Applications Development and Emerging Technologies

MODULE 1 SUBTOPIC 2

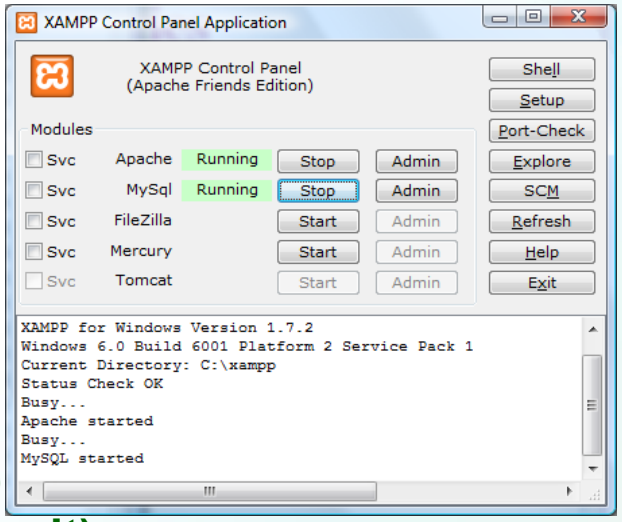
PHP Environment

Objectives

- To run PHP application on a web browser.
- To know the basic syntax of PHP for outputting to browser.
- To know the datatypes that are available on PHP and on how these datatypes are being used.
- To know some predefined type function for data manipulation.
- To know the proper casting values of a given variables.

Running PHP Scripts

XAMPP



1. Download XAMPP
   http://www.apachefriends.org/en/xampp.html
2. Install XAMPP
3. Run XAMPP Control
   - c:\xampp (by default)
   - Start the Apache and MySQL(for database)
4. Create a folder under c:\xampp\htdocs\ (by default)
   Note: All php files must be save on that folder
5. Test PHP script sample file.
   5. Open some internet browser
   6. Type localhost/[folder name] (by default)
   7. Select file from the directory list (if there's

Running PHP Scripts

```
<!-- welcome.php -->
<html>
<head>
<title>My PHP</title>
</head>
<body>

<?php
    echo 'Welcome to PHP';
?>

</body>
<html>
```

All php scripts must enclosed with

<?php … ?> (standard tag)

<? … ?> (short open tag)

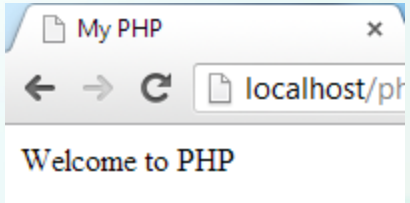
<script language="php">

….

</script> (script tag)

<% … %> (asp style tag)

OUTPUT:



Running PHP Scripts

opening tag for php script

Statement terminator

```
<?php
    echo 'Welcome to PHP';
?>
```

closing tag for php script

echo – is a language construct that use to output one or more string.

print – is a function also use for outputting strings.

PHP Comments

```
<?php
    // single line comment (C syntax)
    # single line comment (PERL style)
    /*
        multiline
        or block
        comments
    */
?>
```

PHP Variables

- Variables are used as a container for values that can be used and manipulate PHP scripts.
- Variables in PHP must begin with a $ symbol.
- Variable might contain a string, numbers, arrays and objects.
- Setting the identifier for a variable must follow some rules.
  1. First character can be '_' or a letter.
  2. It can only contain alpha-numeric characters and underscore (a-z,A-Z,0-9,_).
  3. Don't use white space in naming a variable.
- Variables in PHP is not explicitly declared
- Variables are case sensitive

Example

```
$age, $firstName, $totalSalary, $_myValue,
$str1, $tmp01
```

Using ‘ ‘ (pair single quote), “ “ (pair double quotes) and . (dot character) PHP

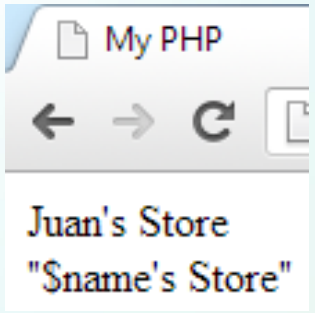
1. Strings inside the ‘ and ‘ are interpreted as literal strings
    1. To display a ‘ character (single quote) as string use the escape character \ (backslash).
2. Strings inside the “ and “ are interpreted as literal strings with some exceptions
    1. $ inside “ and “ are interpreted as variables thus it will display the value rather than the string that starts with $.
    2. To display the string that starts with $ and to display the double quote as string use the escape character \ (backslash).
3. To concatenate string values use the . (dot or period) character thus “abc“.”def” yields to a value “abcdef”

