M1 & M2 UNIX

Wednesday, March 8, 2017 2:20 AM

ls –l * lists all files and directories, and the contents of the directories
ls –ld * lists all the files and directories, but NOT the contents of the directories.
ls –a will show all files including hidden

mkdir directorypath
rmdir directorypath
pwd present working directory
mv sourcepath destinationpath
cp sourcepath destinationpath

3 operational modes
while using the vi editor:

- Command Mode (default mode / you start here)
  User presses letter for a command

- Input Mode
  Lets user enter or edit text.
  ESC to return to command mode.

- Last-line Mode
  Pressing colon ":"
  opens a prompt to enter letter or word commands.

To save and exit
enter ZZ (i.e. two capital z's) OR :x in last line mode
exit without modifying the contents of your file :q!

chmod who=permissions filename
x – allows access to files inside
In order to have access to directory contents, at least the "x" permission is necessary.
called the "pass-through" permission

umask (user file-creation mode mask)

wc [word count] option [filename]
Options:
-l count lines
-w count words (delimited by whitespace)
-m count characters

grep utility
Useful grep options:
-i ignores case
-n numbers lines in the output
-v reverse match
Matches lines that do NOT contain the pattern
-c displays the count of matched lines

UNIX processes
process
Almost everything that is "running" on a UNIX
ps (process status) command
top command
provides a continuous update including resource usage

bg command
Restarting in the background
bg PID
bg job_number

kill
Terminates Background process

Kill -9 PID
pkill -9 firefox (name)

Head
Display the beginning of a file
E.g. head [-line_count] file
example: head –3 users.log

Tail
Display the end of a file
tail -20 tmp
Displays last 20 lines of file tmp

cut –d, –f1–2
cuts first 2 fields delimited with a comma
cut –d" " –f1
space is the field delimiter

sort command
Popular options:
-f ignore case in comparisons
-n numeric sort (i.e. sort the numbers, don't sort alphabetically)
-u display unique entries
-r reverse sort

0	Standard Input	stdin
1	Standard Output	stdout
2	Standard Error	stderr

Redirect with > or append >>

tr 'a-z' 'A-Z' < ls.txt
tr [OPTION]... SET1 [SET2]
takes two sets of characters and replaces occurrences of the characters in the first set with the corresponding elements from the other set

/dev/null file (AKA the bit bucket or black hole)
**remove stuff by sending them here
**like cleaning recycle bin on windows

```
find / -name “homer” 2> /dev/null
```

Create an alias

```
alias dir=ls
```

.bashrc and .bash_profile
Located in the user’s home directory
executed every time a user logs in or creates a new shell
Things vary depending whether the shell is interactive or not

Standard ERROR redirection operators :
2> or 2>>

Pipe (|)

```
ls | less
ls | tee unsorted.txt | sort
```

Hard Link Example

```
ln myfile link-name
```

M3 PHP

Wednesday, March 8, 2017 3:45 AM

Two Protocols
HTTP
Protocol for communication between WEB Server and browser
TCP/IP
Suite of protocols For network communication

```
<?php
   // get the data from the request
   $first_name = $_GET['first_name'];
   $last_name = $_GET['last_name'];
?>

<p>First name: <?php echo $first_name; ?></p>
 <p>Last name: <?php echo $last_name; ?></p>
```

Use double quotes for variable substitution

```
$name = "Name: $first_name";        // Name: Bob
$name = "$first_name $last_name";    // Bob Roberts
```

Concatenation

```
$name = $first_name . ' ' . $last_name;  // Bob Roberts
```

A function for formatting numbers
number_format($number[, $decimals])

```
$nf = number_format(12345.674, 2);     // 12,345.67
```

A function for getting the current date
date($format)

Character	Description
Y	A four-digit year such as 2010.
y	A two-digit year such as 10.
m	Numeric representation of the month with leading zeroes (01-12).
d	Numeric representation of the day of the month with leading zeroes (01-31).

Statements that format a date

```
$date = date('Y-m-d');   // 2010-06-12
$date = date('m/d/y');   // 06/12/10
$date = date('m.d.Y');   // 06.12.2010
$date = date('Y');       // 2010
```

Casting

```
$NewVariable = (new_type) $OldVariable;
```

```
$variable_name = value;

define("CONSTANT_NAME", value);
```

Constant names DON'T begin with a dollar sign ($)
Constant names use all uppercase letters

primitive types
Data types that can be assigned only a single value
E.g. float, numbers, bool, string, NULL
- In PHP bool can only be TRUE or FALSE
  - NOT 1 or 0

```
<script language = “php”>
        echo “I Love PHP”;
</script>
```

The six PHP data types
integer
double
boolean
string
Object
Array

There is 3 types of arrays:
Associative (pair of key and value, where the key could be declared)
Indexed Array (pair of key and value, where the key is an index 0,1,2…) => Most common array
Multidimensional Array (Array inside an array)

```
$array_name = array(values);

  $Provinces = array(
     "Newfoundland and Labrador",
     "Prince Edward Island",
     );

$array_name[ ]
```

count()
Count array elements

print_r()
displays the index and value of each element in an array

var_dump()
displays the index, value, data type and number of characters in the value

```
<form action="display_discount.php" method="post">
<input type="submit" value="Calculate Discount" />
```

M3 Handling Input

Wednesday, March 8, 2017 10:18 PM

$_SERVER["SCRIPT_NAME"];

