CS100: Software Tools & Technologies Lab I

PHP Forms and Files

Vishwesh Jatala

Assistant Professor

Department of EECS

Indian Institute of Technology Bhilai

vishwesh@iitbhilai.ac.in

2022-23 W

File Handling

- File Handling
  - Open and Close
  - Read and Modify
  - Directory operations

Open/Close a File

- A file is opened with fopen() as a “stream”, and PHP returns a ‘handle’ to the file that can be used to reference the open file in other functions.
- Each file is opened in a particular **mode**.
- A file is closed with fclose() or when your script ends.

File Open Modes

‘r’	Open for reading only. Start at beginning of file.
‘r+’	Open for reading and writing. Start at beginning of file.
‘w’	Open for writing only. Remove all previous content, if file doesn’t exist, create it.
‘a’	Open writing, but start at END of current content.
‘a+’	Open for reading and writing, start at END and create file if necessary.

File Open/Close Example

```
<?php
// open file to read
$toread = fopen('some/file.ext','r');
// open (possibly new) file to write
$towrite = fopen('some/file.ext','w');
// close both files
fclose($toread);
fclose($towrite);
?>
```

Now what..?

- If you open a file to read, you can use more in-built PHP functions to read data..
- If you open the file to write, you can use more in-built PHP functions to write..

Reading Data

- There are two main functions to read data:
- **`fgets($handle)`**
  - function returns a line from an open file.
- **`fread($handle,$bytes)`**
  - Reads up to $bytes of data, stops at EOF.

Reading Data

- We need to be aware of the End Of File (EOF) point..
- `feof($handle)`
  - Whether the file has reached the EOF point. Returns true if have reached EOF.

Data Reading Example

```
$handle = fopen('people.txt', 'r');

while (!feof($handle)) {
   echo fgets($handle);
   echo '<br />';
   }

fclose($handle);
```

File Open shortcuts..

- There are two ‘shortcut’ functions that don’t require a file to be opened:
- **`$lines = file($filename)`**
  - Reads entire file into an array with each line a separate entry in the array.
- **`$str = file_get_contents($filename)`**
  - Reads entire file into a single string.

Writing Data

- To write data to a file use:
- `fwrite($handle,$data)`
  - Write $data to the file.

Data Writing Example

```php
$handle = fopen('people.txt', 'a');

fwrite($handle, “\nFred:Male”);

fclose($handle);
```

Other File Operations

- Delete file
  - `unlink('filename');`
- Rename (file or directory)
  - `rename('old name', 'new name');`
- Copy file
  - `copy('source', 'destination');`
- And many, many more!
  - www.php.net/manual/en/ref.filesystem.php

Dealing With Directories

- Open a directory
  - `$handle = opendir('dirname');`
    - `$handle` 'points' to the directory
- Read contents of directory
  - `readdir($handle)`
    - Returns name of next file in directory
    - Files are sorted as on filesystem
- Close a directory
  - `closedir($handle)`
    - Closes directory 'stream'

Directory Example

```
$handle = opendir('./');

while(false !== ($file=readdir($handle)))
 {
 echo "$file<br />";
 }

closedir($handle);
```

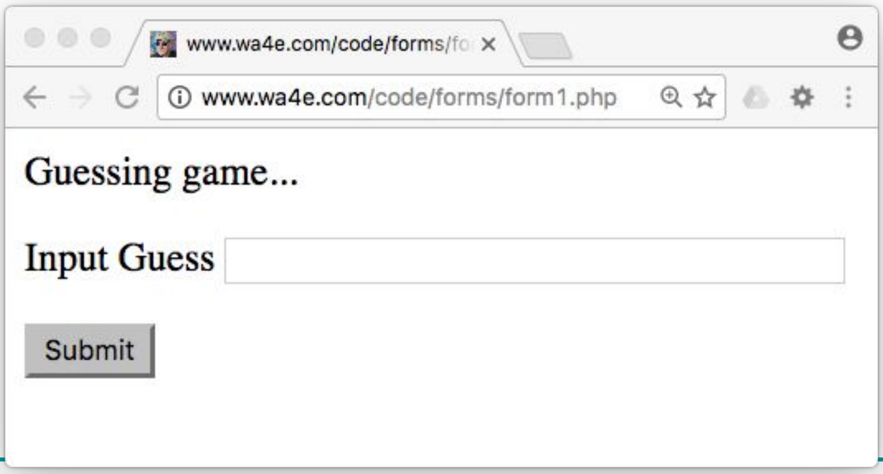
Other Directory Operations

- Get current directory
  - `getcwd()`
- Change Directory
  - `chdir('dirname');`
- Create directory
  - `mkdir('dirname');`
- Delete directory (MUST be empty)
  - `rmdir('dirname');`
- And more!
  - www.php.net/manual/en/ref.dir.php

PHP Handling Forms

Forms – User Input / Action

```
<p>Guessing game...</p>
<form>
   <p><label for="guess">Input Guess</label>
   <input type="text" name="guess" id="guess"/></p>
   <input type="submit"/>
</form>
```



Guessing game...

