

A guided tour to PHP

© Oge Marques, PhD - 2016

Introduction

This document guides you through the most important aspects of PHP. It is structured as a step-by-step guide. It is best that you follow it in the intended sequence.

Part 1- Setup

1. Test your PHP setup at the *lamp.cse.fau.edu* server using the example below (save it as **test.php**):

```
<?php
    echo 'Hello PHP!';
?>
```

2. (OPTIONAL) Choose ***one*** Apache/MySQL/PHP local server solution for your personal computer / OS. Possible alternatives include: XAMPP, EasyPHP, WAMPServer, MAMP (Pro), and many others.
3. (OPTIONAL) Download, install, configure, and test your WAMP/LAMP/MAMP server setup.
4. Check which version of PHP is being used in your local setup (as well as its settings) and compare it with the one being used at the *lamp.cse.fau.edu* server. Use the script below (save it as **info.php**):

```
<?php
    phpinfo();
?>
```

5. Select an editor / IDE for PHP development. Here are some recommendations:
 - Atom (<https://atom.io/>)
 - Brackets (<http://brackets.io/>)
 - Sublime Text (<http://www.sublimetext.com/>)
 - PhpStorm (<http://www.jetbrains.com/phpstorm/>).
6. Download a zip file containing all examples in the (Nixon 2014) book¹, from <http://lpmj.net/>
7. Keep good PHP references handy, particularly <http://php.net/> .

¹ “Learning PHP, MySQL, JavaScript, CSS & HTML5” 4th Edition By Robin Nixon (O'Reilly 2014, ISBN 978-1491918661)

Part 2- Basic PHP

This part covers selected examples from Chapters 3-7 in the (Nixon 2014) book².

1. Run **example 3-1** on your server setup; it should display a "Hello world" message.
2. Modify **example 3-3** to add an 'echo' for each variable and run it to check each variable's contents displayed on screen and run the modified version.

```
<?php
$mycounter = 1;
echo $mycounter;
echo "<br />";
$mystring = "Hello";
echo $mystring;
echo "<br />";
$myarray = array("One", "Two", "Three");
echo $myarray; // not quite what you expected, right?
echo "<br />";
foreach ($myarray as $item)
{
    echo $item;
    echo "<br />";
}
?>
```

3. Run **example 3-4** and make sure you understand what it's doing (and how).
4. Run **example 3-5** and make sure you understand what it's doing (and how).
5. Modify **example 3-5** to add an 'echo' to display the entire array and run the modified version

```
<?php
$oxo = array(array('x', ' ', 'o'),
              array('o', 'o', 'x'),
              array('x', 'o', ' '));
echo "<pre>";
foreach ($oxo as $row)
{
    foreach ($row as $symbol)
        echo "$symbol ";
    echo "<br />";
}
echo "</pre>";
?>
```

² "Learning PHP, MySQL, JavaScript, CSS & HTML5" 4th Edition By Robin Nixon (O'Reilly 2014, ISBN 978-1491918661)

6. Run **examples 3-6 through 3-9** and make sure you understand what they are doing (and how).
7. Run **example 3-10** and make sure you understand what it's doing (and how).
8. Run **example 3-11** and make sure you understand what it's doing (and how).
9. Run **examples 3-12 through 3-16** and make sure you understand what they are doing (and how).
10. Run **example 3-17** and make sure you understand what it's doing (and how).
11. Modify **example 3-17** to add an 'echo' to display the results of calling test() and run the modified version.

```
<?php
$temp = "Calling function test()... ";
echo $temp;
test();
echo "<br />";

$temp = "Calling function test() again... ";
echo $temp;
test();
echo "<br />";

$temp = "... and again ... ";
echo $temp;
test();
echo "<br />";

function test()
{
    static $count = 0;
    echo $count;
    $count++;
}
?>
```

12. Run **examples 4-1 through 4-4** and make sure you understand what they are doing (and how).
13. Run **examples 4-12 through 4-16** and make sure you understand what they are doing (and how).
14. Glance through **examples 4-17 through 4-36** (they are similar to what you've seen in JavaScript and other languages) and make sure you understand what they are doing (and how).
15. Run **example 4-37** and make sure you understand what it's doing (and how).
16. Run **example 5-1** and make sure you understand what it's doing (and how).
17. Run **examples 5-2 through 5-5** and make sure you understand what they are doing (and how).