“post” method
embeds the form data in the request message, more secure

$_POST array

PHP automatically creates and populates it

“get” method
appends the form data to the URL specified in the form’s action attribute

$_GET array

PHP creates and populates it

URL Legend
question mark (?)
separates form data from the URL

ampersand (&)
separates individual elements

equal sign (=)
separates element name from value

plus signs (+)
Represents spaces in the name and value fields

percent sign (%) followed by 2-digit hex. representation of character’s ASCII value
Encodes all other characters
except letters, numbers, hyphens (-), underscores (_) and periods (.)

```
if (isset ($_POST['Submit'])) {
// Process the data
}
else {
// Display the Web form
}

//Include depending on the page
if (isset($_GET['page'])) {
        switch ($_GET['page']) {
                case 'About Me':
                        include('inc_about.html');
                        break;
                default:
                        include('inc_home.html');
                        break;
   }
}
```

```
 $firstName = $_POST['fName'];
  $lastName = $_POST['lName'];
     echo "Thank you for filling out the scholarship form, ".
$firstName." ".$lastName  . ".";
```

empty() function
used to determine if a variable contains a value

is_numeric() function
used to determine if a variable contains a number

round() function
can be used to see if appropriate number of decimal places

stripslashes() function
removes the leading slashes for escape sequences

trim() function
removes any leading or trailing white space from a string

```
  <p>First Name: <input type="text" name="fName" value="<?php
echo $firstName; ?>" /></p>
```

mail() function
sends an e-mail message containing form data in PHP

syntax:

```
        mail(recipient(s), subject, message)
```

isset() function
used to determine if $_POST[‘Submit’] variable has been set
i.e. if submit button pressed

```
if (isset($Submit)) {
// Validate the data
}

if (isset($_POST[‘Submit’])) {
// Validate the data
}
```

The is_*() family of functions determines if the entered value is of the required data type

M4 PHP FnCtrl

Wednesday, March 8, 2017 4:02 AM

```
function displayCompanyName($Company1, $Company2,
$Company3) {
        echo "<p>$Company1</p>";
        echo "<p>$Company2</p>";
        echo "<p>$Company3</p>";
}

displayCompanyName("Course Technology");
```

rand(min, max) -> random number

How to modify a string that's passed by reference

```
function wrap_in_tag(&$text, $tag) {
   $before = '<'  . $tag . '>';
   $after  = '</' . $tag . '>';
   $text = $before . $text . $after;
}
```

A variable with global scope

```
$a = 10;          // $a has global scope
function show_a() {
   echo $a;       // Inside function, $a is NULL
}
show_a();         // Displays nothing
```

How to access a global variable from a function

```
$b = 10;            // $b has global scope
function show_b() {
   global $b;       // $b refers global variable $b
   echo $b;
}
show_b();           // Displays 10
```

AND, OR, and NOT operators

```
!$old_customer ||
   $loan_amount >= 10000 && $score
< $min_score + 200
```

```
foreach ($array_name as $variable_name) {
statements;
}
```

Use continue and break in loops

include and require statements reuse content by allowing you to insert the content of an external file on multiple Web pages

include statement
generates a warning if the include file cannot be found

require statement
halts the processing of the Web page and displays an error if the include file cannot be found

include_once and require_once statements assure that the external file is added to the script only one time

M4 Regex

Wednesday, March 8, 2017 4:01 AM

https://courses.cs.washington.edu/courses/cse190m/12sp/cheat-sheets/php-regex-cheat-sheet.pdf

```
//Regex Variables Examples
    // Step 1 $postalRegex = '/^[A-Za-z]\d[A-Za-z]\d[A-Za-z]\d$';
    /* Step 2 $postalRegex = '/^[A-Za-z]\d[A-Za-z]\s?\d[A-Za-z]\d$/';
    /* Step 3 */ $postalRegex = '/^\s*([A-Za-z]\d[A-Za-z]\s?\d[A-Za-z]\d)\s*$/';
    /* Step 4 */ $courseRegex = '/^\s*([A-Z]{3}\d{3}[A-Z]{1,3})\s*$/';
    /* Step 5 $phoneRegex = '/^\s*(\d{3}-\d{3}-\d{4})\s*$/'; */
    /* Step 6 */ $phoneRegex = '/^\s*(\d{3}-\d{3}-\d{4}|[(]?\d{3}[)]?[\s]*\d{3}([\s]*|[-]?)\d{4})\s*$/';

//Implementation
if(preg_match($postalRegex,$postalCode,$matches)){
            $isValidPostal = true;
            $postalCodeError = "<span style='border-bottom: 1px solid green;'>✓</span>";
          }
          else{
            $isValidPostal = false;
            $postalCodeError = '<- Must be X9X9X9 or X9X 9X9';
          }
```

preg_match(pattern, subject[, submatches])
preg_match_all(pattern, subject[, submatches])
preg_replace(pattern, replacement, subject)
preg_replace_callback(pattern, callback, subject)
preg_grep(pattern, array)
preg_split(pattern, subject)

n* Zero or more of n
n+ One or more of n
n? Zero or one occurrences of n
{n} n occurrences exactly
{n,} At least n occurrences
{,m} At most m occurrences
{n,m} Between n and m occurrences (inclusive)

^ Start of subject (or line in multiline mode)
$ End of subject (or line in multiline mode)
[Start character class definition
] End character class definition
| Alternates, eg (a|b) matches a or b
(Start subpattern
) End subpattern
\ Escape character