Input Guess

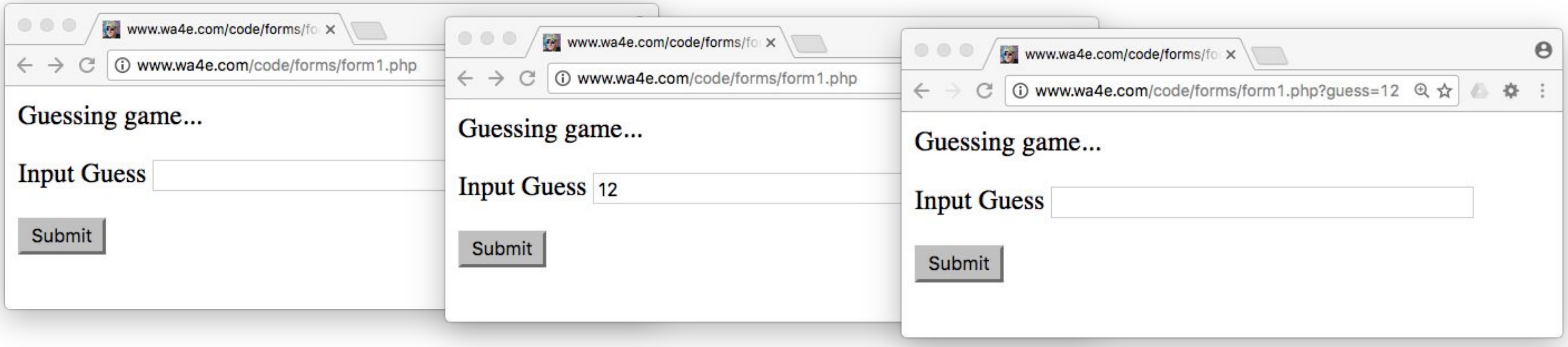
Forms GET vs. POST

Two ways the browser can send parameters to the web server

- GET - Parameters are placed on the URL which is retrieved.
- POST - The URL is retrieved and parameters are appended to the request in the HTTP connection.

Forms Submit Data

```
<p>Guessing game...</p>
<form>
   <p><label for="guess">Input Guess</label>
   <input type="text" name="guess"
id="guess"/></p>
   <input type="submit"/>
</form>
```



www.wa4e.com/code/forms/form1.php

Guessing game...

Input Guess

Submit

www.wa4e.com/code/forms/form1.php

Guessing game...

Input Guess 12

Submit

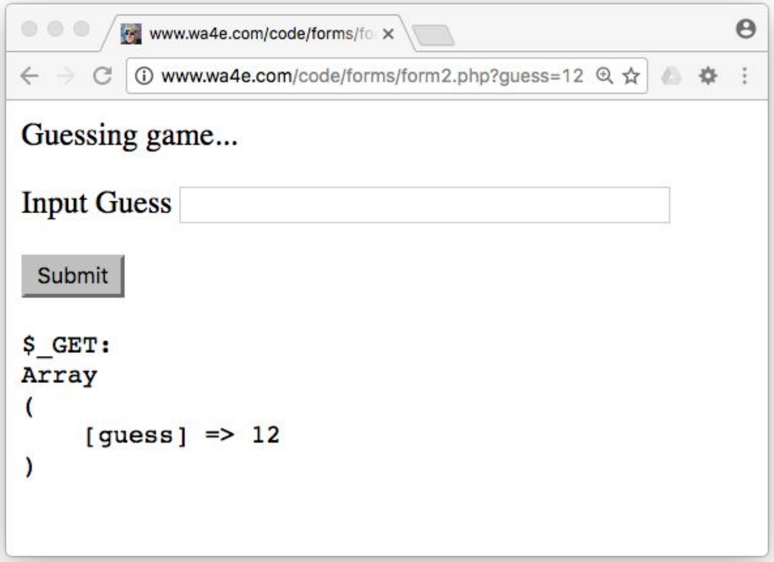
www.wa4e.com/code/forms/form1.php?guess=12

Guessing game...

Input Guess

Submit

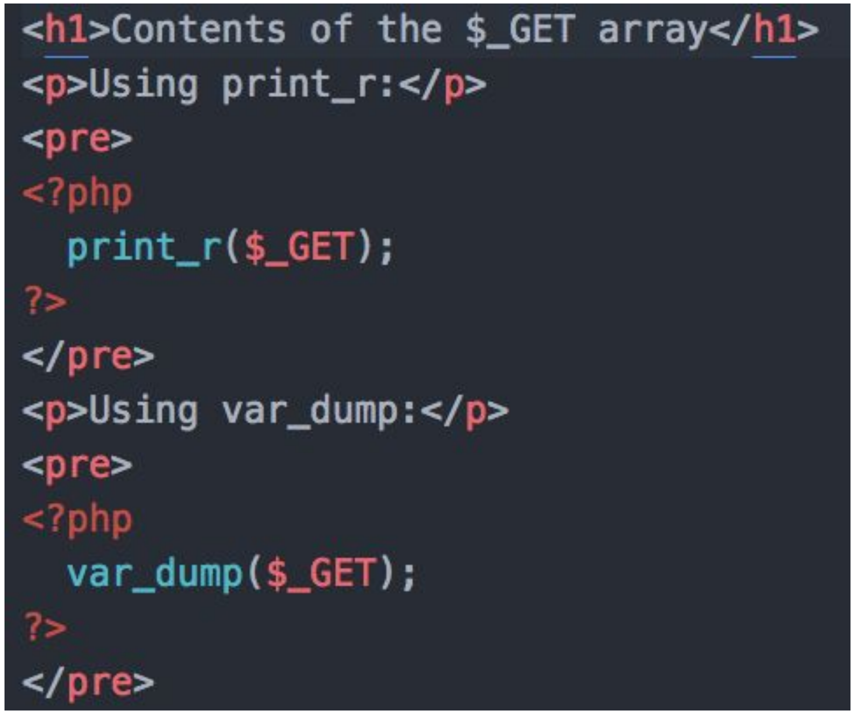
```
<p>Guessing game...</p>
<form>
   <p><label for="guess">Input Guess</label>
   <input type="text" name="guess" id="guess"/></p>
   <input type="submit"/>
</form>
<pre>
$_GET:
<?php
   print_r($_GET);
?>
</pre>
```


