selection structures

SELECTION IN PHP

if construct.

- *if* is a PHP reserved words to perform selection.
- There is another selection, that is *switch…case*. We will only discuss *if* for this course.
- There are 4 ways of using *if*.
  - *i)* Simple *if*
  - *ii)* *If … else*
  - *iii)* *if … else if … else if … else*
  - iv) Nested *if*

<u>A. simple *if*</u>
(one choice selection)

The format:
```
    if (condition) {
        statements;
    }
```

- Condition is a Boolean expression that will produce ***true*** or ***false***.
- All the statements between the if *block* will only be executed if the value produced by the condition is true.

<u>Example</u>: *makan* and *kenyang*.
Study the sentence below;
        If I eat, then full.

If there is a programming language in plain English, the sentence above will become like this if it is translated into program.
```
    If I eat, then
        I'm full.
```

I'm full will happen if only *I eat* is true.

Another <u>example</u> :
Below is a simple algorithm,
        If *gred* is 'A', then display "You are great!".

gred is a character variable to hold a character for a student's grade. If the character inside *gred* is 'A', then the program will display message "You are great".

In PHP, the program will be like this;
```
    if ($gred = = 'A')
        echo ("You are great!");
```

B. *if …. else*

(Used when the selection have only two choices)

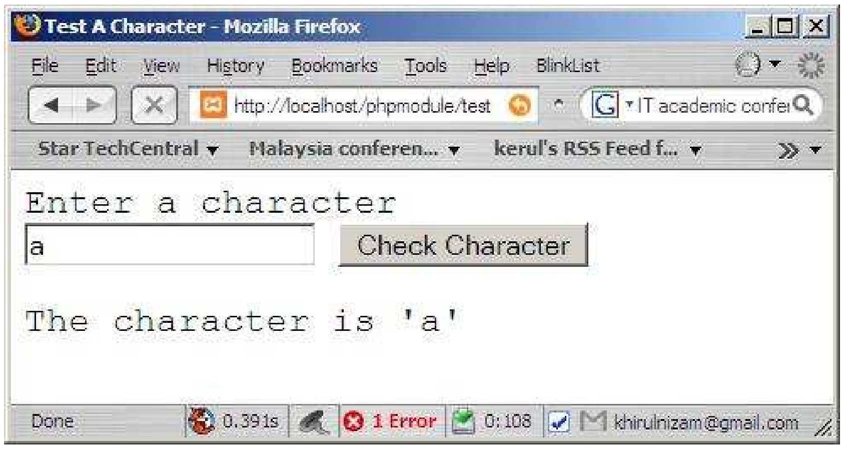
Example :
The example below shows the usage of simple *if...else* .
Algorithm :

```
1. receive a character from a user
2. if the character is 'a'
     then display "Huruf yang anda masukkan ialah a"
3. if the character is other than 'a'
     then display "Huruf yang anda masukkan bukan a"
```

Below is the complete program

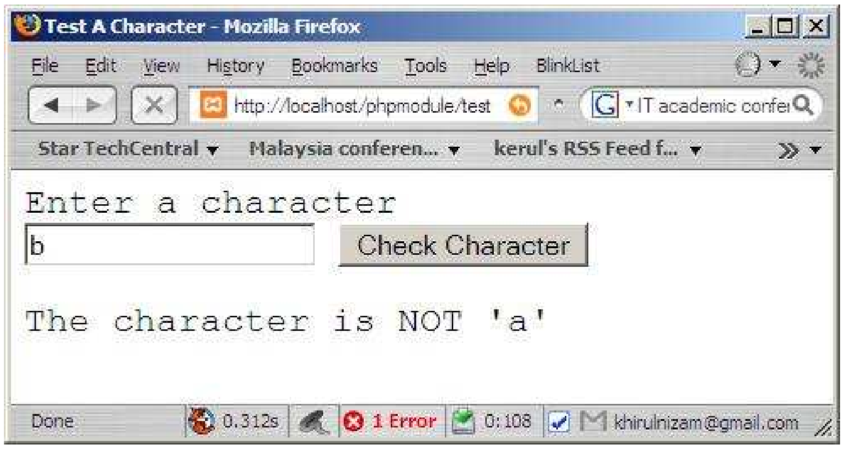
```
<html>
<head><title>Test An Alphabet</title></head>
<body>
Enter a small alphabet<br>
<form name="frmChar" method="get" action="testCharacter.php">
  <input name="txtchar" type="text">
  <input name="btnSubmit" type="submit"  value="Check Char">
</form>

<?php
if ($_GET['txtchar'] !=NULL){
    if ($_GET['txtchar']=='a'){
        echo "The alphabet is 'a'<br>";
    }
    else{
        echo "The alphabet is NOT 'a'<br>";
    }
}
?>

</body>
</html>
```

