Appendix A: HTML Tags

Tag	Description
<!--...-->	Defines a comment
<!DOCTYPE>	Defines the document type
<a>	Defines an anchor
<abbr>	Defines an abbreviation
<acronym>	Defines an acronym
<address>	Defines an address element
<applet>	Defines an applet
<area>	Defines an area inside an image map
<b>	Defines bold text
<base>	Defines a base URL for all the links in a page
<basefont>	Defines a base font
<bdo>	Defines the direction of text display
<big>	Defines big text
<blockquote>	Defines a long quotation
<body>	Defines the body element
<br>	Inserts a single line break
<button>	Defines a push button
<caption>	Defines a table caption
<center>	Defines centered text
<cite>	Defines a citation
<code>	Defines computer code text
<col>	Defines attributes for table columns
<colgroup>	Defines groups of table columns

<dd>	Defines a definition description
<del>	Defines deleted text
<dir>	Defines a directory list
<div>	Defines a section in a document
<dfn>	Defines a definition term
<dl>	Defines a definition list
<dt>	Defines a definition term
<em>	Defines emphasized text
<fieldset>	Defines a fieldset
<font>	Defines text font, size, and color
<form>	Defines a form
<frame>	Defines a sub window (a frame)
<frameset>	Defines a set of frames
<h1> to <h6>	Defines header 1 to header 6
<head>	Defines information about the document
<hr>	Defines a horizontal rule
<html>	Defines an html document
<i>	Defines italic text
<iframe>	Defines an inline sub window (frame)
<img>	Defines an image
<input>	Defines an input field
<ins>	Defines inserted text
<isindex>	Defines a single-line input field
<kbd>	Defines keyboard text
<label>	Defines a label for a form control

<legend>	Defines a title in a fieldset
<li>	Defines a list item
<link>	Defines a resource reference
<map>	Defines an image map
<menu>	Defines a menu list
<meta>	Defines meta information
<noframes>	Defines a noframe section
<noscript>	Defines a noscript section
<object>	Defines an embedded object
<ol>	Defines an ordered list
<optgroup>	Defines an option group
<option>	Defines an option in a drop-down list
<p>	Defines a paragraph
<param>	Defines a parameter for an object
<pre>	Defines preformatted text
<q>	Defines a short quotation
<s>	Defines strikethrough text
<samp>	Defines sample computer code
<script>	Defines a script
<select>	Defines a selectable list
<small>	Defines small text
<span>	Defines a section in a document
<strike>	Defines strikethrough text
<strong>	Defines strong text
<style>	Defines a style definition

<sub>	Defines subscripted text
<sup>	Defines superscripted text
<table>	Defines a table
<tbody>	Defines a table body
<td>	Defines a table cell
<textarea>	Defines a text area
<tfoot>	Defines a table footer
<th>	Defines a table header
<thead>	Defines a table header
<title>	Defines the document title
<tr>	Defines a table row
<tt>	Defines teletype text
<u>	Defines underlined text
<ul>	Defines an unordered list
<var>	Defines a variable

Appendix B: Special Characters

Character	Code	Name
“	&#34;	&quot;
&	&#38;	&amp;
<	&#60;	&lt;
>	&#62;	&gt;
Non-breaking space	&#160;	&nbsp;
¡	&#161;	&iexcl;
¢	&#162;	&cent;
£	&#163;	&pound;
¤	&#164;	&curren;
¥	&#165;	&yen;
¦	&#166;	&brvbar;
§	&#167;	&sect;
¨	&#168;	&uml;
©	&#169;	&copy;
ª	&#170;	&ordf;
«	&#171;	&laquo;
¬	&#172;	&not;
®	&#173;	&reg;
¯	&#174;	&macr;
°	&#175;	&deg;
°	&#176;	&deg;
±	&#177;	&plusmn

²	&#178;	&sup2;
³	&#179;	&sup3;
´	&#180;	&acute;
µ	&#181;	&micro;
¶	&#182;	&para;
·	&#183;	&middot;
¸	&#184;	&cedil;
¹	&#185;	&sup1;
º	&#186;	&ordm;
»	&#187;	&raquo;
¼	&#188;	&frac14;
½	&#189;	&frac12;
¾	&#190;	&frac34;
¿	&#191;	&iquest;
À	&#192;	&Agrave;
Á	&#193;	&Aacute;
Â	&#194;	&Acirc;
Ã	&#195;	&Atilde;
Ä	&#196;	&Auml;
Å	&#197;	&Aring;
Æ	&#198;	&AElig;
Ç	&#199;	&Ccedil;
È	&#200;	&Egrave;
É	&#201;	&Eacute;
Ê	&#202;	&Ecirc;
Ë	&#203;	&Euml;
Ì	&#204;	&Igrave;
Í	&#205;	&Iacute;
Î	&#206;	&Icirc;

Ï	&#207;	&Iuml;
Ð	&#208;	&ETH;
Ñ	&#209;	&Ntilde;
Ò	&#210;	&Ograve;
Ó	&#211;	&Oacute;
Ô	&#212;	&Ocirc;
Õ	&#213;	&Otilde;
Ö	&#214;	&Ouml;
×	&#215;	&times;
Ø	&#216;	&Oslash;
Ù	&#217;	&Ugrave;
Ú	&#218;	&Uacute;
Û	&#219;	&Ucirc;
Ü	&#220;	&Uuml;
Ý	&#221	&Yacute;
Þ	&#222;	&THORN;
ß	&#223;	&szlig;
à	&#224;	&agrave;
á	&#225;	&aacute;
â	&#226;	&acirc;
ã	&#227;	&atilde;
ä	&#228;	&auml;
å	&#229;	&aring;
æ	&#230;	&aelig;
ç	&#231;	&ccedil;
è	&#232;	&egrave;
é	&#233;	&eacute;
ê	&#234;	&ecirc;
ë	&#235;	&euml;

