PHP Basic

Fajiang Yu, fjyu@whu.edu.cn

School of Computer, Wuhan University

2017.4

Agenda

1. Introduction
2. Variable and Operation
3. Control Structure
4. Function

Agenda

1. Introduction
2. Variable and Operation
3. Control Structure
4. Function

Introduction

- PHP stands for “PHP: Hypertext Preprocessor”
- PHP is server side scripting system
- PHP manual: `http://www.php.net/`

History

- Started as a Perl hack in 1994 by Rasmus Lerdorf (to handle his resume), developed to PHP/FI 2.0
- By 1997 up to PHP 3.0 with a new parser engine by Zeev Suraski and Andi Gutmans
- Version 5.0, rewritten by Zend (www.zend.com) to include a number of features, such as an object model
- PHP is one of the premier examples of what an open source project can be

Introduction

LAMP: Linux, Apache, MySql, PHP

- Apache httpd, `./configure --enable-so`
  - `LoadModule php5_module modules/libphp5.so`
  - `LoadModule php7_module modules/libphp7.so`
- PHP
  - `./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql`
  - `./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-mysqli`
  - apxs - APache eXtenSion tool

Hello, World!

- Typically file ends in .php – this is set by the web server configuration Separated in files with the <?php ?> tag
- PHP commands can make up an entire file, or can be contained in html
- Program lines end in ";" or you get an error
- Server recognizes embedded script and executes Result is passed to browser, source isn't visible

Hello, World!

```
<p>
<?php $myvar = "Hello World!";
   echo $myvar;
 ?>
</p>

<?php
   echo "<html><head><title>Howdy</title>
   ......
 ?>
```

phpinfo()

- The `phpinfo()` function shows the php environment
- Use this to read system and server variables, setting stored in php.ini, versions, and modules
- Notice that many of these data are in arrays
- This is the first script we should write

comments

```
<?php
   // this is a single-line comment
   /* and this is a
      multi-line
      comment */
 ?>
```

Agenda

1 Introduction

2 Variable and Operation

3 Control Structure

4 Function

variable

- it automatically determines variable type by the context in which it is being used
- a variable name is preceded by a dollar ($) symbol and must begin with a letter or underscore, optionally followed by more letters, numbers andor underscores
- For example, $popeye, $one and $INCOME are all valid PHP variable names, while $123 and $48hrs are invalid
- Note that variable names in PHP are case sensitive, so $me is different from $Me or $ME

array

```
<?php
   // define an array
   $pizzaToppings = array('onion', 'tomato', 'cheese', 'anchovies',
                          'ham', 'pepperoni');
   print_r($pizzaToppings);
 ?>

<?php
   // define an array
   $fruits = array('red' => 'apple', 'yellow' => 'banana',
                   'purple' => 'plum', 'green' => 'grape');
   print_r($fruits);
 ?>
```

array

```
<?php
   // define an array
   $pasta = array('spaghetti', 'penne', 'macaroni');
 ?>
```

```
<?php
   // define an array
   $pasta[0] = 'spaghetti';
   $pasta[1] = 'penne';
   $pasta[2] = 'macaroni';
 ?>
```

```
<?php
   // define an array
   $menu['breakfast'] = 'bacon and eggs';
   $menu['lunch'] = 'roast beef';
   $menu['dinner'] = 'lasagna';
 ?>
```

array

```
<?php
   // add an element to an array
   $pizzaToppings[3] = 'green olives';
 ?>

<?php
   // modify an array
   $pizzaToppings[4] = 'chicken';
 ?>

<?php
   // modify an array
   $menu['lunch'] = 'steak with mashed potatoes';
 ?>
```

array

```
<?php
   // define an array
   $menu = array('breakfast' => 'bacon and eggs',
                 'lunch' => 'roast beef',
                 'dinner' => 'lasagna');
   /* returns the array ('breakfast', 'lunch', 'dinner')
      with numeric indices */
   $result = array_keys($menu);
   print_r($result);
   print "<br />";

   /* returns the array ('bacon and eggs', 'roast beef', 'lasagna')
      with numeric indices */
   $result = array_values($menu);
   print_r($result);
 ?>
```