If user enter an ‘a’, then the message “The character is a ” will appear. However if the input is other than ‘a’, the message is “The character is NOT a ”.


Test A Character - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/phpmodule/test
IT academic confe
Star TechCentral Malaysia conferen... kerul's RSS Feed f...
Check Character
Done 0.391s 1 Error 0:108 khirulnizam@gmail.com

```
Enter a character
a
The character is 'a'
```

This is the output if user enters ‘a’ for the input.


Test A Character - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/phpmodule/test
Check Character
Done

Enter a character

b

The character is NOT 'a'

This is the output if user enters other than 'a' for the input.

C. *if …. else if …..else if …..else if*

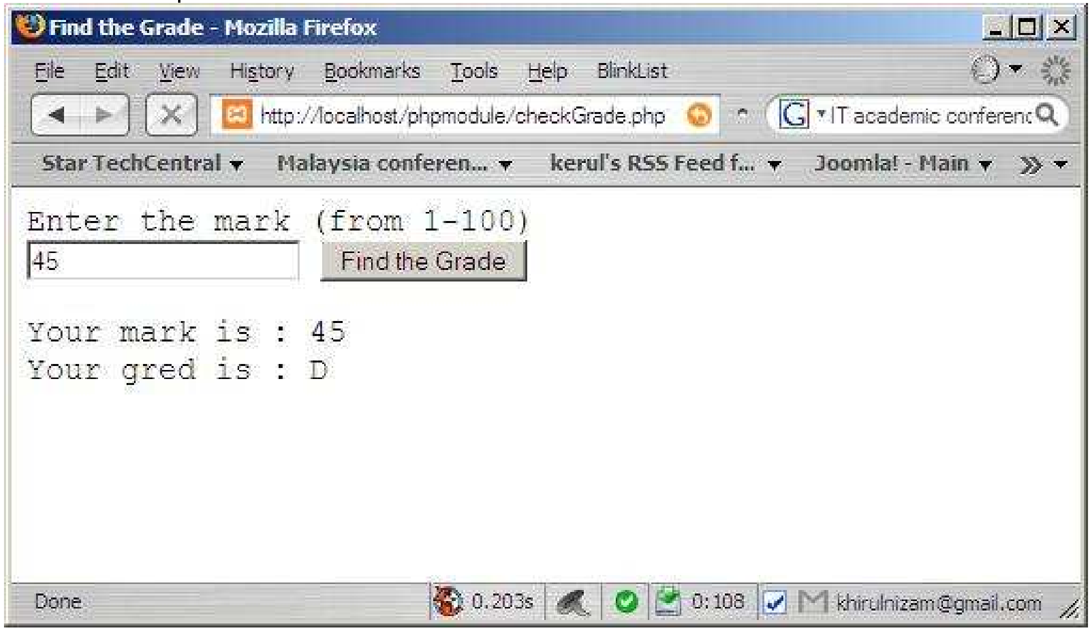
(Used to handle multiple-choice selection. No limitation on the number of the selection)