ì	&#236;	&igrave;
í	&#237;	&iacute;
î	&#238;	&icirc;
ï	&#239;	&iuml;
ð	&#240;	&eth;
ñ	&#241;	&ntilde;
ò	&#242;	&ograve;
ó	&#243;	&oacute;
ô	&#244;	&ocirc;
õ	&#245;	&otilde;
ö	&#246;	&ouml;
÷	&#247;	&divide;
ø	&#248;	&oslash;
ù	&#249;	&ugrave;
ú	&#250;	&uacute;
û	&#251;	&ucirc;
ü	&#252;	&uuml;
ý	&#253;	&yacute;
Þ	&#254;	&thorn;
ÿ	&#255;	&yuml;

Appendix C: PHP Functions, Constants, Filters, & Parameters

Array Functions

Function	Description
array()	Creates an array
array_change_key_case()	Returns an array with all keys in lowercase or uppercase
array_chunk()	Splits an array into chunks of arrays
Array_column()	Return the values from a single column in the input array
array_combine()	Creates an array by using one array for keys and another for values
array_count_values()	Returns an array with the number of occurrences for each value
array_diff()	Compares array values, and returns the differences
array_diff_assoc()	Compares array keys and values, and returns the differences
array_diff_key()	Compares array keys, and returns the differences
array_diff_uassoc()	Compares array keys and values, with an additional user-made function check, and returns the differences
array_diff_ukey()	Compares array keys, with an additional user-made function check, and returns the differences
array_fill()	Fills an array with values
array_fill_keys()	Fill an array with values, specifying keys

array_filter()	Filters elements of an array using a user-made function
array_flip()	Exchanges all keys with their associated values in an array
array_intersect()	Compares array values, and returns the matches
array_intersect_assoc()	Compares array keys and values, and returns the matches
array_intersect_key()	Compares array keys, and returns the matches
array_intersect_uassoc()	Compares array keys and values, with an additional user-made function check, and returns the matches
array_intersect_ukey()	Compares array keys, with an additional user-made function check, and returns the matches
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
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 elements from passed arrays into the first array
array_replace_recursive	Replaces elements from passed arrays into the first array recursive
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()	Compares array values in a user-made function and returns an array
array_udiff_assoc()	Compares array keys, and compares array values in a user-made function, and returns an array
array_udiff_uassoc()	Compares array keys and array values in user-made functions, and returns an array
array_uintersect()	Compares array values in a user-made function and returns an array
array_uintersect_assoc()	Compares array keys, and compares array values in a user-made function, and returns an array
array_uintersect_uassoc()	Compares array keys and array values in user-made functions, and returns an array
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 array in reverse order and maintains index association
asort()	Sorts an array and maintains index association
compact()	Creates an array containing variables and their values
count()	Counts elements in an array, or properties in an object
current()	Returns the current element in an array
each()	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
key_exists()	Alias of array_key_exists
krsort()	Sorts an array by key in reverse order
ksort()	Sorts an array by 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 array in reverse order
shuffle()	Shuffles an array
sizeof()	Alias of count()
sort()	Sorts an array
uasort()	Sorts an array with a user-defined function and maintains index association
uksort()	Sorts an array by keys using a user-defined function
usort()	Sorts an array by values using a user-defined function

Array Constants

Constant	Description
CASE_LOWER	Used with array_change_key_case() to convert array keys to lowercase
CASE_UPPER	Used with array_change_key_case() to convert array keys to uppercase
SORT_ASC	Used with array_multisort() to sort in ascending order
SORT_DESC	Used with array_multisort() to sort in descending order

SORT_REGULAR	Used to compare items normally
SORT_NUMERIC	Used to compare items numerically
SORT_STRING	Used to compare items as strings
SORT_LOCALE_STRING	Used to compare items as strings, based on the current locale
SORT_NATURAL	Used to compare items as strings using “natural ordering”
COUNT_NORMAL	Used to return the number of elements in an array
COUNT_RECURSIVE	Used to return the number of recursive elements in an array
Extract Type and Prefix Parameters	**Description**
EXTR_OVERWRITE	If there is a collision, overwrite the existing variable.
EXTR_SKIP	If there is a collision, don't overwrite the existing variable.
EXTR_PREFIX_SAME	If there is a collision, prefix the variable name with *prefix*
EXTR_PREFIX_ALL	Prefix all variable names with *prefix* .
EXTR_PREFIX_INVALID	Only prefix invalid/numeric variable names with *prefix* .
EXTR_PREFIX_IF_EXISTS	Only create prefixed variable names if the non-prefixed version of the same variable exists in the current symbol table.
EXTR_IF_EXISTS	Only overwrite the variable if it already exists in the current symbol table, otherwise do nothing.
EXTR_REFS	Extracts variables as references.