array

```
<?php
   // define an array
   $pasta = array('spaghetti', 'penne', 'macaroni');
   // add an element to the end
   array_push($pasta, 'tagliatelle');
   print_r($pasta);
 ?>
```

```
<?php
   // define an array
   $pasta = array('spaghetti', 'penne', 'macaroni');
   // remove an element from the end
   array_pop($pasta);
   print_r($pasta);
 ?>
```

array

```
<?php
   // define an array
   $pasta = array('spaghetti', 'penne', 'macaroni');
   // take an element off the top
   array_shift($pasta);
   print_r($pasta);
 ?>
```

```
<?php
   // define an array
   $pasta = array('spaghetti', 'penne', 'macaroni');
   // add an element to the beginning
   array_unshift($pasta, 'tagliatelle');
   print_r($pasta);
 ?>
```

array

```
<?php
   // define CSV string
   $str = 'red, blue, green, yellow';
   // split into individual words
   $colors = explode(', ', $str);
   print_r($colors);
 ?>
```

```
<?php
   // define array
   $colors = array ('red', 'blue', 'green', 'yellow');
   // join into single string with 'and'
   // returns 'red and blue and green and yellow'
   $str = implode(' and ', $colors);
   print $str;
 ?>
```

array

```
<?php
   // define an array
   $pasta = array('spaghetti', 'penne', 'macaroni');
   // returns the array sorted alphabetically
   sort($pasta);
   print_r($pasta);
   print "<br />";

   // returns the array sorted alphabetically in reverse
   rsort($pasta);
   print_r($pasta);
 ?>
```

form

```
<html>
<head></head>

<body>
<form action="message.php" method="post">
   Enter your message: <input type="text" name="msg" size="30">
   <input type="submit" value="Send">
</form>
</body>

</html>
```

form

```
<html>
<head></head>

<body>
<?php
   // retrieve form data
   $input = $_POST['msg'];
   // use it
   echo "You said: <i>$input</i>";
 ?>
</body>

</html>
```

operation

```
<html>
<head></head>

<body>
Agent: So who do you think you are, anyhow? <br />
<?php
   // define variables
   $name = 'Neo';
   $rank = 'Anomaly';
   $serialNumber = 1;
   // print output
   echo "Neo: I am <b>$name</b>, the <b>$rank</b>.
         You can call me by my serial number,
         <b>$serialNumber</b>.";
 ?>
</body>

</html>
```

operation

```
<?php
   $age = $dob + 15;
 ?>
```

```
<?php
   $angle1 = $angle2 = $angle3 = 60;
 ?>
```

```
<?php
   $identity = 'James Bond';
   $car = 'BMW';
   // this would contain the string "James Bond drives a BMW"
   $sentence = "$identity drives a $car";
   echo $sentence;
 ?>
```

operation

```
<html>
<head> </head>
<body>
<?php
   // set quantity
   $quantity = 1000;
   // set original and current unit price
   $origPrice = 100;
   $currPrice = 25;
   // calculate difference in price
   $diffPrice = $currPrice - $origPrice;
   // calculate percentage change in price
  $diffPricePercent = (($currPrice - $origPrice) * 100)/$origPrice
 ?>
```

operation

```
<table border="1" cellpadding="5" cellspacing="0">
<tr>
<td>Quantity</td>
<td>Cost price</td>
<td>Current price</td>
<td>Absolute change in price</td>
<td>Percent change in price</td>
</tr>
<tr>
<td><?php echo $quantity ?></td>
<td><?php echo $origPrice ?></td>
<td><?php echo $currPrice ?></td>
<td><?php echo $diffPrice ?></td>
<td><?php echo $diffPricePercent ?>%</td>
</tr>
</table>
</body>
</html>
```

operation

```
<?php
   // this...
   $a = 5;
   $a = $a + 10;

   // ... is the same as this
   $a = 5;
   $a += 10;
 ?>
```

operation

```
<?php
   // define $total as 10
   $total = 10;
   // increment it
   $total++;
   // $total is now 11
   echo $total;
 ?>
```

```
<?php
   // define $total as 10
   $total = 10;
   // decrement it
   $total--;
   // $total is now 9
   echo $total;
 ?>
```