\w Any "word" character (a-z 0-9 _)
\W Any non "word" character
\s Whitespace (space, tab CRLF)
\S Any non whitepsace character
\d Digits (0-9)
\D Any non digit character
. (Period) – Any character except newline

Regular Expressions are used by
Vi
Grep grep “howdy” myFile.txt
Awk
Sed

grep “he.lo” MyFile.txt
grep "^Hello" data
grep “Bye$" data

[/g]	– global substitution
[address]	– specifies a line range.

preg_match(pattern, string);

Pattern
Pass regular expression pattern as first argument

String
the text you want to search

Returns
1 = match
0 = no match

```
preg_match("/\.com$/", $Identifier);
                    // \ escapes the . So it becomes a literal.
```

M5 MySQL & DBs

Wednesday, March 8, 2017 4:01 AM

To start command line:

```
c:\xampp\mysql\bin\mysql.exe –u root –p
```

-u is the user you want to log in as
-p the password will be prompted
-h the host address
Don't need since localhost is default.
Can use to logon remotely

MySQL

```
INSERT INTO orders (orderDate)
VALUES (NOW());
INSERT INTO orders (customerID, orderDate)
VALUES (100, '2010-11-05');
INSERT INTO orders (customerID, orderDate)
VALUES (201, '2002-01-25 6:26:12');
```

```
UPDATE orders
SET customerID = 101
WHERE customerID = 201
```

```
DELETE FROM products
WHERE `categoryID` > 1
```

Create a user name destroyer who has access to the local host and only the ability to destroy the the my_guitar_shop2 DB contents and structure. His name is his passport.

Sol'n:

```
GRANT DELETE, DROP
ON my_guitar_shop2.*
TO destroyer@localhost
IDENTIFIED BY 'destroyer‘
```

```
mysql> CREATE TABLE autos
 (license VARCHAR(10), make VARCHAR(25),
  model VARCHAR(50), miles FLOAT,
assigned_to VARCHAR(40));
```

```
mysql> ALTER TABLE vehicles RENAME TO
  company_cars;
```

```
<body>
<?php include("includes/header.php");?>
<?php include("includes/navigation.php");?>
This is the content of the page
<?php include("includes/footer.php");?>
</body>
```

Creating a user with limited privileges on one table

```
GRANT SELECT
ON my_guitar_shop1.products
TO mgs_tester@localhost
IDENTIFIED BY 'pa55word'
```

Creating a user with limited privileges on all tables

```
GRANT SELECT, INSERT, DELETE, UPDATE
ON my_guitar_shop1.*
TO mgs_user@localhost
IDENTIFIED BY 'pa55word'
```

Privilege	Description
SELECT	Lets the user select data.
UPDATE	Lets the user update data.
INSERT	Lets the user insert data.
DELETE	Lets the user delete data.
CREATE TABLE	Lets the user create a table.
DROP TABLE	Lets the user drop a table.

Enter the following command:

```
mysql –h host –u user –p
mysql –u root
```

root account
the primary administrative account for MySQL
created without a password

```
mysql> SELECT DATABASE();
mysql> SHOW databases;
DROP DATABASE DB1;
```

```
mysql> DROP TABLE company_cars;
```

Privileges
GRANT
REVOKE ALL
SHOW GRANTS
DROP USER

include() will output a PHP error to the browser then continue processing the rest of the code.
require() will output a PHP error and then stop. Nothing else will be outputted.

M5 Manipulate MySQL

Thursday, March 9, 2017 12:32 AM

PHP < version 5 automatically supported MySQL

PHP > Version 5 you need to turn on MySQL support
Configure it to use the mysqli or mysql package.

mysqli (MySQL Improved) package
the object-oriented equivalent of the mysql package

mysql_connect()function (Cont.):
 $connection = mysql_connect("host" [, "user", "password"]);

$DBConnect = mysql_connect("localhost", “citizen_kane", "rosebud");
mysql_close($DBConnect);

mysql_error()

$DBConnection = (mysql_connect (…) || die(mysql_error));

error control operator (@)
Use to suppress error messages
can be prepended to any expression
although commonly used with built in functions
E.g., $DBConnect = @mysql_connect("localhost", “citizen_kane", "rosebud");

AUTO_INCREMENT keyword
often used with a primary key to generate a unique ID for each new row in a table

NOT NULL keywords
often used with primary keys to require that a field include a value

mysql_num_rows() function
Use to find the number of records returned from the query
With queries that return results
(SELECT queries)

mysql_affected_rows() function
Use to determine the number of affected rows
With queries that modify tables
but do not return results
(INSERT, UPDATE, and DELETE queries),

The mysql_fetch_assoc() function returns the fields in the current row of a resultset into an associative array and moves the result pointer to the next row

result = mysql_create_db("dbname" [, connection]);

mysql_select_db(database [, connection]);

mysql_drop_db()

mysql_query(query [, connection]);

```
$SQLstring = "CREATE TABLE drivers (name VARCHAR(100), "
        . "emp_no SMALLINT, hire_date DATE, "
        . "stop_date DATE)";

$QueryResult = @mysql_query($SQLstring, $DBConnect);

if ($QueryResult===FALSE)
    echo "<p>Unable to execute the query.</p>"
            . "<p>Error code " . mysql_errno($DBConnect)
            . ": " . mysql_error($DBConnect) . "</p>";
else
    echo "<p>Successfully created the table.</p>";
```