www.wa4e.com/code/forms/fo
www.wa4e.com/code/forms/form2.php?guess=12
Submit

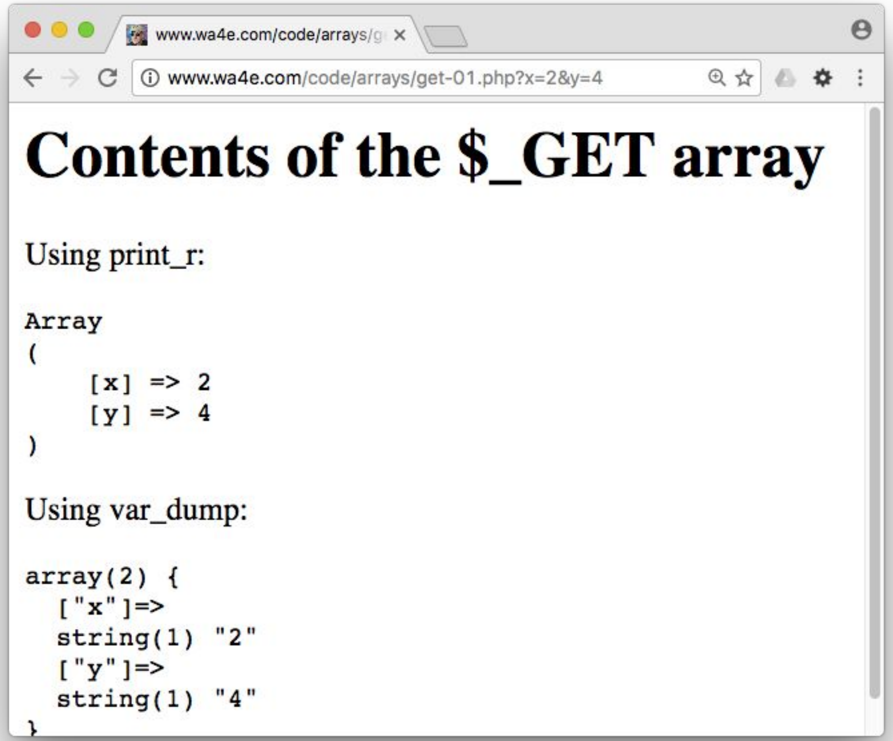
Guessing game...

Input Guess

```
$_GET:
Array
(
    [guess] => 12
)
```



```
<h1>Contents of the $_GET array</h1>
<p>Using print_r:</p>
<pre>
<?php
  print_r($_GET);
?>
</pre>
<p>Using var_dump:</p>
<pre>
<?php
  var_dump($_GET);
?>
</pre>
```



www.wa4e.com/code/arrays/get-01.php?x=2&y=4

Contents of the $_GET array

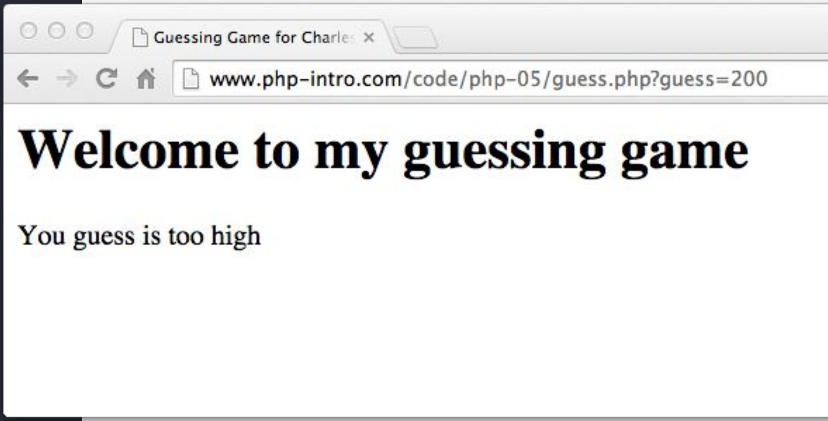
Using print_r:

```
Array
(
    [x] => 2
    [y] => 4
)
```

Using var_dump:

```
array(2) {
  ["x"]=>
  string(1) "2"
  ["y"]=>
  string(1) "4"
}
```

```
<html>
<head>
<title>Guessing Game for Charles Severance</title>
</head>
<body>
<h1>Welcome to my guessing game</h1>
<p>
<?php
  if ( ! isset($_GET['guess']) ) {
    echo("Missing guess parameter");
  } else if ( strlen($_GET['guess']) < 1 ) {
    echo("Your guess is too short");
  } else if ( ! is_numeric($_GET['guess']) ) {
    echo("Your guess is not a number");
  } else if ( $_GET['guess'] < 42 ) {
    echo("Your guess is too low");
  } else if ( $_GET['guess'] > 42 ) {
    echo("Your guess is too high");
  } else {
    echo("Congratulations - You are right");
  }
?>
</p>
</body>
</html>
```

