Three basic ways to provide input to a PHP script

* GET
* POST
* Cookies

HTTP is the protocol used for communication between web clients and web servers, it allows clients to request webpages from servers.

The communicate with messages which have header and body

Header and body are separated by a blank line

|  |  |
| --- | --- |
| **GET** | **POST** |
| Appends any data to the URL via a query string | Includes data submitted via a form in the body of the HTTP request |
| GET / index.html?id=1000 HTTP/1.1  Host: www.educator.com | HTTP/1.1 200 OK  Date: Wed, 3 Aug 2011 23:49:23 GMT  Content-Length: 31  Content-Type: text/html  <html><body>Hello</body></html> |
| Access data $\_GET | Access data $\_POST |
| Simpler and less data in stream | More data in stream, but more security. For password transmission or updating databases  When a PHP script is requested, APACHE/PHP generates an HTTP response message whose header contains the necessary header and output script. |

|  |  |
| --- | --- |
| **Origin** | **Destiny: PostForm.php** |
| **<html>**  **<head>**  **<title>Using POST Method</title>**  **</head>**  **<body>**  **<h2>Using the POST Method</h2>**    <form action="PostForm.php" method="post">  First Name: <**input** type="text" name="firstName"/><br/>  Last Name: <**input** type="text" name="lastName"/><br/>  <br/>  <input type="submit" value="**Submit"**/>  **</form>**  **</body>**  **</html>** | **<html>**  **<head>**  **<title>Accessing POST</title>**  **</head>**  **<body>**    **<h2>Accessing</h2>**  <?php  $firstName = $\_POST['firstName'];  $lastName = $\_POST['lastName'];  echo "Welcome, $firstName $lastName! ";  ?>  **</body>**  **</html>** |

The HTTP GET method provides a query string to input data in the redirected page

<http://localhost/process.php?name1=value1>

**CREATE QUERY STRING MANUALLY**

In query strings the “ “ and [] are changed

El **urlencode** realiza la conversion a un querystring

* person=Joe Smith -> person=Joe+Smith
* grades [] = 99 -> grades %5B%5D= 99 “[“ is %5B and “]” is %5D

$sampleText = “Here is a sample”;

echo ‘<a href=”urlEncodingTest.php?sampleText=’;

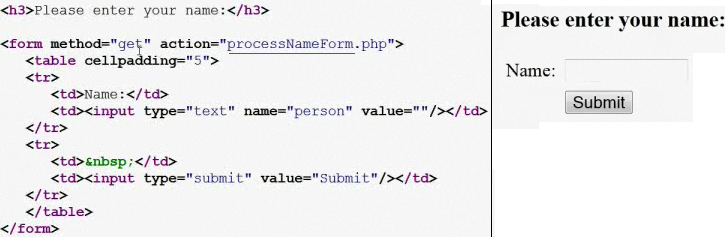
echo urlencode($sampleText);

echo ‘ “>Url Link</a>;

<a href=”urlEncodingTest.php?sampleText=Here+is+a+sample”>URL Link</a>

= = = = = = =

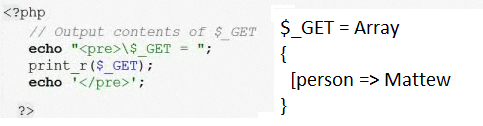
****



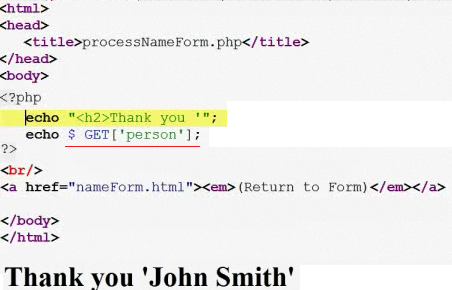
When submitting the form it redirects to processNameForm.php with a query string having the input in person because it is the only input find, in the processNameForm.php we can get the query string