18. Glance through **examples 5-6 through 5-8** and make sure you understand the difference between *include*, *include_once*, *require*, and *require_once*.
19. Run **examples 5-9 through 5-13** and make sure you understand what they are doing (and how).
20. Glance through **examples 5-14 through 5-20** and make sure you understand what they are doing (and how).
21. Run **example 5-21** and make sure you understand what it's doing (and how).
22. Glance through **examples 5-22 through 5-23** and make sure you understand what they are doing (and how).
23. Run **examples 5-24 through 5-27** and make sure you understand what they are doing (and how).
24. Run **examples 6-1 through 6-15** and make sure you understand what they are doing (and how).
25. Run **examples 7-1 through 7-2** and make sure you understand what they are doing (and how).
26. Run **example 7-3** and make sure you understand what it's doing (and how).
27. Modify **example 7-3** to display "Date is valid".
28. Run **example 7-4** and make sure you understand what it's doing (and how).
29. Locate the newly created file (***testfile.txt***) in your hard drive. Is it in the expected folder? Inspect its contents. Do they make sense?
30. Run **examples 7-5 through 7-6** and make sure you understand what they are doing (and how).
31. Run **example 7-8** and make sure you understand what it's doing (and how).
32. Locate the newly created file (***testfile2.txt***) in your hard drive. Is it in the expected folder? Inspect its contents. Do they make sense?
33. Run **example 7-9** and make sure you understand what it's doing (and how).
34. Locate the newly created file (***testfile2.new***) in your hard drive. Is it in the expected folder? Inspect its contents. Do they make sense?
35. Run **example 7-10** and make sure you understand what it's doing (and how). Did it delete *testfile2.new* successfully?
36. Run **example 7-11** and make sure you understand what it's doing (and how). Did it update the contents of *testfile.txt* successfully?
37. Run **example 7-12** and make sure you understand what it's doing (and how). Did it update the contents of *testfile.txt* successfully?
38. Run **example 7-13** and make sure you understand what it's doing (and how). Did it display the contents of *testfile.txt* successfully?
39. Run **example 7-14** and make sure you understand what it's doing (and how).
40. Run **example 7-15** (you must rename it to ***upload.php*** first) and make sure you understand what it's doing (and how).
41. Run **example 7-16** (you must rename it to ***upload2.php*** first) and make sure you understand what it's doing (and how).
42. Run **example 7-17** and make sure you understand what it's doing (and how).

Part 3- MySQL, MySQLi, and PHP

This part covers selected examples from Chapters 8-10 in the (Nixon 2014) book³.

1. Access your MySQL account on lamp.cse.fau.edu following the instructions at <http://tsg.eng.fau.edu/servers/lamp-cse-fau-edu/accessing-mysql/> and the email received from TSG at the time your account was created (you will need it for the MySQL password).
2. Go to **<https://lamp.cse.fau.edu/phpMyAdmin/>**, select your default database (it has your username), and enter the commands from Example 8.3 using the 'SQL' tab.
 - Notice that the newly created table (classics) will appear on the left sidebar.
3. Double-check it by typing `DESCRIBE classics;`
4. Make sure you understand the names, types, and meaning of each field in the table.
5. Add a new column called `id` to the table `classics` with autoincrementing, following the syntax in example 8.5.
6. Delete the table `classics` with the command `DROP TABLE classics;`
7. Re-create the table `classics` using the syntax in example 8.6.
8. Populate the table `classics` using the syntax in example 8.8.
9. Execute a query to see all contents of the newly populated table: `SELECT * FROM classics;`
10. Rename the table: `ALTER TABLE classics RENAME pre1900;`
11. Rename it again (back to the original name): `ALTER TABLE pre1900 RENAME classics;`
12. Change the data type of a column: `ALTER TABLE classics MODIFY year SMALLINT;`
13. Add a new column: `ALTER TABLE classics ADD pages SMALLINT UNSIGNED;`
14. Inspect the results of the last two steps: `DESCRIBE classics;`
15. Rename a column: `ALTER TABLE classics CHANGE type category VARCHAR(16);`
16. Remove a column: `ALTER TABLE classics DROP pages;`
17. Create an index following the syntax in example 8.10.
18. Delete the table `classics` with the command `DROP TABLE classics;`
19. Re-create the table `classics` using the syntax in example 8.12.
20. Populate the table `classics` using the syntax in example 8.8 (*modified* to replace 'type' with 'category').
21. Try to create a new column using the syntax: `ALTER TABLE classics ADD isbn CHAR(13) PRIMARY KEY;`
22. You should get an error message (see textbook for explanation)
23. Create and populate a new 'isbn' column with data and using a primary key following the syntax in Example 8.13.
24. Add a FULLTEXT index to the table 'classics' using: `ALTER TABLE classics ADD FULLTEXT(author,title);`
25. Execute the two SELECT statements (one at a time) from Example 8.16.

