ACTIVITY ANSWER SHEET

Name	Gabutan, Nica
Section:	3R1

Instructions:
1. Push your output on your GITHUB repository.
2. Use the answer sheet provided save it as PDF file then push it to your GitHub.
3. Answer the ff. problems write it on the answer sheet.
4. Late submissions will no longer be accepted.
5. Caught copying outputs of others will be given sanctions.
6. Failure to follow these instructions will be given sanctions.

Activity 1: Control Structures
1. Write down the syntax in PHP for the ff.

1. if	if (*condition*) {
	   code to be executed if condition is true;
	}

2. if…else	if (*condition*) {
	   code to be executed if condition is true;
	} else {
	   code to be executed if condition is false;
	}

3. if…else if…else	if (*condition*) {
	   code to be executed if this condition is true;
	} elseif (*condition*) {
	   code to be executed if first condition is false and this condition is true;
	} else {
	   code to be executed if all conditions are false;
	}

4. switch…case	switch (*n*) {
	   case *label1:*
	     code to be executed if n=label1;
	     break;
	   case *label2:*
	     code to be executed if n=label2;
	     break;
	   case *label3:*
	     code to be executed if n=label3;
	     break;
	   ...
	   default:
	     code to be executed if n is different from all labels;
	}

5. for loop	for (*init counter; test counter; increment counter*) {
	   code to be executed for each iteration;
	}

6. do while loop	do { *code to be executed;* } while (*condition is true*);

7. while loop	while (*condition is true*) {
	   code to be executed;
	}

8. foreach loop	foreach ($array as $value) {
	  code to be executed;
	}

9. break statement	break; break labelname;
10. continue statement	continue; continue labelname;

11. try…catch	try {
	
	  // Block of code to try
	
	}
	catch(Exception e) {
	
	  // Block of code to handle errors
	
	}

2. Solve the ff. problem using PHP.
  a. Write a program that checks if value is a number (integer).
      Sample input: '1'   Sample input: 1
      Expected output: Not a number   Expected output: A number

```php
<?php
$var_name1='1';
$var_name2=1;
if (is_int($var_name1))
{
echo "A number"."\n"
;
}
else
{
echo "Not a number"."\n"
 ;
}
if (is_int($var_name2))
{
echo "A number";
}
else
{
echo "Not a number" ;
}

?>
```

  b. Write a program that checks if a value is positive or negative and odd or even.
      Sample input: 0   Sample input: -1
      Expected output: Positive & Even   Expected output: Negative and Odd

```php
<?php
function check($number){
   if($number % 2 == 0){
      if($number>=0){
         echo "Positive"." ";
      }
      else{
         echo "Negative"." ";
      }
      echo " and Even";
   }
   else{
```

```php
        if($number>0){
            echo "Positive"." ";
        }
        else{
            echo "Negative"." ";
        }
        echo "Odd";
    }
}

// Driver Code
$number = 0;
check($number)
?>



<?php
function check($number){
    if($number % 2 == 0){
        if($number>=0){
            echo "Positive"." ";
        }
        else{
            echo "Negative"." ";
        }
        echo "and Even";
    }
    else{
        if($number>0){
            echo "Positive"." ";
        }
        else{
            echo "Negative"." ";
        }
        echo "and Odd";
    }
}

// Driver Code
$number = -1;
check($number)
?>
```

c. Write a program that checks if a value is palindrome.

Sample input: Anna	Sample input: Bogart
Expected output: Palindrome	Expected output: Not a Palindrome

```php
<?php
function Palindrome($string)
{
 if ($string == strrev($string))
    return 1;
 else
        return 0;
}

// Driver Code
$original = 'anna';
if (Palindrome($original)){
    echo "Palindrome";
}
else {
echo "Not a Palindrome";
}

?>



<?php
function Palindrome($string)
{
 if ($string == strrev($string))
    return 1;
 else
        return 0;
```

```
}

// Driver Code
$original = 'Bogart';
if (Palindrome($original)){
    echo "Palindrome";
}
else {
echo "Not a Palindrome";
}

?>
```

d. Write a program to calculate and print the factorial of a number using a for loop.
   Sample input: 4
   Expected output: 24

```php
<?php
$n = 4;
$x = 1;
for($i=1;$i<=$n-1;$i++)
{
 $x*=($i+1);
}
echo "$x";
?>
```

e. Write a PHP program to generate and display the first n lines of a Floyd triangle.
   Sample input: 3
   Sample output:
       1
       2 3
       4 5 6

```php
<?php
$n = 3;
$count = 1;
for ($i = $n; $i > 0; $i--)
{
  for ($j = $i; $j < $n + 1; $j++)
  {
    printf("%4s", $count);
    $count++;
  }
        echo "\n";
  }
?>
```

Activity 2: PHP Built-in Functions

Write down the functionalities of the ff. built-in functions in PHP.

Array	The array functions allow you to access and manipulate arrays. Simple and multi-dimensional arrays are supported.
	array() - Creates an array
	array_change_key_case() - Changes all keys in an array to lowercase or uppercase
	array_chunk() - Splits an array into chunks of arrays
	array_column() - Returns the values from a single column in the input array
	array_combine() -Creates an array by using the elements from one "keys" array and one "values" array
	array_count_values() -Counts all the values of an array
	array_diff() - Compare arrays, and returns the differences (compare values only)
	array_diff_assoc()-Compare arrays, and returns the differences (compare keys and values)
	array_diff_key()- Compare arrays, and returns the differences (compare keys only)
	array_diff_uassoc()- Compare arrays, and returns the differences (compare keys and values, using a user-defined key comparison function)
	array_diff_ukey()-Compare arrays, and returns the differences (compare keys only, using a user-defined key comparison function)
	array_fill()-Fills an array with values
	array_fill_keys()-Fills an array with values, specifying keys
	array_filter()-Filters the values of an array using a callback function
	array_flip()-Flips/Exchanges all keys with their associated values in an array
	array_intersect()-Compare arrays, and returns the matches (compare values only)
	array_intersect_assoc()-Compare arrays and returns the matches (compare keys and values)
	array_intersect_key() -Compare arrays, and returns the matches (compare keys only)
	array_intersect_uassoc()-Compare arrays, and returns the matches (compare keys and values, using a user-defined key comparison function)
	array_intersect_ukey()-Compare arrays, and returns the matches (compare keys only, using a user-defined key comparison function)
	array_key_exists()-Checks if the specified key exists in the array
	array_keys()-Returns all the keys of an array
	array_map()-Sends each value of an array to a user-made function, which returns new values
	array_merge()-Merges one or more arrays into one array
	array_merge_recursive()-Merges one or more arrays into one array recursively
	array_multisort()-Sorts multiple or multi-dimensional arrays
	array_pad()-Inserts a specified number of items, with a specified value, to an array
	array_pop()-Deletes the last element of an array
	array_product()-Calculates the product of the values in an array
	array_push()-Inserts one or more elements to the end of an array
	array_rand()-Returns one or more random keys from an array
	array_reduce()-Returns an array as a string, using a user-defined function
	array_replace()-Replaces the values of the first array with the values from following arrays
	array_replace_recursive()-Replaces the values of the first array with the values from following arrays recursively
	array_reverse()-Returns an array in the reverse order
	array_search()-Searches an array for a given value and returns the key
	array_shift()-Removes the first element from an array, and returns the value of the removed element