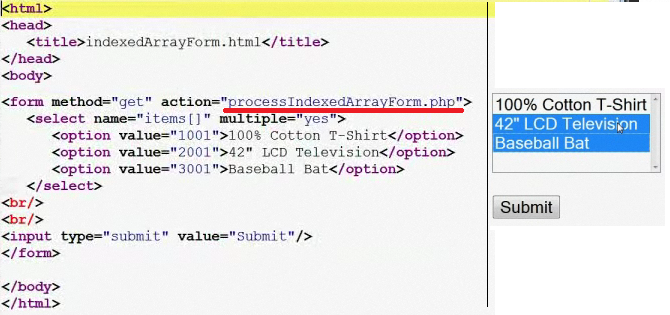
* The $\_GET from PHP holds ALL input data submitted to a PHP script via the GET method
* The print\_r($\_GET); will print in the page the values

http://localhost/intro2php/lecture\_12/processNameForm.php?person=Mattew

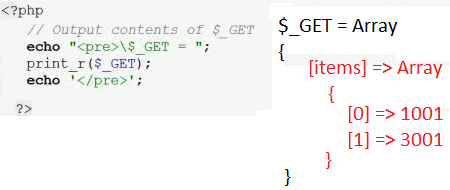


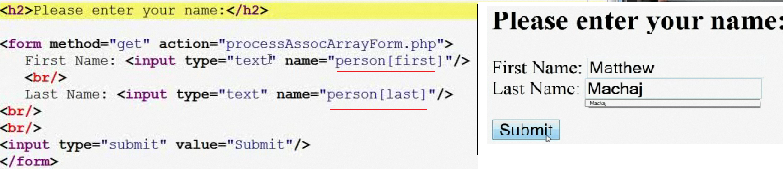
The **processNameForm.php** can show the value from the query string into the html



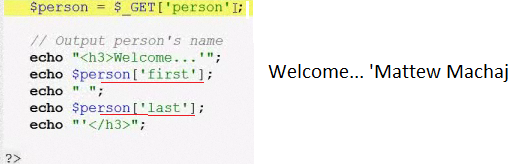


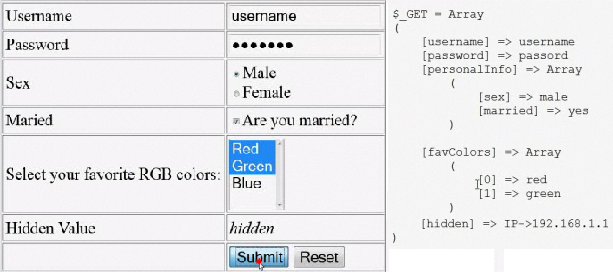
http://localhost/intro2php/lecture\_12/processIndexedArrayForm.php?**items[]=1001&items[]=3001**



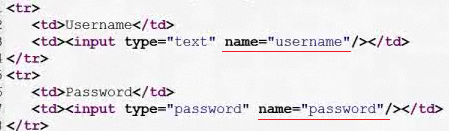


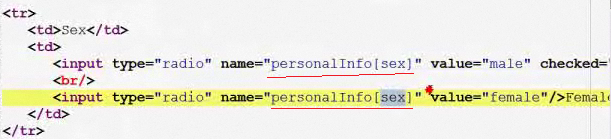
In page **processAssocArrayForm.php** we get the values

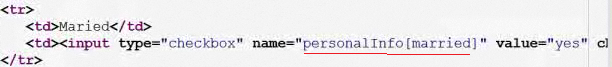


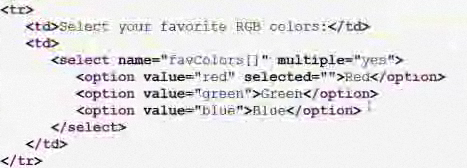


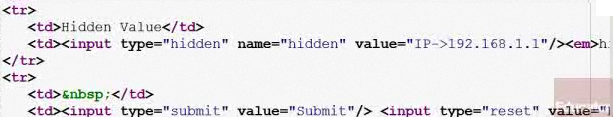








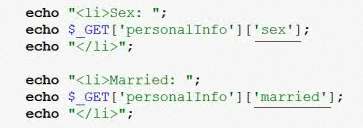




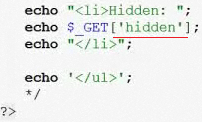
The receiving page **processEchoForm.php** submitted previously

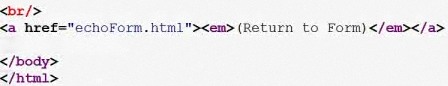


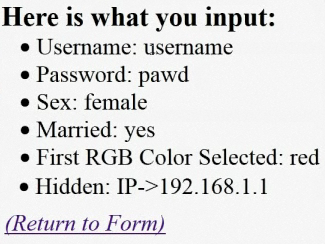












* To output a value in a numeric index echo “array value is $indexNum[1] “;
* To output a value in an associate array echo “array value is $assoc[key] “;
* The complex or curly brace syntax echo “array value is {$assoc[‘key’]} “;