Using ‘ ‘ (pair single quote), “ “ (pair double quotes) and . (dot character) PHP

Example:

```php
<?php
    $name = "Juan";
    echo "$name's Store"."<br/>";
    echo '"$name\'s Store"'."<br/>";
    //statements below will produced errors
    //echo "displaying double quote" ";
    //echo "dispalying single quote ' ";
?>
```

Output:



Outputting Data

`echo` syntax

```
void echo(string argument1[,…string argumentN])
```

`print` syntax

```
int print(argument)
```

`printf` syntax

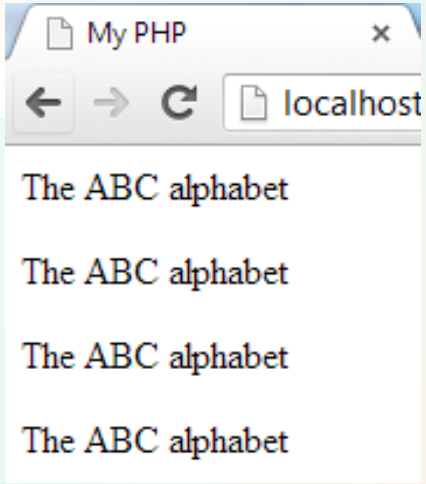
```
boolean printf(string format [, mixed args])
```

Outputting Data

Example:

```php
<?php
    //using echo
    echo "<p>The ABC alphabet</p>";
    echo "<p>","The ","ABC ","alphabet","</p>";
    //using print statement
    print("<p>The ABC alphabet</p>");
    //using printf statement
    printf("<p>%s</p>","The ABC alphabet");
?>
```

Output:



Commonly Used Type Specifiers for printf function

Type	Description
%b	Argument considered an integer; presented as a binary number
%c	Argument considered an integer; presented as a character corresponding to that ASCII value
%d	Argument considered an integer; presented as a signed decimal number
%f	Argument considered a floating-point number; presented as a floating-point number
%o	Argument considered an integer; presented as an octal number
%s	Argument considered a string; presented as a string
%u	Argument considered an integer; presented as an unsigned decimal number
%x	Argument considered an integer; presented as a lowercase hexadecimal number
%X	Argument considered an integer; presented as an uppercase hexadecimal number

Beginning PHP and MySQL 3rd Edition by: W. Jason Gilmore pp.100

PHP Datatypes

Datatypes

is the generic name assigned to any data sharing a common set of characteristics.

Scalar Datatypes

Capable of containing a single item of information
(Boolean, Integer, Float (float, double or real numbers), String)

Compound Datatypes

allow for multiple items of the same type to be aggregated under a single representative entity.
(Array and Objects)

Scalar Datatypes
Example:

```
//Integer
$a = 13;          //13
$a = -12345;      //-12345
$a = 012;         //10
$a = 0x1A;        //26
```

```
//float
$v = 3.2475;      //3.2475
$v = 2.0;         //2
$v = 8.7e4;       //87000
$v = 1.43E+10;    //143E+10
```

```
//Boolean
$a = true;   //$a is true
$a = 1;      //$a is true
$a = -1;     //$a is true
$a = 1056;   //$a is true
$a = 0;      //$a is false
$a = false;  //$a is false
```

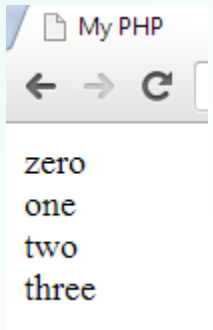
```
//string
$str = "Welcome to PHP";     //Welcome to PHP
$str = "do-re-mi";           //do-re-mi
$str = '$^expr%!';           //$^expr%!
$str = "123$%^789";          //123$%^789
```

Compound Datatypes

Example:

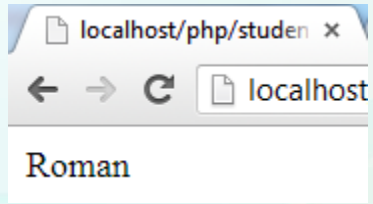
```php
<?php
    //arrays
    $arr_list[0] = "zero";
    $arr_list[1] = "one";
    $arr_list[2] = "two";
    $arr_list[3] = "three";
    for($i=0;$i<count($arr_list);$i++)
        echo $arr_list[$i]."<br/>";
?>
```