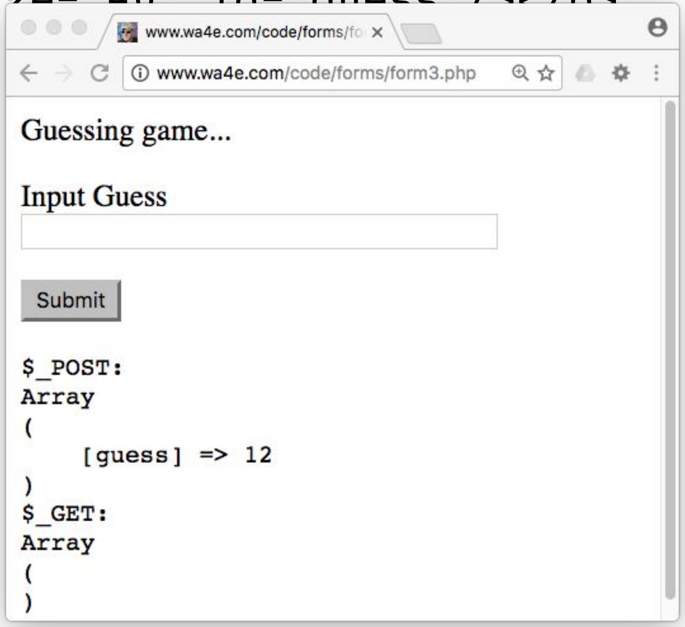


Welcome to my guessing game

You guess is too high

Forms Post Method

```
<p>Guessing game...</p>
<form method="post">
   <p><label for="guess">Input Guess</label>
   <input type="text" name="guess" size="40" id="guess"/></p>
   <input type="submit"/>
</form>
<pre>
$_POST:
<?php
   print_r($_POST);
?>
$_GET:
<?php
   print_r($_GET);
?>
</pre>
```



Guessing game...

Input Guess

```
$_POST:
Array
(
    [guess] => 12
)
$_GET:
Array
(
)
```

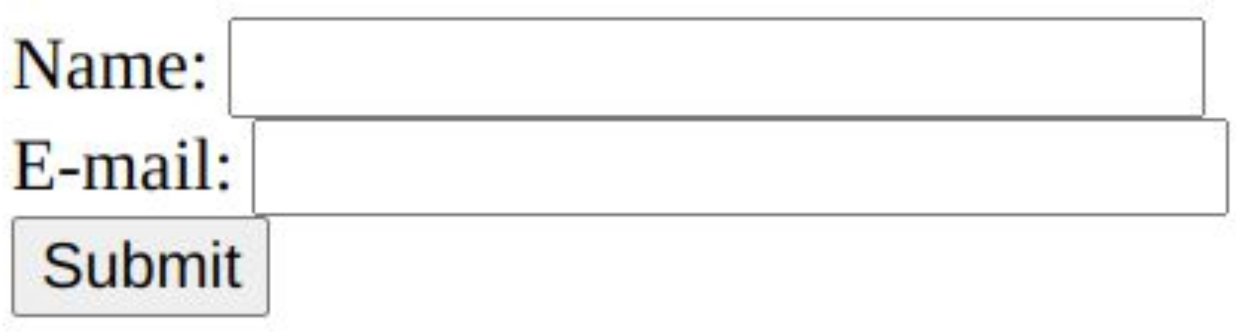
Transfer Data to Another Page

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

Transfer Data to Another Page



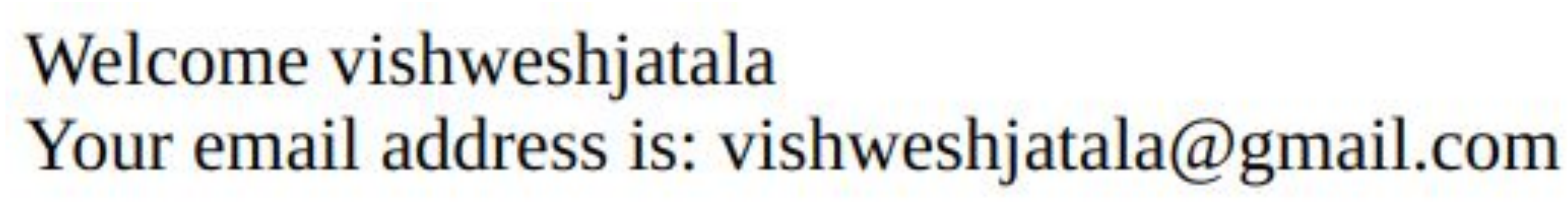
Transfer Data to Another Page

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

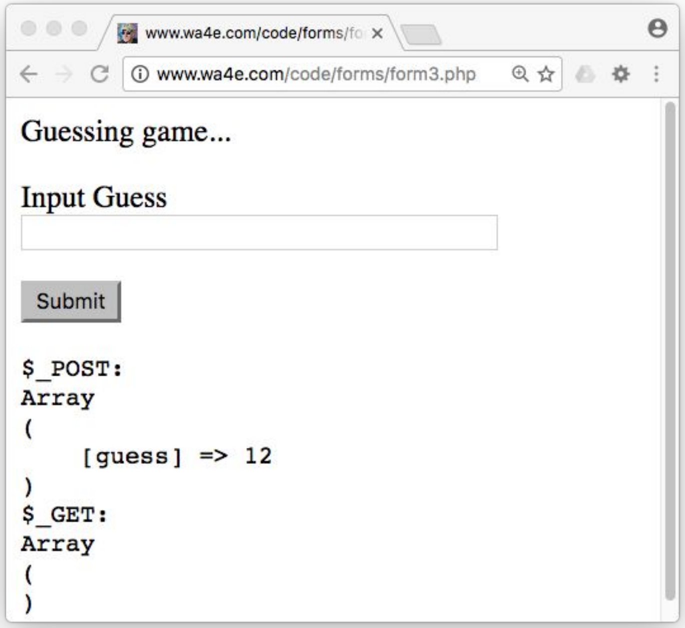
Transfer Data to Another Page



```
Welcome vishweshjatala
Your email address is: vishweshjatala@gmail.com
```