string operation

```
<?php
   // set up some string variables
   $a = 'the';
   $b = 'games';
   $c = 'begin';
   $d = 'now';
   // combine them using the concatenation operator
   // this returns 'the games begin now<br />'
   $statement = $a.' '.$b.' '.$c.' '.$d.'<br />';
   print $statement;
   // and this returns 'begin the games now!'
   $command = $c.' '.$a.' '.$b.' '.$d.'!';
   print $command;
 ?>
```

string operation

```
<?php
   // define string
   $str = 'the';
   // add and assign
   $str .= 'n';
   // str now contains "then"
   echo $str;
 ?>
```

comparison

```
<?php
   /* define some variables */
   $mean = 9;
   $median = 10;
   $mode = 9;

   // less-than operator
   // returns true if left side is less than right
   // returns true here
   $result = ($mean < $median);
   print "result is $result<br />";

   // greater-than operator
   // returns true if left side is greater than right
   // returns false here
   $result = ($mean > $median);
   print "result is $result<br />";
```

comparison

```
// less-than-or-equal-to operator
// returns true if left side is less than or equal to right
// returns false here
$result = ($median <= $mode);
print "result is $result<br />";
// greater-than-or-equal-to operator
// returns true if left side is greater than or equal to right
// returns true here
$result = ($median >= $mode);
print "result is $result<br />";
// equality operator
// returns true if left side is equal to right
// returns true here
$result = ($mean == $mode);
print "result is $result<br />";
```

comparison

```
  // not-equal-to operator
  // returns true if left side is not equal to right
  // returns false here
  $result = ($mean != $mode);
  print "result is $result<br />";

  // inequality operator
  // returns true if left side is not equal to right
  // returns false here
  $result = ($mean <> $mode);
  print "result is $result";
?>
```

comparison

PHP 4.0 also introduced a new comparison operator, which allows you to test both for equality and type: the === operator

```
<?php
   /* define two variables */
   $str = '10';
   $int = 10;

   /* returns true, since both variables contain the same value */
   $result = ($str == $int);
   print "result is $result<br />";
```

comparison

```
  /* returns false, since the variables are not of the same type
     even though they have the same value */
  $result = ($str === $int);
  print "result is $result<br />";

 /* returns true, since the variables are the same type and value */
  $anotherInt = 10;
  $result = ($anotherInt === $int);
  print "result is $result";
?>
```

logical operators

```php
<?php
   /* define some variables */
   $auth = 1;
   $status = 1;
   $role = 4;

   /* logical AND returns true if all conditions are true */
   // returns true
   $result = (($auth == 1) && ($status != 0));
   print "result is $result<br />";

   /* logical OR returns true if any condition is true */
   // returns true
   $result = (($status == 1) || ($role <= 2));
   print "result is $result<br />";
```

logical operators

```
  /* logical NOT returns true if the condition is
     false and vice-versa */
  // returns false
  $result = !($status == 1);
  print "result is $result<br />";

  /* logical XOR returns true if either of two conditions are
     true, or returns false if both conditions are true */
  // returns false
  $result = (($status == 1) xor ($auth == 1));
  print "result is $result<br />";
?>
```

Agenda

1. Introduction

2. Variable and Operation

3. Control Structure

4. Function

if

```
<html>
<head></head>
<body>

<form action="ageist.php" method="post">
   Enter your age: <input name="age" size="2">
</form>

</body>
</html>
```

if

```
<html>
<head></head>
<body>
<?php
   // retrieve form data
   $age = $_POST['age'];
   // check entered value and branch
   if ($age >= 21) {
      echo 'Come on in, we have alcohol and music awaiting you!';
   }
   if ($age < 21) {
      echo "You're too young for this club,
            come back when you're a little older";
   }
 ?>
</body>
</html>
```

if

```
<html>
<head></head>
<body>
<?php
   // retrieve form data
   $age = $_POST['age'];
   // check entered value and branch
   if ($age >= 21) {
      echo 'Come on in, we have alcohol and music awaiting you!';
   }
   else {
      echo "You're too young for this club,
            come back when you're a little older";
   }
 ?>
</body>
</html>
```

if

```
<?php
   if ($numTries > 10) {
      $msg = 'Blocking your account...';
   }
   else {
      $msg = 'Welcome!';
   }
 ?>

<?php
   $msg = $numTries > 10 ?
          'Blocking your account...' : 'Welcome!';
 ?>
```