Calendar Functions

Function	Description
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 given calendar
cal_to_jd()	Converts a date to a 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, on which Easter falls for 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 a 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 a Julian day count to a UNIX timestamp
JewishToJD()	Converts a Jewish date to a Julian day count
JulianToJD()	Converts a Julian date to a Julian day count
unixtojd()	Converts a UNIX timestamp to a Julian day count

Date/Time Functions

Function	Description
checkdate()	Validates a Gregorian date
date_default_timezone_get()	Returns the default time zone
date_default_timezone_set()	Sets the default time zone
date_parse_from_format	Get info about given date formatted according to the specified format
date_parse	Returns associative array with detailed info about given date
date_sunrise()	Returns the time of sunrise for a given day and location
date_sunset()	Returns the time of sunset for a given day and location
date()	Formats a local time/date
getdate()	Returns an array that contains date and time information for a UNIX timestamp

gettimeofday()	Returns an array that contains current time information
gmdate()	Formats a GMT/UTC date/time
gmmktime()	Returns the UNIX timestamp for a GMT date
gmstrftime()	Formats a GMT/UTC time/date according to local settings
idate()	Formats a local time/date as integer
localtime()	Returns an array that contains the time components of a UNIX timestamp
microtime()	Returns the microseconds for the current time
mktime()	Returns the UNIX timestamp for a date
strftime()	Formats a local time/date according to local settings
strptime()	Parses a time/date generated with strftime()
strtotime()	Parses an English textual date or time into a UNIX timestamp
time()	Returns the current time as a UNIX timestamp
timezone_name_from_abbr	Returns the timezone name from abbreviation
timezone_version_get	Gets the version of the timezonedb

Directory Functions

Function	Description
chdir()	Changes the current directory
chroot()	Changes the root directory of the current process
dir()	Opens a directory handle and returns an object
closedir()	Closes a directory handle
getcwd()	Returns the current directory
opendir()	Opens a directory handle
readdir()	Returns an entry from a directory handle
rewinddir()	Resets a directory handle
scandir()	Lists files and directories inside a specified path

Error and Logging Functions

Function	Description
debug_backtrace()	Generates a backtrace
debug_print_backtrace()	Prints a backtrace
error_clear_last()	Clear the most recent error
error_get_last()	Gets the last error that occurred
error_log()	Sends an error to the server error log, to a file, or to a remote destination
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 function to handle errors
set_exception_handler()	Sets a user-defined function to handle exceptions
trigger_error()	Creates a user-defined error message
user_error()	Alias of trigger_error()

Error Function Parameters

Parameter	Description
error_level	Required; specifies the error report level for the user-defined error; corresponds to logging constant numbers
error_message	Required; specifies the error message for the user-defined function
error_file	Optional; specifies the filename in which the error occurred
error_line	Optional; specifies the line number in which the error occurred
error_context	Optional; specifies an array containing all variables and their values in use when the error occurred

Error and Logging Constants

Error Value Constant	Description
1 E_ERROR	Fatal run-time errors; errors that cannot be recovered from; execution of the script halted
2 E_WARNING	Non-fatal run-time errors; execution of the script not halted
4 E_PARSE	Compile-time parse errors; should only be generated by the parser
8 E_NOTICE	Run-time notices; found something that might be an error, but could also happen when running normally
16 E_CORE_ERROR	Fatal errors at PHP startup; like an E_ERROR in the PHP core
32 E_CORE_WARNING	Non-fatal errors at PHP startup; like an E_WARNING in the PHP core
64 E_COMPILE_ERROR	Fatal compile-time errors; like an E_ERROR generated by the Zend Scripting Engine
128 E_COMPILE_WARNING	Non-fatal compile-time errors; like an E_WARNING generated by the Zend Scripting Engine
256 E_USER_ERROR	Fatal user-generated error; like an E_ERROR set by the programmer using the PHP function trigger_error()
512 E_USER_WARNING	Non-fatal user-generated warning; like an E_WARNING set by the programmer using the PHP function trigger_error()
1024 E_USER_NOTICE	User-generated notice; like an E_NOTICE set by the programmer using the PHP function trigger_error()
2048 E_STRICT	Run-time notices; PHP-suggested changes to your code to help interoperability and compatibility of the code
4096 E_RECOVERABLE_ERROR	Catchable fatal error; like an E_ERROR but can be caught by a user-defined handle; see also set_error_handler()
8192 E_DEPRECATED	Run-time notices. Enable this to receive warnings about code that will not work in future

	versions.
16384 E_USER_DEPRECATED	User-generated warning message. This is like E_DEPRECATED, except it is generated in PHP code by using the PHP function trigger_error().
32767 E_ALL	All errors and warnings, except of level E_STRICT

File System Functions

Function	Description
basename()	Returns the filename component of a path
chgrp()	Changes the file group
chmod()	Changes the file mode
chown()	Changes the file owner
clearstatcache()	Clears the file status cache
copy()	Copies a file
delete()	See unlink() or unset()
dirname()	Returns the directory name component of a path
disk_free_space()	Returns the free space of a directory
disk_total_space()	Returns the total size of a directory
diskfreespace()	Alias of disk_free_space()
fclose()	Closes an open file
feof()	Tests for end-of-file on an open file
fflush()	Flushes buffered output to an open file
fgetc()	Returns a character from an open file
fgetcsv()	Parses a line from an open file, checking for CSV fields
fgets()	Returns a line from an open file
fgetss()	Returns a line, with HTML and PHP tags removed, from an open file
file()	Reads a file into an array

