School of Information and Computer Technology

Sirindhorn International Institute of Technology

Thammasat University

ITS351 Database Programming Laboratory

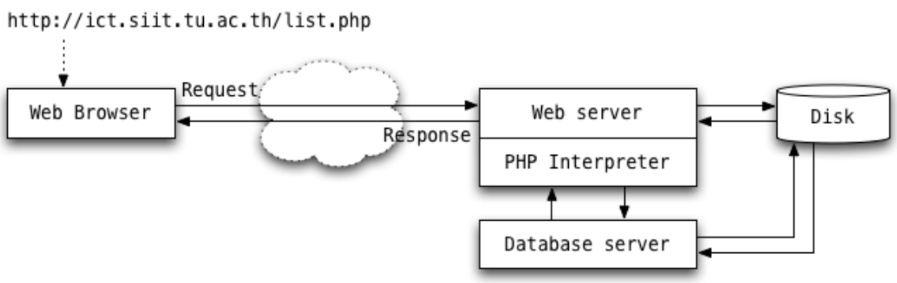
Laboratory #2: PHP

Objective:
- To introduce basic syntax and constructs of PHP scripting language.
- To study how to feed input data to PHP scripts.
- To provide hand-on practices on PHP programming.

Introduction to PHP

1. Basics

PHP is a scripting language designed for web application development. PHP interpreter resides on the server side with the web server. PHP code can be embedded into HTML to generate dynamic contents. PHP statements are enclosed in `<?php` and `?>` tag. Most of PHP syntax is inspired from C programming language.



Here is an example of a snippet of PHP code embedded into a HTML page.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Basic PHP</title>
    </head>
    <body>
        <?php
            echo "<h1>Hello, World</h1>";
            echo "10 + 20 = ";
            echo 10+20;
            echo "\n";
        ?>
    </body>
</html>
```

On a client's request, the above HTML page with embedded PHP code is passed to the PHP interpreter. The interpreter then interprets part of the code surrounded with `<?php` and `?>` (there can be many of these), generates the output according to what the code does (printing "Hello, World" in this case), and puts the result back to the HTML page. In the end, this PHP code yields the following HTML code.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Basic PHP</title>
    </head>
    <body>
        <h1>Hello, World</h1>10 + 20 = 30
    </body>
</html>
```

2. Variables

Variables in PHP are denoted by a dollar sign followed by the variable name. The variable name is case-sensitive, and may compose of alphabets, digits or underscore, but it must not start with a digit.

```
<?php
    $x = 10;
    $y = 'Hello';
    $z = $x.$y; // . is used to concatenate strings
    echo $z;
    //output: 10Hello
?>
```

3. Strings

We can use both single quotes and double quotes to enclose string literals in PHP. However, they yield different results.

- A single-quoted string is used for representing a simple string constant. It does not recognize escape sequences such as `\r`, `\n`, except `\'` for a single quote, and `\\` for a backslash.
- A double-quoted string recognizes all types of escape sequences, as well as variable substituting.

```
<?php
    $name = 'John';
    $age = 20;
    $txt = "His name is $name. He is \"{$age}\" years old.";
    echo $txt;
    //output: His name is John. He is "20" years old.
?>
```

Strings can be accessed the same way as in the case of arrays.

```
<?php
      $str = "SIIT, TU";
      echo $str[0].$str[1].$str[2].$str[3].$str[strlen($str)-1];
      //output: SIITU
?>
```

Notice the use of `strlen($str)` to find the length of the string `$str`.

4. Arrays

An array in PHP is an ordered map associating **key** and **value**. That is, an array in PHP can act as both usual array and a hash table. In addition to being dynamic in its size, PHP arrays are inhomogeneous i.e., values of different types can be stored in the same array.

```
<?php
    $arr = array('foo' => 'bar', 12 => true);

    $arr[] = 20;                //$arr[13] = 20;
    $arr['Hello'] = 'World';    //$arr['Hello'] = 'World';
    $arr[] = "PHP";             //$arr[14] = "PHP";

    print_r($arr);              //print_r displays the content of the array
    echo "<br>";

    unset($arr[13]);            //remove the element from the array
    print_r($arr);
?>
```

The following code demonstrates how PHP arrays can be used in an index-based manner.

```
<!DOCTYPE html>
<html>
    <body>
        <?php
                    $arr = array('a', 2,'b', 'c', 10);
                    for($i=0; $i < count($arr); ++$i){
                          echo $arr[$i] . '<br>';
                    }
            ?>
    </body>
</html>
```

This prints out `a, 2, b, c, 10` on a separate line. Note that index of arrays in PHP starts from 0. The function `count($arr)` can be used to find the size of array $arr.