PHP Get and Post

```
<p>Guessing game...</p>
<form method="post">
   <p><label for="guess">Input Guess</label>
   <input type="text" name="guess" size="40" id="guess"/></p>
   <input type="submit"/>
</form>
<pre>
$_POST:
<?php
   print_r($_POST);
?>
$_GET:
<?php
   print_r($_GET);
?>
</pre>
```

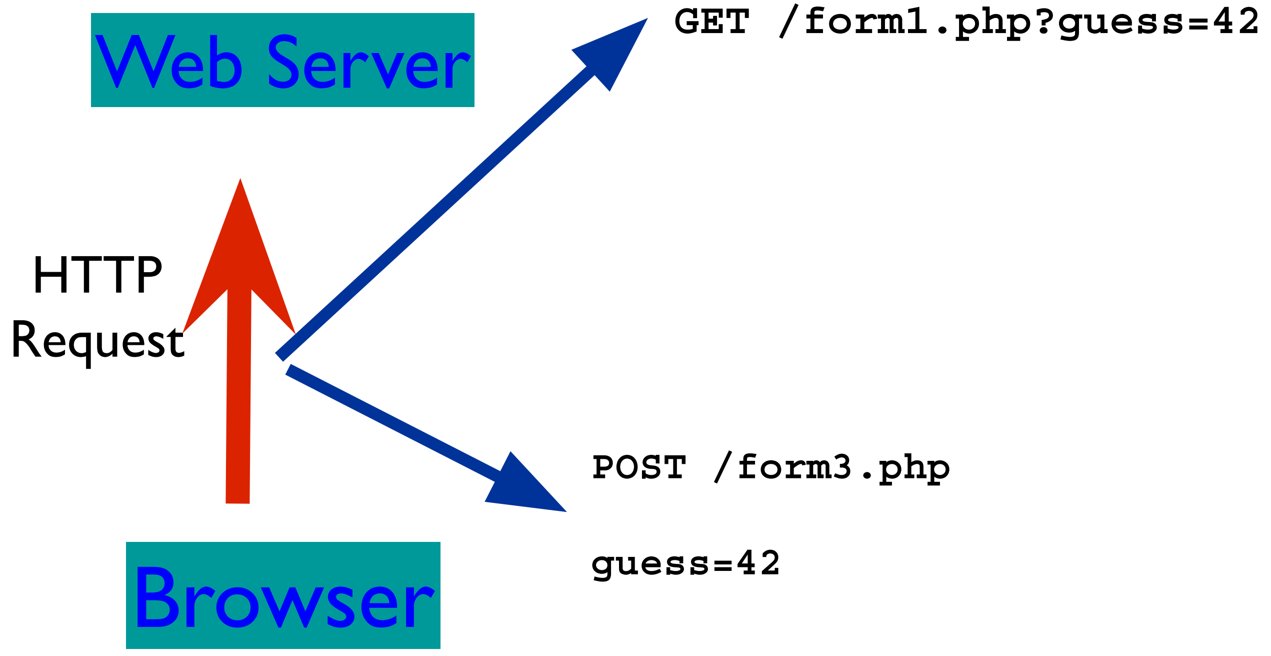


Guessing game...

Input Guess

```
$_POST:
Array
(
    [guess] => 12
)
$_GET:
Array
(
)
```

Passing Parameters to The Server



```
GET /form1.php?guess=42
```

```
POST /form3.php

guess=42
```

<input type="text" name="guess" id="yourid" />

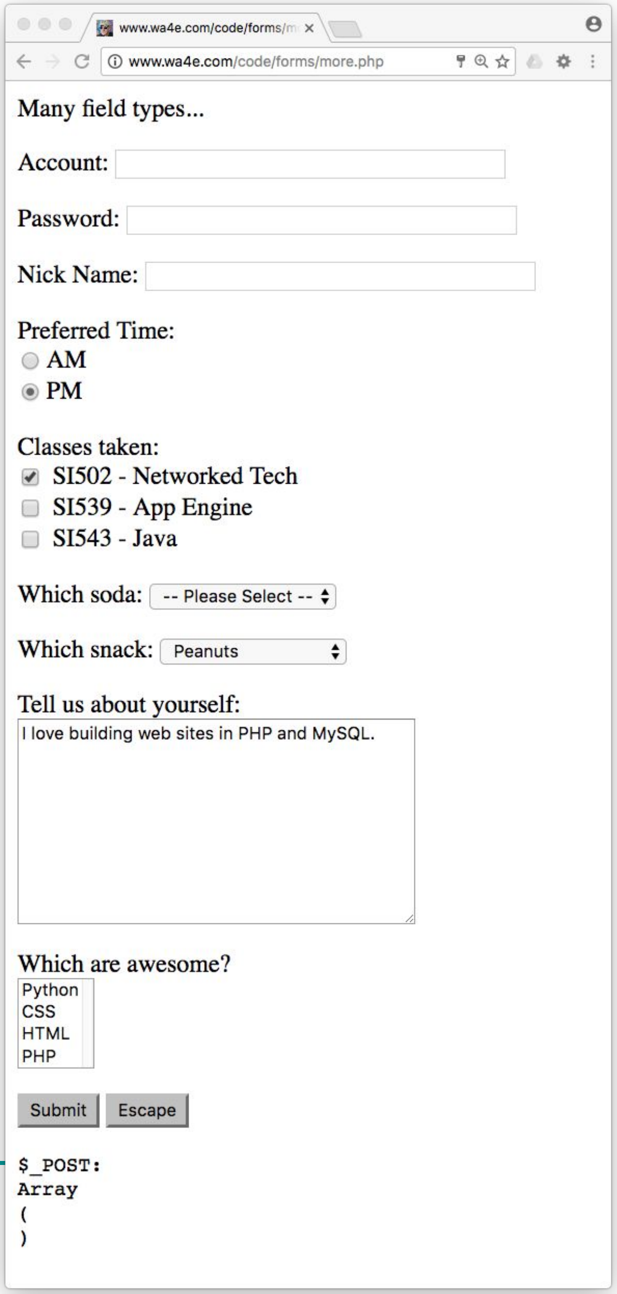
Rules of the POST/GET Choice

- POST is used when data needs to be hidden -- sensitive information
- GET is used when you are reading or searching things.
- Web search spiders will follow GET URLs but generally not POST URLs.
- GET has an upper limit of the number of bytes of parameters and values (think about 2K).

