Overview

String Manipulation

- A string is a collection of characters
    - "This is a string."
- Strings that are delimited by double quotes are preprocessed by PHP.
    - Certain character sequences beginning with backslash (\)
    - Variable names (Starting with $) are replaced with string representations of their values.
- PHP is ideally suited for manipulating strings providing many string manipulation functions.
    - Nearly 100 native functions

String Manipulation (continued)

- strlen() returns the lenght of the string
- substr() returns some part of a string
- strpos returns the first occurrence of a string within another string
- str_replace() replaces characters within a string
- str_gecsv() returns an array of string contents
- strip_tags() removes html and php tags from a string
- trim() removes whitespace from a string

String Manipulation Function

Demonstration

Runtime Error Handling

- Runtime errors are referred to as exceptions.
    - An error that is unexpected, exceptional
- PHP provides a mechanism for managing exceptions that allows for changing the flow of execution in response to the runtime error.
    - If an exception is unmanaged it may impact the appliction's user, it may expose security issues, etc

Runtime Error Handling(continued)

- We use the keywords try and catch create the exception handling mechanism.
  - try encapsulates a code block protecting its execution
    - If a runtime error does occur within the try block then PHP saves the current state of the program and transfers execution to a "handler" for the error
  - catch encapsulates a code block to be executed in response to a runtime error of a specific type
    - A catch block may be written to respond to any runtime error;however, this is not a robust solution
- Handling multiple possible exception types can be achieved using multiple catch blocks.
  - A catch block for each type of exception
  - Most specific exception type first

Runtime Error Handling(continued)

- The Exception object

```
class Exception
{
    protected string $message;
    protected int $code;
    protected string $file;
    protected int $line;
    public __construct($message, $code = 0, Exception $previous = null);
    final public string getMessage(void);
    final public Exception getPrevious(void);
    final public mixed getCode(void);
    final public string getFile(void);
    final public int getLine(void);
    final public array getTrace(void);
    final public string getTraceAsString(void);
    final string __toString(void);
    final private void __clone(void);
}
```

- The SPL provides two categories of Exception-derived classes from which a collection of exceptions are themselves derived.
    - Logic Exception
    - Runtime Exception

Runtime Error Handling

Demonstration

File Resources

- Executing server-side PHP code allows our applications to access resources available on the server.
    - Reading and writing files is only the beginning
    - Modify file security and details, determine disk information, determine directory information, and more ...
- PHP provides robust and easy-to-use file manipulation functions.

Reading a File

- Before a file can be read it must be opened.
  - $fopen(< filename >, < mode >)$
    - File mode for reading only is "r"
    - File mode for reading and writing is "r+"
- In order to control the read process get the file's size.
  - $filesize()$
    - Returns the file size in bytes
- Read the files contents.
- $fclose(< filehandle >)$
  - It is not required to programmatically close an open file;however, it is best practice

Reading a File(continued)

- The *fgets*($<$ *filehandle* $>$) function reads a single line from the file
  - Optionally, you can limit the bytes read using *fgets*($<$ *filehandle* $>$, *length*)
- The *fgetss*($<$ *filehandle* $>$) function reads a single line from the file AND stripts HTML or PHP tags from the data.
- The *fgetcsv*($<$ *filehandle* $>$) will read a comma separated file into an array
- Use *feof*($<$ *filehandler*) to detect

```
      while(!feof())
      {
// do this until the end of the file
      }
```

- Use *fseek*($<$ *filehandle* $>$, $<$ *bytes* $>$) to change position in the file

```
      fseek($file,0); // moves back to beginning of file
```

Writing to a File

- Before writing to a file it must be opened $fopen(< filename >, < mode >)$
  - File mode for writing only is "w"
    - Write from beginning of file
    - Truncates (over writes) file contents
    - If file does not exist it is created
  - File mode for reading and writing is "w+"
    - Write from beginning of file
    - Truncates (over writes) file contents
    - If file does not exist it is created
  - File mode for reading only is "a"
    - Write(append) from end of file
    - If file does not exist it is created
  - File mode for reading and writing is "a+"
    - Write(append) from end of file
    - if file doesnt exist it is created

