ITS351. Database Programming Laboratory
Semester 1, 2018

Lab. 3

HTML form and PHP

Instructor: L. V. Dang (Ph.D researcher)

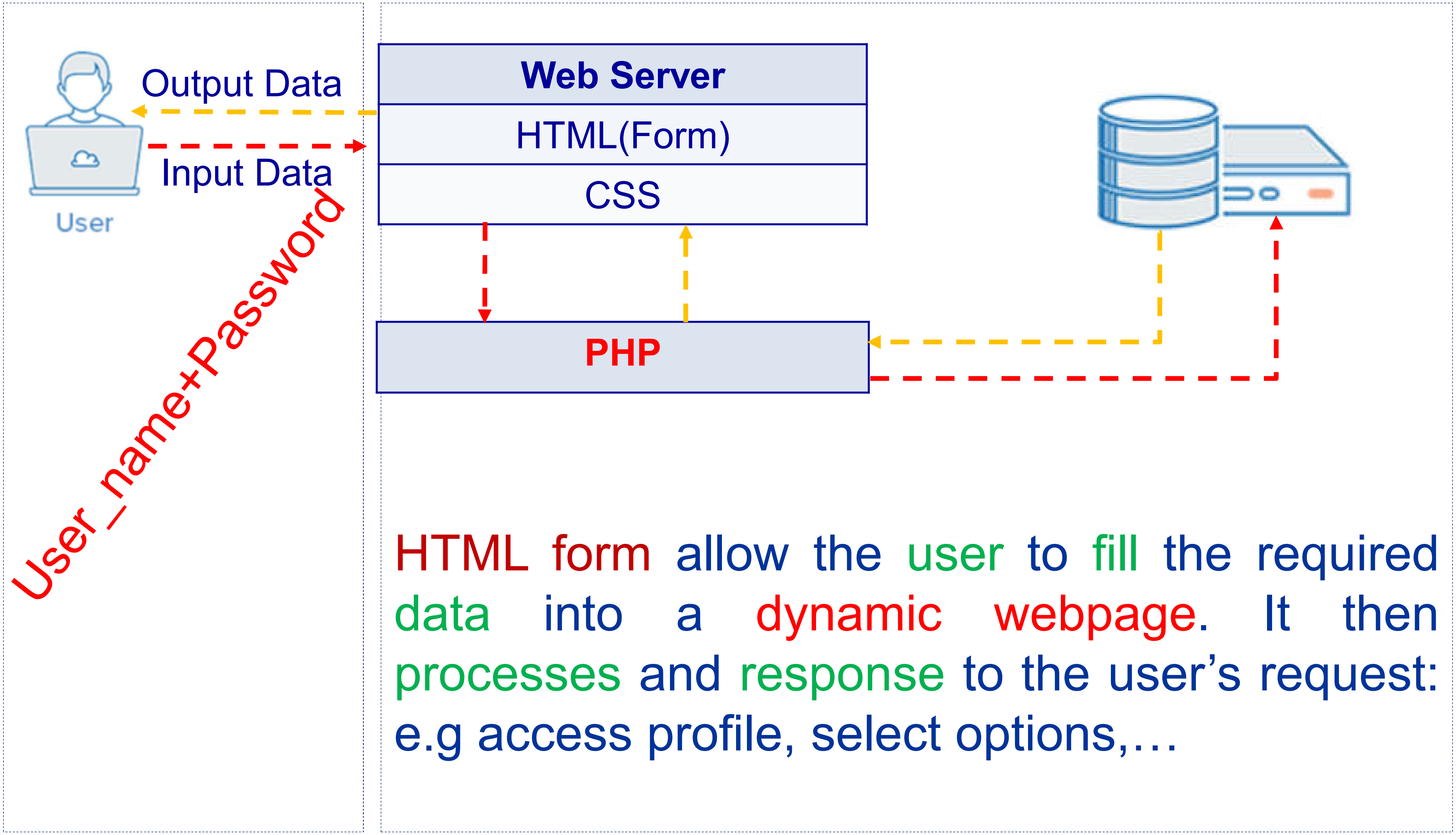
Title page

Contents

1. HTML form.
2. How it works?
   - Action page: action="url"
   - Sending method: method="GET", method="POST"
   - Data received and handle in .php page
3. Element of a form and Property of each element: name, value, size, width, height, disable (enable), multiple, rows, columns, etc.
   - Input type: text, textarea, password
   - Select type: list box, drop down (combo box)
   - Button type: radio, checkbox, file upload, submit, reset,
   - Image and file upload:

