# Introduction to PHP

CSC 242, Web Programming

# PHP: Hypertext Preprocessor

- PHP is a server-side scripting language
- PHP code is *interpreted* not *compiled*
- PHP source code contains `php` "tags"

```
<?php
  /* PHP code */
?>
```

where anything between the '`<?php`' and '`?>`' tokens should be executed by the PHP interpreter

# Basic Syntax

- Statements must end in a semicolon
- Variables must start with a $ symbol
- Line comments are denoted by `//`
- Block comments are denoted by `/* ... */`
- keywords, classes, functions and user-defined functions are *NOT* case sensitive
- variable names *ARE* case sensitive

# Simple Example

```
<?php
$message = "Hello world";
echo "$message";
?>
```

# Variable Naming Rules

- Must start with a dollar sign ($) followed by a letter or the underscore character (_)
- Can only contain `a-z,A-Z,0-9,` and `_`
- Variable names are case sensitive

# PHP Types

- PHP data types:
  - String
  - Number: (Integer and Float)
  - Boolean
  - Array
  - Object
  - NULL
- PHP is dynamically typed – types of variables do not need to be declared
- PHP is generally weakly typed – some type conversions are automatic

# The String Type

- The string type represents a sequence of characters
- Categories of string
  - literal: delimited by single quotes (`'...'`)
  - interpolated: delimited by double quotes (`"..."`)
- The escape character is the backslash (\)
- The dot (.) operator performs string concatenation

# Multi-line strings

- A string can be defined over multiple lines

```
<?php
  $name = "first_name
    last name";
?>
```

- Heredoc syntax preserves white space

```
<?php
  $name = <<<_END
      first_name
      last name
_END;
?>
```

# Heredoc Syntax Rules

- The `<<<TOKEN` starts a heredoc
- The token name is user defined, it typically should be a value not expected to be seen in the string
- The ending token must be the first thing on the line followed by a semicolon (`TOKEN;`)
- There is no need to escape any characters in a heredoc

# Arithmetic Operators

| Operator | Description | Example |
|---|---|---|
| + | Addition | `$a + 3` |
| - | Subtraction | `$a - 3` |
| * | Multiplication | `$a * 3` |
| / | Division | `$a / 3` |
| % | Modulus | `$a % 3` |
| ++ | Increment | `++$a` |
| -- | Decrement | `--$a` |

# Assignment Operators

| Operator | Example | Equivalent to |
|---|---|---|
| = | `$a = 3` | `$a = 3` |
| += | `$a += 3` | `$a = $a + 3` |
| -= | `$a -= 3` | `$a = $a - 3` |
| *= | `$a *= 3` | `$a = $a * 3` |
| /= | `$a /= 3` | `$a = $a / 3` |
| %= | `$a %= 3` | `$a = $a % 3` |
| .= | `$a .= $b` | `$a = $a . $b` |

# PHP Implicit Type Coercion

- The type of a variable is implicitly converted based on the context in which the variable is used
- In PHP this is called "Type Juggling"

```
<?php
  $x = "10"; // string
  $y = 3.14; // float
  $z = $x * $y; // float
?>
```

- The `gettype` function returns a string representation of a variable's type

# Explicit Type Casting

- `(int)`, `(integer)` cast to integer
- `(bool)`, `(boolean)` cast to boolean
- `(float)`, `(double)`, `(real)` cast to float
- `(string)` cast to string
- `(array)` cast to array
- `(object)` cast to object
- `(unset)` cast to NULL

# PHP Constants

- Constants are similar to variables, but the value cannot be changed once set
- Syntax

```
<?php
  define("NAME", "VALUE");
?>
```

- `NAME` is the name of the constant and is traditionally upper case
- `VALUE` is the value assigned to the constant

# Including and Requiring Files

- The `include` statement includes and evaluates the specified file
- The `include_once` statement includes and evaluates the specified file only once
- The `require` statement is identical to the `include` statement, but if a failure occurs, the script is halted
- The `require_once` statement is identical to the `include_once` statement, but if a failure occurs, the script is halted

# Checking Syntax

- The syntax of a PHP file can be checked on the command line:

  `php -l FILE.php`

- The `-l` (lower case L) is short for lint
- If a syntax error exists, then the error and line number are reported