³ "Learning PHP, MySQL, JavaScript, CSS & HTML5" 4th Edition By Robin Nixon (O'Reilly 2014, ISBN 978-1491918661)

26. Count the number of rows using: `SELECT COUNT(*) FROM classics;`
27. Insert a new record following the syntax in Example 8.18.
28. Execute the two `SELECT` statements (one at a time) from Example 8.19 and make sure that you understand the differences between them.
29. Remove the last entry: `DELETE FROM classics WHERE title='Little Dorrit';`
30. Execute the two `SELECT` statements (one at a time) from Example 8.21.
31. Execute the three `SELECT` statements (one at a time) from Example 8.22.
32. Execute the three `SELECT` statements (one at a time) from Example 8.23.
33. Execute the three `SELECT` statements (one at a time) from Example 8.24.
34. Execute the two `SELECT` statements (one at a time) from Example 8.25.
35. Execute the two `UPDATE` statements (one at a time) from Example 8.26.
36. Execute the two `SELECT` statements (one at a time) from Example 8.27.
37. Run the query: `SELECT category, COUNT(author) FROM classics GROUP BY category;` and try to understand what it does.
38. Create a `customers` table following the syntax of example 8.28.
39. Execute the `SELECT` statement from Example 8.29 and notice how it joins information from both tables in a meaningful way.
40. Repeat the previous step using the syntax `SELECT name,author,title FROM customers NATURAL JOIN classics;`
41. Repeat it again, this time using the syntax:
`SELECT name,author,title FROM customers
JOIN classics ON customers.isbn=classics.isbn;`
42. Do it again, this time using:
`SELECT name,author,title from
customers AS cust, classics AS class WHERE cust.isbn=class.isbn;`
43. Execute the three `SELECT` statements (one at a time) from Example 8.30.

Normalization: basic concepts

44. Look at Table 9.1 (below) and make sure you understand why its design is (highly) inefficient.

Table 9-1. A highly inefficient design for a database table

Author 1	Author 2	Title	ISBN	Price (USD)	Customer name	Customer address	Purch. date
David Sklar	Adam Trachtenberg	PHP Cookbook	0596101015	44.99	Emma Brown	1565 Rainbow Road, Los Angeles, CA 90014	Mar 03 2009
Danny Goodman		Dynamic HTML	0596527403	59.99	Darren Ryder	4758 Emily Drive, Richmond, VA 23219	Dec 19 2008
Hugh E Williams	David Lane	PHP and MySQL	0596005436	44.95	Earl B. Thurston	862 Gregory Lane, Frankfort, KY 40601	Jun 22 2009
David Sklar	Adam Trachtenberg	PHP Cookbook	0596101015	44.99	Darren Ryder	4758 Emily Drive, Richmond, VA 23219	Dec 19 2008
Rasmus Lerdorf	Kevin Tatroe & Peter MacIntyre	Programming PHP	0596006815	39.99	David Miller	3647 Cedar Lane, Waltham, MA 02154	Jan 16 2009