array_slice()-Returns selected parts of an array
array_splice()-Removes and replaces specified elements of an array
array_sum()-Returns the sum of the values in an array
array_udiff()-Compare arrays, and returns the differences (compare values only, using a user-defined key comparison function)
array_udiff_assoc()-Compare arrays, and returns the differences (compare keys and values, using a built-in function to compare the keys and a user-defined function to compare the values)
array_udiff_uassoc()-Compare arrays, and returns the differences (compare keys and values, using two user-defined key comparison functions)
array_uintersect()-Compare arrays, and returns the matches (compare values only, using a user-defined key comparison function)
array_uintersect_assoc()-Compare arrays, and returns the matches (compare keys and values, using a built-in function to compare the keys and a user-defined function to compare the values)
array_uintersect_uassoc()-Compare arrays, and returns the matches (compare keys and values, using two user-defined key comparison functions)
array_unique()-Removes duplicate values from an array
array_unshift()-Adds one or more elements to the beginning of an array
array_values()-Returns all the values of an array
array_walk()-Applies a user function to every member of an array
array_walk_recursive()-Applies a user function recursively to every member of an array
arsort()-Sorts an associative array in descending order, according to the value
asort()-Sorts an associative array in ascending order, according to the value
compact()-Create array containing variables and their values
count()-Returns the number of elements in an array
current()-Returns the current element in an array
each()-Deprecated from PHP 7.2. Returns the current key and value pair from an array
end()-Sets the internal pointer of an array to its last element
extract()-Imports variables into the current symbol table from an array
in_array()-Checks if a specified value exists in an array
key()-Fetches a key from an array
krsort()-Sorts an associative array in descending order, according to the key
ksort()-Sorts an associative array in ascending order, according to the key
list()-Assigns variables as if they were an array
natcasesort()-Sorts an array using a case insensitive "natural order" algorithm
natsort()-Sorts an array using a "natural order" algorithm
next()-Advance the internal array pointer of an array
pos()-Alias of current()
prev()-Rewinds the internal array pointer
range()-Creates an array containing a range of elements
reset()-Sets the internal pointer of an array to its first element
rsort()-Sorts an indexed array in descending order
shuffle()-Shuffles an array
sizeof()-Alias of count()
sort()-Sorts an indexed array in ascending order
uasort()-Sorts an array by values using a user-defined comparison function
uksort()-Sorts an array by keys using a user-defined comparison function
usort()-Sorts an array using a user-defined comparison function

Calendar	The calendar extension contains functions that simplifies converting between different calendar formats. **It is based on the Julian Day Count, which is a count of days starting from January 1st, 4713 B.C.** **Note:** To convert between calendar formats, you must first convert to Julian Day Count, then to the calendar of your choice. **Note:** The Julian Day Count is not the same as the Julian Calendar! cal_days_in_month() - Returns the number of days in a month for a specified year and calendar cal_from_jd()-Converts a Julian Day Count into a date of a specified calendar cal_info()-Returns information about a specified calendar cal_to_jd()-Converts a date in a specified calendar to Julian Day Count easter_date()-Returns the Unix timestamp for midnight on Easter of a specified year easter_days()-Returns the number of days after March 21, that the Easter Day is in a specified year frenchtojd()-Converts a French Republican date to a Julian Day Count gregoriantojd()-Converts a Gregorian date to a Julian Day Count jddayofweek()-Returns the day of the week jdmonthname()-Returns a month name jdtofrench()-Converts a Julian Day Count to a French Republican date jdtogregorian()-Converts a Julian Day Count to a Gregorian date jdtojewish()-Converts a Julian Day Count to a Jewish date jdtojulian()-Converts a Julian Day Count to a Julian date jdtounix()-Converts Julian Day Count to Unix timestamp jewishtojd()-Converts a Jewish date to a Julian Day Count juliantojd()-Converts a Julian date to a Julian Day Count unixtojd()-Converts Unix timestamp to Julian Day Count
Date	The date/time functions allow you to get the date and time from the server where your PHP script runs. You can then use the date/time functions to format the date and time in several ways. **Note:** These functions depend on the locale settings of your server. Remember to take daylight saving time and leap years into consideration when working with these functions. checkdate()-Validates a Gregorian date date_add()-Adds days, months, years, hours, minutes, and seconds to a date date_create_from_format()-Returns a new DateTime object formatted according to a specified format date_create()-Returns a new DateTime object date_date_set()-Sets a new date date_default_timezone_get()-Returns the default timezone used by all date/time functions date_default_timezone_set()-Sets the default timezone used by all date/time functions date_diff()-Returns the difference between two dates date_format()-Returns a date formatted according to a specified format date_get_last_errors()-Returns the warnings/errors found in a date string date_interval_create_from_date_string()-Sets up a DateInterval from the relative parts of the string date_interval_format()-Formats the interval date_isodate_set()-Sets the ISO date date_modify()-Modifies the timestamp date_offset_get()-Returns the timezone offset date_parse_from_format()-Returns an associative array with detailed info about a specified date, according to a specified format date_parse()-Returns an associative array with detailed info

		about a specified date
		date_sub()-Subtracts days, months, years, hours, minutes, and seconds from a date
		date_sun_info()-Returns an array containing info about sunset/sunrise and twilight begin/end, for a specified day and location
		date_sunrise()-Returns the sunrise time for a specified day and location
		date_sunset()-Returns the sunset time for a specified day and location
		date_time_set()-Sets the time
		date_timestamp_get()-Returns the Unix timestamp
		date_timestamp_set()-Sets the date and time based on a Unix timestamp
		date_timezone_get()-Returns the time zone of the given DateTime object
		date_timezone_set()-Sets the time zone for the DateTime object
		date()-Formats a local date and time
		getdate()-Returns date/time information of a timestamp or the current local date/time
		gettimeofday()-Returns the current time
		gmdate()-Formats a GMT/UTC date and time
		gmmktime()-Returns the Unix timestamp for a GMT date
		gmstrftime()-Formats a GMT/UTC date and time according to locale settings
		idate()-Formats a local time/date as integer
		localtime()-Returns the local time
		microtime()-Returns the current Unix timestamp with microseconds
		mktime()-Returns the Unix timestamp for a date
		strftime()-Formats a local time and/or date according to locale settings
		strptime()-Parses a time/date generated with strftime()
		strtotime()-Parses an English textual datetime into a Unix timestamp
		time()-Returns the current time as a Unix timestamp
		timezone_abbreviations_list()-Returns an associative array containing dst, offset, and the timezone name
		timezone_identifiers_list()-Returns an indexed array with all timezone identifiers
		timezone_location_get()-Returns location information for a specified timezone
		timezone_name_from_abbr()-Returns the timezone name from abbreviation
		timezone_name_get()-Returns the name of the timezone
		timezone_offset_get()-Returns the timezone offset from GMT
		timezone_open()-Creates new DateTimeZone object
		timezone_transitions_get()-Returns all transitions for the timezone
		timezone_version_get()-Returns the version of the timezonedb
Directory		The directory functions allow you to retrieve information about directories and their contents.
		chdir() -Changes the current directory
		chroot()-Changes the root directory
		closedir()-Closes a directory handle
		dir()-Returns an instance of the Directory class
		getcwd()-Returns the current working directory
		opendir()-Opens a directory handle
		readdir()-Returns an entry from a directory handle
		rewinddir()-Resets a directory handle
		scandir()-Returns an array of files and directories of a specified directory
Error		The error functions are used to deal with error handling and logging. The error functions allow us to define own error handling