Form Input Types

Other Input Types

- Text
- Password
- Radio Button
- Check Box
- Select / Drop-Down
- Textarea



Many field types...

Account:

Password:

Nick Name:

Preferred Time:
- AM
- [✓ PM]

Classes taken:
- [✓] SI502 - Networked Tech
- SI539 - App Engine
- SI543 - Java

Which soda: -- Please Select --

Which snack: Peanuts

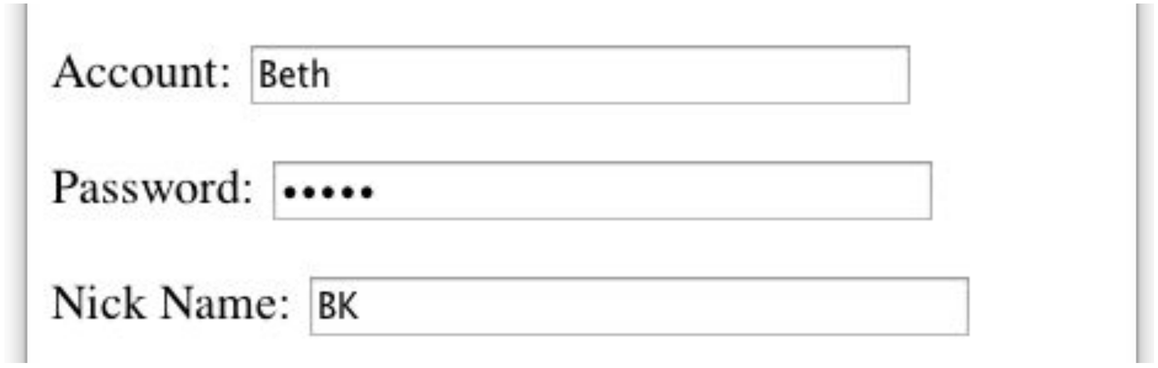
Tell us about yourself:
I love building web sites in PHP and MySQL.

Which are awesome?
Python
CSS
HTML
PHP

```
$_POST:
Array
(
)
```

Input as Text

```
<p>Many field types...</p>
<form method="post" action="more.php">
   <p><label for="inp01">Account:</label>
   <input type="text" name="account" id="inp01" size="40" ></p>
   <p><label for="inp02">Password:</label>
   <input type="password" name="pw" id="inp02" size="40" ></p>
   <p><label for="inp03">Nick Name:</label>
   <input type="text" name="nick" id="inp03" size="40" ></p>
```



Account: Beth

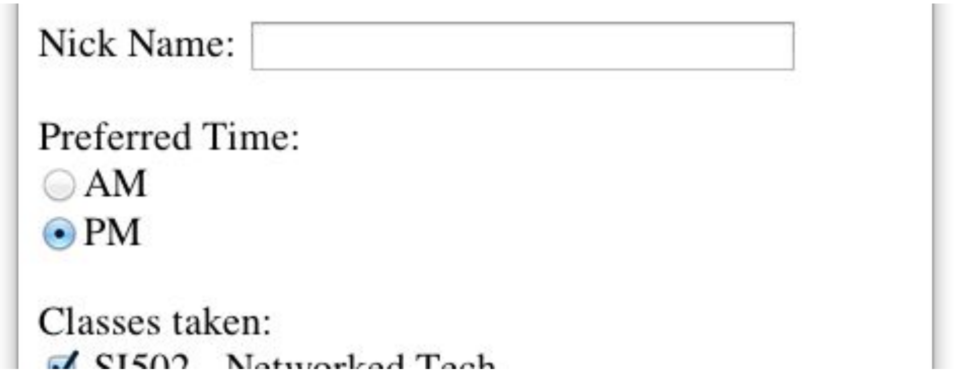
Password: •••••

Nick Name: BK

```
$_POST:
Array
(
    [account] => Beth
    [pw] => 12345
    [nick] => BK
    [when] => pm
  ...
)
```

Input as Radio Button

```
<p>Preferred Time:<br/>
 <input type="radio" name="when" value="am">AM<br>
 <input type="radio" name="when" value="pm" checked>PM
 </p>
```



Nick Name:

Preferred Time:
○ AM
● PM

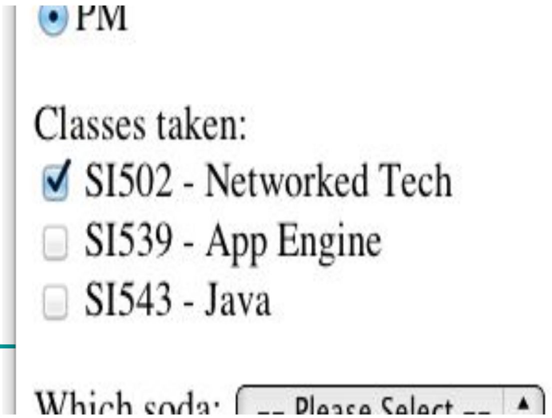
Classes taken:
✓ SI502 – Networked Tech

```
$_POST:
Array(
   ...
      [nick]  => BK
      [when]  => pm
      [class]  => si502
   ...
)
```

