Lab Session 8: HTML and PHP

The content of a Web page can be either statically or dynamically generated. In this lab we will look at both ways of generating HTML. We will start with static HTML code, and then we will develop some PHP scripts to generate HTML dynamically.

Part I: Listing the accounts

1. The following code is for a static HTML page that shows the contents of table *account*.

```
<html>
    <body>
        <h3>Table <em>account</em> (version 1)</h3>
        <table border="1">
            <tr>
                <td><em>account_number</em></td>
                <td><em>branch_name</em></td>
                <td><em>balance</em></td>
            </tr>
            <tr><td>A-101</td><td>Downtown</td><td>500</td></tr>
            <tr><td>A-102</td><td>Perryridge</td><td>400</td></tr>
            <tr><td>A-201</td><td>Brighton</td><td>900</td></tr>
            <tr><td>A-215</td><td>Mianus</td><td>700</td></tr>
            <tr><td>A-217</td><td>Brighton</td><td>750</td></tr>
            <tr><td>A-222</td><td>Redwood</td><td>700</td></tr>
            <tr><td>A-305</td><td>Round Hill</td><td>350</td></tr>
        </table>
    </body>
</html>
```

1. Save this code in a file called **accounts1.php** on your local machine.

2. Use an SFTP client (such as **WinSCP** or **FileZilla**) to open your home directory on sigma.ist.utl.pt. Use the following login info:
   - Host: sigma.ist.utl.pt | Port: 22
   - Username: (your username of Fénix)
   - Password: (your password of Fénix)

3. Locate the folder "web" inside your home directory. This is where you should put your HTML pages and/or PHP scripts.

4. Move or copy the file **accounts1.php** into your "web" folder.

5. Open the Web browser on your local machine and navigate to:
   http://web.ist.utl.pt/istxxxxxx/accounts1.php

where **istxxxxxx** is to be replaced by your username.

6. You should see the following content appearing in your Web browser:

Table *account* (version 1)

account_number	*branch_name*	*balance*
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

7. Right-click somewhere on the Web page and select "View Page Source". Confirm that the HTML code that was sent to the Web browser is the same as what you have written in **accounts1.php**.

8. We are going to change this page so that the same content is generated dynamically from a PHP array. For that purpose, we will use the following PHP script:

```
<html>
    <body>
        <h3>Table <em>account</em> (version 2)</h3>
        <table border="1">
            <tr>
                <td><em>account_number</em></td>
                <td><em>branch_name</em></td>
                <td><em>balance</em></td>
            </tr>
<?php

    $accounts[0] = array('A-101', 'Downtown', 500);
    $accounts[1] = array('A-102', 'Perryridge', 400);
    $accounts[2] = array('A-201', 'Brighton', 900);
    $accounts[3] = array('A-215', 'Mianus', 700);
    $accounts[4] = array('A-217', 'Brighton', 750);
    $accounts[5] = array('A-222', 'Redwood', 700);
    $accounts[6] = array('A-305', 'Round Hill', 350);

    for ($i = 0; $i < count($accounts); $i++)
    {
        echo("<tr>");
        for ($j = 0; $j < count($accounts[$i]); $j++)
        {
            echo("<td>{$accounts[$i][$j]}</td>");
        }
        echo("</tr>\n");
    }

?>
        </table>
    </body>
</html>
```

9. Note the following things about the script above:
   - This code is a mix of static and dynamic content. The dynamic content is being generated inside the PHP tags **<?php** and **?>**. Everything else outside those tags is static content.
   - The accounts are being placed in an array called **$accounts**. There are 7 accounts and these are being placed in consecutive positions of the array, starting at position 0.
   - Each account is also an array, and each of these arrays has 3 elements: an account number, a branch name, and a balance. Therefore, **$accounts** is actually an array of arrays. In PHP, this is called a *multidimensional array*.
   - In the first **for** loop, we are using **count($accounts)** to get the number of accounts in the array. The variable **$i** will go from 0 to 6. Each account can be accessed as **$accounts[$i]**.
   - In the second **for** loop, we are using **count($accounts[$i])** to get the number of data elements inside each account. The variable **$j** will go from 0 to 2. Each data element can be accessed as **$accounts[$i][$j]**.
   - Inside the second **for** loop, there is the instruction **echo("<td>{$accounts[$i][$j]}</td>")**. Note the use of curly braces inside the string. This is necessary for PHP to know that we want to print **$accounts[$i][$j]** and not **$account[$i]** followed by **$j** in square brackets.
   - Some **echo()** calls have a newline (**\n**) at the end of the string. This has not effect on what the Web browser will show. However, if you select "View Page Source" you will see that the HTML code has those line breaks.

10. Save the code above in a file called **accounts2.php** on your local machine.

11. Using an SFTP client as you did before, move or copy the file **accounts2.php** to your "web" folder on sigma.ist.utl.pt.