5. If Statement

PHP's if statements work the same way as in C.

```
<!DOCTYPE html>
<html>
    <body>
        <?php $a = 10; ?>
        <?php if ($a == 10) { ?>
            <p>The condition is true.</p>
        <?php } else { ?>
            <p>The condition is false.</p>
        <?php } ?>
    </body>
</html>
```

Notice that PHP also allows a dynamic opening and closing of `<?php` and `?>`. In the code above, "<p>The condition is true.</p>" will be printed out.

We can write an if statement in an alternative version which is different from C syntax. This alternative syntax somehow makes the PHP code become clearer and easier to read when many PHP statements are embedded in HTML page.

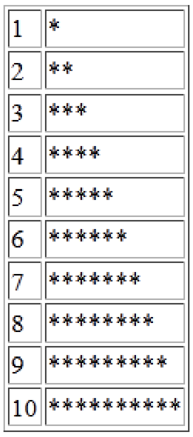
```
<!DOCTYPE html>
<html>
    <body>
        <? $a = 10; ?>
        <? if ($a == 10): ?>
            <p>The condition is true.</p>
        <? else: ?>
            <p>The condition is false.</p>
        <? endif; ?>
    </body>
</html>
```

In multiple nested if-else statements, you may use both "else if" and "elseif" in the C syntax above. However, in the latter one (i.e. colon syntax), only "elseif" is accepted.

6. While Statement

```
<!DOCTYPE html>
<html>
    <body>
        <table border="1">
        <?php $i = 1; ?>
        <?php while($i <= 10) { ?>
            <tr><td><?= $i?></td><td><?= str_repeat("*", $i) ?></td></tr>
            <?php $i++; ?>
        <?php } ?>
        </table>
    </body>
</html>
```

This gives



1	*
2	**
3	***
4	****
5	*****
6	******
7	*******
8	********
9	*********
10	**********

There is also another version of the while statement using : which gives the same result.

```
<!DOCTYPE html>
<html>
    <body>
        <table border="1">
        <? $i = 1; ?>
        <? while($i <= 10): ?>
            <tr><td><?= $i?></td><td><?= str_repeat("*", $i) ?></td></tr>
            <?  $i++; ?>
        <? endwhile; ?>
        </table>
    </body>
</html>
```

7. For Statement

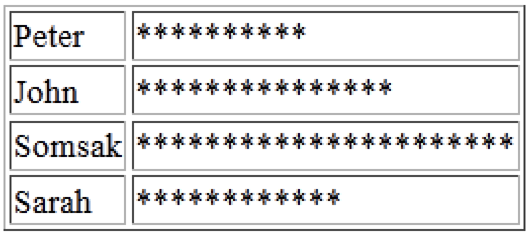
For statements also have a similar syntax as in C.

```
<!DOCTYPE html>
<html>
    <body>
        <table border="1">
        <? for($i=1; $i<=10; $i++): ?>
            <tr><td><?= $i?></td><td><?= str_repeat("*", $i) ?></td></tr>
        <? endfor; ?>
        </table>
    </body>
</html>
```

Additionally, PHP comes with a "foreach" statement.

```
<!DOCTYPE html>
<html>
    <body>
        <? $arr = array('Peter' => 10, 'John' => 15,
                        'Somsak' => 22, 'Sarah' => 12); ?>
        <table border="1">
        <? foreach($arr as $key => $value): ?>
            <tr><td><?= $key?></td><td>
            <?= str_repeat("*", $value) ?></td></tr>
        <? endforeach; ?>
        </table>
    </body>
</html>
```

In each iteration, $key and $value are respectively assigned to the key and the value of an element in the array. Thus, the code gives the following output.