	rules, and modify the way the errors can be logged. The logging functions allow us to send messages directly to other machines, emails, or system logs. The error reporting functions allow us to customize what level and kind of error feedback is given. debug_backtrace()-Generates a backtrace debug_print_backtrace()-Prints a backtrace error_clear_last()-Clears the last error error_get_last()-Returns the last error that occurred error_log()-Sends an error message to a log, to a file, or to a mail account error_reporting()-Specifies which errors are reported restore_error_handler()-Restores the previous error handler restore_exception_handler()-Restores the previous exception handler set_error_handler()-Sets a user-defined error handler function set_exception_handler()-Sets a user-defined exception handler function trigger_error()-Creates a user-level error message user_error()-Alias of trigger_error()
File System	The filesystem functions allow you to access and manipulate the filesystem. basename() chgrp() chchown()mod() clearstatcache() copy() delete() dirname() disk_free_space() disk_total_space() diskfreespace() fclose() feof() fflush() fgetc() fgetcsv() fgets() fgetss() file() file_exists() file_get_contents() file_put_contents() fileatime() filectime() filegroup() fileinode() filemtime() fileowner() fileperms() filesize() filetype() flock() fnmatch() fopen() fpassthru() fputcsv() fputs() fread() fscanf() fseek() fstat() ftell() ftruncate() fwrite() glob() is_dir()

	is_executable() is_file() is_link() is_readable() is_uploaded_file() is_writable() is_writeable() lchgrp() lchown() link() linkinfo() lstat() mkdir() move_uploaded_file() parse_ini_file() parse_ini_string() pathinfo() pclose() popen() readfile() readlink() realpath() realpath_cache_get() realpath_cache_size() rename() rewind() rmdir() set_file_buffer() stat() symlink() tempnam() tmpfile() touch() umask() unlink()
Filter	This PHP filters is used to validate and filter data coming from insecure sources, like user input. filter_has_var() - Checks whether a variable of a specified input type exist filter_id() - Returns the filter ID of a specified filter name filter_input() - Gets an external variable (e.g. from form input) and optionally filters it filter_input_array() - Gets external variables (e.g. from form input) and optionally filters them filter_list() - Returns a list of all supported filter names filter_var() - Filters a variable with a specified filter filter_var_array() - Gets multiple variables and filter them

FTP	The FTP functions give client access to file servers through the File Transfer Protocol (FTP).
	    The FTP functions are used to open, login and close connections, as well as upload, download, rename, delete, and get information on files from file servers. Not all of the FTP functions will work with every server or return the same results. The FTP functions became available with PHP 3.
	    If you only wish to read from or write to a file on an FTP server, consider using the ftp:// wrapper with the Filesystem functions which provide a simpler and more intuitive interface.
	
	ftp_alloc() - Allocates space for a file to be uploaded to the FTP server
	ftp_cdup() - Changes to the parent directory on the FTP server
	ftp_chdir() - Changes the current directory on the FTP server
	ftp_chmod() - Sets permissions on a file via FTP
	ftp_close() - Closes an FTP connection
	ftp_connect() - Opens an FTP connection
	ftp_delete() - Deletes a file on the FTP server
	ftp_exec() - Executes a command on the FTP server

	ftp_fget() - Downloads a file from the FTP server and saves it into an open local file ftp_fput() - Uploads from an open file and saves it to a file on the FTP server ftp_get() - Downloads a file from the FTP server ftp_get_option() - Returns runtime options of the FTP connection ftp_login() - Logs in to the FTP connection ftp_mdtm() - Returns the last modified time of a specified file ftp_mkdir() - Creates a new directory on the FTP server ftp_mlsd() - Returns the list of files in the specified directory ftp_nb_continue() - Continues retrieving/sending a file (non-blocking) ftp_nb_fget() - Downloads a file from the FTP server and saves it into an open file (non-blocking) ftp_nb_fput() - Uploads from an open file and saves it to a file on the FTP server (non-blocking) ftp_nb_get() - Downloads a file from the FTP server (non-blocking) ftp_nb_put() - Uploads a file to the FTP server (non-blocking) ftp_nlist() - Returns a list of files in the specified directory on the FTP server ftp_pasv() - Turns passive mode on or off ftp_put() - Uploads a file to the FTP server ftp_pwd() - Returns the current directory name ftp_quit() - Alias of ftp_close() ftp_raw() - Sends a raw command to the FTP server ftp_rawlist() - Returns a list of files with file information from a specified directory ftp_rename() - Renames a file or directory on the FTP server ftp_rmdir() - Deletes an empty directory on the FTP server ftp_set_option() - Sets runtime options for the FTP connection ftp_site() - Sends an FTP SITE command to the FTP server ftp_size() - Returns the size of the specified file ftp_ssl_connect() - Opens a secure SSL-FTP connection ftp_systype() - Returns the system type identifier of the FTP server
Libxml	The libxml functions and constants are used together with SimpleXML, XSLT and DOM functions. libxml_clear_errors() - Clears the libxml error buffer libxml_disable_entity_loader() - Enables the ability to load external entities libxml_get_errors() - Gets the errors from the the libxml error buffer libxml_get_last_error() - Gets the last error from the the libxml error buffer libxml_set_external_entity_loader() - Changes the default external entity loader libxml_set_streams_context() - Sets the streams context for the next libxml document load or write libxml_use_internal_errors() - Disables the standard libxml errors and enables user error handling
Mail	The mail() function allows you to send emails directly from a script. ezmlm_hash() - Calculates the hash value needed by EZMLM mail() - Allows you to send emails directly from a script
Math	The math functions can handle values within the range of integer and float types. abs() - Returns the absolute (positive) value of a number acos() - Returns the arc cosine of a number acosh() - Returns the inverse hyperbolic cosine of a number asin() - Returns the arc sine of a number asinh() - Returns the inverse hyperbolic sine of a number