45. Look at Tables 9.2 and 9.3 (below) and make sure you understand why this design is better than the one in Table 9.1.

Table 9-2. The result of stripping the author columns from Table 9-1

Title	ISBN	Price (USD)	Customer name	Customer address	Purchase date
PHP Cookbook	0596101015	44.99	Emma Brown	1565 Rainbow Road, Los Angeles, CA 90014	Mar 03 2009
Dynamic HTML	0596527403	59.99	Darren Ryder	4758 Emily Drive, Richmond, VA 23219	Dec 19 2008
PHP and MySQL	0596005436	44.95	Earl B. Thurston	862 Gregory Lane, Frankfort, KY 40601	Jun 22 2009
PHP Cookbook	0596101015	44.99	Darren Ryder	4758 Emily Drive, Richmond, VA 23219	Dec 19 2008
Programming PHP	0596006815	39.99	David Miller	3647 Cedar Lane, Waltham, MA 02154	Jan 16 2009

Table 9-3. The new Authors table

ISBN	Author
0596101015	David Sklar
0596101015	Adam Trachtenberg
0596527403	Danny Goodman
0596005436	Hugh E Williams
0596005436	David Lane
0596006815	Rasmus Lerdorf
0596006815	Kevin Tatroe
0596006815	Peter MacIntyre

46. Look at Tables 9.4 and 9.5 (below) and make sure you understand why this design is better than the one in Table 9.2.

Table 9-4. The new Titles table

ISBN	Title	Price
0596101015	PHP Cookbook	44.99
0596527403	Dynamic HTML	59.99
0596005436	PHP and MySQL	44.95
0596006815	Programming PHP	39.99

Table 9-5. The Customer details from Table 9-2

ISBN	Customer name	Customer address	Purchase date
0596101015	Emma Brown	1565 Rainbow Road, Los Angeles, CA 90014	Mar 03 2009
0596527403	Darren Ryder	4758 Emily Drive, Richmond, VA 23219	Dec 19 2008
0596005436	Earl B. Thurston	862 Gregory Lane, Frankfort, KY 40601	Jun 22 2009
0596101015	Darren Ryder	4758 Emily Drive, Richmond, VA 23219	Dec 19 2008
0596006815	David Miller	3647 Cedar Lane, Waltham, MA 02154	Jan 16 2009

47. Look at Tables 9.6 and 9.7 (below) and make sure you understand why this design is better than the one in Table 9.5.

Table 9-6. The new Customers table

CustNo	Name	Address	City	State	Zip
1	Emma Brown	1565 Rainbow Road	Los Angeles	CA	90014
2	Darren Ryder	4758 Emily Drive	Richmond	VA	23219
3	Earl B. Thurston	862 Gregory Lane	Frankfort	KY	40601
4	David Miller	3647 Cedar Lane	Waltham	MA	02154

Table 9-7. The new Purchases table

CustNo	ISBN	Date
1	0596101015	Mar 03 2009
2	0596527403	Dec 19 2008
2	0596101015	Dec 19 2008

CustNo	ISBN	Date
3	0596005436	Jun 22 2009
4	0596006815	Jan 16 2009

48. Look at Tables 9.8 through 9.11 (below) and make sure you understand why this design is better than the one in Table 9.6.

Table 9-8. Third Normal Form Customers table

CustNo	Name	Address	Zip
1	Emma Brown	1565 Rainbow Road	90014
2	Darren Ryder	4758 Emily Drive	23219
3	Earl B. Thurston	862 Gregory Lane	40601
4	David Miller	3647 Cedar Lane	02154

Table 9-9. Third Normal Form Zip codes table

Zip	CityID
90014	1234
23219	5678
40601	4321
02154	8765

Table 9-10. Third Normal Form Cities table

CityID	Name	StateID
1234	Los Angeles	5
5678	Richmond	46
4321	Frankfort	17
8765	Waltham	21

Table 9-11. Third Normal Form States table

StateID	Name	Abbreviation
5	California	CA
46	Virginia	VA
17	Kentucky	KY
21	Massachusetts	MA

49. (OPTIONAL) create those tables in MySQL (via phpMyAdmin) and "play" with them.

PHP and MySQL

50. You will build a simple web-based application, using PHP to enable access to some of the previously created tables.
- You have two options: running them on your local (MAMP/LAMP/WAMP) server (recommended) or uploading them and running them on lamp.cse.fau.edu.

The Process

The process of using MySQL with PHP is:

1. Connect to MySQL.
2. Select the database to use.
3. Build a query string.
4. Perform the query.
5. Retrieve the results and output them to a web page.
6. Repeat Steps 3 through 5 until all desired data has been retrieved.
7. Disconnect from MySQL.