To add multiple records to a database:
use the LOAD DATA statement
with the name of the local text file
containing the records you want to add

```
$SQLstring = "LOAD DATA INFILE 'company_cars.txt' "
. "INTO TABLE company_cars"
```

mysql_fetch_row() function
returns the fields in the current row of a resultset into an indexed array
and
moves the result pointer to the next row

```
$SQLstring = "SELECT * FROM company_cars";
$QueryResult = @mysql_query($SQLstring, $DBConnect);
echo "<table width='100%' border='1'>\n";
echo "<tr><th>License</th><th>Make</th><th>Model</th>
     <th>Mileage</th><th>Year</th></tr>\n";
while (($Row = mysql_fetch_assoc($QueryResult)) !== FALSE) {
     echo "<tr><td>{$Row['license']}</td>";
     echo "<td>{$Row['make']}</td>";
     echo "<td>{$Row['model']}</td>";
     echo "<td align='right'>{$Row['mileage']}</td>";
     echo "<td>{$Row['year']}</td></tr>\n";
}
     echo "</table>\n";
```

M6 FilesDir

Sunday, April 9, 2017 5:14 AM

\n
end a line on a UNIX/Linux operating system
E.g., SQL is awesome\n

\n\r (pair)
end a line on a Windows operating system
E.g., The best things in life are PHP\n\r

\r
end a line on a Macintosh operating system
OS X has a Linux Core\r

File permission are a four digit octal
0 - first number is always
u - second owner permission
g - third group permissions
o - other permision

Read => 4 Write => 2 Execute => 1

chmod($filename, $mode)
chmod(“index.html, 0644)

fileperms() function
reads permissions associated with a file

scandir() function
returns names of the entries in a directory to an array

DON’T need opendir(), readdir(), closedir()

```
$Dir = "/var/html/uploads";
$DirEntries = scandir($Dir);
foreach ($DirEntries as $Entry) {
        echo $Entry . "<br />\n";
  }
```

Example:
Display a directory listing

```
$path = getcwd();
$items = scandir($path);
echo "<p>Contents of $path</p>";
echo '<ul>';
foreach ($items as $item) {
   echo '<li>' . $item . '</li>';
}
echo '</ul>';
```

Reading Directories
3 steps:

1- opendir(directory)
   used to iterate through entries in a directory
2- readdir(handle) function
   Each time called it moves directory pointer to next entry in directory
3- closedir()
   Close directory

Example:

```
$Dir = "/var/html/uploads";
$DirOpen = opendir($Dir);
while ($CurFile = readdir($DirOpen)) {
                        echo $CurFile . "<br />\n";
                }
closedir($DirOpen);
```

strcmp() function can be used to exclude entries like . (current directory)
Or .. (parent directory)

```
if ((strcmp($CurFile, '.') != 0) && (strcmp($CurFile, '..') != 0))
                echo "<a href=\"files/" . $CurFile . "\">" .
```

mkdir() - function to create a directory

Three functions to test if a file or directory exists
is_file($path)
is_dir($path)
file_exists($path)

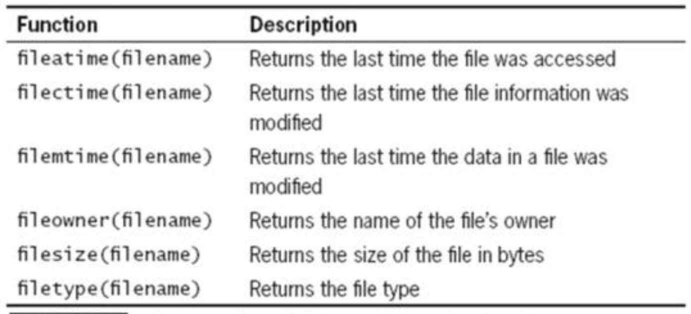
A function to get the current directory
getcwd()

A function to get a directory listing
Scandir($path)

A constant that contains the path separator
DIRECTORY_SEPARATOR

Uploading and Downloading Files

Other Functions



Function	Description
`fileatime(filename)`	Returns the last time the file was accessed
`filectime(filename)`	Returns the last time the file information was modified
`filemtime(filename)`	Returns the last time the data in a file was modified
`fileowner(filename)`	Returns the name of the file's owner
`filesize(filename)`	Returns the size of the file in bytes
`filetype(filename)`	Returns the file type

Table 5-5 Common file and directory information functions

<form action="FileUploader.php" method="POST"
enctype="multipart/form-data">

“multipart/form-data”
instructs browser to post multiple sections
one for regular form data
and one for the file contents

<input type="file" name="FileName" />

MAX_FILE_SIZE (uppercase) attribute of a hidden form field
specifies max. number of bytes allowed in uploaded file
MUST appear BEFORE the FILE INPUT field
Should be type hidden to prevent it from being:
Seen or
Changed
e.g., <input type="hidden" name="MAX_FILE_SIZE"
value="25000" />

Storing the file
Public files
are freely available to anyone visiting the Web site
Private files
are only available to authorized visitors

depending on whether they should be:
immediately available
or verified first
E.g. to make sure it is virus free, right type, etc.
Stored in sandbox area (outside publicly accessible folders)

