Tutorial-11

Forms, Functions and File Handling in PHP

Sending Data using Forms

form-php.php

```
<form action = "form-data.php" method = "post">
    UserName: <input type = "text" name = "user-name"><br>
    Password: <input type = "password" name="password"><br>
    <input type = "submit" value = "Login">
</form>
```

form-data.php

```
#using method post and get
$userName = $_POST['user-name'];
echo "<h1>Welcome <h1>".$userName;
```

When the user fills out the form above and clicks the submit button, the form data is sent for processing to a PHP file named “form-data.php”.

The form data is sent with the HTTP POST method.

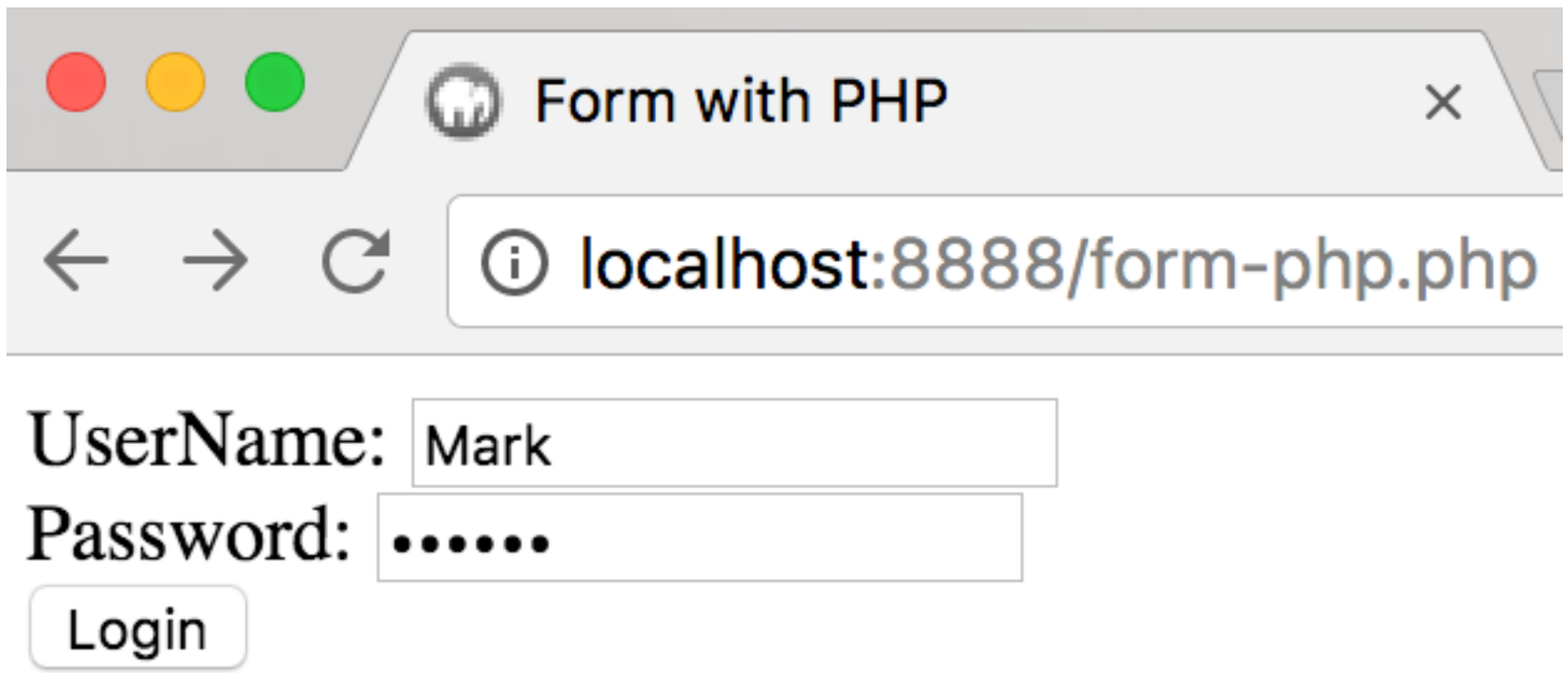
Form Fields

1. **Action**: This is the page on which your form redirects when you submit your form.
   In our case, it is **form-data.php** file.

