Running PHP
You can run a php script directly from the server cli by using the php command:
```
paul@odin:~/public_html/cs3680$ php myscript.php
```
You can also execute a php script from the browser by providing the full URL path:
```
https://cs.csub.edu/~paul/cs3680/myscript.php
```
Notice that when you access PHP from the browser, you will only be able to see the output.
PHP code is processed by the server BEFORE a response is sent.

DataTypes
PHP is a loosely typed language. However, there are a few functions to work with types when necessary:
- get or set the state of a variable:
  ```
  isset, unset, empty
  ```
- get or set the type of a variable:
  ```
  gettype, settype
  ```
- type checkers:
  ```
  is_int, is_float, is_string, is_bool, is_array, is_object, is_null, is_resource
  ```

Scope
- **Superglobals**: global scope, already defined
  ```
  $_GET, $_POST, $GLOBALS, $_COOKIE, $_SESSION, $_FILE, $_ENV, $_REQUEST, $_SERVER
  ```
- **Constants**: global scope, using `define` function
- **Global Variables**: global scope of script, and `func` if declared
- **Static Variables**: local to function/class definition, retains value
- **Function Variables**: local to function

Operators
- Assignment: `=`
- Arithmetic: `+ - * / %`
- Arithmetic Assignment: `+= -= *= /= %=`
- String ops: `. .=`
- Increment/decrement (pre & post): `++ --`
- Comparison: `< > <= >= == === != !== <>`
- Logic: `&& || ! and or xor`
- Ternary/Conditional: `? :`

Control Structures
- `if(expression) { ... }
- if(expression) { ... } else { ... }
- if(expression) { ... } elseif(expression) { ... } else { ... }
- switch($var) { case <int, float, char, string>: ... break; default: ... }

Looping
- while(expression) { ... }
- do { ... }while(expression);
- for(init; test; update) { ... }
- foreach($arr as $key => $val) { ... }

Developer Mode
define('DEVELOPER', false);
if(DEVELOPER) {
    error_reporting(E_ALL);
    ini_set("display_errors",1);
    ini_set("error_log","myLogFile.log");
}

Useful Info
<?= $value ?> : Shorthand for <?php echo $value ?>