file_exists()	Checks whether or not a file or directory exists
file_get_contents()	Reads a file into a string
file_put_contents	Writes a string to a file
fileatime()	Returns the last access time of a file
filectime()	Returns the last change time of a file
filegroup()	Returns the group ID of a file
fileinode()	Returns the inode number of a file
filemtime()	Returns the last modification time of a file
fileowner()	Returns the user ID (owner) of a file
fileperms()	Returns the permissions of a file
filesize()	Returns the file size
filetype()	Returns the file type
flock()	Locks or releases a file
fnmatch()	Matches a filename or string against a specified pattern
fopen()	Opens a file or URL
fpassthru()	Reads from an open file until EOF, and writes the result to the output buffer
fputcsv()	Formats a line as CSV and writes it to an open file
fputs()	Alias of fwrite()
fread()	Reads from an open file
fscanf()	Parses input from an open file according to a specified format
fseek()	Seeks in an open file
fstat()	Returns information about an open file
ftell()	Returns the current position in an open file
ftruncate()	Truncates an open file to a specified length
fwrite()	Writes to an open file
glob()	Returns an array of filenames/directories matching a specified pattern
is_dir()	Checks whether a file is a directory
is_executable()	Checks whether a file is executable

is_file()	Checks whether a file is a regular file
is_link()	Checks whether a file is a link
is_readable()	Checks whether a file is readable
is_uploaded_file()	Checks whether a file was uploaded via HTTP POST
is_writable()	Checks whether a file is writeable
is_writeable()	Alias of is_writable()
lchgrp	Changes group ownership symlink
lchown	Changes user ownership of symlink
link()	Creates a hard link
linkinfo()	Returns information about a hard link
lstat()	Returns information about a file or symbolic link
mkdir()	Creates a directory
move_uploaded_file()	Moves an uploaded file to a new location
parse_ini_file()	Parses a configuration file
parse_ini_string	Parse a configuration string
pathinfo()	Returns information about a file path
pclose()	Closes a pipe opened by popen()
popen()	Opens a pipe
readfile()	Reads a file and writes it to the output buffer
readlink()	Returns the target of a symbolic link
realpath()	Returns the absolute pathname
realpath_cache_get	Get realpath cache entries
rrealpath_cache_size	Get realpath cache size
rename()	Renames a file or directory
rewind()	Rewinds a file pointer
rmdir()	Removes an empty directory
set_file_buffer()	Sets the buffer size of an open file
stat()	Returns information about a file
symlink()	Creates a symbolic link

tempnam()	Creates a unique temporary file
tmpfile()	Creates a unique temporary file
touch()	Sets access and modification time of a file
umask()	Changes file permissions for files
unlink()	Deletes a file

File System Constants

Constant	Description
FILE_USE_INCLUDE_PATH	Search for the file in the include path
FILE_APPEND	Append file
FILE_BINARY	Binary mode
FILE_IGNORE_NEW_LINES	Do not add newline at the end of each array element
FILE_SKIP_EMPTY_LINES	Skip empty lines
GLOB_BRACE	Expands {a,b,c} to match 'a', 'b', or 'c'
GLOB_ONLYDIR	Return only directory entries which match the pattern
GLOB_MARK	Adds a slash to each item returned
GLOB_NOSORT	Return files as they appear in the directory (no sorting)
GLOB_NOCHECK	Return the search pattern if no files matching it were found
GLOB_NOESCAPE	Backslashes do not quote metacharacters
LOCK_SH	To acquire a shared lock (reader)
LOCK_EX	To acquire an exclusive lock (writer)
LOCK_UN	To release a lock (shared or exclusive)
LOCK_NB	If you don’t want flock() to block while locking. Not support on windows.
PATHINFO_DIRNAME	The path being checked returns directory name
PATHINFO_BASENAME	The path being checked returns basename
PATHINFO_EXTENSION	The path being checked returns extension
PATHINFO_FILENAME	The path being checked returns filename

SEEK_SET	Set position equal to offset (deault)
SEEK_CUR	Set position to current location plus offset
SEEK_END	Set position to EOF plus offset

Filter Functions

Function	Description
filter_has_var()	Checks if a variable of a specified input type exist
filter_id()	Returns the ID number of a specified filter
filter_input()	Gets input from outside the script and filters it
filter_input_array()	Gets multiple inputs from outside the script and filters them
filter_list()	Returns an array of all supported filters
filter_var()	Gets a variable and filters it
filter_var_array()	Gets multiple variables and filters them

Filters

<table>
<tr><th>Filter</th><th>Description</th></tr>
<tr><td>FILTER_CALLBACK</td><td>Calls a user-defined function to filter data</td></tr>
<tr><td>FILTER_SANITIZE_STRING</td><td>Strips tags, optionally strips or encodes special characters</td></tr>
<tr><td>FILTER_SANITIZE_STRIPPED</td><td>Alias of "string" filter</td></tr>
<tr><td>FILTER_SANITIZE_ENCODED</td><td>URL-encodes string, optionally strips or encodes special characters</td></tr>
<tr><td>FILTER_SANITIZE_SPECIAL_CHARS</td><td>HTML-escapes '"<>& and characters with ASCII values less than 32</td></tr>
<tr><td>FILTER_SANITIZE_EMAIL</td><td>Removes all characters, except letters, digits, and the following special characters: !#$%&'*+-/=?^_`{|}~@.[]</td></tr>
<tr><td>FILTER_SANITIZE_URL</td><td>Removes all characters except letters, digits, and the following special characters: $-_.+!*'(),{}|\\^~[]`<>#%";/?:@&=</td></tr>
</table>

