully submitted to the SMTP server's queue for sending.

Program 5.22: Program to demonstrate use of mail( ) function.

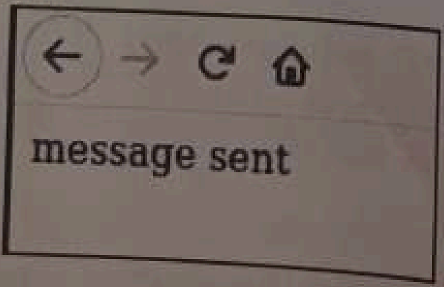
email.php

```
<?php
$address="username@gmail.com";
$subject=" Greetings of the day";
if($_POST['submit'])
$name=$_POST['name1'];
$message=$_POST['message'];
?>
<html>
<form action="sendmail.php" method="post">
Enter Your Name<br><input type="text" name="name1" maxlength="20">
Enter Your Message <br><textarea name="message"></textarea><br>
<input type="submit" name="submit" value="Send an Email">
</form>
</html>
```

sendmail.php

```
<?php
if($_POST['submit'])
$name=$_POST['name1'];
$message=$_POST['message'];
$sender="username@gmail.com";
//echo $name.$message <br>;

if($name && $message){
if(strlen($name<=20) && strlen($message<=200))
{
$to="username@gmail.com";
$subject="test mail";
$body=$message;
$headers="From:". $sender;
ini_set("SMTP","mail.google.com");
if(mail($to,$subject,$body,$headers)){
echo "message sent";
}
else{
echo "message not sent";
}
die();
}
else{
die("max length for name is= $name and max length for message is=
$message");
}
}
```

Output:



```
message sent
```

5.5.1 SwiftMailer

- Swiftmailer can be used with any php framework, and it can be easily integrated with external SMTP like Gmail as well as with other service providers.
- SwiftMailer is a component based library, used with webapps. It is a open source software.
- SwiftMailer is used as main mailing option in frameworks. To use swiftmailer it recommended to use Composer.
- Composer is a dependency manager for PHP. It is a simple and reliable tool that developers use to manage and integrate external packages or libraries into their PHP-based projects.
- First you need to install curl in order to install composer. First updates the package manager using command on terminal in linux.

```
Sudo apt-get update
```

- To install curl type the following command,

```
sudo apt-get install curl
sudo curl -s https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/composer.
```

- Now to test installation of composer type,

```
$composer.
```

- Then you need to change the file permission of composer.json file.

```
chmod 777 composer.json
```

- And now restart your server using following command,

```
sudo service apache2 restart.
```

- And now to install SwiftMailer type the command,

```
$composer require "swiftmailer/swiftmailer:^6.0"
```

- After successful installation of SwiftMailer, we have to specify transport and mailer.
- In SwiftMailer messages are composed with Swift_Message class. Its instance have various methods to manage attachments, media.

```
//create the message
    $message=(new swift_Message())
// add subject
    ->setSubject('here we need to add subject')
// set from address
    ->setFrom ([usrename@example.com])
    ]);
```

Including attachments:

- You can attach files saved locally, or add dynamic content, with attach() method.

```
    $message->attach(Swift_Attachment::fromPath('/path/to/image.jpg'));
```

Sending Message with SwiftMailer:

- When we define subject of message, its recipients, and the contents. Now we can use send() method to send message.

