

## CS320 Web and Internet Programming

Introduction to PHP

Chengyu Sun  
California State University, Los Angeles

## PHP

- ◆ *Personal Home Page Tools* by Rasmus Lerdorf in 1995
- ◆ PHP: Hypertext Preprocessor by Andi Gutmans and Zeev Suraski
  - 1998 PHP 3
  - 2000 PHP 4
  - 2004 PHP 5
- ◆ PHP usage - <http://www.php.net/usage.php>
  - 20+ million domains
  - 1.2+ million ip addresses

## Hello World

```
<html>
<head><title>PHP Hello World</title></head>
<body>

<?php
echo "<p>Hello World!</p>";
?>

</body>
</html>
```

## Development Environment

- ◆ Text editor + Web Server + PHP
  - On CS3
    - ↳ `$HOME/public_html/hello.php`
    - ↳ <http://cs3.calstatela.edu/~cs320stu31/hello.php>
- ◆ Eclipse PHP IDE
  - <http://www.eclipse.org/pdt/>
  - Zend Debugger
    - ↳ <http://downloads.zend.com/pdt>

## PHP as a Programming Language

- ◆ Comments
- ◆ Literals
- ◆ Variables
- ◆ Operators
- ◆ Expressions
- ◆ Statements
- ◆ Functions
- ◆ Classes

## Comments

- ◆ Script style: `#` single-line comment
- ◆ C++ style: `//` single-line comment
- ◆ C style:
  - `/*` single-line comment `*/`
  - `/*` multiple-line comments `*/`

## Literals

- ◆ 123, 0123, 0x123
- ◆ 12.3
- ◆ "123", '123'
- ◆ true, false
- ◆ null

## Heredoc Syntax for String

- ◆ A "Here document" Example:

```
echo <<<HTML
Username: <input type="text" name="username" /> <br />
Password: <input type="password" name="password" /> <br />
<input type="submit" name="submit" value="Login" />
HTML;
```

## Variables and Constants

- ◆ Starts with a \$
- ◆ Variable names are *case-sensitive*
- ◆ Variable *reference* &
- ◆ Constant
  - `define( name, value )`
  - E.g. `define( TITLE, "PHP Test Page" )`

## Example: Variable and Variable Reference

<pre>\$a = "hello"; \$b = "world";  \$a = \$b; \$b = 10;  echo \$a;</pre>	<pre>\$a = "hello"; \$b = "world";  \$a = &amp;\$b; \$b = 10;  echo \$a;</pre>
---	--

## Operators

- |  |  |
|--|--|
| ◆ Arithmetic <ul style="list-style-type: none"><li>▫ +, -, *, /, %</li></ul>                               | ◆ Concatenation <ul style="list-style-type: none"><li>▫ .</li></ul>                                      |
| ◆ Assignment <ul style="list-style-type: none"><li>▫ =</li><li>▫ +=, -=, *=, /=, %=</li><li>▫ .=</li></ul> | ◆ Comparison <ul style="list-style-type: none"><li>▫ ==, !=</li><li>▫ &gt;, &gt;=, &lt;, &lt;=</li></ul> |
| ◆ Increment/decrement <ul style="list-style-type: none"><li>▫ ++, --</li></ul>                             | ◆ Logical <ul style="list-style-type: none"><li>▫ &amp;&amp;,   , !</li></ul>                            |
|  | ◆ Conditional <ul style="list-style-type: none"><li>▫ ?:</li></ul>                                       |

## Example: Operators

```
$a = "hello";
$b = "world";

if( $a < $b ) echo "hello &lt; world <br>";

$c = $a + " " + $b;
echo $c;

$c = $a . " " . $b;
echo $c;

$a .= $b;
```

## Control Statements

### ◆ Branch

- if
- if ... else
- if...elseif...else

### ◆ Switch

- switch

### ◆ Loop

- while
- do ... while
- for
- foreach

### ◆ Break and continue

- break
- continue

## Example: Switch Statement

```
switch( $fruit )
{
  case "apple":
    echo "The fruit is an apple";
    break;

  case "pear":
    echo "The fruit is a pear.";
    break;

  default:
    echo "The fruit is not an apple or a pear.";
}
```

## Arrays

### ◆ *Associative* arrays, like *Map* in Java

- <key, value>
- Key must be either integer or string

## Example: Arrays

```
$courses = array( "cs320" => "web", "cs122" => "SQL" );

$strings["a"] = "abc";
$strings[1] = "def";

# when key is omitted

$numbers1 = array( 1, 2, 3, 4, 5 );
$numbers2 = array( 1 => 1, 2 => 2, 3, 4, 5 => 5 );

$strings[] = "xyz";
$strings[] = "uvw";
```