FILTER_SANITIZE_NUMBER_INT	Removes all characters except digits and +-
FILTER_SANITIZE_NUMBER_FLOAT	Removes all characters except digits, +-, and optionally eE
FILTER_SANITIZE_MAGIC_QUOTES	Applies addslashes()
FILTER_UNSAFE_RAW	Does nothing, optionally strips or encodes special characters
FILTER_VALIDATE_INT	Validates a value as an integer, optionally from the specified range
FILTER_VALIDATE_BOOLEAN	Returns TRUE for "1," "true," "on," or "yes"; FALSE for "0," "false," "off," "no," or ""; NULL otherwise
FILTER_VALIDATE_FLOAT	Validates a value as float
FILTER_VALIDATE_REGEXP	Validates a value against regexp, a Perl-compatible regular expression
FILTER_VALIDATE_URL	Validates a value as a URL, optionally with required components
FILTER_VALIDATE_EMAIL	Validates a value as email
FILTER_VALIDATE_IP	Validates a value as an IP address, optionally only IPv4 or IPv6 or not from private or reserved ranges

FTP Functions

Function	**Description**
ftp_alloc()	Allocates space for a file to be uploaded to the FTP server
ftp_cdup()	Changes the current directory 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 program/command on the FTP server
ftp_fget()	Downloads a file from the FTP server and saves it to an open file

ftp_fput()	Uploads from an open file and saves it to a file on the FTP server
ftp_get_option()	Returns runtime behaviors of the FTP connection
ftp_get()	Downloads a file from the FTP server
ftp_login()	Logs on to an FTP connection
ftp_mdtm()	Returns the last modified time of a specified file
ftp_mkdir()	Creates a new directory on the FTP server
ftp_nb_continue()	Continues retrieving/sending a file (non-blocking)
ftp_nb_fget()	Downloads a file from the FTP server and saves it to 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()	Lists the files in a 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 detailed list of files in the specified directory
ftp_rename()	Renames a file or directory on the FTP server
ftp_rmdir()	Removes a directory on the FTP server
ftp_set_option()	Sets runtime options for the FTP connection
ftp_site()	Sends a SITE command to the 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

FTP Constants

Constant	Description
FTP_ASCII	FTP transfer mode ASCII
FTP_TEXT	FTP transfer mode Text
FTP_BINARY	FTP transfer mode binary
FTP_IMAGE	FTP transfer mode image
FTP_TIMEOUT_SEC	Set FTP timeout
FTP_AUTOSEEK	When enabled, GET or PUT requests with a resumepos or startpos parameter will first seek to the requested position within the file. FTP_AUTOSEEK is enabled by deault.
FTP_AUTORESUME	Determines resume position and start position for get and put requests automatically
FTP_FAILED	Indicates asynchronous transfer has failed
FTP_FINISHED	Indicates asynchronous transfer has finished
FTP_MOREDATA	Indicates asynchronous transfer is still active

HTTP Functions

Function	Description
header()	Sends a raw HTTP header to a client
headers_list()	Returns a list of response headers sent (or ready to send)
headers_sent()	Checks if/where the HTTP headers have been sent
setcookie()	Sends an HTTP cookie to a client
setrawcookie()	Sends an HTTP cookie without URL-encoding the cookie value

Mail Functions

Function	Description
ezmlm_hash()	Calculates the hash value needed by the EZMLM mailing list system
mail()	Allows you to send emails directly from a script

Mail Parameters

Parameter	Description
To	Required; specifies the receiver(s) of the email
Subject	Required; specifies the subject of the email; cannot contain any newline characters
Message	Required; defines the message to be sent; each line should be separated with a LF (\n); lines should not exceed 70 characters
Headers	Optional; specifies additional headers, like From, Cc, and Bcc; additional headers should be separated with a CRLF (\r\n)
Parameters	Optional; specifies an additional parameter to the sendmail program

Math Functions

Function	Description
abs()	Returns the absolute value of a number
acos()	Returns the arccosine of a number
acosh()	Returns the inverse hyperbolic cosine of a number
asin()	Returns the arcsine of a number
asinh()	Returns the inverse hyperbolic sine of a number
atan()	Returns the arctangent of a number as a numeric value between -PI/2 and PI/2 radians
atan2()	Returns the angle theta of an (x,y) point as a numeric value between -PI and PI radians
atanh()	Returns the inverse hyperbolic tangent of a number
base_convert()	Converts a number from one base to another
bindec()	Converts a binary number to a decimal number
ceil()	Returns the value of a number rounded upwards 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 to a radian number
exp()	Returns the value of Ex
expm1()	Returns the value of Ex – 1
floor()	Returns the value of a number rounded downwards to the nearest integer
fmod()	Returns the remainder (modulo) of the division of the arguments
getrandmax()	Returns the maximum random number that can be returned by a call to the rand() function
hexdec()	Converts a hexadecimal number to a decimal number
hypot()	Returns the length of the hypotenuse of a right-angle triangle
Intdiv	Integer division
is_finite()	Returns true if a value is a finite number
is_infinite()	Returns true if a value is an infinite number
is_nan()	Returns true if a value is not a number
lcg_value()	Returns a pseudo-random number in the range of (0,1)
log()	Returns the natural logarithm (base-E) of a number
log10()	Returns the base-10 logarithm of a number
log1p()	Returns log (1+number)
max()	Returns the number with the highest value of two specified numbers
min()	Returns the number with the lowest value of two specified numbers
mt_getrandmax()	Returns the largest possible value that can be returned by mt_rand()
mt_rand()	Returns 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 the value of x to the power of y
rad2deg()	Converts a radian number to a degree
rand()	Returns a random integer
round()	Rounds a number to the nearest integer
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 an angle
tanh()	Returns the hyperbolic tangent of an angle

