**CSIS 3280:**

**Lecture 01: Learning PHP**

**Intro to PHP**

* *Static Websites:* the content does not change and is fixed. The content is the same for all visitors. E.g., personal websites.
* *Dynamic Websites:* pictures and contents are different for different visitors. E.g., Amazon.com
* PHP is a programming language for building dynamic websites.
* PHP is a server-side language.
  + Example: JavaScript is a client-side language.
  + Example: ASP.NET is a server-side language.
* PHP is free.
* OS X and most Linux distributions come with PHP already installed.

**Static Websites:**

* PHP runs on the server not on the client.

Diagram

Description automatically generated

**Dynamic Websites:**

Diagram

Description automatically generated

**PHP & PHP Engine:**

* PHP is a language.
* PHP Engine is the software.
  + Running on a Web Server.
  + Understanding PHP language and executes the commands.
  + For example, talking to DBMS, retrieving data and generating pages
* PHP Engine is written in the C programming language.
* PHP works with a web server running on Windows, Mac OS X, Linux, and many other versions of Unix.
* PHP works on Web Servers such as Apache, nginx, MS IIS, or any web server that supports CGI standard.
* PHP works on many DBMSs: MySQL, PostgreSQL, Oracle, MS SQL Server, SQLite, Redis, and MongoDB.
* PHP is used on more than 200 million different websites, including giants like Facebook, Wikipedia, and Yahoo.

**Basics of PHP:**

* It can be part of a HTML file.
* It starts with **<?** and ends with **?>**.
* PHP engine executes only code between, text out of them is ignored.
* If there is no code at the end of the file, **?>** end tag is optional.
* There can be multiple blocks of PHP code in an HTML file.
* PHP is a case-sensitive language.
* But language keywords (such as print) and function names are not case-sensitive.

Text

Description automatically generated

* Comments:
  + Inline: //.
  + Inline: #.
  + Multiline: /\* (…) \*/.

Graphical user interface, text, application, chat or text message

Description automatically generated

* Every program is composed of statements and every statement ends with a (;).
* You can write multiple PHP statements on the same line of a program if they are separated with a semicolon.
* You can put as many blank lines between statements as you want. The PHP engine ignores them.
* It is recommended to put one statement on a line and blank lines between statements only when it improves the readability.

**Lecture 02: Data: Working with Text and Numbers**

**Data Types:**

* String: a sequence of bytes (represented by characters).
* Strings can contain:
  + Letters a-z, and A-Z.
  + Numbers 0-9.
  + Punctuation . ; ? ! , ( ) :
  + Spaces.
  + Tabs.
  + Or any other character.

**Defining Strings:**

* String: a sequence of bytes (represented by characters).
* Surround the string with single-quote or double-quote.
  + There are differences between using ‘ and “.
* If you want to include a single quote inside a string, put a backslash (\) before it.
* Word processors often change straight quotes like ' and " into curly quotes like ‘, ’, “, and ”. The PHP engine only understands straight quotes as string delimiters.

Graphical user interface, text

Description automatically generated

**Escape Sequence:**

* Backslash (\) is an escape character in PHP (like C, C++, C#, JavaScript, Java).
* Inside single-quoted strings, backslash and single quote are only special characters. Everything else is treated literally.
* Double-quoted strings provides more special characters.

Graphical user interface, text, application

Description automatically generated

**HERE document:**

* You can define strings with the here document syntax, especially HTML code string.
* Start with <<< + a delimiter word and finish with the same work used at start.
* Delimiters can contain:
  + Letters, numbers, and the underscore character.
  + The first character of the delimiter must be a letter or underscore.
  + For readability, recommended writing uppercase.
* The delimiter can’t be **indented**, and no **whitespace**, comments, or other characters are allowed after it, only **semi-colon**.

Text

Description automatically generated

* NOTE: **print** command must be used, but single or double quote not needed.

Text

Description automatically generated

**Combining Strings:**

* Use a period (.) to combine strings.

Text

Description automatically generated

**PHP String Functions:**

* **trim()** function removes whitespace from the beginning and end of a string.
* **strlen()** function tells you the length of a string.

**Text

Description automatically generated**

* Use **==** to compare strings.

**A picture containing text

Description automatically generated**

* **strcasecmp()** function compares two strings without considering their cases (upper or lower case).

**A picture containing text

Description automatically generated**

* **strtolower()** and **strtoupper()** changes a string to all-lowercase or all-uppercase versions, respectively**.**

**Text

Description automatically generated**

* The **ucwords()** function uppercases the first letter of each word in a string.

**Text

Description automatically generated with medium confidence**

* **substr()** function, you can extract just part of a string.
  + 0 to 30 means first 30 characters.
  + If you put negative number, it starts from the end. (-4 means ‘less 4 characters from the end’)

Graphical user interface, text, application

Description automatically generated

* **str\_replace()** function changes parts of a string. It looks for a substring and replaces the substring with a new string.

Text

Description automatically generated

**Numbers:**

* No special notation needed.
* NOTE: Integers are stored precisely but floating-point numbers may not.

**Arithmetic Operators:**

* You can use arithmetic operators ( +, -, \*, / ) as you did in primary school.
* New operators:
  + \*\* for exponentiation.
  + % for modulus division (return the reminder of a division).

**Variables:**

* Variables hold the data in the memory of computer while your program uses and manipulates it.
* In PHP, variables are denoted by a $ followed by the variable’s name.
* To assign a value to a variable, use an equal’s sign (=). This is known as the assignment operator.