Output:




```php
<?php
    //objects
    class Student{
        private $_name;
        function setName($name){
            $this->_name = $name;
        }
        function getName(){
            return $this->_name;
        }
    }
    $stud = new Student;
    $stud->setName('Roman');
    echo $stud->getName();
?>
```

Output:



PHP Type Casting Operators

Cast Operators	Conversion
(array)	Array
(bool) or (boolean)	Boolean
(int) or (integer)	Integer
(int64)	64-bit integer (introduced in PHP 6)
(object)	Object
(real) or (double) or (float)	Float
(string)	String

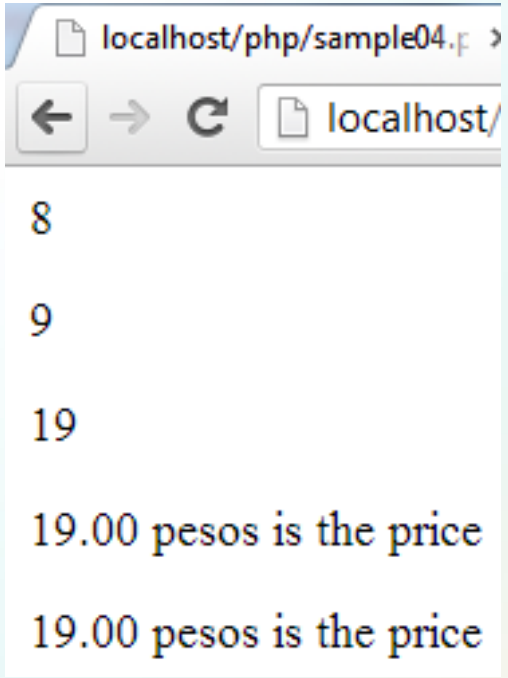
Beginning PHP and MySQL 3rd Edition by: W. Jason Gilmore pp.105

PHP Type Casting Operators

Example:

```php
<?php
    $x = (double) 8;
    echo "<p>$x</p>";
    $x = (int) 9.99;
    echo "<p>$x</p>";
    $str = "19.00 pesos is the price";
    $x = (int) $str;
    echo "<p>$x</p>";
    $x = (array) $str;
    echo "<p>$x[0]</p>";
    $obj = (object) $str;
    echo "<p>$obj->scalar</p>";
?>
```

Output:



PHP Type Juggling

Example:

```php
<?php
    $a = 10;
    $b = "20";
    $c = $a + $b;
    echo "<p>$c</p>";

    $a = "100 apples";
    $b = "50 apples";
    $c = $a + $b;
    echo "<p>$c apples</p>";

    $a = "314e-2";
    $b = 100;
    $c = $a * $b;
    echo "<p>$c</p>";
?>
```

Output:

Predefined Type Functions

`gettype()` function

returns the type of the variable. Possible return values are integer, double, boolean, string, array, object, resource, unknown types.

prototype:

```
string gettype(mixed var)
```

`settype()` function

converts a given variable to a specific type.

prototype:

```
boolean settype(mixed var, string type)
```

Predefined Type Identifier Functions

```
is_array(), is_bool(), is_float(), is_integer(),
is_null(), is_numeric(), is_object(), is_resource(),
is_scalar(), and is_string()
```

Type Functions

Example:

```php
<?php
    $x = 10;
    printf("Variable type: %s <br>", gettype($x));
    settype($x, double);
    printf("Variable type: %s <br>", gettype($x));

    if(is_integer($x)){
        echo "Integer: True<br/>";
    } else {
        echo "Integer: False<br/>";
    }

    if(is_double($x)){
        echo "Double: True<br/>";
    } else {
        echo "Double: False<br/>";
    }

    if(is_numeric($x)){
        echo "Numeric: True<br/>";
    } else {
        echo "Numeric: False<br/>";
    }
?>
```

Output:

Variable type: integer
Variable type: double
Integer: False
Double: True
Numeric: True

Summary

- *You can integrate database easily in PHP.*
- *Standard PHP script are enclosed by <?php and ?> tags.*
- *Each statement in PHP must be terminated with a semi-colon(;).*
- *All variable in PHP must have a dollar ($) character before its identifier.*
- *PHP variables are not explicitly declared.*
- *You can use echo, print, or printf function for outputting strings to internet browser.*
- *You can enclosed string using a pair of single quote or pair of double quote.*
- *String inside a pair of single quote interpreted as literal except for the single quote itself.*

Summary (Continue)