51. Modify Example 10.1 to refer to your database name and user credentials and save it to a separate directory with the name *login.php*.
52. Populate the 'classics' table according to the examples in Chapter 8.
53. Copy Example 10.5 to your test directory and rename it to *query.php*.
54. Test *query.php*. You should see results similar to Fig. 10.1 in the book.

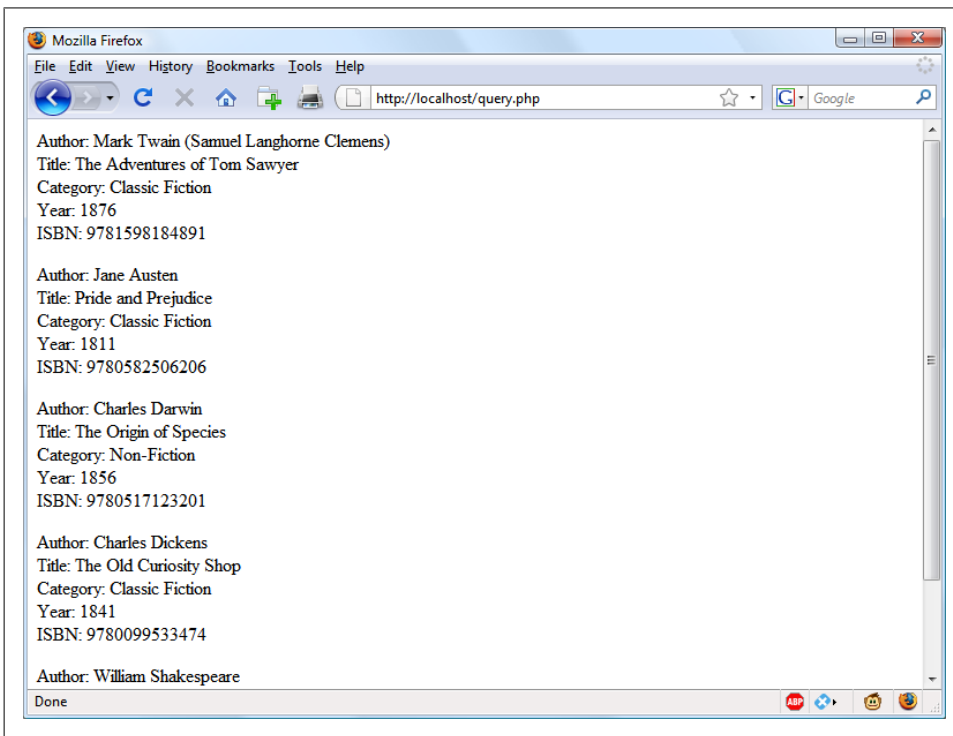


Figure 10-1. The output from the *query.php* program in Example 10-5

55. Copy Example 10.8 to your test directory and rename it to *sqltest.php*.
56. Run *sqltest.php*. You should see results similar to Fig. 10.2 in the book.