Another Example:
Display the files from a directory listing

```
$path = getcwd();
$items = scandir($path);
$files = array();
foreach ($items as $item) {
   $item_path = $path . DIRECTORY_SEPARATOR . $item;
   if (is_file($item_path)) {
      $files[] = $item;
   }
}
echo "<p>Files in $path</p>";
echo '<ul>';
foreach ($files as $file) {
   echo '<li>' . $file . '</li>';
}
echo '</ul>';
```

$_FILES autoglobal array
(after uploading the file, this will contain all the info about files stored)

$_FILES['FileName']['error']
Contains error code associated with file

$_FILES['FileName']['tmp_name']
Contains temporary location of file contents

$_FILES['FileName']['name']
Contains the name of the original file

$_FILES['FileName']['size']
Contains size of uploaded file in bytes

$_FILES['FileName']['type']
Contains the type of the file

move_uploaded_file() function
moves uploaded file
from its temporary location to a permanent destination

First argument -> file to be moved
Second argument -> location

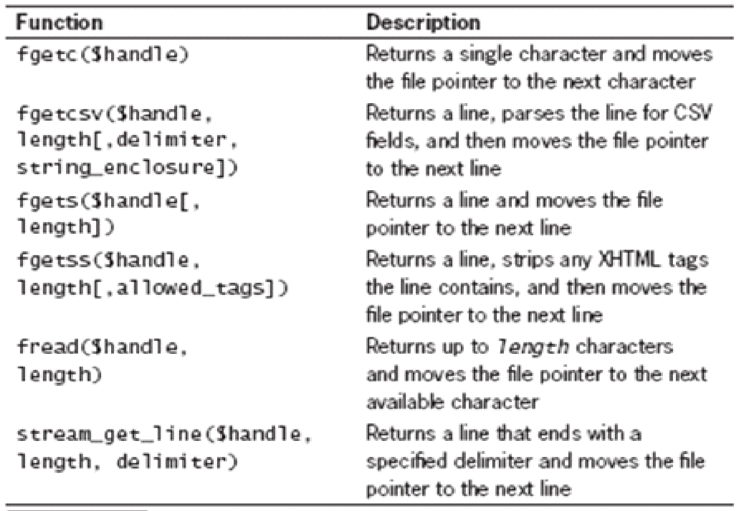
```
chmod("uploads/" . $_FILES['FileName']['name'], 0644);
```

Writing to a file

```
$EventVolunteers = " Blair, Dennis\n ";
$EventVolunteers .= " Hernandez, Louis\n ";
$EventVolunteers .= " Miller, Erica\n ";
$EventVolunteers .= " Morinaga, Scott\n ";
$EventVolunteers .= " Picard, Raymond\n ";

$VolunteersFile = " volunteers.txt ";

if (file_put_contents($VolunteersFile, $EventVolunteers) > 0)
            echo "<p>Data was successfully written to the
                  $VolunteersFile file.</p>";
else
            echo "<p>No data was written to the
            $VolunteersFile file.</p>";
```



Function	Description
fgetc($handle)	Returns a single character and moves the file pointer to the next character
fgetcsv($handle, length[,delimiter, string_enclosure])	Returns a line, parses the line for CSV fields, and then moves the file pointer to the next line
fgets($handle[, length])	Returns a line and moves the file pointer to the next line
fgetss($handle, length[,allowed_tags])	Returns a line, strips any XHTML tags the line contains, and then moves the file pointer to the next line
fread($handle, length)	Returns up to *length* characters and moves the file pointer to the next available character
stream_get_line($handle, length, delimiter)	Returns a line that ends with a specified delimiter and moves the file pointer to the next line

Table 5-11 PHP functions that iterate through a text file

How to read and write arrays
Read a file into an array

```
$names = file('usernames.txt');
foreach ($names as $name) {
   echo '<div>' . $name . '</div>';
}
```

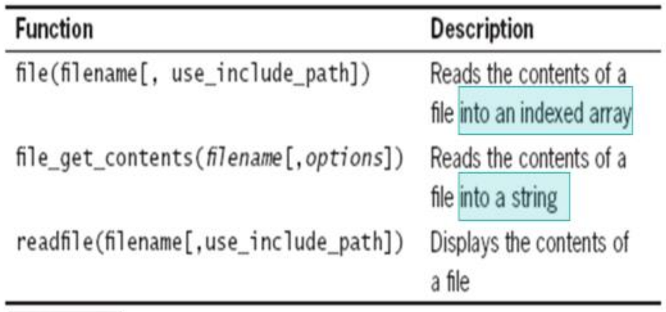
Write an array to a file

```
$names = array('joelmurach', 'rayharris', 'mikemurach');
$names = implode("\n", $names);
file_put_contents('usernames.txt', $names);
```

```
if (move_uploaded_file($_FILES['FileName']['tmp_name'],
"uploads/" . $_FILES['FileName']['name']) === FALSE)

      echo "Could not move uploaded file to \"uploads/" .
      htmlentities($_FILES['FileName']['name']) . "\"<br />\n";
else    Reading an Entire File
      echo "Successfully uploaded \"uploads/" .
      htmlentities($_FILES['FileName']['name']) . "\"<br />\n";
```

Htmlentities() - Convert all applicable characters to HTML entities



Function	Description
file(filename[, use_include_path])	Reads the contents of a file into an indexed array
file_get_contents(*filename*[,*options*])	Reads the contents of a file into a string
readfile(filename[,use_include_path])	Displays the contents of a file

Table 5-8 PHP functions that read the entire contents of a text file

Explode() function => Will separate files by a delimiter

Example:

```
for ($i=0; $i<count($JanuaryTemps); ++$i) {
            $CurDay = explode(", ", $JanuaryTemps[$i]);
}
```