Example:
The program will display the grade for the entered mark.
Algorithm:

```
1. Get the mark entered by user.
2. if mark is between 100 to 80,
         then grade is A
   else, if mark is between 76 to 60,
         then grade is B
   else, if mark is between 56 to 50,
         then grade is C
   else, if mark is between 46 to 40,
         then grade is D
   else, if mark is between 36 to 0,
         then grade is F
   else,
         then "INPUT NOT VALID!!!!"
3. Display mark and grade.
```

```
<html>
<head><title>Find the Grade</title></head>
<body>
Enter the mark (from 1-100)<br>
<form name="frmMark" method="get" action="checkGrade.php">
    <input name="txtmark" type="text">
    <input name="btnSubmit" type="submit"  value="Find the Grade">
</form>

<?php
$mark=$_GET['txtmark'];
if ($mark != NULL){
    //mark is between 80-100
    if (($mark >=80)&&($mark <=100))
        $gred='A';
    //mark is between 60-80
    else if (($mark >=60)&&($mark <80))
        $gred='B';
    //mark is between 50-60
    else if (($mark >=50)&&($mark <60))
        $gred='C';
    //mark is between 40-50
    else if (($mark >=40)&&($mark <50))
        $gred='D';
    //mark is between 0-40
    else if (($mark >=0)&&($mark <40))
        $gred='F';
    //mark is out of range
    else
        $gred='input is not valid';

  //display result
  echo "Your mark is : $mark <br>";
  echo "Your gred is : $gred <br>";
}
?>
</body>
</html>
```

This is the output



```
Enter the mark (from 1-100)
45        Find the Grade

Your mark is : 45
Your gred is : D
```

D. *if….else* nested.

There are another *if* statements inside an *if* statement.

```
if (…..){
      …….
      if (……..){
             ……….
             ……….
       }
      else if (…….){
             ……….
             ……….
       }
      else {
            ……….
       }
}
else if (…..){
       ………….
       ………….
}
else {
       ………….
       ………….
}
```

Example:

The program in page 6 will ask the user to enter two integers. Then it will decide whether both numbers are odd or even or the numbers either one is odd or even.

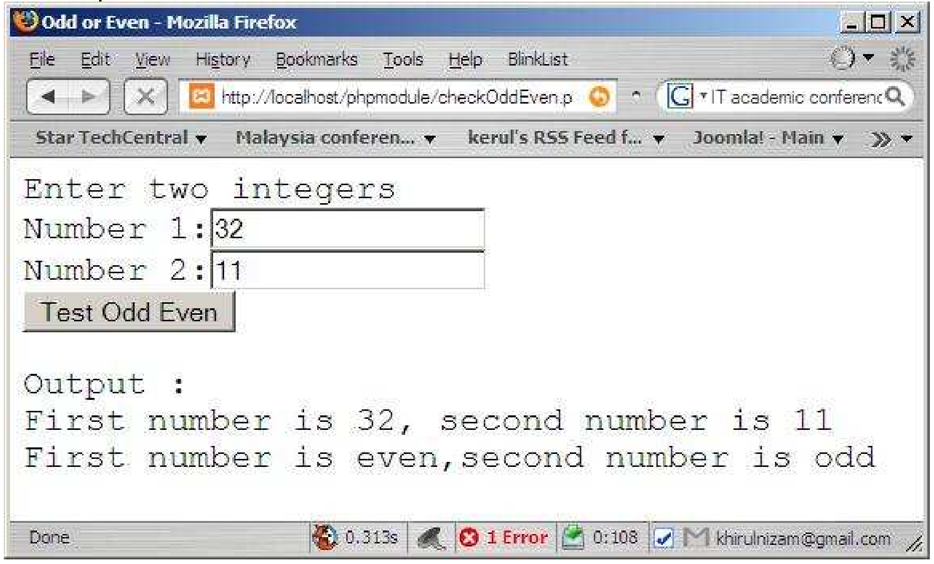
Algorithm:

```
1.    receive 2 integers.
2.    if the first number is even
              and the second number is even, then
                     display "Kedua-duanya genap",
              but if the second number is odd, then
                     display "Nombor pertama genap, kedua ganjil",
3.    else if the first number is odd
              and the second number is odd, then
                     display "Kedua-duanya ganjil",
              but if the second number is even, then
                     display "Nombor pertama ganjil, kedua genap",
```

```
<html>
<head>
<title>Odd or Even</title>
</head>
<body>
Enter two integers<br>
<form name="frmMark" method="get" action="checkOddEven.php">
```

```
    Number 1:<input name="txtnum1" type="text">
    <br>
    Number 2:<input name="txtnum2" type="text">
    <br>
    <input name="btnSubmit" type="submit"  value="Test Odd Even">
</form>
<?php
$num1=$_GET['txtnum1'];
$num2=$_GET['txtnum2'];

if ($num1 != NULL || $num2 != NULL){
      echo "Output :<br>";
      echo "First number is $num1, second number is $num2<br>";
      if ($num1%2 == 0){//first number is even
            if ($num2%2 == 0)//and second number also even
                  echo("Both numbers are even.");
            else //second number is odd
                  echo("First number is even, second number is odd");
      }
      else {//first number is odd
            if ($num2%2 != 0)//and second also odd
                  echo("Both numbers are odd.");
            else //second number even
                  echo("First number is odd, second number is even");
      }
}
?>
</body>
</html>
```