Misc	atan() - Returns the arc tangent of a number in radians atan2() - Returns the arc tangent of two variables x and y atanh() - Returns the inverse hyperbolic tangent of a number base_convert() - Converts a number from one number base to another bindec() - Converts a binary number to a decimal number ceil() - Rounds a number up to the nearest integer cos() - Returns the cosine of a number cosh() - Returns the hyperbolic cosine of a number decbin() - Converts a decimal number to a binary number dechex() - Converts a decimal number to a hexadecimal number decoct() - Converts a decimal number to an octal number deg2rad() - Converts a degree value to a radian value exp() - Calculates the exponent of e expm1() - Returns exp(x) - 1 floor() - Rounds a number down to the nearest integer fmod() - Returns the remainder of x/y getrandmax() - Returns the largest possible value returned by rand() hexdec() - Converts a hexadecimal number to a decimal number hypot() - Calculates the hypotenuse of a right-angle triangle intdiv() - Performs integer division is_finite() - Checks whether a value is finite or not is_infinite() - Checks whe ther a value is infinite or not is_nan() - Checks whether a value is 'not-a-number' lcg_value() - Returns a pseudo random number in a range between 0 and 1 log() - Returns the natural logarithm of a number log10() - Returns the base-10 logarithm of a number log1p() - Returns log(1+number) max() - Returns the highest value in an array, or the highest value of several specified values min() - Returns the lowest value in an array, or the lowest value of several specified values mt_getrandmax() - Returns the largest possible value returned by mt_rand() mt_rand() - Generates a random integer using Mersenne Twister algorithm mt_srand() - Seeds the Mersenne Twister random number generator octdec() - Converts an octal number to a decimal number pi() - Returns the value of PI pow() - Returns x raised to the power of y rad2deg() - Converts a radian value to a degree value rand() - Generates a random integer round() - Rounds a floating-point number sin() - Returns the sine of a number sinh() - Returns the hyperbolic sine of a number sqrt() - Returns the square root of a number srand() - Seeds the random number generator tan() - Returns the tangent of a number tanh() - Returns the hyperbolic tangent of a number
Misc	The misc. functions were only placed here because none of the other categories seemed to fit. connection_aborted() - Checks whether the client has disconnected connection_status() - Returns the current connection status connection_timeout() - Deprecated from PHP 4.0.5. Checks whether the script has timed out constant() - Returns the value of a constant define() - Defines a constant defined() - Checks whether a constant exists die() - Alias of exit() eval() - Evaluates a string as PHP code exit() - Prints a message and exits the current script get_browser() - Returns the capabilities of the user's browser __halt_compiler() - Halts the compiler execution

	highlight_file() - Outputs a file with the PHP syntax highlighted highlight_string() - Outputs a string with the PHP syntax highlighted hrtime() - Returns the system's high resolution time ignore_user_abort() - Sets whether a remote client can abort the running of a script pack() - Packs data into a binary string php_strip_whitespace() - Returns the source code of a file with PHP comments and whitespace removed show_source() - Alias of highlight_file() sleep() - Delays code execution for a number of seconds sys_getloadavg() - Returns the system load average time_nanosleep() - Delays code execution for a number of seconds and nanoseconds time_sleep_until() - Makes a script sleep until the specified time uniqid() - Generates a unique ID unpack() - Unpacks data from a binary string usleep() - Delays code execution for a number of microseconds
MySQLi	The MySQLi functions allows you to access MySQL database servers. **Note:** The MySQLi extension is designed to work with MySQL version 4.1.13 or newer. affected_rows() - Returns the number of affected rows in the previous MySQL operation autocommit() - Turns on or off auto-committing database modifications begin_transaction() - Starts a transaction change_user() - Changes the user of the specified database connection character_set_name() - Returns the default character set for the database connection close() - Closes a previously opened database connection commit() - Commits the current transaction connect() - Opens a new connection to the MySQL server connect_errno() - Returns the error code from the last connection error connect_error() - Returns the error description from the last connection error data_seek() - Adjusts the result pointer to an arbitrary row in the result-set debug() - Performs debugging operations dump_debug_info() - Dumps debugging info into the log errno() - Returns the last error code for the most recent function call error() - Returns the last error description for the most recent function call error_list() - Returns a list of errors for the most recent function call fetch_all() - Fetches all result rows as an associative array, a numeric array, or both fetch_array() - Fetches a result row as an associative, a numeric array, or both fetch_assoc() - Fetches a result row as an associative array fetch_field() - Returns the next field in the result-set, as an object fetch_field_direct() - Returns meta-data for a single field in the result-set, as an object fetch_fields() - Returns an array of objects that

represent the fields in a result-set

fetch_lengths() - Returns the lengths of the columns of the current row in the result-set

fetch_object() - Returns the current row of a result-set, as an object

fetch_row() - Fetches one row from a result-set and returns it as an enumerated array

field_count() - Returns the number of columns for the most recent query

field_seek() - Sets the field cursor to the given field offset

get_charset() - Returns a character set object

get_client_info() - Returns the MySQL client library version

get_client_stats() - Returns statistics about client per-process

get_client_version() - Returns the MySQL client library version as an integer

get_connection_stats() - Returns statistics about the client connection

get_host_info() - Returns the MySQL server hostname and the connection type

get_proto_info() - Returns the MySQL protocol version

get_server_info() - Returns the MySQL server version

get_server_version() - Returns the MySQL server version as an integer

info() - Returns information about the last executed query

init() - Initializes MySQLi and returns a resource for use with real_connect()

insert_id() - Returns the auto-generated id from the last query

kill() - Asks the server to kill a MySQL thread

more_results() - Checks if there are more results from a multi query

multi_query() - Performs one or more queries on the database

next_result() - Prepares the next result-set from multi_query()

options() - Sets extra connect options and affect behavior for a connection

ping() - Pings a server connection, or tries to reconnect if the connection has gone down

poll() - Polls connections

prepare() - Prepares an SQL statement for execution

query() - Performs a query against a database

real_connect() - Opens a new connection to the MySQL server

real_escape_string() - Escapes special characters in a string for use in an SQL statement

real_query() - Executes a single SQL query

reap_async_query() - Returns result from an async SQL query

refresh() - Refreshes/flushes tables or caches, or resets the replication server information

rollback() - Rolls back the current transaction for the database

select_db() - Select the default database for database queries

set_charset() - Sets the default client character set

set_local_infile_default() - Unsets user defined

	handler for load local infile command set_local_infile_handler() - Set callback function for LOAD DATA LOCAL INFILE command sqlstate() - Returns the SQLSTATE error code for the error ssl_set() - Used to establish secure connections using SSL stat() - Returns the current system status stmt_init() - Initializes a statement and returns an object for use with stmt_prepare() store_result() - Transfers a result-set from the last query thread_id() - Returns the thread ID for the current connection thread_safe() - Returns whether the client library is compiled as thread-safe use_result() - Initiates the retrieval of a result-set from the last query executed warning_count() - Returns the number of warnings from the last query in the connection
Network	The Network functions contains various network function and let you manipulate information sent to the browser by the Web server, before any other output has been sent. checkdnsrr() - Checks DNS records for *type* corresponding to *host* closelog() - Closes the connection of system logger define_syslog_variables() - Deprecated and removed in PHP 5.4. Initializes the variables used in syslog functions dns_check_record() - Alias of checkdnsrr() dns_get_mx() - Alias of getmxrr() dns_get_record() - Gets the DNS resource records associated with the specified hostname fsockopen() - Opens an Internet or Unix domain socket connection gethostbyaddr() - Returns the domain name for a given IP address gethostbyname() - Returns the IPv4 address for a given domain/host name gethostbynamel() - Returns a list of IPv4 address for a given domain/host name gethostname() - Returns the host name getmxrr() - Returns the MX records for the specified internet host name getprotobyname() - Returns the protocol number for a given protocol name getprotobynumber() - Returns the protocol name for a given protocol number getservbyname() - Returns the port number for a given Internet service and protocol getservbyport() - Returns the Internet service for a given port and protocol header_register_callback() - Calls a header function header_remove() - Removes an HTTP header previously set with the header()function header() - Sends a raw HTTP header to a client headers_list() - Returns a list of response headers to be sent to the browser headers_sent() - Checks if/where headers have been