Math Constants

Constant	Description
M_E	Returns e (approx. 2.718)
M_EULER	Returns Euler's constant (approx. 0.577)
M_LNPI	Returns the natural logarithm of PI (approx. 1.144)
M_LN2	Returns the natural logarithm of 2 (approx. 0.693)
M_LN10	Returns the natural logarithm of 10 (approx. 2.302)
M_LOG2E	Returns the base-2 logarithm of E (approx. 1.442)
M_LOG10E	Returns the base-10 logarithm of E (approx. 0.434)
M_PI	Returns PI (approx. 3.14159)
M_PI_2	Returns PI/2 (approx. 1.570)
M_PI_4	Returns PI/4 (approx. 0.785)
M_1_PI	Returns 1/PI (approx. 0.318)
M_2_PI	Returns 2/PI (approx. 0.636)
M_SQRTPI	Returns the square root of PI (approx. 1.772)
M_2_SQRTPI	Returns 2/square root of PI (approx. 1.128)
M_SQRT1_2	Returns the square root of 1/2 (approx. 0.707)
M_SQRT2	Returns the square root of 2 (approx. 1.414)
M_SQRT3	Returns the square root of 3 (approx. 1.732)

Miscellaneous Functions

Function	Description
connection_aborted()	Checks whether the client has disconnected
connection_status()	Returns the current connection status
constant()	Returns the value of a constant
define()	Defines a constant
defined()	Checks whether a constant exists
die()	Prints a message and exits the current script
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
ignore_user_abort()	Sets whether a remote client can abort the running of a script
pack()	Packs data into a binary string
php_check_syntax()	Deprecated in PHP 5.0.5
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	Gets system load average
time_nanosleep()	Delays code execution for a number of seconds and nanoseconds
time_sleep_until()	Delays code execution until a specified time
uniqid()	Generates a unique ID
unpack()	Unpacks data from a binary string
usleep()	Delays code execution for a number of microseconds

MySQL Functions

Function	Description
mysql_affected_rows()	Returns the number of affected rows in the previous MySQL operation
mysql_client_encoding()	Returns the name of the character set for the current connection
mysql_close()	Closes a non-persistent MySQL connection
mysql_connect()	Opens a non-persistent MySQL connection
mysql_create_db()	Deprecated; creates a new MySQL database; use mysql_query() instead
mysql_data_seek()	Moves the record pointer
mysql_db_name()	Returns a database name from a call to mysql_list_dbs()
mysql_db_query()	Deprecated; sends a MySQL query; use mysql_select_db() and mysql_query() instead
mysql_drop_db()	Deprecated; deletes a MySQL database; use mysql_query() instead
mysql_errno()	Returns the error number of the last MySQL operation
mysql_error()	Returns the error description of the last MySQL operation
mysql_escape_string()	Deprecated; escapes a string for use in a mysql_query; use mysql_real_escape_string() instead
mysql_fetch_array()	Returns a row from a recordset as an associative array and/or a numeric array
mysql_fetch_assoc()	Returns a row from a recordset as an associative array
mysql_fetch_field()	Returns column information from a recordset as an object
mysql_fetch_lengths()	Returns the length of the contents of each field in a result row
mysql_fetch_object()	Returns a row from a recordset as an object
mysql_fetch_row()	Returns a row from a recordset as a numeric array
mysql_field_flags()	Returns the flags associated with a field in a recordset
mysql_field_len()	Returns the maximum length of a field in a recordset
mysql_field_name()	Returns the name of a field in a recordset
mysql_field_seek()	Moves the result pointer to a specified field

mysql_field_table()	Returns the name of the table the specified field is in
mysql_field_type()	Returns the type of a field in a recordset
mysql_free_result()	Frees result memory
mysql_get_client_info()	Returns MySQL client info
mysql_get_host_info()	Returns MySQL host info
mysql_get_proto_info()	Returns MySQL protocol info
mysql_get_server_info()	Returns MySQL server info
mysql_info()	Returns information about the last query
mysql_insert_id()	Returns the AUTO_INCREMENT ID generated from the previous INSERT operation
mysql_list_dbs()	Lists available databases on a MySQL server
mysql_list_fields()	Deprecated; lists MySQL table fields; use mysql_query() instead
mysql_list_processes()	Lists MySQL processes
mysql_list_tables()	Deprecated. Lists tables in a MySQL database. Use mysql_query() instead
mysql_num_fields()	Returns the number of fields in a recordset
mysql_num_rows()	Returns the number of rows in a recordset
mysql_pconnect()	Opens a persistent MySQL connection
mysql_ping()	Pings a server connection or reconnects if there is no connection
mysql_query()	Executes a query on a MySQL database
mysql_real_escape_string()	Escapes a string for use in SQL statements
mysql_result()	Returns the value of a field in a recordset
mysql_select_db()	Sets the active MySQL database
Mysql_set_charset	Sets the client character set
mysql_stat()	Returns the current system status of the MySQL server
mysql_tablename()	Deprecated; returns the table name of a field; use mysql_query() instead
mysql_thread_id()	Returns the current thread ID
mysql_unbuffered_query()	Executes a query on a MySQL database (without

	fetching/buffering the result)

MySQL Constants

Constant	Description
MYSQL_CLIENT_COMPRESS	Uses compression protocol
MYSQL_CLIENT_IGNORE_SPACE	Allows space after function names
MYSQL_CLIENT_INTERACTIVE	Allows interactive timeout seconds of inactivity before closing the connection
MYSQL_CLIENT_SSL	Uses SSL encryption (only available with version 4+ of the MySQL client library)
MYSQL_ASSOC	Returns columns into the array with the fieldname as the array index
MYSQL_BOTH	Returns columns into the array having both a numerical index and the fieldname as the array index
MYSQL_NUM	Returns columns into the array having a numerical index (starting at zero)

