Announcements

- Submit Server
  - Remember that the summer submit server link is different from regular semester (use class web page link)

Include functions

- Four include functions
  - include → Any error will cause a warning
  - require → Any error will cause a fatal error and termination of script
  - include_once → includes file only once per script. Any error will cause a warning
  - require_once → includes file only once per script. Any error will cause a fatal error and termination of script
- Include functions pass context as text
- If you include a file in PHP mode the contents of the file is not processed in PHP mode (it is processed in HTML mode)
- To process the contents in PHP mode you must add <? php ?> to the file
- **Example**: includeProblem.php
- **IMPORTANT:** Make sure that you do not leave any blank lines or spaces at the end a of file (after the ?>) you are planning to include in scripts
- If you use include in a loop the file will be included as many times as the loop is executed
- You can use include functions to create libraries
- The include_path directive in php.ini allows you to set directories with php scripts

heredoc

- heredoc
  - Alternative to specify an string
  - Allow us to change the delimiter used for defining strings from “” to the specified label
    - Useful as we will not need to escape “
- Format

```
$theString = <<<LABEL
PUT HERE LINES OF CHARACTERS YOU WANT TO MAKE
PART OF THE STRING
LABEL;
```

- Ending label must be followed by a semicolon but nothing else
- You can use any LABEL (e.g., EO, EOT, etc.)
- Do not add any spaces after the opening LABEL
- The last LABEL must not be indented at all
- Interpolation of variables works as in double-quoted strings
- IMPORTANT: Superglobals do not work with heredoc
- **Example:** heredoc.php

Switching Modes

- **Example**: switchingModes.php
- Why do we want to switch between modes?

Our Approach to Output Generation

- **Example:** support.php, main.php

Type Conversions

- **PHP Conversions**
  - Boolean → integer/double: true → 1, false → 0
  - Boolean → string : true → 1, false → empty string
  - null → number : 0
  - null → boolean : false
  - string → number : equivalent to reading a number from the string. If a number cannot be read, the generated value is zero
  - string → boolean : false if empty string or string is “0”; true otherwise
  - number → boolean : 0 → false, otherwise → true
  - float → integer : fractional part dropped
  - integer → float : corresponding float is created
- **Conversion Functions**
  - intval → converts argument to an integer
  - floatval → converts argument to a float
  - strval → converts argument to a string

About Variables

- A variable is either set or unset
- Assigning a value to a variable makes it a set variable
- isset() → returns true if a variable is set
- To unset a variable → use unset or set the variable to null
- **A variable present in the URL query string is set even if it has no value assigned to it**
- Empty variable → has value that evaluates to false. Those values are:
  - double O.O
  - Integer O
  - false
  - "O"
  - "" → Empty string
  - Object with no properties
  - Array with no elements
  - NULL
- All unset variables are empty
- Empty vs. false → emptiness applies only to variables
- **Example:** variables.php

Self-Referencing Scripts

- Often is convenient to merge the HTML form and the form handler in the same file
- Advantages
  - Easier to display error messages and pre-filled form fields
  - Better control of variable namespace
  - Changing filename easier
- Form-handling code should come before the form
- Example: selfReference.php

Header function

- **Header function**
  - Allow us to send arbitrary HTTP header
  - Takes a string argument
- **Redirection via header function**
  - Provide a string with “Location: “ followed by URL
  - Example: header(“Location: http://www.cs.umd.edu”);
  - **Example:** exHeader.php
- **Authentication via header function**
  - Header function can also be used to ask for a username and password via a pop-up window
  - **Example:** authentication.php

Files

- Files we are referring to should be accessible by the web server (e.g., available in htdocs or directories with appropriate permissions)
- You may use files when the amount of data you want to manipulate is small and may not require a database
- **Opening**: through the fopen() function.
  - If opening is successful a resource ID ("file pointer" or "file descriptor") will be returned; otherwise false will be returned
  - Some opening modes
    - "r" → read-only
    - "w" → write-only, creates file if it does not exist and erases the file contents if it exists
    - "a" → appends to the end of the file
- **Reading**: through the fread() function
    - Takes as arguments a file-pointer identifier and a file size in bytes
    - You can use the function filesize() to determine the size of a file (remember to pass the filename not the file descriptor)

Files

- You can read data line by line using fgets(). It takes a file-pointer and a line length as parameters
- The line length defaults to 1024 bytes if you don't specify a value
- **Example:** fileReading.php
  - The file to read is called myData1.txt and it is located in the same directory as fileReading.php
- **Closing files:** via fclose() function
- **Writing**: via fwrite() function
  - Takes as arguments a file-pointer identifier, a string, and an optional length in bytes
  - Returns the number of bytes written
- **Example:** fileWriting.php
  - The file will be created in the same directory fileWriting.php resides

File/Directory Functions

- feof → takes a file descriptor as parameter
- file_exists → takes a file name as parameter. Verifies whether the file exists
- filesize → returns the size of a file in bytes
- is_dir → returns true if the filename exists and is a directory
- is_file → returns true if the filename exists and is a file
- touch → sets access and modification time of a file. Creates file if it does not exist
- unlink → deletes a file

Hidden Fields

- You can use a form hidden field to pass information between scripts
- A better alternative is to use session variables which we will see soon
- **Example:** hidden.php

Date Functions

- **time()**
  - Provides all the information about current date and time
  - Returns an integer representing number of seconds since midnight GMT on Jan 1, 1970 (this moment is known as the UNIX epoch)
  - time stamp → number of seconds since Jan 1, 1970
  - **Example:** dateAndTime.php
- **getdate()**
  - Returns associative array with date/time information for today's date
  - It can take a time stamp as a parameter
  - **Example:** dateAndTime.php
- **date()** → allow us to format date information via a string parameter
  - If only a string is provided uses current date, otherwise the second parameter (time stamp) is used
  - **Example:** dateAndTime.php
- **mktime()** → allow us to create a date
  - **Example:** dateAndTime.php
- End of Unix Time (January 19, 2038 03:14:07 GMT) → http://unixepoch.com/
- More information about options for date are available at: http://us2.php.net/manual/en/function.date.php