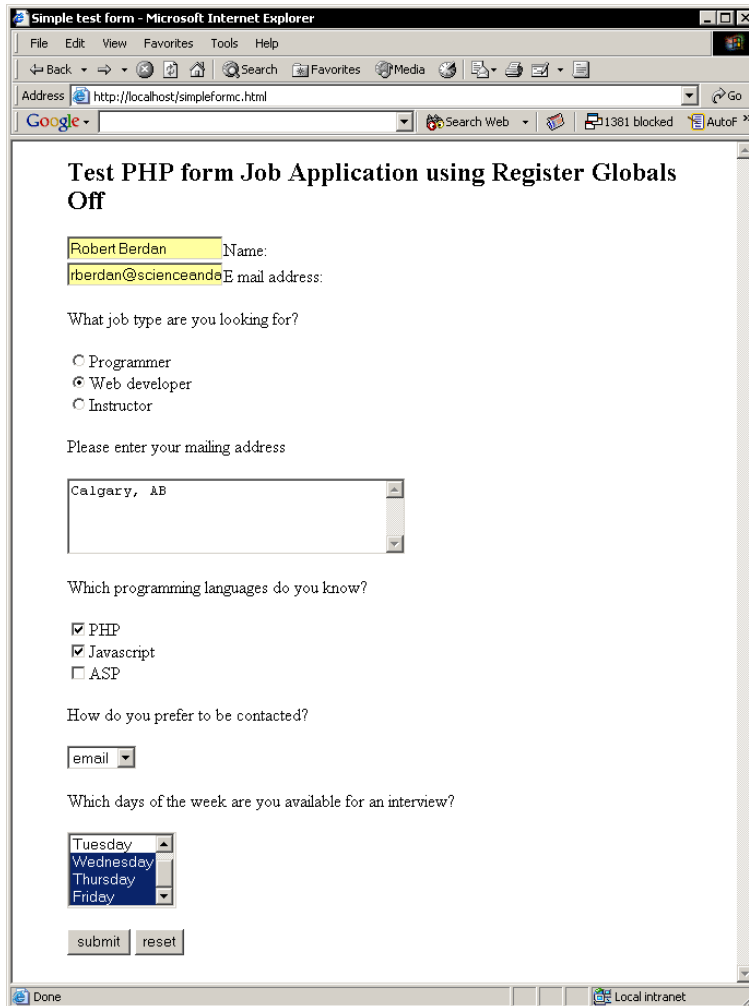


Creating Forms and processing your forms using a Mailto command or PHP – R. Berdan May 21, 2006

Forms are important, they permit the collection of specific information, they can be used to create password protected areas, shopping carts, guest books, allow users to upload files and more. Below is an example containing most of the different elements you might want to add to a form. You can add or delete elements to customize the form to suit your needs.



Simple test form - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://localhost/simpleformc.html

Google Search Web 1381 blocked AutoF

Test PHP form Job Application using Register Globals Off

Robert Berdan Name:
rberdan@scienceandE mail address:

What job type are you looking for?

Programmer
 Web developer
 Instructor

Please enter your mailing address

Calgary, AB

Which programming languages do you know?

PHP
 Javascript
 ASP

How do you prefer to be contacted?

email

Which days of the week are you available for an interview?

Tuesday
Wednesday
Thursday
Friday

submit reset

Done Local intranet

Sample form with an example of all elements See the code on the next page.

Form HTML code uses mailto: command to send email – no other programming required.

```
<html>
<head>
<title>Simple test form</title>
</head>
<body>

<blockquote>
<h2>Generic Form – modify title and code as needed </h2>

<form method="post" enctype="text/plain" action="mailto:youremail@company.com">
<input type="hidden" name="hidden1" value="Job Application for S&A">
<input type="text" name="name">Name:<br>
<input type="text" name="email">E mail address: <br>
<p>What job type are you looking for?</p>
<input type="radio" name="job" value="programmer">Programmer<br>
<input type="radio" name="job" value="webdeveloper">Web developer<br>
<input type="radio" name="job" value="instructor">Instructor<br><br>
<p>Please enter your mailing address</p>
<textarea rows="4" cols="36" name="address"></textarea><br><br>
<p>Which programming languages do you know?</p>
<input type="checkbox" name="choice1" value="PHP">PHP<br>
<input type="checkbox" name="choice2" value="Javascript">Javascript<br>
<input type="checkbox" name="choice3" value="ASP">ASP<br><br>
<p>How do you prefer to be contacted?</p>
<select name="contact">
<option value="email">email</option>
<option value="mail">mail</option>
<option value="phone">phone</option>
</select>
<br>
<br>
<p>Which days of the week are you available for an interview?</p>
<select name="week[]" Multiple>
<option value="Monday">Monday</option>
<option value="Tuesday">Tuesday</option>
<option value="Wednesday">Wednesday</option>
<option value="Thursday">Thursday</option>
<option value="Friday">Friday</option>
</select><br><br />

<input type="submit" name="submit" value="submit">
<input type="reset" name="reset" value="reset">
</form>
</blockquote>

</body>
```