- *String inside a pair of double quote interpreted as literal with some exception like using dollar character.*
- *To explicitly display the character on the internet browser use backslash (\) character like \$, \', \" and etc.*
- *Datatypes in PHP can be categorized as scalar or compound datatypes.*
- *You can explicitly change the type of a variable in PHP.*
- *PHP support some predefined type function that can be used in manipulating type variables.*

Q&A SESSION

Applications Development and Emerging Technologies

MODULE 1 (SUBTOPIC 1)

Introduction to PHP

Objectives

- To understand the history of PHP
- To know the advantages of PHP as a server side scripting language
- To know what are the software's needed in developing web application using PHP.
- To know the different requirements before using PHP

HISTORY

- *1994/1995 (PHP 1.0)*
  - *Developed by Rasmus Lerdorf*
  - *To know how many visitors were reading his online resume based on PERL/CGI script*
  - *Personal Home Page (PHP)*
- *1997 (PHP 2.0)*
  - *PHP is based on C rather than PERL*
  - *Personal Home Page/Form Interpreter*
- *1998 (PHP 3.0)*
  - *50,000 users were using PHP to enhance their Web pages*
  - *Developers joined Lerdorf*
- *1999 (PHP 4.0)*
  - *With core developers Zeev Suraski and Andi Gutmans*
  - *PHP makes the most popular scripting language with morethan one million user base by Netcraff*

HISTORY (Con't.)

- Hundreds of functions being added
- Dubbed the Zend scripting engine
- May 22, 2000 (PHP 4.0)
  - PHP: Hypertext Preprocessor (recursive acronym)
  - 3.6 million domain PHP installed
  - Enterprise development
    - Improved resource handling (scalability)
    - Object-oriented support (Classes and Objects)
    - Native session-handling support (session)
    - Encryption (encryption algoritms)
    - ISAPI Support (for IIS)
    - Native COM/DOM (Windows applications)
    - Native Java Support: (binding to java objects)
    - PERL Compatible Regular Expressions
  - New features, power, and scalabilty

HISTORY (Con't.)

- PHP 5 (July 13, 2004)
  - Vastly improved object-oriented capabilities: OOP improvement
  - Try/catch exception handling
  - Improved XML and Web Services support (Simple XML Support, using SOAP)
  - Native support for SQLite
  - Installed on 19 million domains
  - 54 percent on all Apache module
- PHP 6 (2007 not yet released)
  - Unicode Support (Native Unicode support for multilingual applications
  - Security improvements
  - New language features and constructs (64-bit type integer, foreach looping)