Input as Checkbox

```
   <p>Classes taken:<br/>
    <input type="checkbox" name="class1" value="si502"
checked>
        SI502 - Networked Tech<br>
    <input type="checkbox" name="class2" value="si539">
        SI539 - App Engine<br>
    <input type="checkbox" name="class3">
        SI543 - Java<br>    </p>
```

```
$_POST:
Array(
  ...
    [when] => pm
    [class1] => si502
    [soda] => 0
  ...
)
```



● PM

Classes taken:

✓ SI502 - Networked Tech

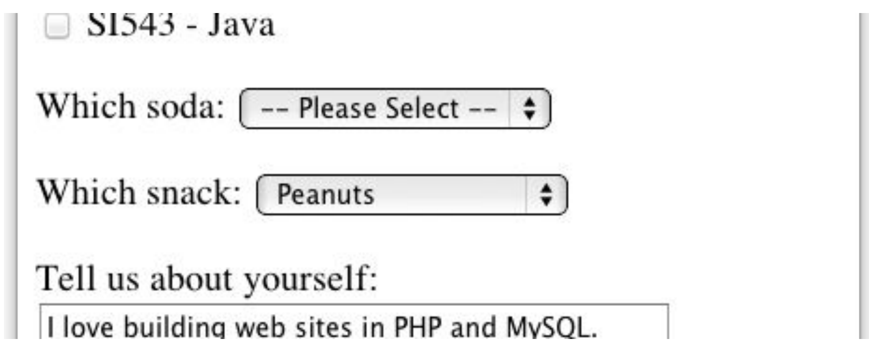
☐ SI539 - App Engine

☐ SI543 - Java

Which soda: -- Please Select --

```
$_POST:
Array(
  ...
    [when] => pm
    [class3] => on
    [soda] => 0
  ...
)
```

```
<p><label for="inp06">Which soda:
 <select name="soda" id="inp06">
   <option value="0">-- Please Select --</option>
   <option value="1">Coke</option>
   <option value="2">Pepsi</option>
   <option value="3">Mountain Dew</option>
   <option value="4">Orange Juice</option>
   <option value="5">Lemonade</option>
 </select>
</p>
```



SI543 - Java

Which soda: -- Please Select --

Which snack: Peanuts

Tell us about yourself:

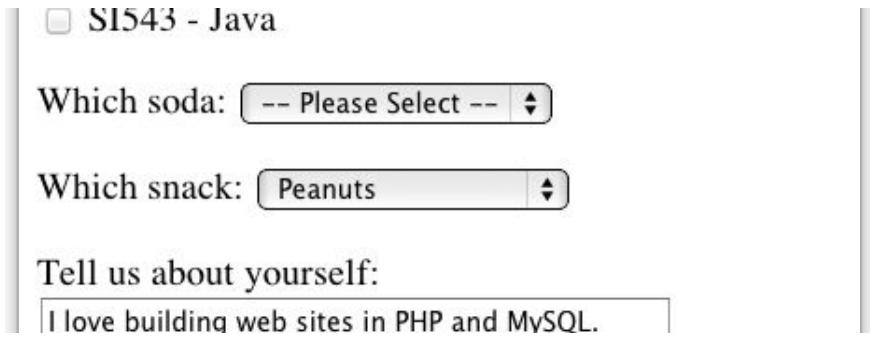
I love building web sites in PHP and MySQL.

```
$_POST:
Array(
   ...
     [class] => si502
     [soda] => 0
     [snack] => peanuts
   ...
)
```

The values can be any string, but numbers are used quite often.

Input as Dropbox

```
<p><label for="inp07">Which snack:
  <select name="snack" id="inp07">
    <option value="">-- Please Select --</option>
    <option value="chips">Chips</option>
    <option value="peanuts" selected>Peanuts</option>
    <option value="cookie">Cookie</option>
  </select>
</p>
```


SI543 - Java

Which soda: -- Please Select --

Which snack: Peanuts

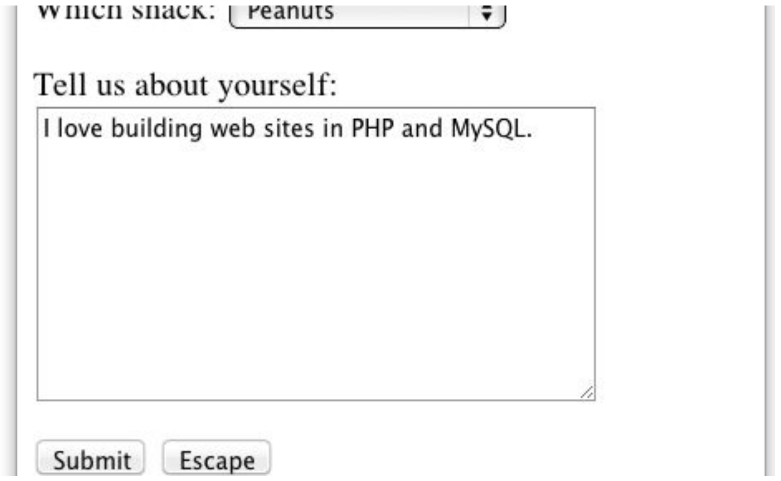
Tell us about yourself:

I love building web sites in PHP and MySQL.

```
$_POST:
Array(
   ...
      [class] => si502
      [soda] => 0
      [snack] => peanuts
   ...
)
```

Input as Textarea

```
<p><label for="inp08">Tell us about yourself:<br/>
  <textarea rows="10" cols="40" id="inp08" name="about">
    I love building web sites in PHP and MySQL.
  </textarea>
</p>
```