SimpleXML Functions

Function	Description
__construct()	Creates a new SimpleXMLElement object
addAttribute()	Adds an attribute to the SimpleXML element
addChild()	Adds a child element the SimpleXML element
asXML()	Gets an XML string from a SimpleXML element
attributes()	Gets a SimpleXML element's attributes
children()	Gets the children of a specified node
count()	Counts the children of a specified document
getDocNamespaces()	Gets the namespaces of an XML document
getName()	Gets the name of a SimpleXML element
getNamespaces()	Gets the namespaces from XML data
registerXPathNamespace()	Creates a namespace context for the next XPath query

saveXML()	Alias of asXML()
simplexml_import_dom()	Gets a SimpleXMLElement object from a DOM node
simplexml_load_file()	Gets a SimpleXMLElement object from an XML document
simplexml_load_string()	Gets a SimpleXMLElement object from an XML string
xpath()	Runs an XPath query on XML data

String Functions

Function	Description
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()	Alias of rtrim()
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 how many times an ASCII character occurs within a string and returns the information
crc32()	Calculates a 32-bit CRC for a string
crypt()	One-way string encryption (hashing)
echo()	Outputs 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

	  tags
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()
levenshtein()	Returns the Levenshtein distance between two strings
localeconv()	Returns local numeric and monetary formatting information
ltrim()	Strips whitespace 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 a string
printf()	Outputs a formatted string
quoted_printable_decode()	Decodes a quoted-printable string
quoted_printable_encode()	Convert 8 bit string to quoted printable string
quotemeta()	Quotes meta characters
rtrim()	Strips whitespace 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	Parse 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()	Counts 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()	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 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 part of a string with another string
trim()	Strips whitespace 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

String Constants

Constant	Description
CRYPT_SALT_LENGTH	Contains the length of the default encryption method for the system; for standard DES encryption, the length is two
CRYPT_STD_DES	Set to one if the standard DES-based encryption with a two-character salt is supported; zero otherwise
CRYPT_EXT_DES	Set to one if the extended DES-based encryption with a nine-character salt is supported; zero otherwise
CRYPT_MD5	Set to one if the MD5 encryption with a 12-character salt starting with 1 is supported; zero otherwise
CRYPT_BLOWFISH	Set to one if the Blowfish encryption with a 16-character salt starting with 2 or $2a$ is supported; zero otherwise

XML Parser Functions

Function	Description
utf8_decode()	Decodes a UTF-8 string to ISO-8859-1
utf8_encode()	Encodes an ISO-8859-1 string to UTF-8
xml_error_string()	Gets an error string from the XML parser
xml_get_current_byte_index()	Gets the current byte index from the XML parser
xml_get_current_column_number()	Gets the current column number from the XML parser
xml_get_current_line_number()	Gets the current line number from the XML parser
xml_get_error_code()	Gets 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()	Gets options from an XML parser
xml_parser_set_option()	Sets options in an parser

xml_set_character_data_handler()	Sets handler function for character data
xml_set_default_handler()	Sets default handler function
xml_set_element_handler()	Sets handler function for the start and end element of elements
xml_set_end_namespace_decl_handler()	Sets handler function for the end-of-namespace declarations
xml_set_external_entity_ref_handler()	Sets handler function for external entities
xml_set_notation_decl_handler()	Sets handler function for notation declarations
xml_set_object()	Uses the XML parser within an object
xml_set_processing_instruction_handler()	Sets the handler function for processing instructions
xml_set_start_namespace_decl_handler()	Sets the handler function for the start-of-namespace declarations
xml_set_unparsed_entity_decl_handler()	Sets the handler function for unparsed entity declarations

Appendix D: JDBC Data Types

Type	Recommended Method	Allowed Methods
BIGINT	getLong	getByte, getShort, getInt, getFloat, getDouble, getBigDecimal, getBoolean, getString, getObject
BINARY	getBytes	getString, getAsciiStream, getUnicodeStream, getBinaryStream, getObject
BIT	getBoolean	getByte, getShort, getInt, getLong, getFloat, getDouble, getBigDecimal, getString, getObject
CHAR	getStgring	getByte, getShort, getInt, getLong, getFloat, getDouble, getBigDecimal, getBoolean, getDate, getTime, getTimestamp, getAsciiStream, getUnicodeStream, getObject
DATE	getDate	getString, getTimestamp, getObject
DECIMAL	getBigDecimal	getByte, getShort, getInt, getLong, getFloat, getDouble, getBoolean, getString, getObject
DOUBLE	getDouble	getByte, getShort, getInt, getLong, getFloat, getBigDecimal, getBoolean, getString, getObject
FLOAT	getDouble	getByte, getShort, getInt, getLong, getFloat, getBigDecimal, getBoolean, getString, getObject
INTEGER	getInt	getByte, getShort, getLong, getFloat, getDouble, getBigDecimal, getBoolean, getString, getObject
LONGVARBINARY	getBinaryStream	getString, getBytes, getAsciiStream, getUnicodeStream, getObject