Peter	**********
John	***************
Somsak	**********************
Sarah	************

8. Query String

Query string is a part of URL that comes after the file name. It is used to pass parameters to a PHP script file. For example, `http://localhost/test.php` calls '`test.php`' without passing any parameter. But, `http://localhost/test.php?a=10&b=20` passes two parameters: `a=10` and `b=20`. Passing parameters to the script in this way is referred to as a "GET request". These two parameters can be accessed in PHP script through a global variable `$_GET`.

```
<!DOCTYPE html>
<html>
    <body>
        <? if (isset($_GET['a']) && isset($_GET['b'])): ?>
            <?= $_GET['a'] ?> + <?= $_GET['b'] ?>
            = <?= $_GET['a']+$_GET['b'] ?>
        <? else: ?>
            Please specify the values of a and b.
        <? endif; ?>
    </body>
</html>
```

isset() is used to check if a variable is available, or if a value has been passed into the script. When passing `a=10` and `b=20` to the code above, the output is 10 + 20 = 30

9. Useful Functions

str_ireplace($target, $replacement, $text)

Replace all occurrences of a substring in an input string with a specified replacement For example,

```
str_ireplace("red", "black", "Change Red to Black");
```

This replaces "red" with "black" (case-insensitive) and thus gives a new string "Change black to Black".

str_replace($target, $replacement, $text)

Same as str_ireplace() but case-sensitively searches for the target string.

trim($str)

Strip whitespaces (including newlines and tabs) from the beginning and end of a string.

rand($min, $max)

Return a random integer from $min to $max (inclusive). rand() is automatically seeded.

Worksheet

1. Create a PHP file named "multiply.php" that shows a multiplication table sized 12x12 as shown below

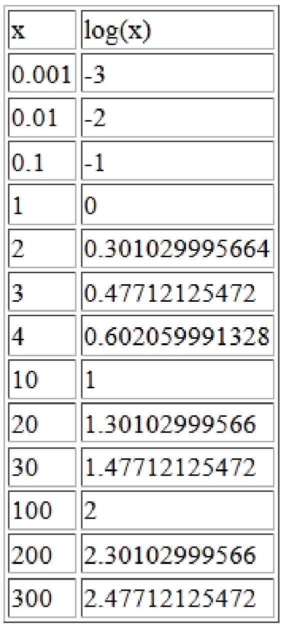
x	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	14	16	18	20	22	24
3	3	6	9	12	15	18	21	24	27	30	33	36
4	4	8	12	16	20	24	28	32	36	40	44	48
5	5	10	15	20	25	30	35	40	45	50	55	60
6	6	12	18	24	30	36	42	48	54	60	66	72
7	7	14	21	28	35	42	49	56	63	70	77	84
8	8	16	24	32	40	48	56	64	72	80	88	96
9	9	18	27	36	45	54	63	72	81	90	99	108
10	10	20	30	40	50	60	70	80	90	100	110	120
11	11	22	33	44	55	66	77	88	99	110	121	132
12	12	24	36	48	60	72	84	96	108	120	132	144

The following code prints the first heading row. You may use the code as the starting point.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Multiplication Table</title>
    </head>
    <body>
        <table style="text-align:center;" border="1">
            <tr>
                <th>x</th>
                <? for($i=1; $i<=12; $i++): ?>
                    <th><?= $i ?></th>
                <? endfor; ?>
            </tr>
        </table>
    </body>
</html>
```

Add a nested for loop to make the full table.

2. Write a PHP code to display a table of logarithm as shown.



x	log(x)
0.001	-3
0.01	-2
0.1	-1
1	0
2	0.301029995664
3	0.47712125472
4	0.602059991328
10	1
20	1.30102999566
30	1.47712125472
100	2
200	2.30102999566
300	2.47712125472

Use an array to store x, and loop through the array to compute log. Use log($x, 10) for log base 10. DO NOT hard code the whole table in HTML.

3. Display the following shape with loops in PHP

```
O
OO
OOO
OOOO
OOOOO
OOOOOO
OOOOOOO
OOOOOOOO
```

Exercises

To be announced.