Writing to a File(continued)

- Write to the file
  - $fwrite(< filehandle >, < datatowrite >)$
- $fclose(< filehandle >)$
  - It is not required to programmatically close an open file;however,it is best practice

Reading a Configuration File

- Using an external file for application configuraiton is preferred to hard-coding values within the application
  - PHP provides a built-in function for reading key-value pairs from external configuraiton files
    - *parse_ini_file*(< *filename* >) returns an associative array of key-values pairs as listed in the configuration file.

```
    //  The expected format of the file

     [debug]
     debug_on = 1
     [directory]
     root = c:\app_root
```

  - PHP also provides for using XML-based configuration files.

XML File I/O

- Extensible Markup Language(XML) is a structural and semantic language.
  - Not a formatting lanaguage
  - Self-describing data
- XML provides for the creation of common information formats and the data.
  - Separates presentation, structure, and meaning from the actual content
- XML provides for information interchange.
  - Web services

XML File I/O (continued)

- An XML document's structure is defined by the W3C XML specification AND a Document TYpe Definition (DTD)
- DTD defines the structure for data interchange.
  - Many industries are defining their own DTD specifications
- The fundamental components of an XML document are the elements and attributes.
  - Elements mark sections of the document
  - Elements may be nested
  - Attributes provide additional information about and element

XML is case sensitive

XML File I/O (continued)

```
<movies>
<film>
<film_id lastwatched="2014-10-19">42</film_id>
<title>2001: A Space Odyssey</title>
<year>1968</year>
<director>Stanley Kubrick</director>
<actor>Keir Dullea, Gary Lockwood</actor>
<rating>G</rating>

</film>
...
</movies>
```

XML File I/O (continued)

- SimpleXML provides built-in functions for reading and parsing XML files.
  - Easy-to-use extension for getting XML element values
- SimpleXML reads the XML document into an object where the elements are converted into string attributes of the object.
  - The entire DOM tree is ready into memory

```
$movies = simplexml_load_file("movies.xml");
echo $movies->film[0]->title."<br>";
echo $movies->film[0]->year."<br>";
echo $movies->film[0]->director."<br>";
echo $movies->film[0]->actor."<br>";
echo $movies->film[0]->rating
```

Database Storage

- PHP allows us to store information in variables during application execution;however,information may need to be stored and retrieved long after the user's interaction with the application ends.
  - File storage provides a solution
    - Slow and unorganized
- Databases allow PHP applications to store and retrieve information in near real0time during application execution
  - MySQL, SQL Server, MariaDB, etc

Database Storage (continued)

- There are four basic commands to interact with a database.
  - Select
    - Retrieves data

```
Select * from movies ;returns all of the data for all of the movies in
```

  - Update
    - Changes value of stored data

```
Update movies set rating="PG—13" where title="Blade␣Runners"
```

  - Insert
    - Inserts new values in a table

```
insert into movies values("The␣Matrix",
"Keanu␣Reeves,Laurence␣Fishborne",
"Wachowski␣Bros",1999,"R");
```

  - Delete
    - Removes stored data

```
Delete from movies where ID=2;
```

Database Storage (continued)

- In order to send commands to the database the application must establish a connection to the database.
- The application must have user access on the database server in order to establish connection
  - Username and password
- Creating the connection
  - MySQL
    - $\$DB = mysql_connect(< Server >, < username >, < password >)$
  - SQL Server
    - $connectioninfoarray(< databasename >, < username >, < password >)$
    - $\$DB = sqlsrv\_connect(< Server >, < connectioninfoarray >)$

Database Storage (continued)

- Once a connection to the database is established the application can send SQL commands to the database
    - $sqlsrv\_query(< databaseconnection >, < query >, < parameters >)$
    - $mysql_query(< query >, < databaseconnection >)$
- Queries can be constructed and sent to the datbase from the application.
    - Referred to as **ad hoc** queries
    - PreventSQL injection attacks
        - Contents must be sanitized and validated
- Stored procedure (SRPROC) or prepared statements are predefined queries residing on the datbase called by the application.
    - $CALL < nameofprocedure > (< parameter >, < parameter >)$
    - Pass parameters as needed
    - Assign return value to variable

```
$returnValue = CALL <name of procedure>(<parameters>,<parameters>)
```