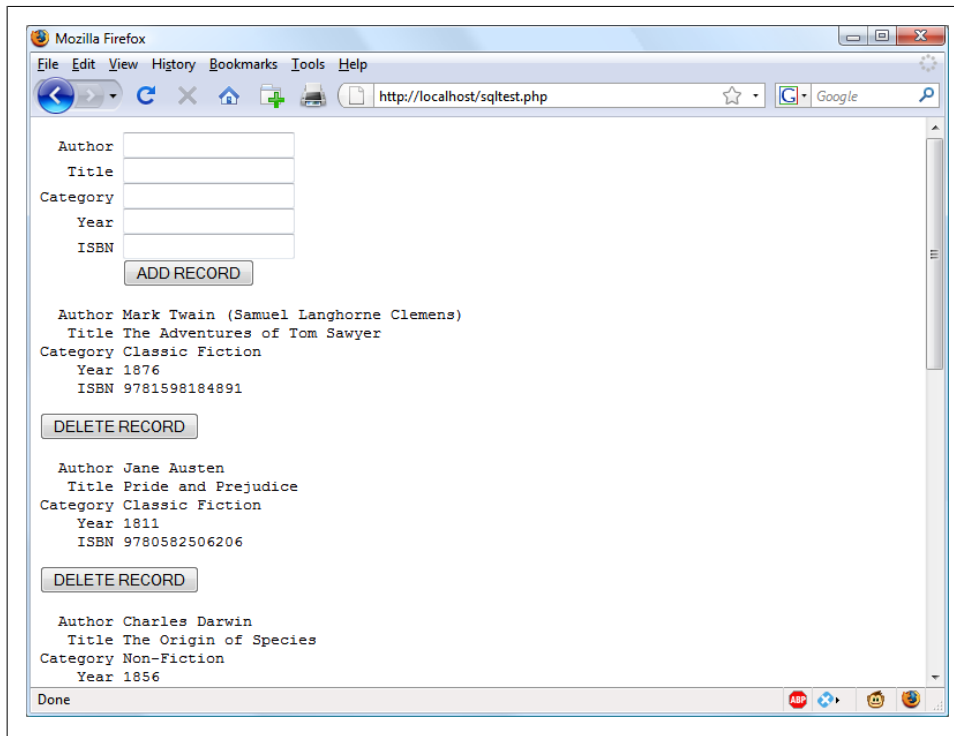


Figure 10-2. The output from *Example 10-8*, *sqltest.php*

57. Play with *sqltest.php* and test its add/delete capabilities.
58. Run Example 10.9 and ensure that it created a table (cats) as expected.
59. Run Example 10.10 and ensure that it works as expected.
60. Run Example 10.11 and ensure that it drops the table (cats) as expected.
61. Run Example 10.9 again and re-create the table.
62. Run Example 10.12 to populate the table with contents.
63. Add more records to the 'cats' table (either by editing Example 10.12 or manually via phpMyAdmin).
64. Run Examples 10.13 through 10.16 and ensure that they work as expected.
65. Ensure that you (still) have tables 'customers' and 'classics' from Chapter 8. If not, re-create them.
66. Run Example 10.17 and ensure that it works as expected.

Part 4- Form processing and AJAX

This part uses resources from W3Schools.

1. (OPTIONAL) Review PHP concepts, following the steps from http://www.w3schools.com/php/php_intro.asp to http://www.w3schools.com/php/php_superglobals.asp
2. Learn about form submission (including the important aspects of validation and sanitization) and processing, following the steps from http://www.w3schools.com/php/php_forms.asp to http://www.w3schools.com/php/php_form_complete.asp
3. Learn more about the use of PHP and MySQL to store, retrieve, and organize information in databases, following the steps from http://www.w3schools.com/php/php_mysql_intro.asp to http://www.w3schools.com/php/php_mysql_select_limit.asp
4. Learn more about how to implement AJAX functionality using PHP, especially the *name suggestion* example, following the steps from http://www.w3schools.com/php/php_ajax_intro.asp to http://www.w3schools.com/php/php_ajax_poll.asp
5. (OPTIONAL) Take the PHP quiz to test your knowledge: http://www.w3schools.com/php/php_quiz.asp

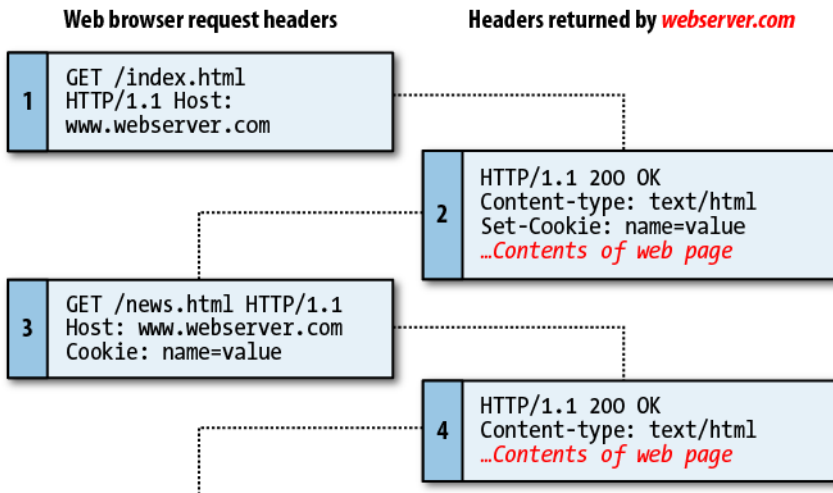
Part 5- Cookies, sessions, authentication, and more...