The output:


Odd or Even - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/phpmodule/checkOddEven.p
Star TechCentral
Malaysia conferen...
kerul's RSS Feed f...
Joomla! - Main
Done
0.313s
1 Error
0:108
khirulnizam@gmail.com

```
Enter two integers
Number 1:32
Number 2:11
Test Odd Even

Output :
First number is 32, second number is 11
First number is even,second number is odd
```

FORM VALIDATION

Validating input/data from a form is a very important step in order to make sure the input given by the user is not garbage / junk. The is no point saving the input from a form to the database if the is no data inside the input element. Or trying to verify user's username and password if there are no username or password entered by the user.

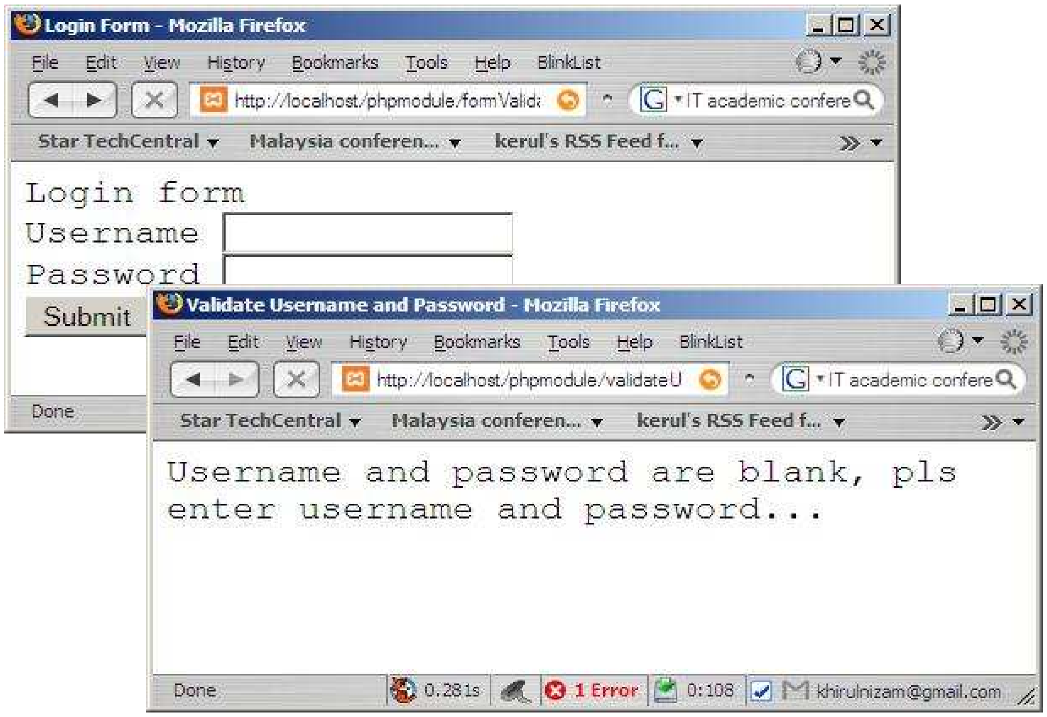
Form Validating Example 1 : Validating Username and Password
The following example is to make sure the user key-in the username and password, not sending a blank form.

```
<html>
<head><title>Login Form</title></head>
<body>
Login form <br>
<form name="frmLogin"  method="get" action="validateUP.php">
  Username
     <input name="txtUsername"  type="text"><br>
  Password
     <input name="txtPassword"  type="password"><br>
     <input name="btnSubmit"  type="submit"  value="Submit">
  <input name="btnReset"  type="reset"  value="Reset">
</form>
</body>
</html>
```

```
Target file : validateUP.php
<html>
<head>
     <title>Validate Username and Password</title>
</head>
<body>
<?php
  $username=$_GET["txtUsername"];
  $pword=$_GET["txtPassword"];
  if($username==NULL && $password==NULL){
     echo "Username and password blank,
           pls enter username and password...";
  }
  else if($username==NULL){
     echo "Username is blank, pls enter username...";
  }
  else if($password==NULL){
     echo "Password is blank, pls enter password...";
  }
  else{
     echo "Your username is : $username <br>";
     echo "Your password is : $pword <br>";
  }
?>

</body>
</html>
```