4. Exercise:

1. HTML Form

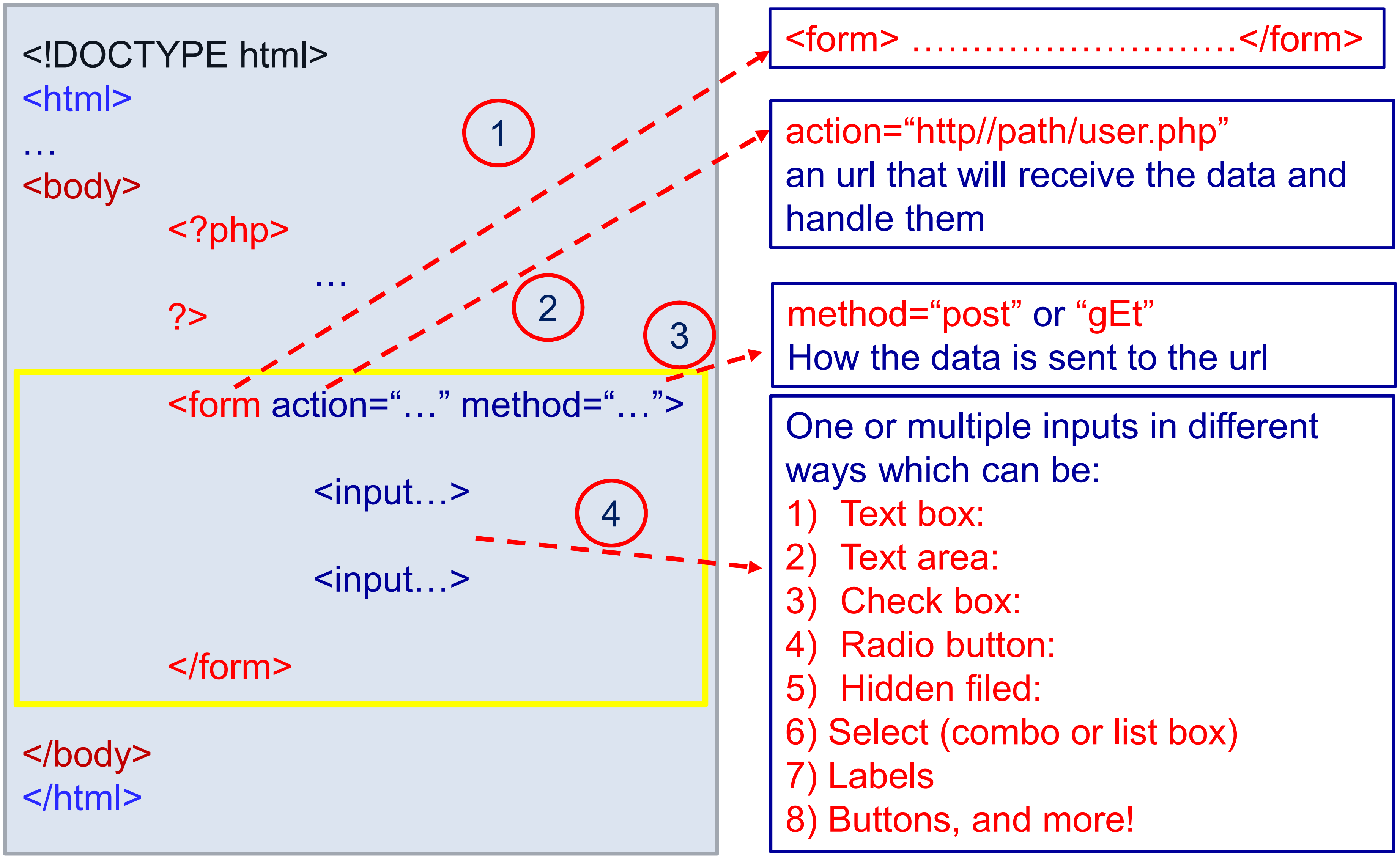


HTML form allow the user to fill the required data into a dynamic webpage. It then processes and response to the user's request: e.g access profile, select options,…