Another Example:
file_exists (return T or F)
— Checks whether a file or directory exists
is_dir (returns T or F)
— Tells whether the filename is a directory
Copy(source,destination)
— Copies file

Opening and Closing File Streams
open_file = fopen("text file", " mode");
$VolunteersFile = fopen(“volunteers.txt", “r+");

Use the rename() function to rename a file or directory with PHP
rename(old_name, new_name)

Use the unlink() function to delete files and the rmdir() function to delete directories

The is_readable(), is_writeable(), and is_executable() functions check the the file or directory to determine if the PHP scripting engine has read, write, or execute permissions, respectively

```
if (file_exists(" sfweather.txt ")) {
  if(is_dir(" history ")) {
        if (copy(" sfweather.txt ",
        " history\\sfweather01-27-2006.txt "))
                echo " <p>File copied
                successfully.</p> ";
        else
                echo " <p>Unable to copy the
                file!</p> ";
           }
     else
                echo (" <p>The directory does not
                exist!</p> ");
}
else
     echo (" <p>The file does not exist!</p> ");
```

In PHP, a file can be one of two types: binary or text
A binary file is a series of characters or bytes for which PHP attaches no special meaning
A text file has only printable characters and a small set of control of formatting characters

MIME (Multipurpose Internet Mail Extension) generally classifies the file upload as in “image.gif”, “image.jpg”, “text/plain,” or “text/html”

M7 StateInfo

Sunday, April 9, 2017 5:17 AM

HTTP was originally designed to be stateless
browsers store NO persistent data about website visit

Four tools for maintaining state information:
Hidden form fields
Query strings
Cookies
Sessions

<input type="hidden">

Using Hidden Form Fields to Save State Information
```
echo "<input type='hidden' name='internID' " .
        " value='$InternID'>\n";
```

Using Query Strings to Save State Information
Example:
```
<a href="http://www.example.com/TargetPage
.php?firstName=Don&lastName=Gosselin&
occupation=writer">Link Text</a>
```

```
echo "{$_GET['firstName']} {$_GET['lastName']}
 is a {$_GET['occupation']}. ";
```

The expires Argument
setcookie("firstName", "Don", time()+3600);
Means: firstName cookie expires 3600 seconds from current time

The path Argument
setcookie("firstName", "Don", time()+3600, "/marketing/");
*make cookie named firstName available to all Web pages in /marketing directory

Using Cookies to Save State Information
2 Types:

Temporary cookies
available ONLY for the CURRENT BROWSER SESSION
They only take name and value

Persistent cookies
available beyond current browser session
stored in a text file on a client computer

setcookie() function syntax :
setcookie(name [,value ,expires, path, domain, secure])
setcookie("firstName", "Don");

```
setcookie(“arrayName[index]”, “value”)
setcookie("professional['firstName']", "Don");
setcookie("professional['lastName']", "Gosselin");
setcookie("professional['occupation']", "writer");
```

```
setcookie("professional[0]", "Don");
setcookie("professional[1]", "Gosselin");
setcookie("professional[2]", "writer");
```

The domain argument
setcookie("firstName", "Don”, time()+3600, "/", ".gosselin.com");
*Used usually for levels of domains

The secure argument
setcookie("firstName”, "Don", time()+3600, "/", ".gosselin.com", 1);
assign a value of 1 (for TRUE) or 0 (for FALSE)

- If TRUE will pass throught a secure connection only

Reading Cookies

$_COOKIE autoglobal
Access cookie by using the cookie name as a key
e.g., echo $_COOKIE['firstName'];

setcookie("firstName", "Don");

```
if (isset($_COOKIE['firstName'])
    echo "{$_COOKIE['firstName']}
```

setcookie("professional[0]", "Don");

```
if (isset($_COOKIE['professional']))
        echo "{$_COOKIE['professional'][0]}
```

To delete, substract time from expiration
setcookie("firstName", "", time()-3600);

Sessions

allow you to maintain state information EVEN WHEN clients DISABLE COOKIES in their Web browsers

session_start() function
starts a new session or
continues existing session

```
<input type="hidden" name="PHPSESSID"
 value='<?php echo session_id() ?>' />
```

isset() function
Use to ensure that a session variable is set
before you attempt to use it

```
<?php
session_start();
if (isset($_SESSION['firstName']))
        echo "<p>" . $_SESSION['firstName'] . "</p>";
?>
```

$_SESSION[] autoglobal

Example:

```
<?php
session_start();
$_SESSION['firstName'] = "Don";
$_SESSION['lastName'] = "Gosselin";
$_SESSION['occupation'] = "writer";
?>

<p><a href='<?php echo "Occupation.php?"
 . session_id() ?>'>Occupation</a></p>
```

Deleting a session

```
<?php
session_start();
$_SESSION = array();
session_destroy();
?>
```

M8 OOP

Sunday, April 9, 2017 5:17 AM

Object oriented is the idea of classes and components, to keep DRY code

Data
The information contained within:
variables or
other types of storage structures

Methods
The functions associated with an object

properties or attributes
The variables that are associated with an object

syntax for instantiating an object is:

```
        $ObjectName = new ClassName();
```

class constructor
a special function with the SAME name as its class
It is called automatically when an object from the class is instantiated
primarily used to initialize properties.

```
  $DBConnect = @new mysqli("php_db", "Citizen_Kane",
                         "rosebud");
                 @ will ignore errors
```