	sent http_response_code() - Sets or returns the HTTP response status code inet_ntop() - Converts a 32bit IPv4 or 128bit IPv6 address into a readable format inet_pton() - Converts a readable IP address into a packed 32bit IPv4 or 128bit IPv6 format ip2long() - Converts an IPv4 address into a long integer long2ip() - Converts a long integer address into a string in IPv4 format openlog() - Opens the connection of system logger pfsockopen() - Opens a persistent Internet or Unix domain socket connection setcookie() - Defines a cookie to be sent along with the rest of the HTTP headers setrawcookie() - Defines a cookie (without URL encoding) to be sent along with the rest of the HTTP headers socket_get_status() - Alias of stream_get_meta_data() socket_set_blocking() - Alias of stream_set_blocking() socket_set_timeout() - Alias of stream_set_timeout() syslog() - Generates a system log message
SimpleXML	SimpleXML is an extension that allows us to easily manipulate and get XML data. SimpleXML provides an easy way of getting an element's name, attributes and textual content if you know the XML document's structure or layout. SimpleXML turns an XML document into a data structure you can iterate through like a collection of arrays and objects. __construct() - Creates a new SimpleXMLElement object __toString() - Returns the string content of an element addAttribute() - Appends an attribute to the SimpleXML element addChild() - Appends a child element the SimpleXML element asXML() - Returns a well-formed XML string (XML version 1.0) from a SimpleXML object attributes() - Returns the attributes/values of an element children() - Returns the children of a specified node count() - Counts the children of a specified node getDocNamespaces() - Returns the namespaces declared in document getName() - Returns the name of an element getNamespaces() - Returns the namespaces used in document registerXPathNamespace() - Creates a namespace context for the next XPath query saveXML() - Alias of asXML() simplexml_import_dom() - Returns a SimpleXMLElement object from a DOM node simplexml_load_file() - Converts an XML document to an object simplexml_load_string() - Converts an XML string to an object

	xpath() - Runs an XPath query on XML data PHP SimpleXML Iteration Functions current() - Returns the current element getChildren() - Returns the child elements of the current element hasChildren() - Checks whether the current element has children key() - Returns the XML tag name of the current element next() - Moves to the next element rewind() - Rewinds to the first element valid() - Checks whether the current element is valid
Stream	The Stream functions .... Streams are the way of generalizing file, network, data compression, and other operations which share a common set of functions and uses. In its simplest definition, a stream is a resource object which exhibits streamable behavior. That is, it can be read from or written to in a linear fashion, and may be able to fseek() to an arbitrary location within the stream. A wrapper is additional code which tells the stream how to handle specific protocols/encodings. set_socket_blocking() - Deprecated in PHP 5.4, and removed in PHP 7.0. Alias of stream_set_blocking() stream_bucket_prepend() stream_context_create() stream_context_get_default() stream_context_get_options() stream_context_get_params() stream_context_set_default() stream_context_set_options() stream_context_set_params() stream_copy_to_stream() - Copies data from one stream to another stream_filter_append() - Appends a filter to a stream stream_filter_prepend() stream_filter_register() stream_filter_remove() stream_get_contents() stream_get_filters() stream_get_line() stream_get_meta_data() stream_get_transports() stream_get_wrappers() stream_is_local() stream_isatty() stream_notification_callback() tream_register_wrapper() - Alias of stream_wrapper_register() stream_resolve_include_path() stream_select() stream_set_blocking() stream_set_chunk_size() stream_set_read_buffer() stream_set_timeout() stream_set_write_buffer() stream_socket_accept()

	stream_socket_client() stream_socket_enable_crypto() stream_socket_get_name() stream_socket_pair() stream_socket_recvfrom() stream_socket_sendto() stream_socket_server() stream_socket_shutdown() stream_supports_lock() stream_wrapper_register() stream_wrapper_restore() stream_wrapper_unregister()
String	The PHP string functions are part of the PHP core. No installation is required to use these functions. addcslashes()-Returns a string with backslashes in front of the specified characters addslashes() -Returns a string with backslashes in front of predefined characters bin2hex() -Converts a string of ASCII characters to hexadecimal values chop() -Removes whitespace or other characters from the right end of a string chr() -Returns a character from a specified ASCII value chunk_split()-Splits a string into a series of smaller parts convert_cyr_string()-Converts a string from one Cyrillic character-set to another convert_uudecode()-Decodes a uuencoded string convert_uuencode() -Encodes a string using the uuencode algorithm count_chars()-Returns information about characters used in a string crc32() -Calculates a 32-bit CRC for a string crypt() -One-way string hashing echo() -Outputs one or more strings explode()-Breaks a string into an array fprintf()-Writes a formatted string to a specified output stream get_html_translation_table() -Returns the translation table used by htmlspecialchars() and htmlentities() hebrev()-Converts Hebrew text to visual text hebrevc() -Converts Hebrew text to visual text and new lines (\n) into hex2bin() -Converts a string of hexadecimal values to ASCII characters html_entity_decode()-Converts HTML entities to characters htmlentities() -Converts characters to HTML entities htmlspecialchars_decode()-Converts some predefined HTML entities to characters htmlspecialchars() -Converts some predefined characters to HTML entities implode()-Returns a string from the elements of an array join()-Alias of implode() lcfirst()-Converts the first character of a string to lowercase levenshtein() -Returns the Levenshtein distance between two strings

	localeconv()-Returns locale numeric and monetary formatting information
	ltrim()-Removes whitespace or other characters from the left side of a string
	md5() -Calculates the MD5 hash of a string
	md5_file() -Calculates the MD5 hash of a file
	metaphone() -Calculates the metaphone key of a string
	money_format()-Returns a string formatted as a currency string
	nl_langinfo() -Returns specific local information
	nl2br()-Inserts HTML line breaks in front of each newline in a string
	number_format() -Formats a number with grouped thousands
	ord()-Returns the ASCII value of the first character of a string
	parse_str()-Parses a query string into variables
	print()-Outputs one or more strings
	printf() -Outputs a formatted string
	quoted_printable_decode() -Converts a quoted-printable string to an 8-bit string
	quoted_printable_encode()-Converts an 8-bit string to a quoted printable string
	quotemeta()-Quotes meta characters
	rtrim() -Removes whitespace or other characters from the right side of a string
	setlocale() -Sets locale information
	sha1() -Calculates the SHA-1 hash of a string
	sha1_file()-Calculates the SHA-1 hash of a file
	similar_text() -Calculates the similarity between two strings
	soundex() -Calculates the soundex key of a string
	sprintf()-Writes a formatted string to a variable
	sscanf()-Parses input from a string according to a format
	str_getcsv()-Parses a CSV string into an array
	str_ireplace()-Replaces some characters in a string (case-insensitive)
	str_pad()-Pads a string to a new length
	str_repeat()-Repeats a string a specified number of times
	str_replace() -Replaces some characters in a string (case-sensitive)
	str_rot13()-Performs the ROT13 encoding on a string
	str_shuffle() -Randomly shuffles all characters in a string
	str_split()-Splits a string into an array
	str_word_count() -Count the number of words in a string
	strcasecmp()-Compares two strings (case-insensitive)
	strchr()-Finds the first occurrence of a string inside another string (alias of strstr())
	strcmp()-Compares two strings (case-sensitive)
	strcoll() -compares two strings (locale based string comparison)
	strcspn() - Returns the number of characters found in a string before any part of some specified characters are found
	strip_tags()-Strips HTML and PHP tags from a string
	stripcslashes()-Unquotes a string quoted with