Client side

Host side

1. HTML form components



```
<!DOCTYPE html>
<html>
…
<body>
        <?php>
                …
        ?>
        <form action="…" method="…">
                <input…>
                <input…>
        </form>
</body>
</html>
```

<form> ………………………</form>

action="http//path/user.php"
an url that will receive the data and handle them

method="post" or "gEt"
How the data is sent to the url

One or multiple inputs in different ways which can be:

1) Text box:
2) Text area:
3) Check box:
4) Radio button:
5) Hidden filed:
6) Select (combo or list box)
7) Labels
8) Buttons, and more!

1. HTML form components

(2) Action and (3) Method

2. Action specifies the address (.php page) that the inputs will be sent to and handle there.

- If the action is omitted, the action page is the current .html/.php page
- Submitted result will open in a new browser page

```
<form action="/action_page.php" target="_blank">
```

- Submitted result will open in the current browser page (default)

```
<form action="/action_page.php" target="_self">
```

1. HTML form components

(2) Action and (3) Method

3. If method="GET" (default) the input data will appear on the url of the action page while method="POST" is not. So when should use GET and POST?

- GET method allow user to bookmark the result page while POST does not.
- In html form: method ="post" // case insensitive
- In .php form target: $x=$_GET['variable name in form']

  $x=$_POST['variable name in form']

  // case insensitive

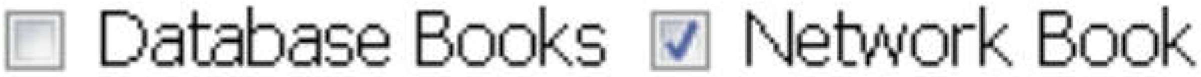
1. HTML form components

(4) Element of an html form

1) Text box: smith@gmail.com

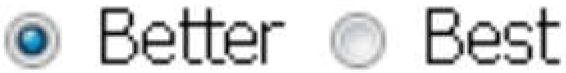
2) Text area: This article is about….
……………………………………………………………

3) Check box:



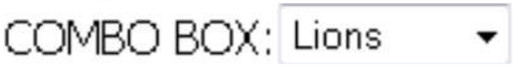
□ Database Books ☑ Network Book

4) Radio button:

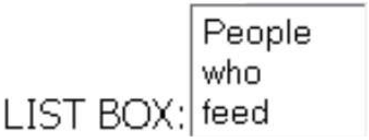


● Better ○ Best

5) Select (combo or list box)

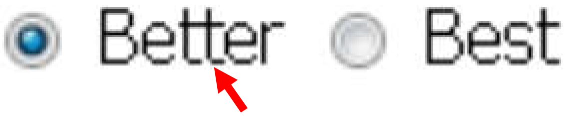


COMBO BOX: Lions ▾



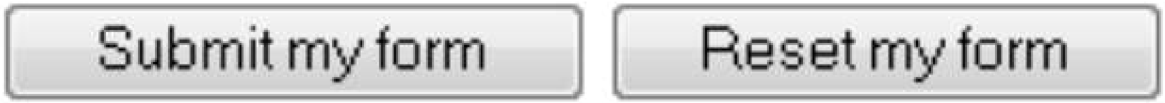
LIST BOX: People
who
feed

6) Labels



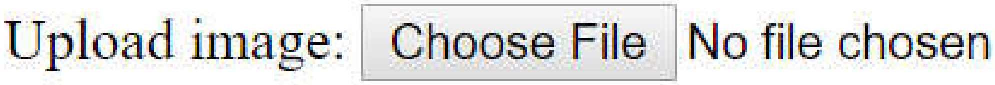
● Better ○ Best

7) Buttons



8) Hidden filed:

