Learning PHP: Hypertext Preprocessor

Edward Sharick - Week 10 and 11 (10/30/23 and 11/6/23)

What is PHP?

- PHP is an open-source scripting language
- PHP scripts are executed on the server. The result of the script can:
    - Return data to the browser as plain HTML.
    - Output images, PDF files, or text in other XML files.
    - Send and receive cookies.
    - Add, delete, modify data in a database (MYSQL)

Setting up my computer as a local web server

I downloaded and installed XAMMP. I then modified the Apache > Config > httpd.conf file to change the DocumentRoot element to my projects working directory.

PHP Syntax

- A php script can be placed anywhere in the .php document. The script is executed on the web server and the result is sent back to the browser.

```
<?php

// PHP code goes here

?>
```

- PHP keywords are not case-sensitive, but variable names are.
- PHP is loosely typed.
- Comments:
    - // or # for single line comments
    - /* multi-line comment */
- Variables:
    - $x=5; //declares variable with value 5
    - $y=”John”; //declares variable with value “John”
    - Variables start with $ followed by a letter or _
    - Variables cannot contain symbols
    - You can assign multiple variables in one line: $x = $y = $z = “Fruit”;
    - Variables have local (inside function), global (outside function), or static scope
        - Access global variables in a function: global $x;
        - All global variables are stored in an index $GLOBALS[index]
        - Static variables are defined inside a function, one time, and their data is saved each time the function is called.
    - Superglobal variables:
        - $GLOBAL

        - $_SERVER – holds information about headers, paths, and script locations
        - $_REQUEST, $_POST, and $_GET – Used to collect data after submitting an HTML form
        - Use $_GET for non-sensitive data, $_POST for sensitive data
        - We can use PHP scripts to get the from data and validate it
        - PHP Complete Form Examp.le (w3schools.com)
- Data types
    - String, Integer, Float, Boolean, Array, Object, NULL, and Resource
    - var_dump($x); //returns the type of $x
- Strings
    - Can use ‘ or “ to define
    - strlen(‘hello’);
    - str_word_count(‘hello world!’); //counts the number of words in the string
    - strrev(x); //reverse string
    - strops(‘word’, ‘wo’); //returns character position or false
    - str_replace(“world”, “dolly”, “hello world”); //replace 1st param with 2nd param in 3rd param
    - PHP String Functions (w3schools.com) more string functions here…
- Integers
    - Between -2^31 and 2^31-1
    - PHP_INT_MAX/MIN/SIZE – predefined integer constants
        - Size is size in bytes (8 on a 32 bit system)
    - is_int($x)
    - (int)$x; //cast $x (float or string) to an int
- Float
    - Decimal or number in exponential form (1.9e 10)
    - PHP_FLOAT_MAX/MIN/DIG/EPSILON
        - DIG – the number of decimal digits that can be rounded into a float and back without precision loss
        - EPSILON – smallest representable positive number x such that x + 1.0 != 1.0
    - is_float(), is_nan()
    - is_numeric($x) – returns true if number or numeric string
- Boolean
    - True, or false
- Arrays
    - $letters = array(“A”, “B”, “C”)
    - $letters[0];
    - count($letters); //3
    - PHP Array Functions (w3schools.com)
    - Associative arrays
        - $x = array(“P”=>”1”, “Q”=>3);
        - $x[“P”];
        - foreach($x as $key => $value)
    - Multidimensional Array
        - array (array(…), array(…), array(…));

  - sort(), rsort(); //sort the array in ascending/descending order
  - asort(), ksort(), arsort(), krsort(); //sort associative array by value/keys in ascending/descending order
  -
- Object
  - Instances of classes
  - Uses ‘new’ keyword to call the __construct() function for the class
- Null
  - $x=null;
- Math functions
  - pi(); //returns 3.141592…etc
  - min(0, 1, 2, 3); //returns 0
  - max(0, 1, 2, 3); //returns 3
  - abs(-3); //returns 3
  - sqrt(64);
  - round(0.60);
  - rand(); //random integer
  - rand(10, 100); //random integer between [10, 100]
  - https://www.w3schools.com/php/php_ref_math.asp

- Constants
  - Use the define(*name, value, case-insensitive=false*) function
    - *name* – name of the constant
    - *value* – value of the constant
  - Or use the const keyword:
    - const CONST_NAME = “value”;
  - Constants are global
  - 9 predefined constants (magical):
    - __CLASS__ //class name
    - __DIR__ //directory of the file
    - __FILE__ //file path
    - __FUNCTION__ //function name
    - __LINE__ //line number
    - __METHOD__ //class and function name
    - __NAMESPACE__ //name of namespace
    - __TRAIT__
    - ClassName::class //name of specified class and namespace

- Output
  - echo – prints the string to the HTML file; no return value
  - print – returns 1 so it can be used in expressions
- Operators
  - Arithmetic: +, -, *, /, %, **
  - Assignment: =, +=, -=, *=, /=, %=

    - Comparison: ==, === (same value and type), !=, <> (not equal), !== (not identical), <, >, <=, >=
        - $x <=> $y //returns and integer less than, equal to, or greater than zero, depending on if $x is less than, equal to, or greater than $y.
    - Increment/decrement: ++$x, $x++, --$x, $x—
    - Logical: and, or, xor; &&, ||, !
    - String:
        - $x = $x . $y //concatenation of $x and $y
        - $x .= $y //same as above
    - Array:
        - + Union
        - == Equality – same key/value pairs
        - === Identity – same key/value pairs and in same order and of same types
        - !=
        - <>
        - !== - non-identity
    - ?: - Ternary
        - $x = expr1 ? expr2 : expr3
    - ?? – Null coalescing
        - $x = expr1 ?? expr2 //if expr1 isn't null, $x gets the value, otherwise it gets expr2