The outputs:

Form both with username and password are blank.


Login Form - Mozilla Firefox

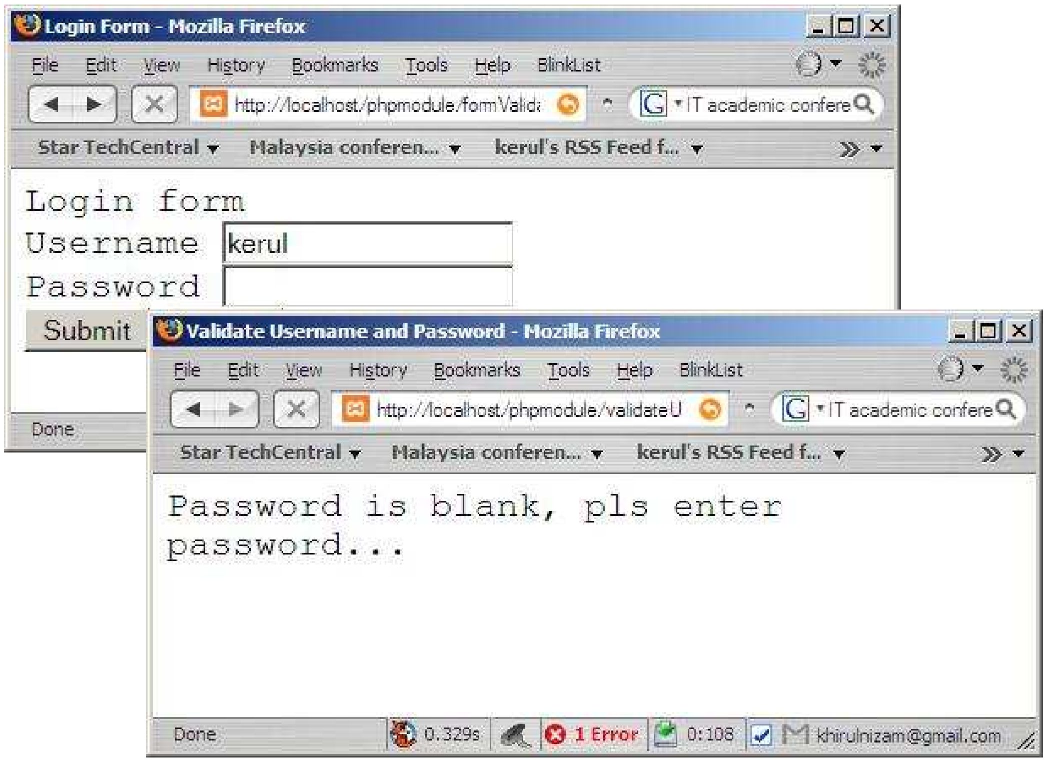
Login form
Username
Password
Submit

Validate Username and Password - Mozilla Firefox

http://localhost/phpmodule/validateU

```
Username and password are blank, pls
enter username and password...
```

Form both with password blank.