9) Uploading



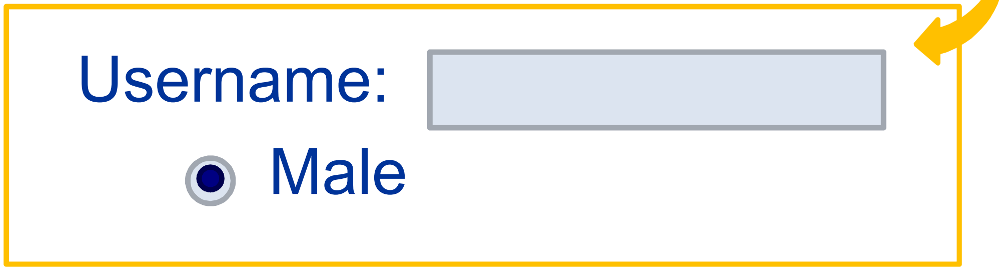
Upload image: Choose File No file chosen

2. How it work?

Form input page

- Different input type requires/has different properties. However, many of them usually requires: name and value property.
- For example,

```
<form action=“user.php” method=“POST”>
        Username: <input type="text" name="name" size="size"
                maxlength="length" value="value“ required><br>
        <input type="checkbox" name="name" value="value"
                checked="checked"> Male <br>
</form>
```



Username:

● Male

2. How it work?

Form input page

```
<!DOCTYPE>
<html>
    <head> <title>My Form</title></head>
<body>

<form action=“handle.php” method=“PosT” >
          Username: <input type="text“ name=“username"><br>
          Password: <input type="text“ name=“pass"><br>
          <input type="submit" name=“something” value=“register">
</form>

</body>
</html>
```

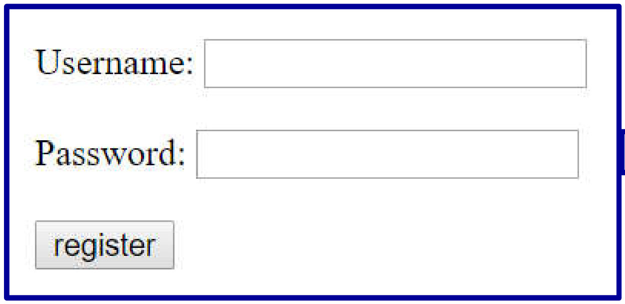
Save this html page as: ITS351-html-form.html

2. How it work?

Target .php page

```
<?php
// receive variable from toward url
        $n=$_POST[“username”]; // will receive $n as username
        $p=$_POST[“pass”]; //if method in form is GET, use $_GET
        //Assume available data is: username=smith, pass=000;
        if ($n==“smith” && $p==000)
                echo “Welcome Smith!”;
        else
                echo “Name or Password is mismatch! Try again!”;
?>
```

Save this html page as:ITS351-form-handle.php



```
Welcome Smith!
```

```
Name or Password is mismatch! Try again!
```

3. Input method

3.1-3.2 Text box & Text area

```
<html>
<form action="handle.php" method="PosT" >
         …

Textbox <input type="text" name="title" size=20 width=20 (is default)
value="your title">


Text area <textarea name="content" rows=7 cols=56 > Text area
can be allocated outside the form, but still consider as a part of
html form </textarea>

         …
</form>
```

3. Input method

3.3-3.4 Check button & Radio button

```
<html>
<form action=“handle.php” method=“PosT” >
          …

Married <input type=“checkbox” name=“status” value=“married”>
Single  <input type=“checkbox” name=“status” value=“single” checked>

Student <input type=“radio” name=“career” value=“student”>
Other <input type=“radio” name=“career” value=“other”>


          …
</form>
</html>
```

3. Input method

3.5 Select Method: Combo box

```
<html>
<form action=“handle.php” method=“PosT” >
        Username: <input type="text“ name=“username"><br>
        Password: <input type="text“ name=“pass"><br>

        Combo Box   <select name=“gender”>
                <option value=“M”>Male</option>
                <option value=“F”>Female</option>
                <option value=“O”>Other</option>
        </select><br>

        <input type="submit" name=“log” value=“register">
</form>
</html>
```

3. Input method

3.5 Select Method: List box

```
<html>
<form action="handle.php" method="PosT" >

        .....

        List Box <select name="gender_list []" size=3 multiple>
                <option value="M">Male</option>
                <option value="F">Female</option>
                <option value="O" selected>Other</option>
        </select><br>

        <input type="submit" name="log" value="register">
</form>
</html>
```