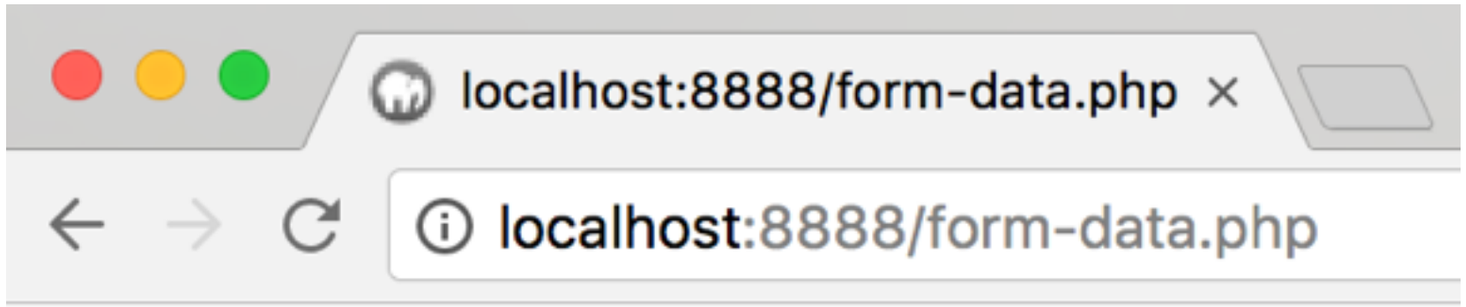
2. **method**: It can be post or get. Both are Http methods to send data to the server.

3. **When we use Get**: The data sent to the server is visible on the URL which is a security breach. We use method = post in such cases.

form-php.php on browser



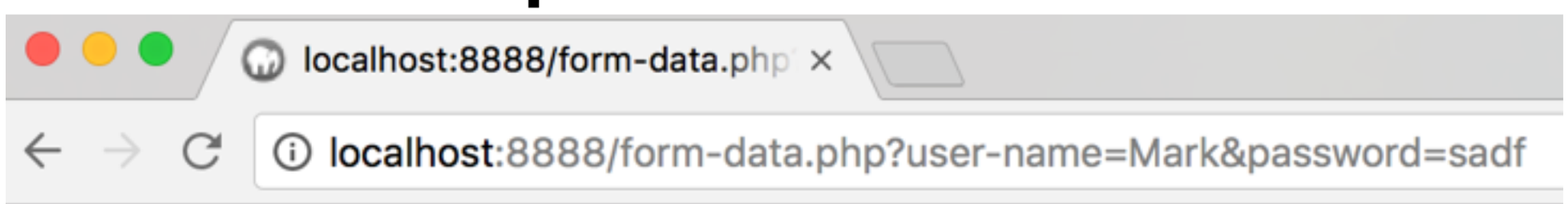
Response with POST



Welcome

Mark

Response with GET



Welcome

Mark

Functions

```php
<?php
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Example function.\n";
    return $retval;
}
?>
```

Any valid PHP code may appear inside a function

Function names follow the same rules as other labels in PHP.

A valid function name starts with a letter or underscore, followed by any number of letters, numbers, or underscores.

Simple Function example

```
<?php
function writeMsg() {
    echo "Hello world!";
}

writeMsg();
?>
```

→ Hello world!

We can't call writeMsg() before the function is defined!!

File Handling in PHP

PHP has several functions for creating, reading, uploading, and editing files.

1. file

```
// Get a file into an array.  In this example we'll go through HTTP to get
// the HTML source of a URL.
$lines = file('http://www.example.com/');
```

file — Reads entire file into an array

Returns the file in an array. Each element of the array corresponds to a line in the file, with the newline still attached. Upon failure, file() returns **FALSE**.

2. fopen — Opens file or URL

```php
<?php
$handle = fopen("c:\\folder\\resource.txt", "r");
?>
```

The **mode** parameter specifies the type of access you require to the stream.

File open modes

mode	Description
'r'	Open for reading only; place the file pointer at the beginning of the file.
'r+'	Open for reading and writing; place the file pointer at the beginning of the file.
'w'	Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
'w+'	Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
'a'	Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it. In this mode, fseek() has no effect, writes are always appended.
'a+'	Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it. In this mode, fseek() only affects the reading position, writes are always appended.

fwrite

fwrite() writes the contents of **string** to the file stream pointed to by **handle**.

```
fwrite($handle_w, "First line\n");
fwrite($handle_w, "Second line\n");
fwrite($handle_w, "Third line\n");
```

fread

fread() reads up to **length** bytes from the file pointer referenced by **handle**.

```
#printing only first five letters of the file
$s = fread($handle_r, 5);
echo "Using fread: ".$s."<br><br><br>";
```

fclose

```
bool fclose ( resource $handle )
```

```
#to free the pointer $handle_r and $handle_w
fclose($handle_w);
fclose($handle_r);
```

The file pointed to by **handle** is closed.