	addcslashes() stripslashes()-Unquotes a string quoted with addslashes() stripos() -Returns the position of the first occurrence of a string inside another string (case-insensitive) stristr() -Finds the first occurrence of a string inside another string (case-insensitive) strlen() -Returns the length of a string strnatcasecmp() -Compares two strings using a "natural order" algorithm (case-insensitive) strnatcmp()-Compares two strings using a "natural order" algorithm (case-sensitive) strncasecmp()-String comparison of the first n characters (case-insensitive) strncmp()-String comparison of the first n characters (case-sensitive) strpbrk()-Searches a string for any of a set of characters strpos()-Returns the position of the first occurrence of a string inside another string (case-sensitive) strrchr()-Finds the last occurrence of a string inside another string strrev()-Reverses a string strripos() - Finds the position of the last occurrence of a string inside another string (case-insensitive) strrpos()-Finds the position of the last occurrence of a string inside another string (case-sensitive) strspn()-Returns the number of characters found in a string that contains only characters from a specified charlist strstr() -Finds the first occurrence of a string inside another string (case-sensitive) strtok() -Splits a string into smaller strings strtolower()-Converts a string to lowercase letters strtoupper() -Converts a string to uppercase letters strtr()-Translates certain characters in a string substr()-Returns a part of a string substr_compare()-Compares two strings from a specified start position (binary safe and optionally case-sensitive) substr_count()-Counts the number of times a substring occurs in a string substr_replace()-Replaces a part of a string with another string trim()-Removes whitespace or other characters from both sides of a string ucfirst()-Converts the first character of a string to uppercase ucwords()-Converts the first character of each word in a string to uppercase vfprintf()-Writes a formatted string to a specified output stream vprintf()-Outputs a formatted string vsprintf() - Writes a formatted string to a variable wordwrap() - Wraps a string to a given number of characters
XML Parser	The XML functions lets you parse, but not validate, XML documents. XML is a data format for standardized structured document exchange. More information on XML can be found in our XML Tutorial.

	This extension uses the Expat XML parser. Expat is an event-based parser, it views an XML document as a series of events. When an event occurs, it calls a specified function to handle it. Expat is a non-validating parser, and ignores any DTDs linked to a document. However, if the document is not well formed it will end with an error message. Because it is an event-based, non validating parser, Expat is fast and well suited for web applications. The XML parser functions lets you create XML parsers and define handlers for XML events. utf8_decode() -Decodes an UTF-8 string to ISO-8859-1 utf8_encode() -Encodes an ISO-8859-1 string to UTF-8 xml_error_string() -Returns an error string from the XML parser xml_get_current_byte_index() -Returns the current byte index from the XML parser xml_get_current_column_number() -Returns the current column number from the XML parser xml_get_current_line_number() -Returns the current line number from the XML parser xml_get_error_code() -Returns an error code from the XML parser xml_parse() -Parses an XML document xml_parse_into_struct() -Parses XML data into an array xml_parser_create_ns()-Creates an XML parser with namespace support xml_parser_create()-Creates an XML parser xml_parser_free() -Frees an XML parser xml_parser_get_option() -Returns options from an XML parser xml_parser_set_option() -Sets options in an XML parser xml_set_character_data_handler() -Sets up the character data handler for the XML parser xml_set_default_handler() - Sets up the default data handler for the XML parser xml_set_element_handler() -Sets up start and end element handlers for the XML parser xml_set_end_namespace_decl_handler() -Sets up the end namespace declaration handler xml_set_external_entity_ref_handler() -Sets up the external entity reference handler for the XML parser xml_set_notation_decl_handler() - Sets up notation declaration handler for the XML parser xml_set_object() - Allows to use XML parser within an object xml_set_processing_instruction_handler()-Sets up processing instruction handler xml_set_start_namespace_decl_handler() - Sets up the start namespace declaration handler xml_set_unparsed_entity_decl_handler() -Sets handler function for unparsed entity declarations
Zip	The Zip files functions allows you to read ZIP files. zip_close() - Closes a ZIP file archive

	zip_entry_close() - Closes a ZIP directory entry zip_entry_compressedsize() - Returns the compressed file size of a ZIP directory entry zip_entry_compressionmethod() - Returns the compression method of a ZIP directory entry zip_entry_filesize() - Returns the actual file size of a ZIP directory entry zip_entry_name() - Returns the name of a ZIP directory entry zip_entry_open() - Opens a directory entry in a ZIP file for reading zip_entry_read() - Reads from an open directory entry in the ZIP file zip_open() - Opens a ZIP file archive zip_read() - Reads the next file in a open ZIP file archive
Timezones	PHP Supported Timezones Below is a complete list of the timezones supported by PHP, which are useful with several PHP date functions. • Africa • America • Antarctica • Arctic • Asia • Atlantic • Australia • Europe • Indian • Pacific AFRICA Africa/Abidjan Africa/Accra Africa/Addis_Ababa Africa/Algiers Africa/Asmara Africa/Asmera Africa/Bamako Africa/Bangui Africa/Banjul Africa/Bissau Africa/Blantyre Africa/Brazzaville Africa/Bujumbura Africa/Cairo Africa/Casablanca Africa/Ceuta Africa/Conakry Africa/Dakar Africa/Dar_es_Salaam Africa/Djibouti Africa/Douala Africa/El_Aaiun Africa/Freetown Africa/Gaborone Africa/Harare Africa/Johannesburg Africa/Juba Africa/Kampala Africa/Khartoum Africa/Kigali

	Africa/Kinshasa Africa/Lagos Africa/Libreville Africa/Lome Africa/Luanda Africa/Lubumbashi Africa/Lusaka Africa/Malabo Africa/Maputo Africa/Maseru Africa/Mbabane Africa/Mogadishu Africa/Monrovia Africa/Nairobi Africa/Ndjamena Africa/Niamey Africa/Nouakchott Africa/Ouagadougou Africa/Porto-Novo Africa/Sao_Tome Africa/Timbuktu Africa/Tripoli Africa/Tunis Africa/Windhoek America America/Adak America/Anchorage America/Anguilla America/Antigua America/Araguaina America/Argentina/Buenos_Aires America/Argentina/Catamarca America/Argentina/ComodRivadavia America/Argentina/Cordoba America/Argentina/Jujuy America/Argentina/La_Rioja America/Argentina/Mendoza America/Argentina/Rio_Gallegos America/Argentina/Salta America/Argentina/San_Juan America/Argentina/San_Luis America/Argentina/Tucuman America/Argentina/Ushuaia America/Aruba America/Asuncion America/Atikokan America/Atka America/Bahia America/Bahia_Banderas America/Barbados America/Belem America/Belize America/Blanc-Sablon America/Boa_Vista America/Bogota America/Boise America/Buenos_Aires America/Cambridge_Bay