3 Input method

3.6-3.7 Sending Method: label, submit & reset button

```
<html>
<form action=“handle.php” method=“PosT” >
          …

<label> Submit </label>
<input type=“submit” name=“send” value=“sending”>

<input type=“image” src=“path/name.png” width=“40” alt=“…”>


<button type=“reset” value=“clean_all”> Reset </button>
          …
</form>
</html>
```

3. Input method

on the target page

```
<?php
        $n=$_POST[“username”]; $p=$_POST[“pass”];
        $g=$_POST[‘gender’]; $log=$_POST[“log”];
//Assume available data is: username=smith, pass=000, gender=M;
        $gl=$_POST[‘gender_list’]; // gl now is an array
        if ($n==“smith” && $p==000 && $g==“M”)
                {
                echo “Welcome Smith!”;
                display_profile($n,$p,$g,$gl,$log);
                }
        elseif (isset($n) && $n!=“smith” or isset($p) && $p!=000)
                echo “Name or Password is mismatch! Try again!”;
        else
                echo “Cannot Access. Sorry!”;
?>
```

3. Input method

3.8 hidden value

Why?

- In many cases, we will need some variables to use in our webserver, but it is unnecessary to the user.
- Hidden type can be used in these situations.
- Hidden variable can be constant or dynamic

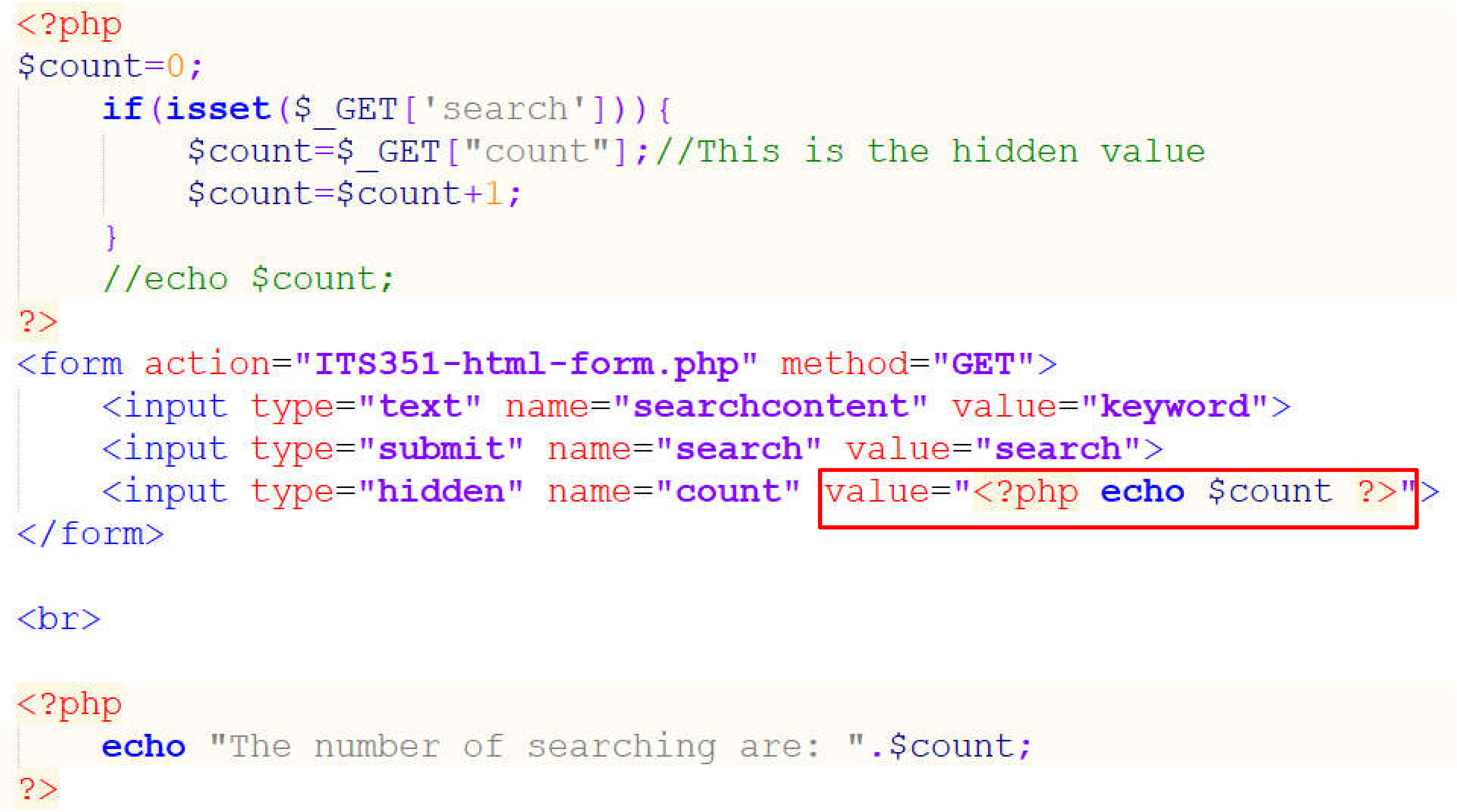
EX1

```
<form action="ITS351-form-get.php" method="GET">
        <input type="text" name="searchbox" value="keyword">
        <input type="submit" name="search" value="search">
        <input type="hidden" name="lang", value=“Thai">
</form>
```