## Example: Access Array Elements

```
echo $numbers1[2], "<br>";
echo $courses["cs320"], "<br>";

foreach( $courses as $key => $value )
  echo "$key, $value <br>";

foreach( $numbers as $n )
  echo "$n <br>";

for( $i = 0 ; $i < count($numbers) ; ++$i )
  echo $numbers[$i];
```

## Some Useful Array Functions

### ◆ Remove an element

- unset()

### ◆ Number of elements

- count(), sizeof()

### ◆ Sort

- By value: asort(), arsort()
- By key: ksort(), krsort()

## Example: Array Functions

```
foreach( $strings as $key => $value )
    echo $key, " ", $value, "<br>";

asort($strings);

foreach( $strings as $key => $value )
    echo $key, " ", $value, "<br>";

ksort($strings);

foreach( $strings as $key => $value )
    echo $key, " ", $value, "<br>";
```

## Functions

```
function sum( $op1, $op2 )
{
    return $op1 + $op2;
}

function say_hello()
{
    echo "hello.";
}
```

## About Functions

- ◆ Functions can be defined pretty much anywhere in the code
  - E.g. in other functions or in an if statement
- ◆ Functions have global scope
- ◆ No function overloading, and no way to undefine or redefine a function
- ◆ Support *variable functions*

## Example: Variable Function

```
function sum ($a, $b)
{
    return $a + $b;
}

$v = "s"."u"."m";

echo $v(1,3);
```

## Basic Class Syntax

```
class GuestBookEntry {

    private $name;
    public $comment;

    function __construct( $name, $comment )
    {
        $this->name = $name;
        $this->comment = $comment;
    }

    function getName() { return $this->name; }
    function setName( $name ) { $this->name = $name; }
}
```

## Using Objects

```
$entry = new GuestBookEntry( "cysun", "hello" );

echo $entry->getName(), " says ", $entry->comment;
```

## Include Other Files

- ◆ `include()`, `include_once()`
- ◆ `require()`, `require_once()`

## PHP as a Web Programming Language

- ◆ Handle HTTP requests
- ◆ Generate HTTP response
- ◆ Session tracking
- ◆ Database access

## Example: GuestBook without Database

- ◆ Store *an array of GuestBookEntry* in session scope – there is no application scope in PHP

## Pre-defined Variables

- ◆ `$_GET`
- ◆ `$_POST`
- ◆ `$_REQUEST`
- ◆ `$_COOKIE`
- ◆ `$_SESSION`
- ◆ `$_SERVER`

<http://www.php.net/manual/en/reserved.variables.php>

## Access Request and Session Variables

```
if( ! empty($_REQUEST["submit"]) )
{
    $name = $_SESSION["name"];
    if( empty($name) )
    {
        $name = $_REQUEST["name"];
        $_SESSION["name"] = $name;
    }
}
```

## Override Default Settings

<u>php.ini</u>	<u>code</u>
<code>display_errors = Off</code>	<code>ini_set("display_errors", 1);</code>
<code>error_reporting = E_ALL</code>	<code>error_reporting(E_ERROR);</code>
<code>session.auto_start = 0</code>	<code>session_start(); = 0</code>

## Example: GuestBook with Database

### ◆ Using MySQL database

```
$connection = mysql_connect("localhost", "cs320stu31", "abcd");  
mysql_select_db("cs320stu31", $connection);  
$result = mysql_query("select * from guest_book");
```

```
$entries = array();  
while( $row = mysql_fetch_array($result) )  
    $entries[] =  
        new GuestBookEntry( $row["name"], $row["comments"] );
```

```
mysql_close( $connection );
```

## Handle Query Results

\$result  $\xrightarrow{\text{mysql\_fetch\_array()}}$  \$row

name	comment
cysun	Hello
steve	Nice page!

```
"name" => "cysun"  
"comment" => "Hello"  
1 => "cysun"  
2 => "Hello"
```

## Variations of mysql\_fetch\_array()

### ◆ Fetch a row as an associative array

- mysql\_fetch\_array(\$result, MYSQL\_ASSOC)
- mysql\_fetch\_assoc(\$result)

### ◆ Fetch a row as an integer-indexed array

- mysql\_fetch\_array(\$result, MYSQL\_NUM)
- mysql\_fetch\_row(\$result)

## More MySQL Functions

- ◆ <http://www.php.net/manual/en/book.mysql.php>

## About PHP

### ◆ Advantages

- Easy to learn
- Easy to use
- Dynamic typing
- Web-friendly language features

### ◆ Disadvantages??

## Resources

- ◆ [www.php.net](http://www.php.net)
  - PHP Manual - <http://www.php.net/manual/en/index.php>