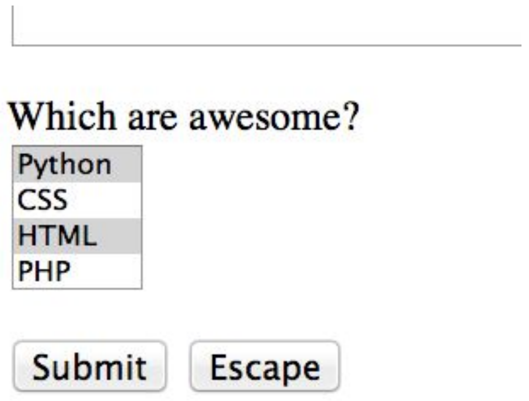


Tell us about yourself:
I love building web sites in PHP and MySQL.

```
$_POST:
Array(
   ...
      [about] => I love
building web sites in PHP
and MySQL.
      [dopost] => Submit
   ...
)
```

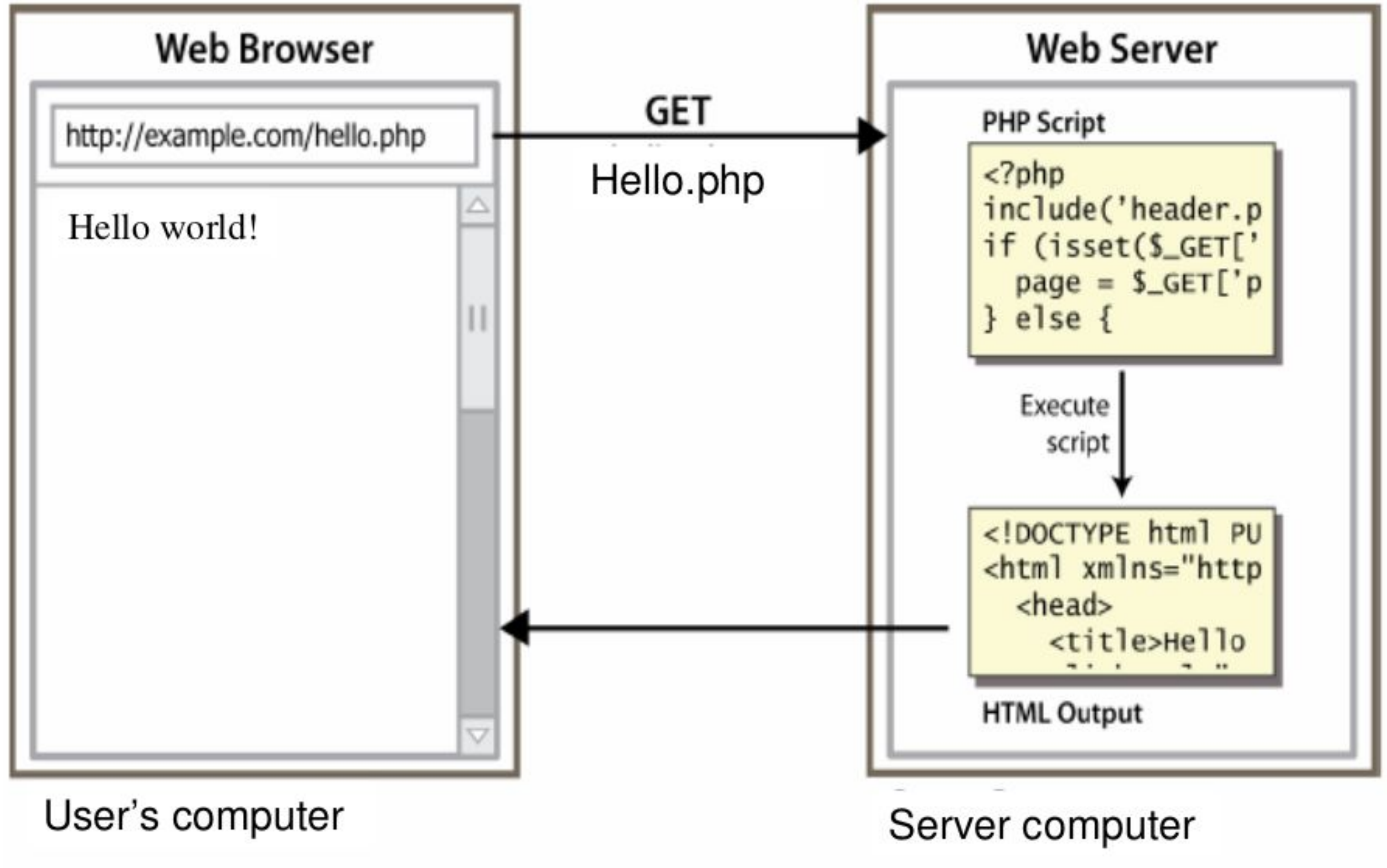
Input as Multiple Options

```
<p><label for="inp09">Which are awesome?<br/>
<select multiple="multiple" name="code[]" id="inp09">
  <option value="python">Python</option>
  <option value="css">CSS</option>
  <option value="html">HTML</option>
  <option value="php">PHP</option>
</select>
```



```
$_POST:
Array(
  ...
    [code] => Array
        (
            [0] => python
            [1] => html
        )
    [dopost] => Submit
  ...
)
```

Client Server Model



```
<?php
include('header.p
if (isset($_GET['
  page = $_GET['p
} else {
```

```
<!DOCTYPE html PU
<html xmlns="http
  <head>
    <title>Hello
```

PHP More Features

- Form validation using regular expressions
- Handling session management
- Database Connection

References

- Miscellaneous resource from internet

Thank you!