3. Input method

3.8 hidden value

EX2: Using as dynamic variable

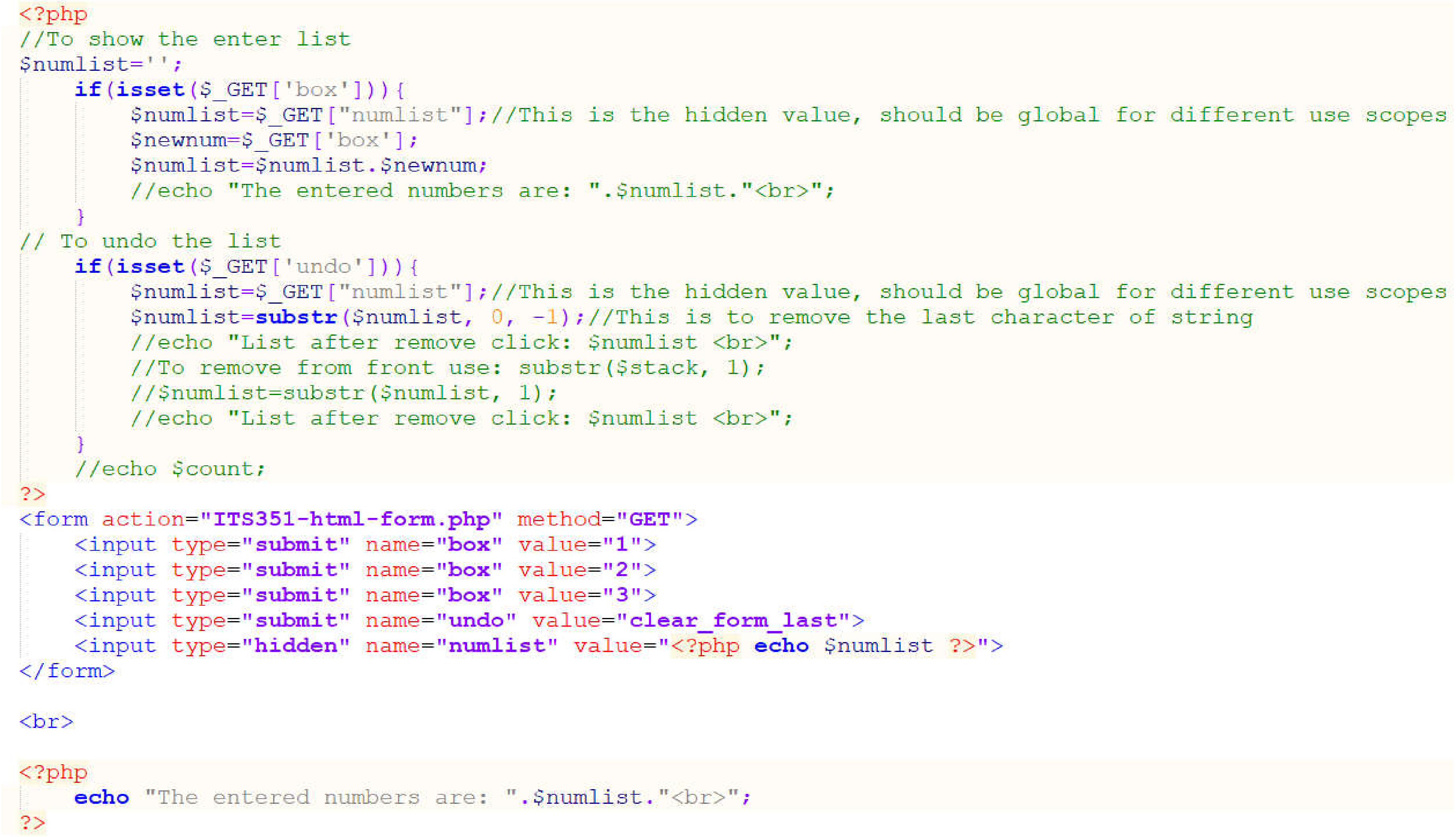


```
<?php
$count=0;
    if(isset($_GET['search'])){
        $count=$_GET["count"];//This is the hidden value
        $count=$count+1;
    }
    //echo $count;
?>
<form action="ITS351-html-form.php" method="GET">
    <input type="text" name="searchcontent" value="keyword">
    <input type="submit" name="search" value="search">
    <input type="hidden" name="count" value="<?php echo $count ?>">
</form>

<br>

<?php
    echo "The number of searching are: ".$count;
?>
```