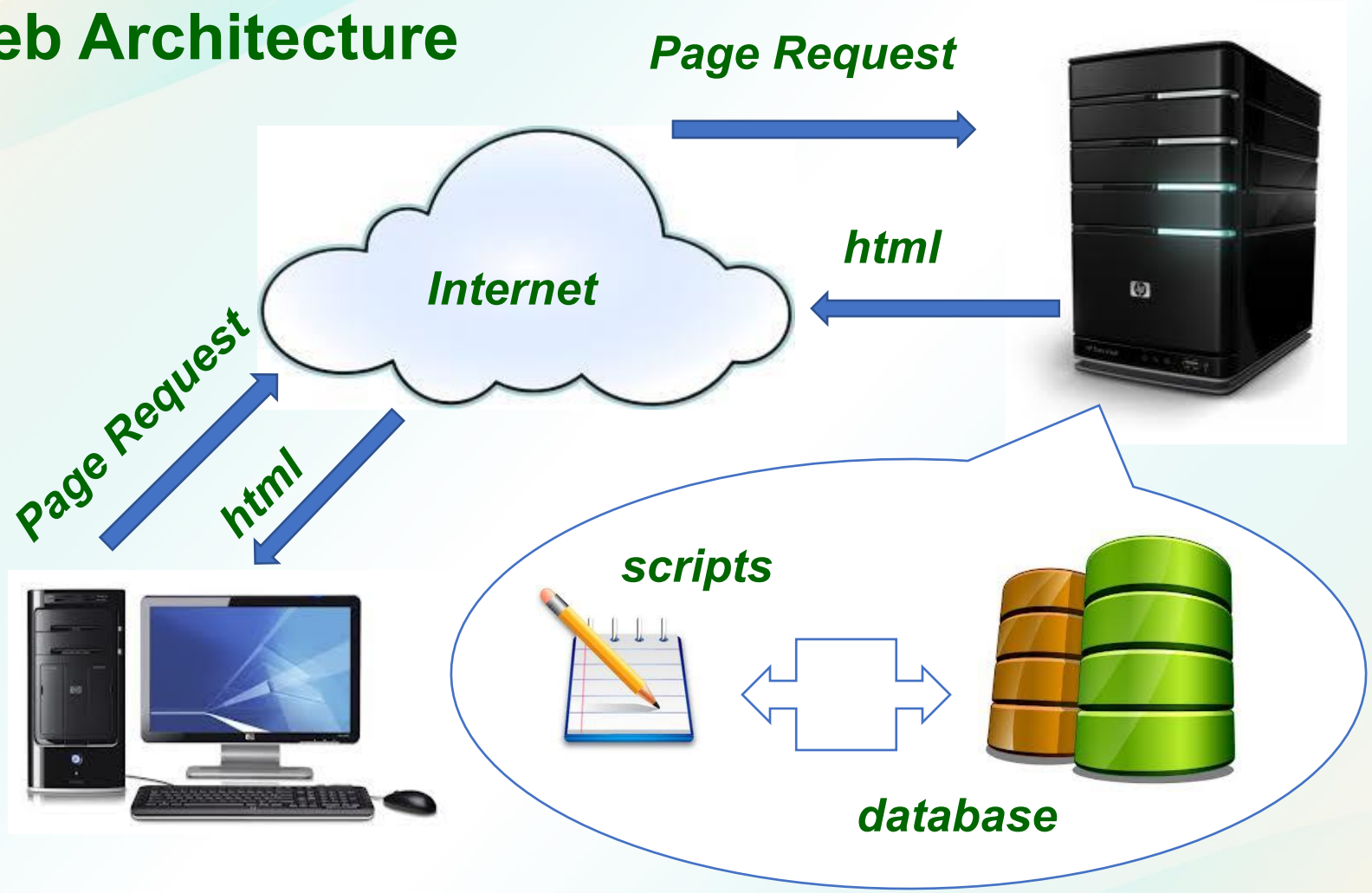
http://en.wikipedia.org/wiki/PHP

HISTORY (Con't.)

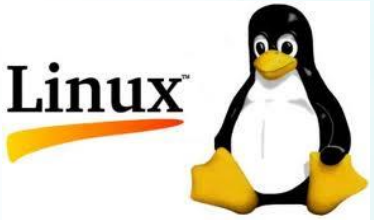
PHP Key categories

- *Practicality*
  - *PHP is a loosely type language (no explicitly create, typecast, or destroy a variable)*
- *Power*
  - *More libraries and thousands of functions*
- *Possibility*
  - *Native support is offered for more than 25 database products, including Adabas D, dBase, Empress, FilePro, FrontBase, Hyperwave, IBM DB2, Informix, Ingres, InterBase, mSQL, Microsoft SQL Server, MySQL, Oracle, Ovrimos, PostgreSQL, Solid, Sybase, Unix dbm, and Velocis.*
  - *Both structured and Object Oriented approach*
- *Price*
  - *Free of charge*

Web Architecture



Software Requirements
Operating System: Linux



Software Requirements
Operating System: Windows

Software Requirements
Operating System: Mac

Cheetah

Panther

Puma

Jaguar

Tiger

Leopard

Snow leopard

Lion

Software Requirements
Internet browser

Google Chrome

Mozilla Firefox

Internet Explorer

Opera

Safari

Maxthon

Rockmelt

SeaMonkey

Deepnet Explorer

Avant Browser

Software Requirements
Web Server

Apache

NGINX™

Google Web Server (GWS)

Software Requirements
Server side scripting language

Java (JavaServer Pages: JSP)

Ruby (Ruby on Rails)

Active Server Pages (ASP)

Python (Django)

PERL CGI

Software Requirements
Database

Software Requirements
Code editor

aptana

eclipse

Bluefish

Dreamweaver

NetBeans

Software Requirements
Other

CSS

HTML

Software Requirements
Other

Operating System
Windows 7 / Windows 8

Web Server
Apache

Language Script
PHP

Database
MySQL

XAMPP / WAMP

Code Editor
Notepad++ / Eclipse / Netbeans / Dreamweaver

Summary

- *PHP was developed originally by Rasmus Lerdorf*
- *Seev Suraski and Andi Gutsman were the one who contribute most in PHP development.*
- *PHP key categories are practicality, power, possibility, and price.*
- *PHP is a embedded language that is commonly embedded on HTML*
- *PHP is a server-side scripting language that runs its application on a Web server.*
- *Much of the PHP syntax were derive from C programming Language.*

Q&A SESSION