if

```
<?php
   if ($day == 'Thursday') {
      if ($time == '0800') {
         if ($country == 'UK') {
            $meal = 'bacon and eggs';
         }
      }
   }
 ?>
```

```
<?php
   if ($day == 'Thursday' && $time == '0800' &&
       $country == 'UK') {
      $meal = 'bacon and eggs';
   }
 ?>
```

if

```
<html>
<head></head>

<body>
<h2>Today's Special</h2>
<p>
<form method="get" action="cooking.php">
<select name="day">
   <option value="1">Monday/Wednesday
   <option value="2">Tuesday/Thursday
   <option value="3">Friday/Sunday
   <option value="4">Saturday
</select>
<input type="submit" value="Send">
</form>
</body>

</html>
```

if

```
<html>
<head></head>

<body>
<?php
   // get form selection
   $day = $_GET['day'];
   // check value and select appropriate item
   if ($day == 1) {
      $special = 'Chicken in oyster sauce';
   }
   elseif ($day == 2) {
      $special = 'French onion soup';
   }
   elseif ($day == 3) {
     $special = 'Pork chops with mashed potatoes and green salad';
   }
```

if

```
   else {
      $special = 'Fish and chips';
   }
 ?>

<h2>Today's special is:</h2>
<?php echo $special; ?>
</body>
</html>
```

switch

```
<html>
<head></head>

<body>
<?php
   // get form selection
   $day = $_GET['day'];
   // check value and select appropriate item
   switch ($day) {
      case 1:
         $special = 'Chicken in oyster sauce';
         break;
      case 2:
         $special = 'French onion soup';
         break;
      case 3:
       $special = 'Pork chops with mashed potatoes and green salad';
         break;
```

switch

```
      default:
         $special = 'Fish and chips';
         break;
   }
 ?>
<h2>Today's special is:</h2>
<?php echo $special ?>
</body>
</html>
```

Combine creating and processing forms

```
<html>
<head></head>

<body>
<?php
   /* if the "submit" variable does not exist,
      the form has not been submitted - display initial page */
   if (!isset($_POST['submit'])) {
 ?>
      <form action="<?php echo $_SERVER['PHP_SELF']; ?>"
            method="post">
         Enter your age: <input name="age" size="2">
         <input type="submit" name="submit" value="Go">
      </form>
```

Combine creating and processing forms

```
<?php
   }
   else {
     /* if the "submit" variable exists, the form has been
        submitted - look for and process form data */
     // display result
     $age = $_POST['age'];
     if ($age >= 21) {
       echo 'Come on in, we have alcohol and music awaiting you!';
     }
     else {
        echo 'You're too young for this club,
              come back when you're a little older';
     }
   }
 ?>
</body>
</html>
```

Combine creating and processing forms

```
<html>
<head></head>
<body>
<?php
   // check for submit
   if (!isset($_POST['submit'])) {
       // and display form
 ?>
      <form action="<?php echo $_SERVER['PHP_SELF']; ?>"
            method="POST">
         <input type="checkbox" name="artist[]"
                value="Bon Jovi">Bon Jovi
         <input type="checkbox" name="artist[]"
                value="N'Sync">N'Sync
         <input type="checkbox" name="artist[]"
                value="Boyzone">Boyzone
```

Combine creating and processing forms

```
  <input type="checkbox" name="artist[]"
         value="Britney Spears">Britney Spears
  <input type="checkbox" name="artist[]"
         value="Jethro Tull">Jethro Tull
  <input type="checkbox" name="artist[]"
         value="Britney Spears">Britney Spears
  <input type="checkbox" name="artist[]"
         value="Jethro Tull">Jethro Tull
  <input type="checkbox" name="artist[]"
         value="Crosby, Stills & Nash">
         Crosby, Stills & Nash
  <input type="submit" name="submit" value="Select">
</form>
```