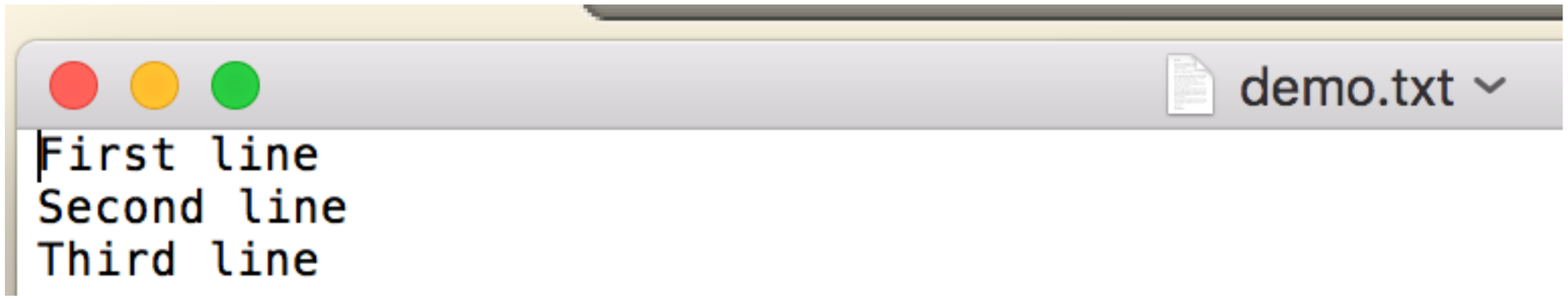
Returns **TRUE** on success or **FALSE** on failure.

A file should always be closed after use so that the file pointer is set free.

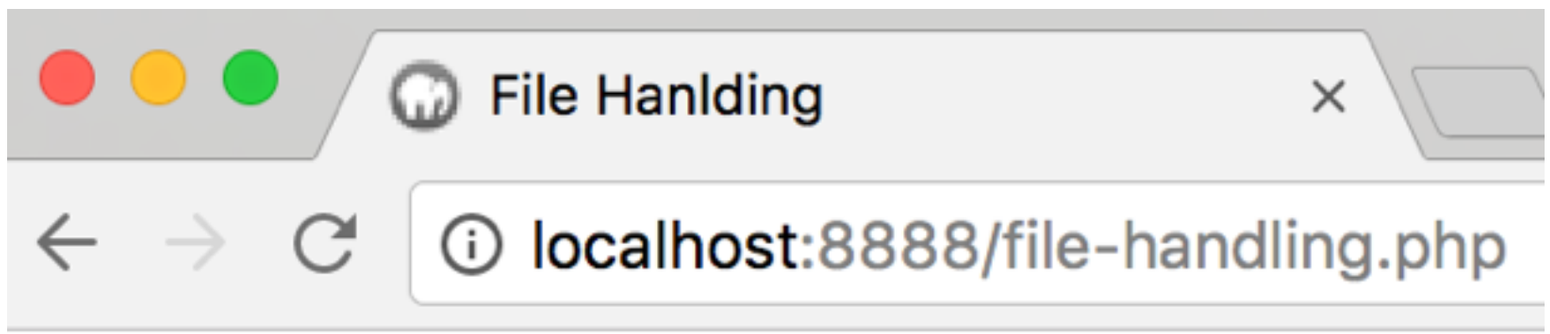
File Handling Example

```php
<?php
    $handle_w = fopen("demo.txt", "w");
    $handle_r = fopen("demo.txt", "r");

    if ($handle_r == false) {
        echo "File not created . Some Error !";
    }

    fwrite($handle_w, "First line\n");
    fwrite($handle_w, "Second line\n");
    fwrite($handle_w, "Third line\n");

    #printing only first five letters of the file
    $s = fread($handle_r, 5);
    echo "Using fread: ".$s."<br><br><br>";

    #reading the file into an array
    $theData = file("demo.txt");

    #printing the file using array
    echo "<h3>Output of the file: </h3><br>";

    foreach($theData as $line)
        {
            $line = rtrim($line);
            echo $line."<br>";
        } //end foreach

    #to free the pointer $handle_r and $handle_w
    fclose($handle_w);
    fclose($handle_r);
?>
```

File create in htdocs/ www



Output



Using fread: First

Output of the file:

First line
Second line
Third line

Common File Functions

- fclose — Closes an open file pointer
- feof — Tests for end-of-file on a file pointer
- fflush — Flushes the output to a file
- fgetc — Gets character from file pointer
- fgetcsv — Gets line from file pointer and parse for CSV fields
- fgets — Gets line from file pointer
- fgetss — Gets line from file pointer and strip HTML tags
- file_exists — Checks whether a file or directory exists
- file_get_contents — Reads entire file into a string
- file_put_contents — Write a string to a file
- file — Reads entire file into an array
- filesize — Gets file size
- filetype — Gets file type

http://php.net/manual/en/ref.filesystem.php

Practice Questions

1. Create a small form and send data to other file. Also, redirect your page from one to another when form is submitted. This is similar to Login and Logout except for validations. If you wish to, create a validation function and before Welcoming the user, validate the input.

2. Create a file using file function, write, append, and display file on the browser.