12. Open the Web browser on your local machine and navigate to:
    http://web.ist.utl.pt/istxxxxxx/accounts2.php
    where **istxxxxxx** is to be replaced by your username.

13. You should see the following table appearing in your browser:

Table *account* (version 2)

account_number	*branch_name*	*balance*
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

14. Right-click somewhere on the Web page and select “View Page Source” to view the HTML code that was sent to the Web browser. This HTML code has been dynamically generated by the PHP script.

15. In the previous example we have used an array **$accounts** indexed by position **$i**. Now we will change that to an *associative array*, meaning that the array will be indexed by a key. In this case, the key is the account number.

```
<html>
    <body>
        <h3>Table <em>account</em> (version 3)</h3>
        <table border="1">
            <tr>
                <td><em>account_number</em></td>
                <td><em>branch_name</em></td>
                <td><em>balance</em></td>
            </tr>
<?php

    $accounts['A-101'] = array('Downtown', 500);
    $accounts['A-102'] = array('Perryridge', 400);
    $accounts['A-201'] = array('Brighton', 900);
    $accounts['A-215'] = array('Mianus', 700);
    $accounts['A-217'] = array('Brighton', 750);
    $accounts['A-222'] = array('Redwood', 700);
    $accounts['A-305'] = array('Round Hill', 350);

    foreach ($accounts as $account_number => $account)
    {
        echo("<tr>");
        echo("<td>$account_number</td>");
        for ($j = 0; $j < count($account); $j++)
        {
            echo("<td>{$account[$j]}</td>");
        }
        echo("</tr>\n");
    }

?>
        </table>
    </body>
</html>
```

16. In the code above, note the following:
    - **$accounts** is an associative array indexed by account numbers.
    - The first loop is now a **foreach** loop which goes through all key-value pairs. For each pair, key is stored in the variable **$account_number** and the value is stored in the variable **$account**. Note that the value associated with the key is an array.
    - The second loop is still a **for** loop that goes through all the data elements in each account. However, now there are only two data elements: branch name and balance.

17. Save the code above in a file called **accounts3.php** on your local machine.

18. Using an SFTP client, move or copy the file **accounts3.php** to your "web" folder on sigma.ist.utl.pt.

19. Open the Web browser on your local machine and navigate to:
    http://web.ist.utl.pt/istxxxxxx/accounts3.php
    where **istxxxxxx** is to be replaced by your username.

20. You should see the following table appearing in your browser:

Table *account* (version 3)

account_number	*branch_name*	*balance*
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

21. Right-click somewhere on the Web page and select “View Page Source” to view the HTML code that was sent to the Web browser. This HTML code has been dynamically produced by the PHP script.

22. Now we will turn each account also into an associative array. For this purpose, we will use the following PHP script:

```
<html>
    <body>
        <h3>Table <em>account</em> (version 4)</h3>
        <table border="1">
            <tr>
                <td><em>account_number</em></td>
                <td><em>branch_name</em></td>
                <td><em>balance</em></td>
            </tr>
<?php

    $accounts['A-101'] = array('branch_name' => 'Downtown', 'balance' => 500);
    $accounts['A-102'] = array('branch_name' => 'Perryridge', 'balance' => 400);
    $accounts['A-201'] = array('branch_name' => 'Brighton', 'balance' => 900);
    $accounts['A-215'] = array('branch_name' => 'Mianus', 'balance' => 700);
    $accounts['A-217'] = array('branch_name' => 'Brighton', 'balance' => 750);
    $accounts['A-222'] = array('branch_name' => 'Redwood', 'balance' => 700);
    $accounts['A-305'] = array('branch_name' => 'Round Hill', 'balance' => 350);

    foreach ($accounts as $account_number => $account)
    {
        echo("<tr>");
        echo("<td>$account_number</td>");
        echo("<td>{$account['branch_name']}</td>");
        echo("<td>{$account['balance']}</td>");
        echo("</tr>\n");
    }

?>
        </table>
    </body>
</html>
```

23. In the code above, note the following:
    - Each element in the associative array **$accounts** is itself an associative array with two keys: branch name and balance.
    - The second loop has disappeared and instead we just print the values associated with those two keys. Note the use of curly braces in the **echo()** instructions.

24. Save the code above in a file called **accounts4.php** on your local machine.

25. Using an SFTP client, move or copy the file **accounts4.php** to your "web" folder on sigma.ist.utl.pt.

26. Open the Web browser on your local machine and navigate to:
    http://web.ist.utl.pt/istxxxxxx/accounts4.php
    where **istxxxxxx** is to be replaced by your username.

27. You should see the following table appearing in your browser:

Table *account* (version 4)

account_number	*branch_name*	*balance*
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

28. Right-click somewhere on the Web page and select "View Page Source" to view the HTML code that was sent to the Web browser. This HTML code has been dynamically produced by the PHP script.