Most msqli class methods return TRUE (= success) or FALSE

```
$DBName = "vehicle_fleet";
$Result = @$DBConnect->select_db($DBName);
if ($Result === FALSE){
        //show error}
Else{
        //Successful Code}
```

```
$TableName = "company_cars";
$SQLstring = "SELECT * FROM $TableName";
$QueryResult = @$DBConnect->query($SQLstring);

while (($Row = $QueryResult->fetch_row()) !== FALSE) {
       echo "<tr><td>{$Row[0]}</td>";
       echo "<td>{$Row[1]}</td>";
       echo "<td>{$Row[2]}</td>";
       echo "<td align='right'>{$Row[3]}</td>";
       echo "<td>{$Row[4]}</td></tr>\n";
   }
   echo "</table>\n";}
```

```
$Checking->getBalance();
                $CheckNumber = 1022;
                $Checking->getCheckAmount($CheckNumber);
```

member selection notation (->)
Properties DON'T have ()

```
        $CheckAmount = 52.10;
        $Checking->Balance = $Checking->Balance + $CheckAmount
```

Working with Database Connections as Objects

Procedural way:

```
  $DBConnect = mysql_connect("php_db", "CitizenKane",
                                        "rosebud");
  mysql_select_db("real_estate", $DBConnect);
```

OOP way:

```
$DBConnect = new mysqli("php_db", "CitizenKane",
                               "rosebud", "real_estate");
```

Explicit close the mysqli class

```
$DBConnect->close();
```

Show errors
mysqli_connect_errno() or mysqli_connect_error()

```
if ($DBConnect->connect_errno){
     echo( "Error:" + $DBConnect->connect_errno);
```

Execute SQL Statements

To return the fields in the current row of a resultset into an indexed array use:
The fetch_row() method of the mysqli class

To return the fields in the current row of a resultset into an associative array use:
The fetch_assoc() method of the msqli class

Defining Custom PHP Classes

Classes:
Help make complex programs easier to manage
Hide information
that users of a class do not need to access or know about
Make it easier to:
reuse code or
distribute your code
for use in other programs

```
class ClassName {
        data member definitions
        member function definitions
}
```

ClassName: the name of the new class
begin with an uppercase letter
to distinguish them from other identifiers

```
include(ExternalFilesPathName)
require(ExternalFilesPathName)
include_once(ExternalFilesPathName)
require_once(ExternalFilesPathName)
```

Collecting Garbage
unset(VariableName)

three access specifiers in PHP: public, private, and protected

public access specifier
allows anyone to call a class’s member function or to modify and retrieve a data member

private access specifier
Private member only accessible in class defining it

protected access specifier
In between public and private
Accessible only within:
the class that declared it or
inheriting class

```
class BankAccount {
    data member and member function definitions
    }
    $Checking = new BankAccount();
```

class_exists(ClassName)

```
if (class_exists("BankAccount"))
        $Checking = new BankAccount();
else
        echo "<p>The BankAccount class is not available!</p>\n";
```

```
if ($Checking instanceof BankAccount)
 echo "The \$Checking object is instantiated from the
 BankAccount class.</p>\n";
```

include() and require() functions
insert the contents of an external file, called an include file, into script
Include executes but gives warning if included file not found
Require stops executing causing fatal error

include_once() and require_once() functions
only include an external file once during the processing of a script

Serialization
the process of converting an object into a string that you can store for reuse

serialize(object name)
To serialize an object
$SavedAccount = serialize($Checking);

unserialize(StringVariable) function
converts serialized data back into an object
$Checking = unserialize($SavedAccount);

```
class BankAccount {
        public $Balance = 958.20;
        public function withdrawal($Amount) {
        $this->Balance -= $Amount;
        }
}
if (class_exists("BankAccount"))
        $Checking = new BankAccount();
else
        exit("<p>The BankAccount class is not available!</p>");

printf("<p>Your checking account balance is $%.2f.</p>",
        $Checking->Balance);
        $Cash = 200;
        $Checking->withdrawal(200);
        printf("<p>After withdrawing $%.2f, your checking account
        balance is $%.2f.</p>", $Cash, $Checking->Balance);
```

Setters and Getters
Client can call it
To retrieve or modify the value of a data member

```
class BankAccount {
                private $Balance = 0;
                public function setBalance($NewValue) {
                        $this->Balance = $NewValue;
                }
                public function getBalance() {
                        return $this->Balance;
                }
}

if (class_exists("BankAccount"))
                $Checking = new BankAccount();
else
                exit("<p>The BankAccount class is not
                                available!</p>");
$Checking->setBalance(100);
echo "<p>Your checking account balance is "
                . $Checking->getBalance() . "</p>\n";
```

Initializing with Constructor

```
class BankAccount {
                private $AccountNumber;
                private $CustomerName;
                private $Balance;
        // function BankAccount , same name as the class
                function __construct() {
                        $this->AccountNumber = 0;
                        $this->Balance = 0;
                        $this->CustomerName = "";
                }
}
```

destructor function
called when the object is destroyed
cleans up any resources allocated to object after it is destroyed
To add a destructor function to a PHP class, create a function named __destruct()

```
function __construct() {
        $DBConnect = new mysqli("php_db",
                "CitizenKane","rosebud", "real_estate");
}

function __destruct() {
                $DBConnect->close();
}
```

M9 Security

Sunday, April 9, 2017 5:17 AM

Ways to assign privileges
1. directly manipulate tables

Using INSERT, UPDATE and DELETE statements
MySQL will have to be reloaded
or the privileges flushed
e.g.using FLUSH PRIVILEGES statement

PASSWORD() function
To add a record directly to the password field

2. GRANT statement (easier)

GRANT privilege ON table_or_database_name TO user@hostname IDENTIFIED BY 'password'.