	America/Campo_Grande
	America/Cancun
	America/Caracas
	America/Catamarca
	America/Cayenne
	America/Cayman
	America/Chicago
	America/Chihuahua
	America/Coral_Harbour
	America/Cordoba
	America/Costa_Rica
	America/Creston
	America/Cuiaba
	America/Curacao
	America/Danmarkshavn
	America/Dawson
	America/Dawson_Creek
	America/Denver
	America/Detroit
	America/Dominica
	America/Edmonton
	America/Eirunepe
	America/El_Salvador
	America/Ensenada
	America/Fort_Wayne
	America/Fortaleza
	America/Glace_Bay
	America/Godthab
	America/Goose_Bay
	America/Grand_Turk
	America/Grenada
	America/Guadeloupe
	America/Guatemala
	America/Guayaquil
	America/Guyana
	America/Halifax
	America/Havana
	America/Hermosillo
	America/Indiana/Indianapolis
	America/Indiana/Knox
	America/Indiana/Marengo
	America/Indiana/Petersburg
	America/Indiana/Tell_City
	America/Indiana/Vevay
	America/Indiana/Vincennes
	America/Indiana/Winamac
	America/Indianapolis
	America/Inuvik
	America/Iqaluit
	America/Jamaica
	America/Jujuy
	America/Juneau
	America/Kentucky/Louisville
	America/Kentucky/Monticello
	America/Knox_IN
	America/Kralendijk
	America/La_Paz
	America/Lima
	America/Los_Angeles
	America/Louisville
	America/Lower_Princes

	America/Maceio
	America/Managua
	America/Manaus
	America/Marigot
	America/Martinique
	America/Matamoros
	America/Mazatlan
	America/Mendoza
	America/Menominee
	America/Merida
	America/Metlakatla
	America/Mexico_City
	America/Miquelon
	America/Moncton
	America/Monterrey
	America/Montevideo
	America/Montreal
	America/Montserrat
	America/Nassau
	America/New_York
	America/Nipigon
	America/Nome
	America/Noronha
	America/North_Dakota/Beulah
	America/North_Dakota/Center
	America/North_Dakota/New_Salem
	America/Ojinaga
	America/Panama
	America/Pangnirtung
	America/Paramaribo
	America/Phoenix
	America/Port-au-Prince
	America/Port_of_Spain
	America/Porto_Acre
	America/Porto_Velho
	America/Puerto_Rico
	America/Rainy_River
	America/Rankin_Inlet
	America/Recife
	America/Regina
	America/Resolute
	America/Rio_Branco
	America/Rosario
	America/Santa_Isabel
	America/Santarem
	America/Santiago
	America/Santo_Domingo
	America/Sao_Paulo
	America/Scoresbysund
	America/Shiprock
	America/Sitka
	America/St_Barthelemy
	America/St_Johns
	America/St_Kitts
	America/St_Lucia
	America/St_Thomas
	America/St_Vincent
	America/Swift_Current
	America/Tegucigalpa
	America/Thule
	America/Thunder_Bay

	America/Tijuana America/Toronto America/Tortola America/Vancouver America/Virgin America/Whitehorse America/Winnipeg America/Yakutat America/Yellowknife Antarctica Antarctica/Casey Antarctica/Davis Antarctica/DumontDUrville Antarctica/Macquarie Antarctica/Mawson Antarctica/McMurdo Antarctica/Palmer Antarctica/Rothera Antarctica/South_Pole Antarctica/Syowa Antarctica/Vostok

Activity 3: Regular Expression

1. Define Regular Expression (RegEx) and provide example programming scenario where you can use (RegEx). Provide example syntax in PHP.

Regular Expressions, commonly known as "**regex**" or "**RegExp**", are a specially formatted text strings used to find patterns in text. Regular expressions are one of the most powerful tools available today for effective and efficient text processing and manipulations. For example, it can be used to verify whether the format of data i.e. name, email, phone number, etc. entered by the user was correct or not, find or replace matching string within text content, and so on.

PHP (version 5.3 and above) supports Perl style regular expressions via its preg_ family of functions. Why Perl style regular expressions? Because Perl (*Practical Extraction and Report Language*) was the first mainstream programming language that provided integrated support for regular expressions and it is well known for its strong support of regular expressions and its extraordinary text processing and manipulation capabilities.

Regular Expression Syntax

Regular expression syntax includes the use of special characters (do not confuse with the HTML special characters). The characters that are given special meaning within a regular expression, are: . * ? + [ ] ( ) { } ^ $ | \. You will need to backslash these characters whenever you want to use them literally. For example, if you want to match ".", you'd have to write \.. All other characters automatically assume their literal meanings.

The following sections describe the various options available for formulating patterns:

Character Classes

Square brackets surrounding a pattern of characters are called a character class e.g. [abc]. A character class always matches a single character out of a list of specified characters that means the expression [abc] matches only a, b or c character.

Negated character classes can also be defined that match any character except those contained within the brackets. A negated character class is defined by placing a caret (^) symbol immediately after the opening bracket, like this [^abc].

You can also define a range of characters by using the hyphen (-) character inside a character class, like [0-9]. Let's look at some examples of character classes:

RegExp	What it Does
[abc]	Matches any one of the characters a, b, or c.
[^abc]	Matches any one character other than a, b, or c.
[a-z]	Matches any one character from lowercase a to lowercase z.
[A-Z]	Matches any one character from uppercase a to uppercase z.
[a-Z]	Matches any one character from lowercase a to uppercase Z.
[0-9]	Matches a single digit between 0 and 9.
[a-z0-9]	Matches a single character between a and z or between 0 and 9.

The following example will show you how to find whether a pattern exists in a string or not using the regular expression and PHP `preg_match()` function:

Example

```php
<?php
$pattern = "/ca[kf]e/";
$text = "He was eating cake in the cafe.";
if(preg_match($pattern, $text)){
    echo "Match found!";
} else{
    echo "Match not found.";
}
?>
```

Similarly, you can use the `preg_match_all()` function to find all matches within a string:

Example

```php
<?php
$pattern = "/ca[kf]e/";
$text = "He was eating cake in the cafe.";
$matches = preg_match_all($pattern, $text, $array);
echo $matches . " matches were found.";
?>
```

Tip: Regular expressions aren't exclusive to PHP. Languages such as Java, Perl, Python, etc. use the same notation for finding patterns in text.

