

AN INTRODUCTION TO PHP

Qiang Xu and Yuanling Ren



1. What Is PHP

- Programming language created to enable web developers to quickly create dynamic web applications
- PHP stands for "PHP: hypertext preprocessor"
- Html-embedded programming language syntactically similar to C, Perl, and java



2. What a PHP Script Looks Like

```
<HTML>
<HEAD>
<TITLE>A Simple PHP Example</TITLE>
</HEAD>
<BODY>
<?php echo "Hello from PHP!" ?>
</BODY>
</HTML>
```



PHP preprocessor execute all code between <?php and ?> tag and return results as text

```
<HTML>
<HEAD>
<TITLE>A Simple PHP Example</TITLE>
</HEAD>
<BODY> Hello from PHP!</BODY>
</HTML>
```



3. The Syntax of the Language

similar to C, Perl, and Java

● Type

floating-point, integer, string,
array, and object

● Variable

```
$i=5; $f=1.2;
$s="This is a simple string.";
$y[0] = "red"; $y[1] = "green";
$color["blue"] = "#0000FF";
$color["green"] = "#00FF00";
```



● Constant

Predefined constant

HTTP_HOST, HTTP_USER_AGENT ...

Define your own constant

```
<?php
    define ("aString", "xxxxxx");
    define("aNumber", 1);
    print(aNumber . "constant with value
        of" . aString . "<br>");
?>
```



● Functions

Arguments can be passed by value
and by reference

```
<?php
    function ReturnSum(&$a, $b) {
        $c = $a + $b;
        return $c;
    }
?>
```



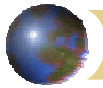
● OO/Classes

support class creation with syntax
similar to C++

Constructor exist, no destructor

Requires \$this variable within classes
to refer to member variables and
methods

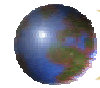
Single inheritance, but not multiple



```
<?php

class ShoppingBasket {
    var $item;
    var $value;

    function ShoppingBasket ($initValue) {
        $this->value = $initValue;
    }
}
```

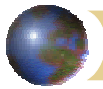


```
function AddItem($aName, $aV, $aQty){
    $this->item[$aName]["Q"] = $aQty;
    $this->item[$aName]["V"] = $aV;
    $this->value += $aV * $aQty;
    return true;
}

...
}

$aBasket = new ShoppingBasket(3.5);
$aBasket->AddItem("Rose", 10, 2);
$aBasket->PrintBasket();

?>
```



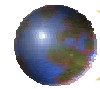
● Pattern Matching

PHP supports two types of pattern-matching functions:

Perl-compatible regular expression functions, function names all start with preg_ string

POSIX-style functions:

ereg(), erigi(), ereg_replace(),
eregi_replace(), split()



4. Forms and Cookies

● Forms

```
<form action = "post1.phtml" method =
"post"> ..... </form>
```

Form values stored in PHP-Generated variables, e.g, HTTP_POST_VARS

Form data validation

regular expression functions in PHP programs



validator class

One of beautiful thing about PHP is its open-source. <http://www.thewebmasters.net> provides great open-source classes and source modules. For data validation, Validator class provides an array of functions, e.g, `is_email()`, `is_url()`, `is_phone()`

- Cookies

`setcookie()`, `HTTP_COOKIE_VARS`.



5. Working With File Objects

- provide C-like functions for managing file objects

```
fopen(),fclose(), fput(), popen()  
fsockopen("208.129.36.164", 17),
```

- upload files whose names are submitted in form data, i.e., it can read a file of client and save it in server side.



6. Working With Databases

- support

Adabas D, dBase, Empress, FilePro, Informix, InterBase, mSQL, MySQL, Oracle, postgresQL, Solid, Sybase, Velocis, Unix dbm, Microsoft SQL Server, ODBC

- With inclusion of ODBC, PHP probably can be used to access any available DBMS.



- An example using MySQL

```
<?php  
    $aLink = mysql_connect("xx.xx.com",  
                          "user", "passwd" );  
    if( !empty($aLink) ) {  
        if( mysql_select_db( "mydb",  
                             $aLink ) == True ) {  
            $Qry = "select * from E";  
            $R = mysql_query($Qry, $aLink);
```



```
if($R == True ) {
  while($aRow =
    mysql_fetch_array($R)){
    $aFName = $aRow["first"];
    $aPos = $aRow["position"];
    print( "$aFName, $aPos <br>" );
  }
}
?>
```



7. Debugging

- `assert()`
`assert('is_array($anArray)');`
- `error_log()`
`error_log("Error here!", 0);`
`error_log("Err!", 3, "/tmp/error.log");`
`error_log("Error!", 1, "yyy@xxx.xx",`
`"From: zzz@myhost.com\r\n");`



- write your own error handler

```
function myErrorHandler( $aErrorNo,
$aErrorStr, $aFile, $aLine, $aContext)
{ }

set_error_handler("myErrorHandler");
```



8. Code Reuse

- Reusing code is significant consideration in any programming language
- PHP provides means for including external files and creating OO classes, it facilitates code reuse.
- With a little bit effort, you can reuse code written in C/C++, Java, COM, and PHP



9. Cool PHP

- Send Non_HTML files to browser
- full scripting language that can be used for any programming task
- support The Web Distributed Data Exchange (WDDX)
- support sockets and network protocols , developing network monitoring utility in PHP is straightforward



- support template system
- For E-Commerce, PHP provides interfaces to several payment processing systems including CyberCash, VeriSign, and CCVS



10. Why Better than Its Alternatives

- PHP is free
- Open Source software
download complete source code
- broad platform support



- syntax and structure resemble C with complexity (e.g. memory management, pointers, and strong typing) taken out.
- support direct access to Java objects on any system with Java Virtual Machine available, as well as Distributed COM on Windows.
- modifiable. designed to allow extension of functionality. It's coded in C and provides a well_defined API



- Enforce cleaner separation of layout and application logic
- Abundant Connectivity
- support most current Internet standards: IMAP, FTP, POP, XML, WDDX, LDAP, NIS, and SNMP
- can be compiled as stand-alone script interpreter, and handle simple system administration tasks



11. Conclusion

- PHP is full-fledged programming language that will enable you to create Web applications with all functionality you need