Login Form - Mozilla Firefox

Login form
Username kerul
Password
Submit

Validate Username and Password - Mozilla Firefox

http://localhost/phpmodule/validateU

```
Password is blank, pls enter
password...
```

Form Validating Example 2 : Validating Numbers

This is how we make sure the user enter only number for the input.

```
<html>
<head>
<title>Add two numbers</title>
</head>
<body>
Enter two numbers<br>
<form method="GET" name="form2numbers"     action="add2numbers-
validate.php">
     Number 1   <input type="text" name="num1"><br>
     Number 2   <input type="text" name="num2"><br>
     <input type="submit" name="btnAdd" value="Add numbers">
</form>
</body>
</html>
```

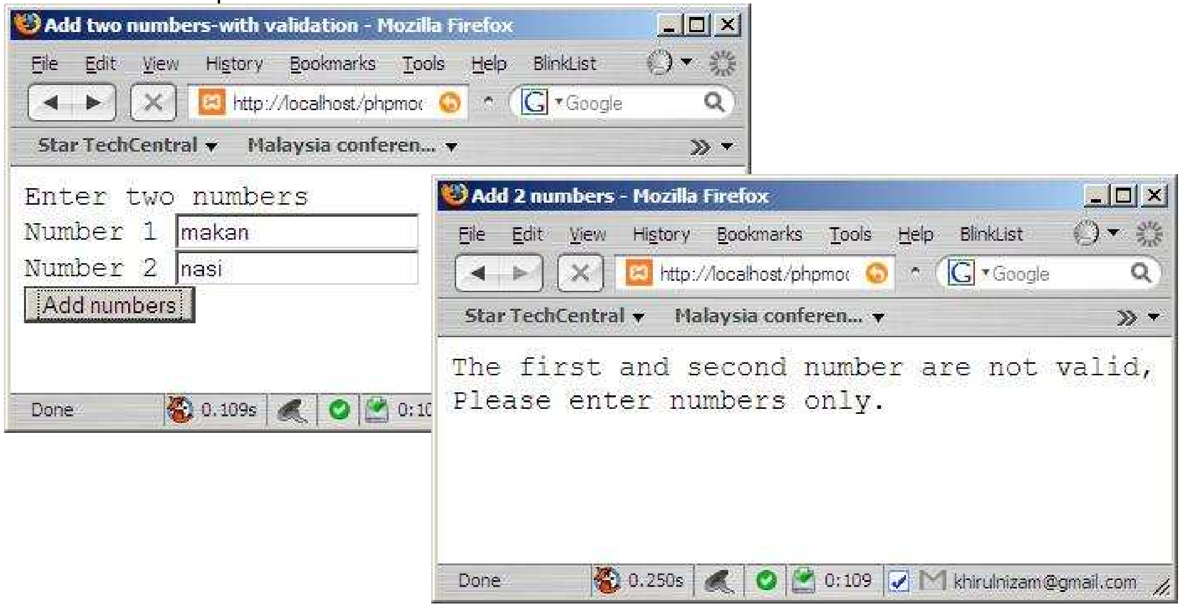
The target file : add2numbers-validate.php

```
<html>
<head>
<title>Add 2 numbers </title>
</head>
<body>
<?php
  $n1=$_GET["num1"]; //retrieve the first number
  $n2=$_GET["num2"];  //retrieve the second number

  //check so that both input are not blank
  if ($n1!=NULL || $n2!=NULL){
      //if both are numbers
      if(ctype_digit($n1) && ctype_digit($n2)){
           echo "The first number is: $n1 <br>";
           echo "The second number is: $n2";
           $hasil=$n1+$n2;
           echo " $n1 + $n2 = $hasil";
      }
      //if both are notnumbers
      else if(!ctype_digit($n1) && !ctype_digit($n2)){
           echo "The first and second number are not valid,<br>
                  Please enter a number only";
      }
      //if the first input is not number
      else if(!ctype_digit($n1)){
           echo "The first number is not a digit <br>";
           echo "Please enter a number only";
      }
      //if the second input is not number
      else if(!ctype_digit($n2)){
           echo "The first second is not a digit <br>";
           echo "Please enter a number only";
```

```
        }
    }
    else{//if both inputs are blanks
        echo "Please make sure to enter the both numbers<br>";
    }
?>
</body>
</html>
```

The outputs:

Form with both input are not numbers.