Predefined Character Classes

Some character classes such as digits, letters, and whitespaces are used so frequently that there are shortcut names for them. The following table lists those predefined character classes:

Shortcut	What it Does
.	Matches any single character except newline \n.
\d	matches any digit character. Same as [0-9]
\D	Matches any non-digit character. Same as [^0-9]
\s	Matches any whitespace character (space, tab, newline or carriage return character). Same as [\t\n\r]
\S	Matches any non-whitespace character. Same as [^ \t\n\r]
\w	Matches any word character (definned as a to z, A to Z,0 to 9, and the underscore). Same as [a-zA-Z_0-9]
\W	Matches any non-word character. Same as [^a-zA-Z_0-9]

The following example will show you how to find and replace space with a hyphen character in a string using regular expression and PHP preg_replace() function:

Example
```php
<?php
$pattern = "/\s/";
$replacement = "-";
$text = "Earth revolves around\nthe\tSun";
// Replace spaces, newlines and tabs
echo preg_replace($pattern, $replacement, $text);
echo "<br>";
// Replace only spaces
echo str_replace(" ", "-", $text);
?>
```

Repetition Quantifiers

In the previous section we've learnt how to match a single character in a variety of fashions. But what if you want to match on more than one character? For example, let's say you want to find out words containing one or more instances of

the letter p, or words containing at least two p's, and so on. This is where quantifiers come into play. With quantifiers you can specify how many times a character in a regular expression should match.

The following table lists the various ways to quantify a particular pattern:

RegExp	What it Does
p+	Matches one or more occurrences of the letter p.
p*	Matches zero or more occurrences of the letter p.
p?	Matches zero or one occurrences of the letter p.
p{2}	Matches exactly two occurrences of the letter p.
p{2,3}	Matches at least two occurrences of the letter p, but not more than three occurrences of the letter p.
p{2,}	Matches two or more occurrences of the letter p.
p{,3}	Matches at most three occurrences of the letter p

The regular expression in the following example will splits the string at comma, sequence of commas, whitespace, or combination thereof using the PHP preg_split() function:

Example
```php
<?php
$pattern = "/[\s,]+/";
$text = "My favourite colors are red, green and blue";
$parts = preg_split($pattern, $text);

// Loop through parts array and display substrings
foreach($parts as $part){
    echo $part . "<br>";
}
?>
```

Position Anchors

There are certain situations where you want to match at the beginning or end of a line, word, or string. To do this you can use anchors. Two common anchors are caret (^) which represent the start of the string, and the dollar ($) sign which represent the end of the string.

RegExp	What it Does
^p	Matches the letter p at the beginning of a line.
p$	Matches the letter p at the end of a line.

The regular expression in the following example will display only those names from the names array which start with the letter "J" using the PHP `preg_grep()` function:

Example
```php
<?php
$pattern = "/^J/";
$names = array("Jhon Carter", "Clark Kent", "John Rambo");
$matches = preg_grep($pattern, $names);

// Loop through matches array and display matched names
foreach($matches as $match){
    echo $match . "<br>";
}
?>
```

Pattern Modifiers

A pattern modifier allows you to control the way a pattern match is handled. Pattern modifiers are placed directly after the regular expression, for example, if you want to search for a pattern in a case-insensitive manner, you can use the i modifier, like this: /pattern/i. The following table lists some of the most commonly used pattern modifiers.

Modifier	What it Does
i	Makes the match case-insensitive manner.
m	Changes the behavior of ^ and $ to match against a newline boundary (i.e. start or end of each line within a multiline string), instead of a string boundary.
g	Perform a global match i.e. finds all occurrences.
o	Evaluates the expression only once.
s	Changes the behavior of . (dot) to match all characters, including newlines.
x	Allows you to use whitespace and comments within a regular expression for clarity.

The following example will show you how to perform a global case-insensitive search using the i modifier and the PHP preg_match_all() function.

```php
<?php
$pattern = "/color/i";
$text = "Color red is more visible than color blue in daylight.";
$matches = preg_match_all($pattern, $text, $array);
echo $matches . " matches were found.";
?>
```

Similarly, the following example shows how to match at the beginning of every line in a multi-line string using ^ anchor and m modifier with PHP preg_match_all() function.

```php
<?php
$pattern = "/^color/im";
$text = "Color red is more visible than \ncolor blue in daylight.";
$matches = preg_match_all($pattern, $text, $array);
echo $matches . " matches were found.";
?>
```

Word Boundaries

A word boundary character (\b) helps you search for the words that begins and/or ends with a pattern. For example, the regexp /\bcar/ matches the words beginning with the pattern car, and would match cart, carrot, or cartoon, but would not match oscar.

Similarly, the regexp /car\b/ matches the words ending with the pattern car, and would match scar, oscar, or supercar, but would not match cart. Likewise, the /\bcar\b/ matches the words beginning and ending with the pattern car, and would match only the word car.

The following example will highlight the words beginning with car in bold:

```php
<?php
$pattern = '/\bcar\w*/';
$replacement = '<b>$0</b>';
```

```php
$text = 'Words begining with car: cart, carrot, cartoon. Words ending with car: scar,
oscar, supercar.';
echo preg_replace($pattern, $replacement, $text);
?>
```

2. Solve the ff. problem using Regular Expressions.
        a. Write a PHP script that checks if a string contains another string
           Sample String: 'The quick brown fox'
           Test input: 'Fox'
           Expected output: Fox is found the string

```php
<?php
$str1 = 'The quick brown fox';
if (strpos($str1,'fox') !== false)
{
    echo 'Fox is found the string';
}
else
{
    echo 'Fox is not found the string';
}
?>
```

        b. Write a PHP script that removes the last word from a string.
           Sample String: 'The quick brown fox'
           Expected output: 'The quick brown'

```php
<?php
$str1 = 'The quick brown fox';
echo preg_replace('/\W\w+\s*(\W*)$/', '$1', $str1)."\n";
?>
```

        c. Write a PHP script to remove nonnumeric characters except comma and dot.
           Sample String: '/$123,34.00A#'
           Expected output: 123,34.00

```php
<?php
$str1 = "$12,334.00A";
echo preg_replace("/[^0-9,.]/", "", $str1)."\n";
?>
```

        d. Write a PHP script to extract text (within parenthesis) from a string.
           Sample String: 'The quick brown [fox].'
           Expected output: Fox

```php
<?php
$my_text = 'The quick brown [fox].';
preg_match('#\[(.*?)\]#', $my_text, $match);
print $match[1]."\n";
?>
```

e. Write a PHP script to remove all characters from a string except a-z A-Z 0-9 or " ".
   Sample String: 'abcde$ddfd @abcd)der]'
   Expected output: abcdeddfd abcd der

```php
<?php

$string = 'abcde$ddfd @abcd )der]';

$newstr = preg_replace("/[^A-Za-z0-9 ]/", '', $string);

echo $newstr."\n";

?>
```

Activity 4: Error Handling

1. List down the different PHP errors. Provide example code on how to handle these errors.

Parse error or Syntax Error:
Fatal Error:
Warning Errors :
Notice Error:

Basic Error Handling: Using the die() function

Example

```php
<?php
if(file_exists("mytestfile.txt")) {
  $file = fopen("mytestfile.txt", "r");
} else {
  die("Error: The file does not exist.");
}
?>
```