Note using action=<mailto:emailaddress> requires the inclusion of **enctype="text/plain"** command. The output won't be fancy but this will work on most servers.

A better way to process your forms is to use a cgi script. These can be provided by your Internet provider if you simply copy and paste action=<http://URL/cgi-bin/program.pl> following their instructions. Alternatively you can write a PHP form script (see below) e.g. formprocessor.php and make the **action="formprocessor.php"**; the server must support PHP.

To test if your server supports PHP create the simple script below, upload it – then type URL of your site/test.php – if PHP is supported you will see a page with the PHP version and all the details. If it does not you will get a blank page, an error or nothing will happen.

Below is a sample PHP script, sentences after // are comments
Save this file into your web folder call it **formprocessor.php** in the same location as your form.html page.

```
<?php

// validate each variable from your form

$hidden1 = $_POST['hidden1'];
$name = $_POST['name'];
$email = $_POST['email'];
$job = $_POST['job'];
$address = $_POST['address'];
$choice1 = $_POST['choice1'];
$choice2 = $_POST['choice2'];
$choice3 = $_POST['choice3'];
$contact = $_POST['contact'];
$week = $_POST['week'];

// add date the form was submitted
$date = gmdate("M d Y");

// Thank the user after they submit the form
print "$date <br>\n";
print "<h2>Thank you $name we received your request and will get back to you shortly </h2>\n ";

// send data to a specified email address substitute your email address
$to = "youremail@whatevercompany.com ";
$subject = "Job Application";
$body = " Date: $date \n Name: $name \n E-mail address: $email \n Job Sought: $job \n Mailing
address: $address \n Programming languages: $choice1 $choice2 $choice3 \n Method of
Contact: $contact \n Hidden Data: $hidden1 \n Days available: $week[0] $week[1] $week[2]
$week[3] $week[4] \n\n";
mail($to,$subject, $body);

?>
```

Note: in order to use this script it must contain exactly the same variables e.g. name, address, email etc - if not change the code of the PHP script so they match the variables i.e. name="variable" in your HTML form.

Alternatively you can use a PHP generic script that works will almost any form – see Below. If you can type this code below without making any mistakes and save it as formprocessor.php include this file with your form and it will process your form.

PHP Generic Form Processor Script tutorial by R. Berdan Jan 25, 2005

Script by S.Herd modified by RB

1. Create a form to collect data with any form elements you like e.g. form.html

Set <form method=post action = "formloop.php">

2. Type in the php script below and save the file in the same directory as your form call it **formloop.php** If you want to make a custom thankyou with the persons name be sure one of the form text boxes has **name=name** to print out custom thankyou.

```
<?php
```

```
$to = "youremail@company.com";  
$subject = "Generic Form processor";  
$body = date("l dS of F Y h:i:s A") . "\n";
```

```
$Line = " "; // Variable to store to info in each form element  
$i=0; // counter  
$filename = "data.txt"; // file location for file object  
$fp = fopen($filename, "a"); // create pointer to the text file
```

```
$Line = each($_POST); // loops through each value until submit is reached
```

```
while($i< count($_POST) && $Line[key] != "submit")  
{  
    if($Line[value] != " ")  
    {  
        //print " <div>$Line[key] = $Line[value]</div>\n"; // prints all the variables to the screen  
        turn off except for testing  
        $body .= "$Line[key] = $Line[value] \n";  
    }  
    $Line = each($_POST);  
    $i++;  
}
```

```
print "<center><h2>Thank you $name for filling out our form</h2></center>";
```

```
?>
```

To change the subject heading just change the \$subject variable above.
Below is what the results should look like in your email.

```
Saturday 29th of January 2005 08:31:13 PM  
name = Donna Berdan  
email = donna@hotmail.com  
phone = 123-4567  
address = 6255-72nd ST. N.W.  
nextvariable etc
```

This script will work with almost any type of form and works with
register_globals=On or Off