Combine creating and processing forms

```
<?php
   }
   else {
      // or display the selected artists
      // use a foreach loop to read and display array elements
      if (is_array($_POST['artist'])) {
         echo 'You selected: <br />';
         foreach ($_POST['artist'] as $a) {
            echo "<i>$a</i><br />";
         }
      }
      else {
         echo 'Nothing selected';
      }
   }
 ?>
</body>
</html>
```

loop

```
<html>

<head></head>

<body>
<form action="squares.php" method="POST">
   Print all the squares between 1 and
   <input type="text" name="limit" size="4" maxlength="4">
   <input type="submit" name="submit" value="Go">
</form>
</body>

</html>
```

loop

```
<html>
<head></head>
<body>
<?php
   // set variables from form input
   $upperLimit = $_POST['limit'];
   $lowerLimit = 1;
   // keep printing squares until lower limit = upper limit
   while ($lowerLimit <= $upperLimit) {
      echo ($lowerLimit * $lowerLimit).'&nbsp;';
      $lowerLimit++;
   }
   // print end marker
   echo 'END';
 ?>
</body>
</html>
```

loop

```
<html>
<head></head>
<body>
<?php
   // set variables from form input
   $upperLimit = $_POST['limit'];
   $lowerLimit = 1;
   // keep printing squares until lower limit = upper limit
   do {
      echo ($lowerLimit * $lowerLimit).'&nbsp;';
      $lowerLimit++;
   } while ($lowerLimit <= $upperLimit);
   // print end marker
   echo ' END';
 ?>
</body>
</html>
```

loop

```
<html>
<head>
<basefont face="Arial">
</head>

<body>
<?php
   // define the number
   $number = 13;
   // use a for loop to calculate tables for that number
   for ($x = 1; $x <= 10; $x++) {
      echo "$number x $x = ".($number * $x)."<br />";
   }
 ?>
</body>

</html>
```

loop

```
<html>
<head></head>
<body>
<form method="post"
      action="<?php echo $_SERVER['PHP_SELF']; ?>">
   Enter number of rows
   <input name="rows" type="text" size="4">
   and columns <input name="columns" type="text" size="4">
   <input type="submit" name="submit" value="Draw Table">
</form>
<?php
   if (isset($_POST['submit'])) {
      echo "<table width = 90% border = '1'
                    cellspacing = '5' cellpadding = '0'>";
      // set variables from form input
      $rows = $_POST['rows'];
      $columns = $_POST['columns'];
```

loop

```
      // loop to create rows
      for ($r = 1; $r <= $rows; $r++) {
         echo "<tr>";
         // loop to create columns
         for ($c = 1; $c <= $columns; $c++) {
            echo "<td>&nbsp;</td>";
         }
         echo "</tr>";
      }
      echo "</table>";
   }
 ?>
</body>
</html>
```

loop

```
<html>
<head></head>
<body>
My favourite bands are:
<ul>
<?php
   // define array
   $artists = array('Metallica', 'Evanescence',
                     'Linkin Park', 'Guns n Roses');
   // loop over it and print array elements
   for ($x = 0; $x < sizeof($artists); $x++) {
      echo '<li>'.$artists[$x];
   }
 ?>
</ul>
</body>
</html>
```

loop

```
<html>
<head></head>
<body>
My favourite bands are:
<ul>
<?php
   // define array
   $artists = array('Metallica', 'Evanescence',
                    'Linkin Park', 'Guns n Roses');
   // loop over it
   // print array elements
   foreach ($artists as $a) {
      echo '<li>'.$a;
   }
 ?>
</ul>
</body>
</html>
```

Agenda

1. Introduction

2. Variable and Operation

3. Control Structure

4. Function

function

```
<?php
   function myStandardResponse() {
      echo "Get lost, jerk!<br /><br />";
   }
   // on the bus
   echo "Hey lady, can you spare a dime? <br />";
   myStandardResponse();
   // at the office
   echo "Can you handle Joe's workload, in addition to your own,
         while he's in Tahiti for a month? You'll probably need
         to come in early and work till midnight, but we are
         confident you can handle it. Oh, and we can't pay you
         extra because of budgetary constraints...<br />";
   myStandardResponse();
   // at the party
   echo "Hi, haven't I seen you somewhere before?<br />";
   myStandardResponse();
 ?>
```

function with argument(s)

```
<?php
   // define a function
   function getCircumference($radius) {
      echo "Circumference of a circle with radius $radius is "
           .sprintf("%4.2f", (2 * $radius * pi()))."<br />";
   }

   // call a function with an argument
   getCircumference(10);
   // call the same function with another argument
   getCircumference(20);
 ?>
```