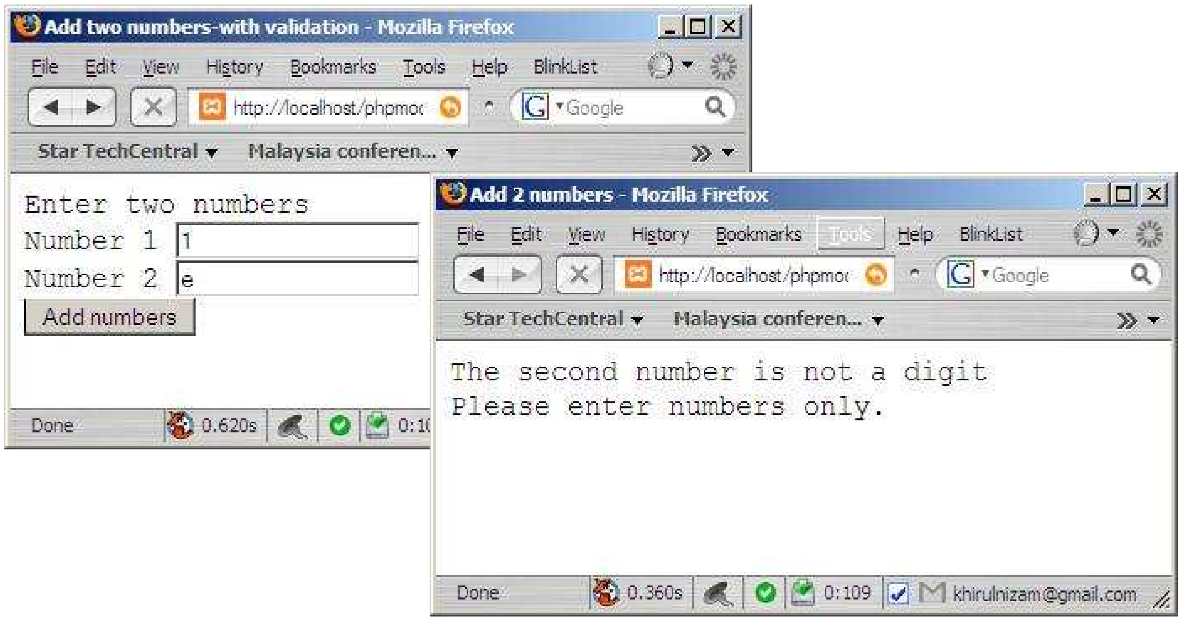


Add two numbers-with validation - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/phpmo
Google
Star TechCentral
Malaysia conferen...
Add numbers
Done
Add 2 numbers - Mozilla Firefox
khirulnizam@gmail.com

```
Enter two numbers
Number 1 makan
Number 2 nasi
```

```
The first and second number are not valid,
Please enter numbers only.
```

Form with both input are not numbers.