- Conditionals
    - if, elseif, else
    - switch, case, default
    - Syntax same as java/c#
- Loops
    - while //same as java
    - do/while //same as java
- for //same as java
    - foreach ($array as $value) //only works on arrays
    - break and continue act the same as Java
- Functions
    - Functions names are NOT case-sensitive

```
function myFunc($x, $y = 5) {
  echo "hello";
  return $x + $y;
}

echo myFunc(5);
```

    - Passing by reference
        - &$x //use & operator
    - Variable number of arguments (…$x)
- Dates
    - date(format)

  - d – day of month 01 to 31
  - m – month 01 to 12
  - Y – year in 4 digits
  - l – day of the week
  - H – 24 hr format 00 to 23
  - h – 12 hr format 01 to 12
  - i – minutes 00 to 59
  - s – seconds 00 to 59
  - a – am or pm
  - date_default_timezone_set(“America/New_York”);
  - mktime(hour, minute, second, month, day, year);
  - Or create a date from a string: strtotime(“next Saturday”)
- Include Files
  - Inserting the content of one PHP file into another PHP file before the server executes it
  - require – produces a fatal error and stops the script
  - include – produces only a warning and the script continues

    ```
    include ‘filename’;
    //or
    require ‘filename’;
    ```

  - You can create templates, like a menu file and then use it in all other files.

    - Example 1:

      ```
      <?php include ‘footer.php’; ?>
      ```

    - Example 2:

      ```
      <div class="menu">
      <?php include 'menu.php';?>
      </div>
      ```
- Manipulating Files
  - readfile() – reads the file and writes it to the output buffer
  - $file = fopen(“webdictionary.txt”, “r/w/a/x/r+/w+/a+/x+”) - opens a file pointer to somewhere in the file
  - fread($file, *bytes*) – reads from an open file
  - fclose($file)
  - filesize($file) – number of bytes in a file
  - feof($file) – check if at the end of the file
  - fgetc($file) – read a single character
  - fwrite($file, $txt) – writes text to file at current pointer; use (“w” to overwrite; “a” to append)
  - PHP Filesystem Functions (w3schools.com)
- PHP cookies
  - setcookie(*name, value, expire, path, domain, secure, httponly);*

  - isset($_COOKIE[$cookie_name]) – returns true if cookie is set
  - Delete a cookie – use setcookie with an expiration date in the past
- PHP Session
  - A session stores information to be used across multiple pages. It is not stored on the users computer.
  - Session variables hold information about a single user
  - session_start(); //starts a session
  - $_SESSION['varname'] = 'value';
  - session_unset(); //remove all session variables
  - session_destroy(); //destroy the session
- PHP Filters
  - PHP has built-in filters for validating (data in proper form) and sanitizing (removing illegal characters from data) data.
  - https://www.w3schools.com/php/php_ref_filter.asp
- Callback functions
  - array_map("my_callback_function_name", $array_of_values);
    - applies the callback to each value of the array
  - You can also pass the name of a function as a variable, then call that function $x()
- PHP and JSON
  - json_encode($associative_array); //converts dictionary to JSON
  - json_decode($json_object); //converts JSON object to associative array
  - To get values from a json object
    - $obj -> *key;*
  - To get values from an array
    - $arr[*key*];
- PHP also has a built-in XML parser

- Exceptions
  - throw new Exception("exception text");
    - The exception class can give information on:
      - $ex -> getCode();
      - $ex -> getMessage();
      - $ex -> getFile();
      - $ex -> getLine();
  - try/catch/finally blocks (same as Java)
- Classes and Objects – OOP with PHP
  - class ClassName { }
  - public $variableName; //Properties
  - $this -> variableName = *value;* //Accessing a property
  - __construct(){ } //defines a constructor for the class
  - __destruct(){ } //function automatically called at the end of the script
  - public function functionName() : returnType { }
  - Access modifiers: public, protected, and private

  - extends keyword for inheritance; final prevents class inheritance or method overriding
  - abstract class ClassName { } //contains abstract or concrete functions
  - interface InterfaceName {} //contains only abstract functions (bud doesn't need abstract keyword)
  - Traits – trait traitName { public function func() {} } //defines a trait
    - use traitName; //allows the class using the trait to call func()
  - const CONSTANT_NAME = *value*; //defines a constant, cannot be changed once it is declared
    - Accessed within class -> echo self::CONSTANT_NAME;
    - Accessed outside class -> echo ClassName::CONSTANT_NAME;
  - static variables and functions can be accessed with ::
- MySQL Database
  - Database system that runs on a server using standard SQL
  - PHP can be combined with MySQL to use SQL queries
  - It is the de-facto standard database system for web sites with large volumes of data and end-users
  - You can use the MySQLi library or PDO to do the following:
    1. Connect to the database
    2. Create the database table
    3. Insert the data into the table
    4. Retrieve data from the table
    5. Update data in the table
    6. Delete data from the table
    7. Search for/filter data
  - Sample code: PHP MySQL Create Database (w3schools.com)
- You can use PHP and AJAX to run server-side scripts and return data to make interactive applications, get data from MySQL databases or XML files, do live searches, create a poll, etc.
  - Example: PHP - AJAX and PHP (w3schools.com)