# PIC 40A

## Lecture 16: Introduction to PHP

# What is PHP?

Server-side scripting language

An interpreted language like JavaScript

Developed by Rasmus Lerdorfin 1994

Open source

Most web servers have PHP processors

# What does "PHP" stand for?

PHP originally meant **P**ersonal **H**ome **P**age.

PHP is now an acronym for

**P**HP: **H**ypertext **P**reprocessor

# What is PHP used for?

Initially used to help track visitors to Lerdorf’s personal web site

Form handling

Database accessing

Creating dynamic web pages

# Where do PHP scripts reside?

PHP scripts are text files residing on a server.

When a web browser requests a document on the server that has a .php extension the file is first sent to a PHP processor.

# How does PHP processor work?

When a client requests a document that has embedded PHP, the Web server calls a **PHP processor** which

- Copies XML code and client-side scripts verbatim to an output document as it finds them
- Interprets the PHP scripts when they occur
- Writes the PHP script output to the output document

The Web server then sends the output XHTML document to the client.

# Always keep in mind!

The output of a PHP script is an XHTML page!!

# PHP Hello world!

```
#!/usr/local/bin/php
<?php
print '<?xml version = "1.0" encoding="utf-8"?>';
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>My First PHP Embedded Page</title>
    </head>
    <body>
        <p>
            <?php
                print 'Hello World! <br/>';
            ?>
        </p>
    </body>
```

# Key points

`#!/usr/local/bin/php`

Tells the server where to find the PHP processor.

The actual script is between `<?php` `and` `?>`

The stuff that is not between those tags is copied verbatim to the output document.

The stuff that is in between the tags has to evaluate to XHTML.

The final output of this document is an XHTML document.

The browser only renders the XHTML document.

It is not possible to see the PHP source if you view source on an XHTML document created from a PHP document.

# Basic PHP Facts

Like JS, PHP is usually purely interpreted

Syntax and semantics are closely related to JS

Like JS, PHP uses dynamic typing

Variables are not declared, they have intrinsic type

Every statement ends in a ;

# Basic PHP Facts cont.

PHP is case sensitive for variable names.

Reserved words and functions are not case sensitive:

While, wHiLe etc. are all same.

# PHP Primitive data types

Four scalar types - Integer, double, string, boolean

Two compound types- array and object

Two special types - resource and null

# Primitive data types

Integers – Corespond to long int type of C++. In most machines not fewer than 10 digits.

Doubles – Corresponds to double type of C++. Any number that has a decimal or exponent form with e or E are legal doubles.

String – There is no char data type in PHP. A single letter is a string of length 1. String literals can be either in single quotes 'text' or double quotes “text”. There is a difference. We will discuss this in more detail later.

Boolean – Only possible values are TRUE and FALSE which are case sensitive. Integer value of 0 is coerced to false and all other values are coerced to TRUE.

# Variables

Variables are not declared

They are dynamically typed

No type declaration is possible

Unassigned variables are also called *unbounded variables* and have type `NULL`

All variable names begin with a $ followed by letters digits or underscores

Variable names are case sensitive

# Example

```php
<?php
$txt="Hello World!";
$x=16;
?>
```

Note that we always use $.

```php
<?php
$x=16.7;
?>
```

# PHP Arithmetic operations

PHP has usual C++ collection of operators

+, - , * , / , % , ++, --

+,-,* : If either operand is a double the result is a double. If both are integers then integer is produced.

/ : If the result is not an integer the result is returned as a double.

% : If both are not integers they are coerced to integers.

# Useful predefined math functions

floor(x) - Argument: double,  returns largest int less than argument

ceil(x) -  Argument: double, returns smallest int greater than argument

round(x) - Argument: double, returns nearest int

mt_srand(n) - Argument: integer,  Initializes random number generator

mt_rand(x,y) - Arguments: Two numbers, returns random integer in the range (x,y)

abs(x) - Argument: Number, returns absolute value of the parameter

min(x,y) - Argument: Two numbers, returns smallest

max(x,y) - Argument: Two numbers, returns largest

# Example

```
<?php

$first = mt_rand(0,10);
$second = mt_rand(0,10);

$smallest = min($first,$second);

?>
```

# PHP operators

Assignment operators

=, +=, -=, *=, /=, %=

Comparison operators

==, !=, <>, >, < , >=, <=

Logic operators

&&, ||, !

# Strings

There is no character type: Single character of data is a string of length 1

String literals are defined with either single quotes ' ' or double quotes " ".

```
$my_text = "Hello!";
```

or

```
$my_text = 'hello';
```

# Single quotes ' ' vs Double quotes ""

## Inside single quotes

- There is no interpolation
- Escaped characters such as \n and variables are not replaced by their values

## Inside double quotes

- Variable values are substituted for their names
- Escaped characters are replaced by their values

# Example

```
$apples = 16;
```

```
print "I have $apples apples";
```

Output: I have 16 apples

```
print 'I have $apples apples';
```

Output: I have $apples apples

What about:

```
print "He said: 'I have $apples apples'";
```

# Concatenation

Concatenate is done with .

Example:

```
$str1 = "Hello!";
$str2 = "What's up?";
$str3 = $str1 . " " . $str2;
```

This is the only string operation. Others are done using predefined functions.

# Predefined string functions

`strlen` - Argument: string
Returns: The number of characters in the string

`strcomp` - Argument: Two strings
Returns: Zero if the two strings are identical.

`strpos` - Argument: Two strings
Returns: The character position in the first string of the first character of the second string.  Or false if not found.

`substr` - Argument:  A string and two integers index and length
Returns: A string starting at index length many characters long

`trim` - Argument: A string
Returns: The parameter string with white space removed from both ends

# Example:

```
$name = "Philip J. Fry";
$name_length = strlen($name);

$last_name = substr($name, 10,3);
```

# Comments

```
// Single-line comment

# This is also single line comment

/*  This is
    a multi-line
    comment */
```

# Include

One of the most useful tools is to insert another php script from a file into the current php script.

The command

include("filename");

This will import contents of a text file called filename and insert it at the include spot.

The included text may be composed of XHTML, PHP or both.

Any PHP in the included text must be inside the <?php tags.