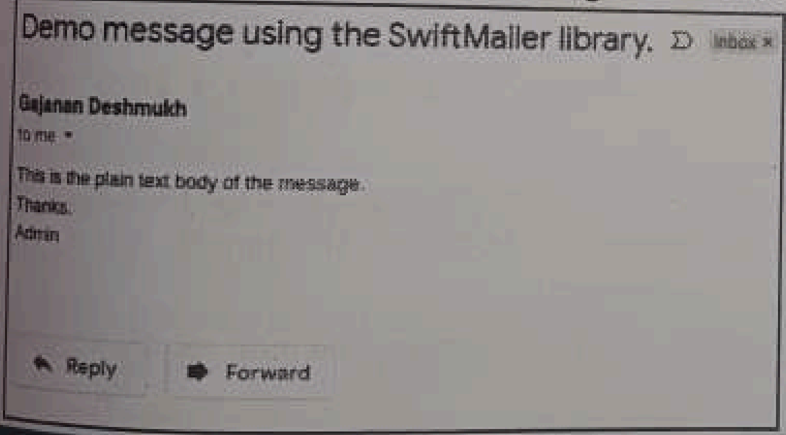
```
    $mailer->send($message);
```

- Now we will compose a whole message and send it with attachments CC and BCC.

```
    <?php
    require_once '/home/ubuntu/vendor/autoload.php';
    try {
    // It Create the SMTP Transport
    $transport = (new Swift_SmtpTransport('smtp.gmail.com', 465,"ssl"))
    ->setUsername('username@gmail.com')
    ->setPassword('yourpassword');
    // It Create the Mailer using your created Transport
    $mailer = new Swift_Mailer($transport);
    // It Creates  a message
    $message = new Swift_Message();
    // Set a "subject"
    $message->setSubject('This  is  Test  message  using  the  SwiftMailer
    library.');
    // Set the "From address"
    $message->setFrom(['senderusername@gmail.com' => 'provide Name here']);
    //  Set  the  "To  address"  [Use  setTo  method  for  multiple  recipients,
    argument should be array]
    $message->addTo('receiverusername@gmail.com','receiver name');
    // Add "CC" address [Use setCc method for multiple recipients, argument
    should be array]
    //$message->addCc('recipient@gmail.com', 'recipient name');
    //  Add  "BCC"  address  [Use  setBcc  method  for  multiple  recipients,
    argument should be array]
    //$message->addBcc('recipient@gmail.com', 'recipient name');
```

```
// Add an "Attachment" (Also, the dynamic data can be attached)
//$attachment = Swift_Attachment::fromPath('example.xls');
//$attachment->setFilename('report.xls');
//$message->attach($attachment);
$message->setBody("Hello This is the test mail sent to verify
swiftmailer.\nThanks,\nAdmin");
// Set a "Body"
// Send the message
$result = $mailer->send($message);
} catch (Exception $e) {
echo $e->getMessage();
}
```

- After successful execution of code you will be able to get following message in inbox.
- Sometimes you get errors related to SMTP in that case you need to check ports whether they are blocked . Also, in your mail account under the security tab you need to ON Less Secure app access.
- Allow less secure apps. If you are sending from localhost.


Demo message using the SwiftMailer library.

Gajanan Deshmukh

to me

This is the plain text body of the message.
Thanks,
Admin

Summary

- Checkbox helps user to choose between one of two mutually exclusive options. We have used empty( ) function to check whether checkbox is empty or not. Isset( ) function is used to heck whether a variable is set or not. If a variable is already unset with unset() function. check whether a variable is set or not.
- To get value of multiple check checkboxes, atrribute name ="checkbox" tag must be initialize with an array.

- The radio buttons are used for single choice from multiple options. All radio buttons in the group have the same name attribute. Only one button can be selected per group.
- Select element <select> is used to create drop down list. It is used commonly to collect input from user. Use of multiple attribute in HTML to select multiple values from drop down list.
- Validation means checking what user has submitted. We have used validations with numbers using is_numeric( ) function, also we have counted length of number to find valid mobile number or not.
- FILTER_VALIDATE_INT filter is used to validate value as integer. It also allows to specify range for integer variable.
- FILTER_VALIDATE_EMAIL filter validates an email address.
- Regular expression are sequence or pattern of characters. They provide us pattern matching functionality. Regular expression syntax includes the use of special characters. The characters that are given special meaning within a regular expression, are: . * ? + [ ] ( ) { } ^ $ |.
- The PCRE library is a set of functions that implement regular expression pattern matching using the same syntax and semantics as Perl. We have discussed preg_match( ) function it is used to test whether a regular expression matches a specific string. preg_match_all( ) function is used to perform a global regular expression match.
- PHP used mail( ) function to send emails this function requires mandatory arguments that specify recepient email adddress,subject,body and attachment. It has syntax mail(to, subject, message, header, parameters).
- Swiftmailer is component based library, it is open source and most commonly used to sending mails.

Check Your Understanding

1. Which one of following should not be used while sending sensitive information?
   (a) GET (b) POST
   (c) REQUEST (d) NEXT
2. To validate email address, which flag is passes to filter_val() function?
   (a) FILTER_VALIDATE_EMAIL (b) FILET_VALIDATE_MAIL
   (c) VALIDATE_EMAIL (d) VALIDATE_MAIL

3. How many functions PHP offers for searching and modifying string using PCRE?
   (a) 7 (b) 8
   (c) 9 (d) 10
4. PHP supports for two regular expression implementations known as __ and __.
   i) perl ii) PEAR
   iii) POSIX iv) regex1
   (a) i and iii (b) ii and iii
   (c) i and iv (d) ii and iii
5. Which of the following is correct about preg_match() function?
   (a) The preg_match( ) function is used to search specified string by string, returns true if the pattern found otherwise false
   (b) The preg_match( ) function searches throughout string specified by pattern for a string specified by string. The search is not case sensitive.
   (c) The preg_match( ) function searches string for pattern, returning true if pattern exist otherwise false.
   (d) None of Above

ANSWER KEY

1. (a)	2. (a)	3. (b)	4. (b)	5. (a)

Practice Questions

Q.I: Answer the following Questions in short.

1. What is validation?
2. Which attribute is used for multiple selection in select tag?
3. Differentiate between checkbox and radio button tag element?
4. Explain select tag element with suitable example?
5. Explain empty( ) and isset( ) function with checkbox.

Q.II: Answer the following Questions.

1. Explain data validations in PHP?
2. Explain sending mail in PHP?
3. How to access data from checkbox? Give example?
4. Explain how to use multiple checkbox?
5. How radio buttons form elements are used in PHP?
6. Explain select tag element in PHP with example?

7. Explain regular expressions in PHP?
8. Write a PHP script to accept username and password, allow three chances for user otherwise display error message.
9. Write a PHP Program to create input form of grocery that displays list of grocery items with chcekboxes and create a bill according to the items selected.

Q.III: Define terms:

1. Regular expressions
2. preg_match()
3. PCRE
4. Preg_match_all()
5. SwiftMailer

❖❖❖