29. In the previous examples, we have used a **for** or a **foreach** loop to iterate through the array **$accounts**. In this example, we will use an iterator and the special functions **reset()**, **current()**, and **next()**. Have a look at the following PHP script:

```
<html>
    <body>
        <h3>Table <em>account</em> (version 5)</h3>
        <table border="1">
            <tr>
                <td><em>account_number</em></td>
                <td><em>branch_name</em></td>
                <td><em>balance</em></td>
            </tr>
<?php

    $accounts = array(
        array('account_number' => 'A-101', 'branch_name' => 'Downtown', 'balance' => 500),
        array('account_number' => 'A-102', 'branch_name' => 'Perryridge', 'balance' => 400),
        array('account_number' => 'A-201', 'branch_name' => 'Brighton', 'balance' => 900),
        array('account_number' => 'A-215', 'branch_name' => 'Mianus', 'balance' => 700),
        array('account_number' => 'A-217', 'branch_name' => 'Brighton', 'balance' => 750),
        array('account_number' => 'A-222', 'branch_name' => 'Redwood', 'balance' => 700),
        array('account_number' => 'A-305', 'branch_name' => 'Round Hill', 'balance' => 350)
    );

    reset($accounts);
    $account = current($accounts);
    while ($account)
    {
        echo("<tr>");
        echo("<td>{$account['account_number']}</td>");
        echo("<td>{$account['branch_name']}</td>");
        echo("<td>{$account['balance']}</td>");
        echo("</tr>\n");
        $account = next($accounts);
    }

?>
        </table>
    </body>
</html>
```

30. In the code above, note the following:
    - The array **$accounts** is being initialized with the function **array()**. However, this is an array of arrays, so each element in the array **$accounts** is itself an array that is also initialized with the function **array()**.
    - The function **reset()** sets the internal pointer of the array to its first element.
    - The function **current()** returns the current element in the array. Since this is being called after **reset()**, the current element in the array is the first element.
    - If the array happened to be empty, then there would be no current element. In that case, **current()** would return **FALSE**, and therefore the **while** loop would not execute.
    - Inside the **while** loop, we print a table row (**<tr>...</tr>**) and then we use the function **next()** to get the next element from the array. If there are no more elements in the array, **next()** returns **FALSE**.
    - At the end of the **while** loop, the pointer is past the end of the array. If you need to iterate through the array again, you will need to call **reset()** to go back to the first element.

31. Save the code above in a file called **accounts5.php** on your local machine.

32. Using an SFTP client, move or copy the file **accounts5.php** to your "web" folder on sigma.ist.utl.pt.

33. Open the Web browser on your local machine and navigate to:
    http://web.ist.utl.pt/istxxxxxx/accounts5.php
    where **istxxxxxx** is to be replaced by your username.

34. You should see the following table appearing in your browser:

Table *account* (version 5)

account_number	*branch_name*	*balance*
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

35. Right-click somewhere on the Web page and select “View Page Source” to view the HTML code that was sent to the Web browser. This HTML code has been dynamically produced by the PHP script.

Part II: Showing the current date and time

36. In this step, we will create a different Web page to show us the current date and time. For this purpose, we will use the following PHP script:

```
<html>
    <body>
<?php
        $date = date("Y-m-d");
        echo("<p>The current date is: $date</p>\n");
        $time = date("H:i:s");
        echo("<p>The current time (on the server) is: $time</p>\n");
?>
    </body>
</html>
```

37. Save the code above in a file called **datetime1.php** on your local machine.

38. Using an SFTP client, move or copy the file **datetime1.php** to your "web" folder on sigma.ist.utl.pt.

39. Open the Web browser on your local machine and navigate to:
    http://web.ist.utl.pt/istxxxxxx/datetime1.php
    where **istxxxxxx** is to be replaced by your username.

40. You should see the date and time being displayed on the browser.

41. On your keyboard, hit F5 (refresh) repeatedly to reload the Web page, so that you see the time being updated.

42. Right-click somewhere on the Web page and select “View Page Source” to view the HTML that was sent to the Web browser.

43. Most content in the previous page is static. In fact, only the date and time are dynamic elements. In this case, it is possible to rewrite the PHP script in a simpler way:

```
<html>
    <body>
        <p>The current date is: <?= date("Y-m-d") ?></p>
        <p>The current time (on the server) is: <?= date("H:i:s") ?></p>
    </body>
</html>
```

44. In the code above, note the following:
    - Almost everything is static HTML code except for the highlighted parts, which contain the PHP instructions.
    - The short tag **<?=** allows printing (i.e. “echoing”) the value of a PHP function or variable without the need to call **echo()**.

45. Save the code above in the file **datetime2.php**.

46. Using an SFTP client, move or copy the file **datetime2.php** to your "web" folder on sigma.ist.utl.pt.

47. Open the Web browser on your local machine and navigate to:
    http://web.ist.utl.pt/istxxxxxx/datetime2.php
    where **istxxxxxx** is to be replaced by your username.

48. You should see the date and time being displayed on the browser, as before.