Database Storage (continued)

- "The PHP Data Objects(PDO) extension defines a lightweight, consistent interface for accessing databases in PHP"
  - Activate extension in php.ini
- PDO is an abstraction layer.
  - Same interface regardless of underlying database system
- Changing underlying database system supporting an application does not require large-scale code rework.
  - Change connection string, then test

Database Storage (continued)

- PDO connections are established by creating a new object of type PDO.

```
    $db= new PDO("mssql:host=<host>,dbname=<databasename>,<username>,<password>");
```

- While developing use *PDO* :: *ERRMODE_WARNING* which produces PHP warning messages.
- In production use *PDO* :: *ERRMODE_EXCEPTION* which will throw exceptions

```
    $db->setAttributes(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
```

Always wrap PDO calls in try...catch blocks

Database Storage

Demonstration

Email

- Composing and sending e-mail messages from a PHP application is a common task.
- PHP has a built-in *mail*() function
    - Requires php.ini configuration settings defining e-mail server and port

```
     $To='someone@theiraddr.com';
     $Subject = 'Intergalactic␣Kegger';
     $Message = 'End␣of␣the␣world␣party:
␣␣␣␣␣"MIB,DELIVER␣THE␣GALAXY␣OR␣EARTH␣WILL␣BE␣DESTROYED."';
     $Headers = "From:K@MIB.gov\r\n".
     "Reply-To:Z@MIB.gov\r\n".
     "Content-type:text/html;charset=UTF-8\r\n";
     mail($To,$Subject,$Message,$Headers);
```

- The *mail*() function lacks SMTP authentication capability

Email(continued)

- Extend e-mail capability by using the PEAR Mail Package
  - Distributed as part of PHP installation
  - Determine availability by using phpinfo()

```
     $From = "Sender's␣name␣<k@MIB.gove>";
     $To='Recipient's name <someone@theiraddr.com>';
␣␣␣␣␣$Subject␣=␣'Intergalactic Kegger';
␣␣␣␣␣$Message␣=␣'End of the world party:
     "MIB,DELIVER␣THE␣GALAXY␣OR␣EARTH␣WILL␣BE␣DESTROYED."';
␣␣␣␣␣$Host␣=␣"mail.mib.gov";
␣␣␣␣␣$Username␣=␣"k";
␣␣␣␣␣$Password␣=␣"123456";
␣␣␣␣␣$Headers=array("From"=>$From,"To"=>$To,"Subject"=>$Subject);
␣␣␣␣␣$SMTP␣=␣Mail::factory('smtp',array('host'=>$Host,'auth'=>true,
␣␣␣␣␣'username'=>$Username,'password'=>$Password));
␣␣␣␣␣$mail␣=␣$SMTP->send($To,$Headers,$Message);
␣␣␣␣␣␣
```

- The *mail*() function lacks SMTP authentication capability

Email(continued)

- Retrieve e-mail with the PHP IMAP extension.
  - Distributed as part of PHP installation;however,must be initialized in php.ini
  - Determine availability by using phpinfo()

```
class EmailReader
{
public function __construct() {...}
public function connect(){...}
public function close(){...}
public function getInbox(){...}
}
```

  - The *mail*() function lacks SMTP authentication capability

PHP with PEAR

- PHP Extension and Application Repository is a distribution system for reusable PHP code
  - Do not need to "reinvent the wheel"
- The components found in PEAR solves common application development issues.
  - Thousands of hours of development effort
- Installing PEAR provides a unified include path and the Package Manager
  - The package manager is used to retrieve and install PHP libraries

PHP with PEAR(continued)

- PEAR packages available for common application functionality.
  - Logging
  - User Authentication
  - CAPTCHA
  - Data Validation
  - Geographic IP Address Mapping
  - Database Interface
  - ...

PHP with PEAR

Demonstration