function with argument(s)

```
<?php
   // define a function
   function changeCase($str, $flag) {
       /* check the flag variable and branch the code */
       switch($flag) {
          case 'U':
             print strtoupper($str)."<br />";
             break;
          case 'L':
             print strtolower($str)."<br />";
             break;
          default:
             print $str."<br />";
             break;
      }
   }
```

function with argument(s)

```
  // call the function
  changeCase("The cow jumped over the moon", "U");
  changeCase("Hello Sam", "L");
?>
```

return value

```
<?php
   // define a function
   function getCircumference($radius) {
      // return value
      return (2 * $radius * pi());
   }

   /* call a function with an argument and
      store the result in a variable */
   $result = getCircumference(10);

   /* call the same function with another argument
      and print the return value */
   print getCircumference(20);
 ?>
```

return array

```
<?php
   /* define a function that can accept a list of email addresses */
   function getUniqueDomains($list) {
      /* iterate over the list, split addresses
         and add domain part to another array */
      $domains = array();
      foreach ($list as $l) {
         $arr = explode("@", $l);
         $domains[] = trim($arr[1]);
      }
      // remove duplicates and return
      return array_unique($domains);
   }
```

return array

```
  // read email addresses from a file into an array
  $fileContents = file("data.txt");
  /* pass the file contents to the function
     and retrieve the result array */
  $returnArray = getUniqueDomains($fileContents);
  // process the return array
  foreach ($returnArray as $d) {
     print "$d, ";
  }
?>
```

default argument(s)

```
<?php
   // define a function
   function introduce($name="John Doe", $place="London") {
      print "Hello, I am $name from $place";
   }

   // call function
   introduce("Moonface");
 ?>
```

argument list

```
<?php
   // define a function
   function someFunc() {
      // get the arguments
      $args = func_get_args();
      // print the arguments
      print "You sent me the following arguments:";
      foreach ($args as $arg) {
         print " $arg ";
      }
      print "<br />";
   }

   // call a function with different arguments
   someFunc("red", "green", "blue");
   someFunc(1, "soap");
 ?>
```

argument list

```
<?php
   // define a function
   function someFunc() {
      // get the number of arguments passed
      $numArgs = func_num_args();
      // get the arguments
      $args = func_get_args();
      // print the arguments
      print "You sent me the following arguments: ";
      for ($x = 0; $x < $numArgs; $x++) {
         print "<br />Argument $x: ";
         /* check if an array was passed and,
            if so, iterate and print contents */
         if (is_array($args[$x])) {
            print " ARRAY ";
            foreach ($args[$x] as $index => $element) {
               print " $index => $element ";
            }
```

argument list

```
            }
            else {
                print " $args[$x] ";
            }
        }
    }

    // call a function with different arguments
    someFunc("red", "green", "blue", array(4,5), "yellow");
?>
```

global and local variable(s)

```
<?php
   // define a variable in the main program
   $today = "Tuesday";

   // define a function
   function getDay() {
      // define a variable inside the function
      $today = "Saturday";
      // print the variable
      print "It is $today inside the function<br />";
   }

   // call the function
   getDay();

   // print the variable
   print "It is $today outside the function";
 ?>
```

global and local variable(s)

```
<?php
   // define a function
   function getDay() {
      // define a variable inside the function
      $today = "Saturday";
   }

   getDay();
   print "Today is $today";
 ?>
```

global and local variable(s)

```
<?php
   // define a variable in the main program
   $today = "Tuesday";

   // define a function
   function getDay() {
      // make the variable global
      global $today;
      // define a variable inside the function
      $today = "Saturday";
      // print the variable
      print "It is $today inside the function<br />";
   }
```

global and local variable(s)

```
  // print the variable
  print "It is $today before running the function<br />";
  // call the function
  getDay();
  // print the variable
  print "It is $today after running the function";
?>
```

passing by reference

```
<?php
   // create a variable
   $today = "Saturday";

   // function to print the value of the variable
   function setDay(&$day) {
      $day = "Tuesday";
      print "It is $day inside the function<br />";
   }

   // call function
   setDay($today);
   // print the value of the variable
   print "It is $today outside the function";
 ?>
```

Thank You!
Any Questions?