Graphical user interface, text

Description automatically generated

**Variable Names:**

* Variable names may only include:
  + Uppercase or lowercase Basic Latin letter (A-Z and a-z).
  + Digits (0-9).
  + Underscore (\_).
  + Any non-Basic Latin character (such as ç), if you’re using a character encoding such as UTF-8 for your program file.
  + NOTE: Even though you can use special characters, but NOT RECOMMENDED.
  + Variable names are case-sensitive.

**Lecture 03: Making Decisions**

**Boolean Expressions**

* Every expression in a PHP program can be evaluated to true or false.
* All non-zero integers or floating-point numbers evaluated to true.
* All non-empty strings are evaluated to true.
* Empty string, or string containing zero (‘0’) is evaluated to false.

* A variable containing null is evaluated to false.

**Floating-point Numbers Comparison**

* To compare two floating-point numbers, check whether the two numbers differ by less than some acceptably small threshold

Text

Description automatically generated

**Comparison Operators - Strings**

* When the PHP engine sees strings containing only numbers, it converts them to numbers for the comparison.

Text

Description automatically generated with low confidence

* the PHP engine converts the string 6 pack to the number 6, and then compares it to the number 55 using numeric order. Since 6 is less than 55, the less than test returns true.

Text

Description automatically generated with medium confidence

* **strcmp()** compares string using dictionary order without any converting to numbers.

Graphical user interface, text

Description automatically generated

**Spaceship Operator**

* The spaceship operator (<=>) does comparison similar to strcmp(), but for any data type.

* It evaluates to a negative number when its left-hand operand is less than the righthand operand.
* A positive number when the righthand operand is bigger.
* 0 when they are equal.

Text

Description automatically generated

**For-Loop**

* You can combine multiple expressions in the initialization expression and the iteration expression of a for() loop by separating each of the individual expressions with a comma.

Graphical user interface, text, application, email

Description automatically generated

**Lecture 04: Arrays**

**Arrays**

* Arrays are collections of related values, such as the data submitted from a form, the names of students in a class, or the populations of a list of cities.
* An array is made up of elements.
* Any string or number value can be an array element key, such as corn, 4, -36, or Salt Baked Squid.

**Creating an Associative Array**

* Associative Arrays: Each element has a key and a value.
* For example, an array holding information about the colors of vegetables has vegetables names for keys and colors for values.
* Any string or number value can be an array element key, such as corn, 4, -36, or Salt Baked Squid.
* Arrays and other non-scalar values can’t be keys, but they can be element values.
* Scalar values are simple values such as 6, 98.76, “Saeed”. Arrays or objects are non-scalar values.

Text

Description automatically generated

Text

Description automatically generated with low confidence

**Shortcut to Create Arrays**

Graphical user interface, text, application

Description automatically generated

**Creating an Array Element by Element**

A picture containing text

Description automatically generated

* If you create an array with [] or array() by specifying only a list of values instead of key/value pairs, the PHP engine automatically assigns a numeric key to each value.
* The keys start at 0 and increase by one for each element.
* PHP automatically uses incrementing numbers for array keys when you create an array or add elements to an array with the empty brackets.

Text

Description automatically generated with medium confidence

Text

Description automatically generated

**Finding the Size of an Array**

* The count() function tells you the number of elements in an array.
* An empty array (an array with no elements in it), count() returns 0.
* An empty array also evaluates to false in an if() test expression.
  + For example: $x = []; if($x) print “$x is an empty array”.

Graphical user interface, text

Description automatically generated with medium confidence

**Looping Through Array – foreach**

Text

Description automatically generated

* Inside the foreach(), changing the values of $key and $value doesn’t affect the elements in the actual array.

* If you want to change the array element values, use the $key variable as an index into the array.

Text

Description automatically generated

Text

Description automatically generated

* If elements of a numeric array were added in a different order than how their keys would usually be ordered, this could produce unexpected results.

Graphical user interface, text, application

Description automatically generated

**Locate a Particular Key in an Array**

* To check for an element with a certain key, use array\_key\_exists().

A picture containing text

Description automatically generated

**Locate a Particular Value in an Array**

* To check for an element with a certain value, use in\_array().

Text

Description automatically generated

* The array\_search() function, if it finds an element, it returns the element key instead of true.

Text

Description automatically generated

**Modifying Arrays**

* You can operate on individual array elements just like regular scalar variables, using arithmetic, logical, and other operators.

Graphical user interface, text, application, email

Description automatically generated

**Using Element Arrays in a String**

* Inside strings, don’t put quotes around the element key.

Text

Description automatically generated

* If you have an array key that has whitespace or other punctuation in it, interpolate it with curly braces.

Text

Description automatically generated

**Remove Elements by unset()**

A picture containing icon

Description automatically generated

* Removing an element with unset() is different than just setting the element value to 0 or the empty string.

**Print All of the Values in an Array**

* Use the implode() function to print all of the values in an array at once.

Graphical user interface, text, application, email

Description automatically generated

**Creating Multidimensional Arrays**

* Use the array() constructor or the [] to create arrays that have more arrays as element values.

Text, calendar

Description automatically generated with medium confidence

**Accessing Multidimensional Array**

Text

Description automatically generated

**Iterating Through a Multidimensional Array**

Text

Description automatically generated with medium confidence