3. Input method

3.8 hidden value

EX3: Using as dynamic variable



```
<?php
//To show the enter list
$numlist='';
    if(isset($_GET['box'])){
        $numlist=$_GET["numlist"];//This is the hidden value, should be global for different use scopes
        $newnum=$_GET['box'];
        $numlist=$numlist.$newnum;
        //echo "The entered numbers are: ".$numlist."<br>";
    }
// To undo the list
    if(isset($_GET['undo'])){
        $numlist=$_GET["numlist"];//This is the hidden value, should be global for different use scopes
        $numlist=substr($numlist, 0, -1);//This is to remove the last character of string
        //echo "List after remove click: $numlist <br>";
        //To remove from front use: substr($stack, 1);
        //$numlist=substr($numlist, 1);
        //echo "List after remove click: $numlist <br>";
    }
    //echo $count;
?>
<form action="ITS351-html-form.php" method="GET">
    <input type="submit" name="box" value="1">
    <input type="submit" name="box" value="2">
    <input type="submit" name="box" value="3">
    <input type="submit" name="undo" value="clear_form_last">
    <input type="hidden" name="numlist" value="<?php echo $numlist ?>">
</form>

<br>

<?php
    echo "The entered numbers are: ".$numlist."<br>";
?>
```

3. Input method

3.9 Input Method: upload image & file

Step 1- In your "php.ini" file, search for the file_uploads directive, and set it to On: `file_uploads = On`

Step 2:

```
<form action="upload.php"method="post"enctype="multipart/form-data">
    Select image to upload:
    <input type="file" name="fileToUpload" id="fileToUpload">
    <input type="submit" value="Upload Image" name="submit">
    //<botton type="submit" value="Upload Image" name="submit">
</form>
```

Some rules to follow for the HTML form above:

1-Make sure that the form uses method="post"

2-The form also needs the following attribute: enctype="multipart/form-data". It specifies which content-type to use when submitting the form

Without the requirements above, the file upload will not work

3. Input method

3.9 Input Method: upload image & file

Step 3:

```
<?php
$target_dir = "uploads/";
$target_file = $target_dir.basename($_FILES["fileToUpload"]["name"]);

//$_FILES["fileToUpload"] similar as $_POST, it give an array with:
name=>…,type=…,tmp_name=>…,error=>…, size=>…
// basename($path,prefix) give the filename-not whole path (of the
prefix if any)
if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file))
   {
        echo "The file ". basename( $_FILES["fileToUpload"]["name"]). " has
               been uploaded.";
   }
else
    {
       echo "Sorry, there was an error uploading your file.";
    }
?>
```

4. Exercises