

# PHP 03

Coding Style

Self-processing pages

Creating Memory

Cookies

Writing to a file

Including Code

Simple String Operations

**Peter Cho | 161B**

Based on a tutorial by Prof. Daniel Sauter, Rasmus Lerdorf & Kevin Tatroe:  
Programming PHP. Sebastopol: O'Reilly, 2002, and <http://php.net>

# PHP Programming Styles

There are various possibilities for mixing HTML and PHP code. If the page contains predominantly HTML tags with only some dynamic PHP elements, the **XML programming style** is recommended. This way, the `<?php . . . ?>` construct is used whenever a dynamic PHP element is required.

The same result can be achieved using the `<?php . . . ?>` construct only once by **echoing** out all content, both dynamic and static. This is appropriate for pages using predominantly PHP code, e.g. to process forms, calculate, etc. Styles can be mixed, but consistency is highly recommended.

Remember that PHP can be used to generate not only HTML content, but also JavaScript, CSS, and even pdfs and images at runtime.

# XML Style see style\_xml.php

```
<html>
<head>
<title>PHP Coding XML Style</title>
</head>
<body>
<form action="next.php" method="post">
<input name="check1" type="checkbox"
value="<?php echo $Check1value; ?>">
<input name="text1" type="text"
value="<?php echo $text1; ?>" size="20">
<input name="submit" type="submit" value="send it">
</form>
</body>
</html>
```

# Echoing content

style\_echo.php

```
<html>
<head>
<title>PHP Coding Style</title>
</head>
<body>
<?php
echo '
<form action="next.php" method="post">
<input name="check1" type="checkbox"
value="' . $Check1value . '">
<input name="text1" type="text"
value="' . $text1 . '" size="20">
<input name="submit" type="submit" value="send it">
</form>
';
?>
</body>
</html>
```

# Self-processing pages

temperature.php

```
<html>
<head><title>temperature conversion</title></head>
<body>

<?php
    $fahrenheit = $_GET['fahrenheit'];
    if (!is_null($fahrenheit)) {
        $celsius = ($fahrenheit - 32) * 5/9.;
        echo($fahrenheit . ' degrees Fahrenheit is the same as '
        . $celsius . ' Celsius. ');
        echo('<br /><br />');
    }
?>

<form action="temperature.php" method="GET">
    Fahrenheit temperature:
    <input type="text" name="fahrenheit" />
    <input type="submit" value="">> convert to Celsius" />
</form>

</body>
</html>
```

# Creating memory

Dynamic web pages are built at runtime.

User input can be processed and direct feedback returned.

In order to remember data provided by users, we need a method to continue to store data from one page to another.

Besides using a database, there are two other ways to store data:

Cookies

External files (e.g. text, html, etc.)

# Cookies

Cookies are one way for storing data in the user's browser.

They are normally used to track or identify returning users.

A cookie requires a unique *name*, a *value*, an *expiration date*

(if not set, the cookie expires when quitting the browser),

a *path*, and a *domain* (only cookies from that domain under this path will be returned).

You can set cookies using the `setcookie()` function.

Cookies are part of the HTTP header, so `setcookie()` must be called before any output is sent to the browser or you will get an error message.

# Setting cookies

```
<?php
    setcookie ("161B_email", "your@email.com",
        time()+(60*60*24*10));
    setcookie ("161B_cookie_date", strftime("%Y-%m-%d"),
        time()+(60*60*24*10));
?>
```

note: The function `time()` returns the current time measured in the number of seconds since the Unix Epoch (January 1 1970 00:00:00 GMT).

# Reading out cookies

The cookies are store in the array `$_COOKIE`.

The key to access the value of the cookie is the cookie name.

```
<?php
// Print an individual cookie
echo $_COOKIE["161B_email"];
echo $_COOKIE["161B_cookie_date"];
echo "<br>";

// Another way to debug/test is to view all cookies
print_r($_COOKIE);
?>
```

# Reading & writing to files

The method of writing to an external file can be an alternative for specific purposes. An example would be a simple text log file, often used for guest book applications.

Depending on the data format of the external file generated by PHP, it can also store XML, variable and values etc. and share the data with other dynamic pages or Flash movies.

The PHP function `include` allows to include the content of a specified file into an PHP page. E.g.

```
include 'variables.php';  
include 'header.html';  
include 'javaScript.js';  
include 'guestbook.txt';
```

# PHP functions

`fopen ()`

Opens a specified file

`fwrite ()`

Used to read and write into the file. Different modes define the way how the file is handled.

`fclose ()`

Closes the file

# fopen() modes

- 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.
- 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.
- x Create and open for writing only; place the file pointer at the beginning of the file. If the file already exists, the fopen() call will fail by returning FALSE and generating an error. If the file does not exist, attempt to create it. PHP 4.3.2 and later, and only works for local files.
- x+ Create and open for reading and writing; place the file pointer at the beginning of the file. If the file already exists, the fopen() call will fail by returning FALSE and generating an error. If the file does not exist, attempt to create it. PHP 4.3.2 and later, and only works for local files.

# Simple guestbook

```
<HTML>
<HEAD>
  <TITLE>Simple Guestbook</TITLE>
</HEAD>

<BODY>
  <FORM ACTION="guestbook.php" METHOD="post">
    <P>Date</P>
    <INPUT TYPE="text" NAME="mydate" VALUE="">
    <P>Text</P>
    <TEXTAREA NAME="mytext" COLS="72" ROWS="5"></textarea>
    <BR>
    <INPUT TYPE="submit" VALUE="send">
  </FORM>

  <?php

  // defining the file where the data is stored. Check writing permissions!
  $source = "guestbook_data.txt";
  // checking if the date has been provided
  if(isset($mytext)) {
    // opening the file, "a" stands for
    $file = fopen($source, "a");
    fwrite($file, "<HR>" . $mytext . "<BR><BR>" . $mydate . "<BR>");
    fclose($file);
  } else {
  }
  include($source);

  ?>
</BODY>
</HTML>
```

# Strings

Double quotes and single quotes:

If a string is defined in a double quote it is subject to variable interpolation, which means the process of replacing the variable names in the string with the values of those variables.

```
$who = 'Tyler';  
$where = 'here';  
echo "$who was $where";
```

-> Tyler was here

```
echo '$who was $where';
```

-> \$who was \$where

# substr()

```
$name = 'Fred Flintstone';  
echo substr($name, 6, 4);
```

-> lint

```
echo substr($name, 11);
```

-> tone

# strlen()

```
$mystring = 'Hello, world';  
$length = strlen($string); // $length is 12
```

# substr\_replace()

```
$greeting = "good morning folks";  
$farewell = substr_replace($greeting, "bye", 5, 7);
```

-> good bye folks

If you use the length value of 0 (last parameter), you can insert into a string without deleting any characters

# Searching strings

`strpos()` function finds the first occurrence of a small string in a larger string

`strrpos()` finds the last occurrence of a character in a string,

```
$record = "Fred,Flintstone,35,Wilma";  
$pos = strrpos($record, ","); // find last comma  
echo ("The last comma in the record is at $pos");
```

-> The last comma in the record is at position 18

see also: `strstr()`, `strchr()`, `strspn()`, `strcspn()`

# Concatenating strings

```
$my_string = "The beginning ";  
$my_string .= "of the end";  
echo $my_string;
```

-> The beginning of the end