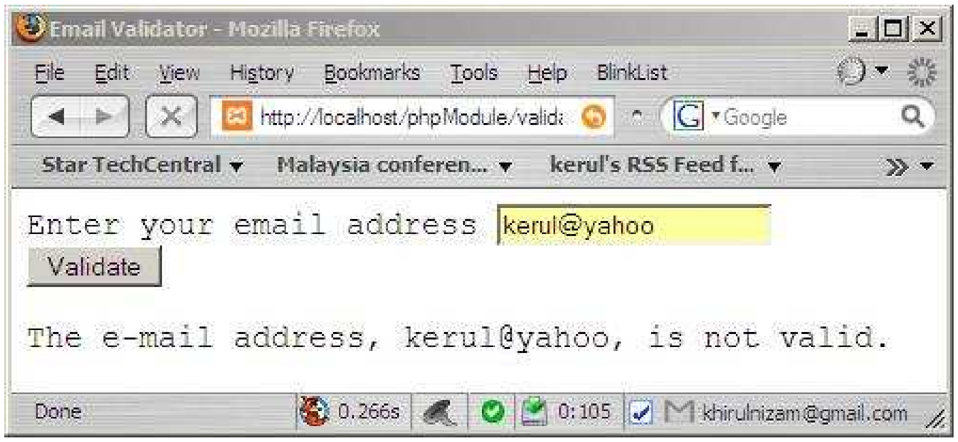
Add two numbers-with validation - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/phpmo
Google
Star TechCentral
Malaysia conferen...
Add numbers
Done
Add 2 numbers - Mozilla Firefox
khirulnizam@gmail.com

```
Enter two numbers
Number 1 1
Number 2 e
```

```
The second number is not a digit
Please enter numbers only.
```

Form Validating Example 3 : Validating Email Address using Regular Expression

```
<html>
<head>
<title>Email Validator</title>
</head>
<body>
<form action="validateEmail.php" method="get">
Enter your email address
      <input name="txtemail" type="text"><br>
      <input name="btnsubmit" type="submit" value="Validate">

</form>
<?php
$emailadd=$_GET['txtemail'];
if ($emailadd != NULL){
   $email=$_GET["txtemail"];
   $emelRegexp ="^[a-z0-9,!#\$%&'\*\+/=\?\^_`\{\|}~-]+(\.[a-z0-
9,!#\$%&'\*\+/=\?\^_`\{\|}~-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*\.([a-z]{2,})$
";//in one line
      if(!eregi($emelRegexp, $emailadd)){
         echo "The e-mail address, $emailadd,  is not valid.";
      }
      else {
         echo "The e-mail address, $emailadd,  is valid.";
      }
}
else{
      echo "No e-mail address yet to validate...";
}
?>
</body>
</html>
```


Email Validator - Mozilla Firefox
File Edit View History Bookmarks Tools Help BlinkList
http://localhost/phpModule/valid
Google
Star TechCentral
Malaysia conferen...
kerul's RSS Feed f...
Validate
Done
0.266s
0:105
khirulnizam@gmail.com

Enter your email address kerul@yahoo

The e-mail address, kerul@yahoo, is not valid.

*For further information on *eregi*, please refer to php.net/eregi.
**The example on regular expression to validate e-mail address is provide by Markus Sipilä from user contributed note in http://www.markussipila.info/pub/emailvalidator.php.

Exercises for Selection in PHP

Question 1
Write a program in PHP to check an integer whether it is odd or even.

Question 2
Write a program in PHP to check whether a number is positive or negative.

Question 3

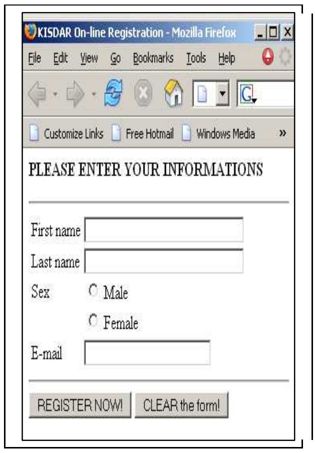

KISDAR On-line Registration - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Customize Links
Free Hotmail
Windows Media
PLEASE ENTER YOUR INFORMATIONS
First name
Last name
Sex
Male
Female
E-mail
REGISTER NOW!
CLEAR the form!

Figure 2

a. You are required to write the HTML code for the HTML document shown in Figure 2 above. Create a form named *formInfo* and send the data to a file named *processInfo.php* and the method is GET. Place the input components of the form inside a table and arrange them neatly. Give a name to each of the input component. “REGISTER NOW!” is the submit button and “CLEAR the form!” is the reset button.

b. Write a validation script in the *processInfo.php* file to make sure the user does not leave any input blank (user must key in all the information). If there is any invalid input or blank, inform the user by giving him/her messages.