LONGVARCHAR	getAsciiStream	getByte, getShort, getInt, getLong, getFloat, getDouble, getBigDecimal, getBoolean, getString, getDate, getTime, getTimestamp, getObject
NUMERIC	getBigDecimal	getByte, getShort, getInt, getLong, getFloat, getDouble, getBoolean, getString, getObject
REAL	getFloat	getByte, getShort, getInt, getLong, getDouble, getBigDecimal, getBoolean, getString, getObject
SMALLINT	getShort	getByte, getInt, getLong, getFloat, getDouble, getBigDecimal, getBoolean, getString, getObject
TIME	getTime	getString, getTimestamp, getObject
TIMESTAMP	getTimestamp	getString, getDate, getTime, getObject
TINYINT	getByte	getShort, getInt, getLong, getFloat, getDouble, getBigDecimal, getBoolean, getString, getObject
VARCHAR	getString	getByte, getShort, getInt, getLong, getFloat, getDouble, getBigDecimal, getBoolean, getDate, getTime, getTimestamp, getAsciiStream, getUnicodeStream, getObject
VARBINARY	getBytes	getString, getAsciiStream, getUnicodeStream, getBinaryStream, getObject

Appendix E: HTML Colors

In the following table of HTML colors, an asterisk (*) after a color name indicates a standard Windows color. A name in bold indicates an HTML 4.0 named color. Browser-safe colors are those with each pair of hex codes in the RGB value selected from the following: 00, 33, 66, 99, CC, or FF.

Named Color	Hexadecimal Code
Aliceblue	#F0F8FF
Antiquewhite	#FAEBD7
Aqua*	#00FFFF
Aquamarine	#7FFFD4
Azure	#F0FFFF
Beige	#F5F5DC
Bisque	#FFE4C4
Black*	#000000
Blanchedalmond	#FFEBCD
Blue*	#0000FF
Blueviolet	#8A2BE2
Brown	#A52A2A
Burlywood	#DEB887
Cadetblue	#5F9EA0
Chartruse	#7FFF00
Chocolate	#D2691E
Coral	#FF7F50
cornflowerblue	#6495ED

Cornsilk	#FFF8DC
Crimson	#DC143C
Cyan (same as aqua)	#00FFFF
Darkblue	#00008B
Darkcyan	#008B8B
Darkgoldenrod	#B8860B
Darkgray	#A9A9A9
Darkgreen	#006400
Darkkhaki	#BDB76B
Darkmagenta	#BD008B
Darkolivegreen	#556B2F
Darkorange	#FF8C00
Darkorchid	#9932CC
Darkred	#8B0000
Darksalmon	#E9967A
Darkseagreen	#8FBC8F
Darkslateblue	#483D8B
Darkslategray	#2F4F4F
Darkturquoise	#00CED1
Darkviolet	#9400D3
Deeppink	#FF1493
Deepskyblue	#00BFFF
Dimgray	#696969
Dogerblue	#1E90FF
Firebrick	#B22222
Floralwhite	#FFFAF0
Forestgreen	#228B22

Fuchsia*	#FF00FF
Gainsboro	#DCDCDC
Ghostwhite	#F8F8FF
Gold	#FFD700
Goldenrod	#DAA520
Gray*	#808080
Green*	#008000
Greenyellow	#ADFF2F
honeydew	#F0FFF0
Hotpink	#FF69B4
Indianred	#CD5C5C
Indigo	#480082
Ivory	#FFFFF0
Khaki	#F0E68C
Lavender	#E6E6FA
Lavenderblush	#FFF0F5
Lawngreen	#7CFC00
Lemonchiffon	#FFFACD
Lightblue	#ADD8E6
Lightcoral	#F08080
Lightcyan	#E0FFFF
Lightgoldenrodyellow	#FAFAD2
Lightgreen	#90EE90
Lightgray	#D3D3D3
Lightpink	#FFB6C1
lightsalmon	#FFA07A
Lightseagreen	#20B2AA

Lightskyblue	#87CEFA
Lightslategray	#778899
Lightsteelblue	#B0C4DE
Lightyellow	#FFFFE0
Lime*	#00FF00
Limegreen	#32CD32
Linen	#FAF0E6
Magenta (same as fuchsia)	#FF00FF
Maroon*	#800000
Mediumaquamarine	#66CDAA
Mediumblue	#0000CD
Mediumorchid	#BA55D3
Mediumpurple	#9370DB
Mediumseagreen	#3CB371
Mediumslateblue	#7B68EE
Mediumspringgreen	#00FA9A
Mediumturquoise	#48D1CC
Mediumvioletred	#C71585
Midnightblue	#191970
Mintcream	#F5FFFA
Mistyrose	#FFE4E1
Moccasin	#FFE4B5
Navajowhite	#FFDEAD
Navy*	#000080
Oldlace	#FDF5E6
Olive*	#808000
Olivedrab	#6B8E23

Orange	#FFA500
Orangered	#FF4500
Orchid	#DA70D6
Palegoldenrod	#EEE8AA
Palegreen	#98FB98
Paleturquoise	#AFEEEE
palevioletred	#DB87093
Papayawhip	#FFEFD5
Peachpuff	#FFDAB9
Peru	#CD853F
Pink	#FFC0CB
Plum	#DDA0DD
Powderblue	#B0E0E6
Purple*	#800080
Red*	#FF0000
Rosybrown	#BC8F8F
Royalblue	#4169E1
Saddlebrown	#8B4513
Salmon	#FA8072
Seagreen	#2E8B57
Seashell	#FFF5EE
Sienna	#A0522D
Silver*	#C0C0C0
Skyblue	#87CEEB
Slateblue	#6A5ACD
Slategray	#708090
Snow	#FFFAFA

Springgreen	#00FF7F
Steelblue	#4682B4
Tan	#D2B486
Teal*	#008080