This part covers selected examples from Chapters 11, 12, and 17 in the (Nixon 2014) book⁴.

1. Open example 11-1 in your favorite editor and save a copy as *formtest.php*.
2. Run *formtest.php* and make sure you understand what it's doing (not a whole test, it turns out) (and how).
3. Open example 11-2 in your favorite editor and save a copy as *formtest2.php*.
4. Run *formtest2.php* and make sure you understand what it's doing (and how).
5. Look at Example 11-9 and ensure that you understand the concept of “input sanitization” and how it’s implemented.
6. Open example 11-10 in your favorite editor and save a copy as *convert.php*.
7. Run *convert.php* and make sure you understand what it's doing (and how).

⁴ “Learning PHP, MySQL, JavaScript, CSS & HTML5” 4th Edition By Robin Nixon (O'Reilly 2014, ISBN 978-1491918661)

8. Study Fig 12-1 below carefully and ensure that you understand it.



9. Run Example 12-1 and make sure you understand what it's doing (and how).
10. Run Example 12-2 and make sure you understand what it's doing (and how).
11. Ensure that you have a valid *login.php* file (with your MySQL information and credentials) in the same directory as the examples in this chapter.
12. Run Example 12-3 and make sure you understand what it's doing (and how). More specifically, look at the differences between the users' actual passwords, their "salted" versions, and the values actually stored in the 'users' table in MySQL.

This is what I got:

forename	surname	username	password
Bill	Smith	bsmith	be9d31ad4315b2ad9900a8526cd3edb1
Pauline	Jones	pjones	b1334d37914cf7561a006f656e27600c

13. Close your browser and reopen it.
14. Open Example 12-4 in your favorite editor and save a copy as *authenticate.php*.
15. Run *authenticate.php* and make sure you understand what it's doing (and how).
16. Close your browser and reopen it.
17. Open Example 12-5 in your favorite editor and save a copy as *authenticate2.php*.
18. Open Example 12-6 in your favorite editor and save a copy as *continue.php*.
19. Run *authenticate2.php* and make sure you understand what it's doing (and how).
20. Open Example 12-8 in your favorite editor and save a copy as *continue.php* (overwriting the previous file with the same name).
21. Close your browser and reopen it.
22. Run *authenticate2.php* again.
23. Press the browser's reload button and see what happens. Can you understand why?

The next steps refer to resources from W3Schools.

24. Learn how to create, delete, retrieve, and check a cookie using PHP at http://www.w3schools.com/php/php_cookies.asp

25. Learn how to create, manage, and destroy sessions in PHP at http://www.w3schools.com/php/php_sessions.asp

Part 6- (OPTIONAL) PHP: latest developments and best practices

This part contains a compilation of useful resources for keeping up-to-date with the latest developments in PHP as well as best practices and miscellaneous recommendations for writing better PHP code.

The following sites/books are strongly recommended:

- “Hacking with PHP” (<http://www.hackingwithphp.com/>): free (ad-supported) online version of the book with the same name (previously titled "Practical PHP Programming").
- “PHP: the right way” (<http://www.phptherightway.com/>): frequently updated "quick reference for PHP popular coding standards, links to authoritative tutorials around the Web and what the contributors consider to be best practices at the present time."
- “Modern PHP: new features and good practices”, book by Josh Lockhart (the creator of "PHP: the right way"): <http://shop.oreilly.com/product/0636920033868.do>. Source code for the book examples are available at: <https://github.com/codeguy/modern-php>

Part 7- (OPTIONAL) PhpStorm

PhpStorm (<https://www.jetbrains.com/phpstorm/>) is a leading IDE for PHP development. As an FAU student, you can get it for free (see <https://www.jetbrains.com/student/> for details).

It has a very rich documentation (<https://www.jetbrains.com/phpstorm/documentation/>) and a collection of 40+ videos (<https://www.youtube.com/playlist?list=PLQ176FUlyIUbfeFz-2EbDzwExRID0Bc-w>) that teach you how to make the best use of its functionality.

THE END