Guidelines:
1. Never give anyone access to the user table
   - except MySQL root accounts
   - b/c it stores the real password (i.e., encrypted password)
2. Learn the MySQL access privilege system.
3. Grant minimal privileges.
   - Never grant privileges to all hosts.
   - Use Show Grant (to check which accounts have access to what).
   - Use Revoke to remove unnecessary privileges
4. Make sure root user has a password
   - mysql -u root should not be able to login (by default it can)
5. DON'T put plain-text passwords in database
   - Sol'n: Use one-way hashing function and store the hash value

e.g., MD5(), SHA1()

Register_globals

```
if (authenticated_user()) {
   $authorized = true;    // define $authorized = true only
if user is authenticated
}
```

SSL (Secure Sockets Layer)

- Developed by Netscape (1994)
- For encryption and identity assurance.
  - makes it hard to intercept and read data
    - data passed between the web server and browser is private and secure

Database and Table names in a GRANT statement:
.
All tables in a database
*
All tables in the current database
dbname.*
All tables in the named database
dbname.tbname
The named table in the named database

REVOKE statement
- Takes away permissions

6. NO dictionary passwords
   - Use min. 8 mixed characters / numbers / symbols
7. Use a firewall.
   - protects you from at least 50% of all types of exploits in any software.
   - Put MySQL behind the firewall
8. Try to scan your ports from the Internet using a tool such as nmap.
   - port 3306
     - MySQL uses it by Default.
     - SHOULDN'T be accessible from untrusted hosts.
   - To check if MySQL port is open: telnet server_host 3306
     - If you get a connection and some garbage characters, the port is open
     - Sol'n: close on firewall or router
     - If telnet hangs or the connection is refused, the port is blocked

Port Forwarding (AKA Tunnelling)
- Intecepting data meant for one IP/Port combination and sending it to another one
  Usually done by program running on host
  Can also be done by intermediate hardware
  e.g., router, proxy server, firewall
- Packets have to be rewritten
- Can hide what services are running on network by:
  Using 1 IP address for all incoming traffic
  And dropping traffic unrelated firewall's services
- Pros:
  saves Public IP addresses
  Limit access in and out of network
  Hide services and servers
  Transparent to sender

- works through a combination of programs and encryption/decryption routines
  - that exist on the:
    - web server and
    - web browsers
- **IP**
  - Two versions: IPv4 and IPv6
    - IPv4 is used by default on most networks b/c
      - older / established
      - Well understood
- **Transmission Control Protocol (TCP)**.
  - most common protocol used in addition to IP
  - protocol that uses a set of rules to exchange messages with other internet points
    - at the data packet level.
  - **Connection Oriented**: ensures reliable, flow-controlled data packet delivery.

SSL Interaction

- Browser checks certificate
  - makes sure it is the real site
- Determine encryption types that both (browser and server) can use to understand each other
- Browser and Server send each other unique codes
  - used encrypting data
- browser & server start talking using the encryption
  - browser shows the encrypting icon
  - pages are processed secured

Internet Technologies and Applications

Hypertext mark up Language (HTML)
specialized coding language used to encode content so it can displayed in a web browser

Web browser
software with a user-friendly, graphics-capable interface that enables users to connect to and navigate websites on the internet.

Hypertext Transfer Protocol (HTTP)
set of rules used in exchanging files (such as text, graphics, sound and video) for display on the world wide web.

Virtual Private Networks (VPN)
a secure and encrypted connection between two points across the internet.
transfer information by encrypting the data in IP packets and sending the packets over the internet by a process called Tunnelling.

VPNs reduce costs: Networking & Staff

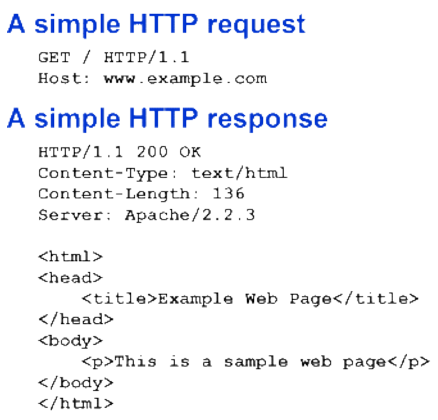
VPNs faster for international networks
Alternative: waiting for links to be established by carriers

How Data travels on VPN:
PC -> firewall (data encrypted) -> Your ISP -> tunnels -> recipient's ISP -> recipient's firewall (decrypted) -> recipient's PC

Extensible Mark up language (XML)
mark up language similar to HTML

Intranets
network that is:
internal to an organization
uses internet technology.

Extranets
Networks that are:
available to users inside and outside of a company and use internet technology.



A simple HTTP request

```
GET / HTTP/1.1
Host: www.example.com
```

A simple HTTP response

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 136
Server: Apache/2.2.3

<html>
<head>
    <title>Example Web Page</title>
</head>
<body>
    <p>This is a sample web page</p>
</body>
</html>
```