

PHP

PHP Standard Functions

Function	Description
<code>isset (el)</code>	Will return false if <code>el</code> has been assigned the constant NULL, <code>el</code> has not been set to any value yet (undefined) <code>el</code> has been deleted using the <code>unset</code> function
<code>print str</code> <code>echo str</code>	Prints <code>str</code>
<code>time()</code>	Returns the current time in seconds
<code>date(format, time)</code>	Converts an optional <code>time</code> in seconds to a date based on <code>format</code>
<code>mt_rand(min, max)</code>	Returns a random integer between <code>min</code> and <code>max</code> (inclusive)
<code>header(string)</code>	Sends a raw HTTP header. Examples include: <code>header("HTTP/1.1 400 Invalid Request");</code> <code>header("Content-type: text/plain");</code> <code>header("Content-type: application/json");</code>
<code>die(message)</code>	Ends execution and sends back optional <code>message</code>
<code>include "path"</code>	Includes and evaluates the specified file <code>path</code> such as <code>"hidden/config.php"</code>

PHP Array Functions

Function	Description
<code>count(arr)</code>	Returns the length of an array <code>arr</code>
<code>print_r(arr)</code>	Prints the <code>arr</code> 's contents
<code>array_pop(arr)</code>	Pops (removes) an element off the end of the array <code>arr</code>
<code>array_shift(arr)</code>	Shifts (removes) an element off the beginning of the array <code>arr</code>
<code>array_push(arr, el)</code>	Pushes (adds) one or more elements onto the end of the array <code>arr</code>
<code>array_unshift(arr, el)</code>	Prepends one or more elements to the beginning of the array <code>arr</code>
<code>sort(arr)</code>	Sorts the array <code>arr</code>
<code>array_reverse(arr)</code>	Returns an array with elements of <code>arr</code> in reverse order
<code>in_array(el, arr)</code>	Returns whether a value <code>el</code> exists in an array <code>arr</code>
<code>list(a, b, ...)</code>	Assigns variables as if they were an array
<code>implode(glue, pieces)</code>	Joins array elements (<code>pieces</code>) with a string (<code>glue</code>)
<code>array_rand(arr)</code>	Randomly selects a random entry from the array and returns the key (or keys) of the random entries.

PHP JSON Functions

Function	Description
<code>json_encode(obj)</code>	Returns JSON equivalent for the given object/array/value
<code>json_decode(string)</code>	Parse the given JSON data string and returns an equivalent associative array object

PHP String Functions

Function	Description
<code>strlen(s)</code>	Returns the length of a string <code>s</code>
<code>strpos(str, substr)</code>	Returns the position of the first occurrence of <code>substr</code> in <code>str</code> , or <code>FALSE</code> if not found
<code>substr(s, start, len)</code>	Returns a substring of <code>s</code> starting at <code>start</code> and up to <code>len</code> characters in length. If <code>s</code> is less than <code>start</code> characters long, <code>FALSE</code> will be returned
<code>trim(s)</code>	Strips whitespace characters from both ends of a string <code>s</code>
<code>strtolower(s)</code>	Returns a lowercase version of <code>s</code>
<code>strtoupper(s)</code>	Returns an uppercase version of <code>s</code>
<code>explode(delimiter, s)</code>	Returns an array of substrings of <code>s</code> split by <code>delimiter</code>

PHP File Functions

Function	Description
<code>file(path, [int flags = 0])</code>	Reads entire file <code>path</code> into an array. Optional <code>flags</code> parameter can be passed in such as <code>FILE_IGNORE_NEW_LINES</code> or <code>FILE_SKIP_EMPTY_LINES</code>
<code>file_exists(path)</code>	Returns whether a file or directory <code>path</code> exists
<code>file_get_contents(path)</code>	Reads entire file <code>path</code> into a string
<code>file_put_contents(path, data)</code>	Writes a string <code>data</code> to a file <code>path</code>
<code>scandir(path)</code>	Returns an array of all files and directories inside the specified <code>path</code> including <code>.</code> and <code>..</code>
<code>glob(pattern)</code>	Returns an array of path names matching <code>pattern</code>
<code>basename(path)</code>	Given a filename <code>path</code> , this function will strip any leading directory from a file path and return just the filename

PHP Superglobals Reference

Variable	Description
<code>\$_GET</code>	Superglobal array which contains query parameters passed in via a <code>GET</code> request
<code>\$_POST</code>	Superglobal array which contains <code>POST</code> parameters passed in via a <code>POST</code> request

PHP PDO Functions (with `mysql`)

Note that for some PDO object `$db`, you can call some function `fxn` using `$db->fxn(...)`.

Function	Description
<code>new PDO('mysql:dbname=database;host=yourhost', username, password)</code>	Constructor, connecting to the database using the given <code>yourhost</code> host value, username, and password
<code>setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION)</code>	Sets PDO error-handling properties
<code>query(sqlquery)</code>	Returns a <code>PDOStatement</code> (that contains a result set) after executing <code>sqlquery</code> in the PDO's connected database
<code>exec(sqlquery)</code>	Executes a SQL statement. Returns the number of affected rows.
<code>prepare(statement)</code>	Prepares an SQL statement to be executed by the <code>execute(arr)</code> method. The SQL statement can contain zero or more named (<code>:name</code>) parameter markers for which real values will be substituted when the statement is executed.

PDOStatement Functions

A `PDOStatement` represents a prepared statement and, after the statement is executed, an associated result set. You can retrieve the rows using a `foreach` loops, `fetch()`, or `fetchAll()`. These functions are also used with `$stmt->fxn(...)` syntax.

Function	Description
<code>execute(arr)</code>	Executes the prepared statement, filling in the named or question mark parameters with real values from the associative array. Returns TRUE if database was changed as a result, otherwise FALSE.
<code>columnCount()</code>	Returns the number of columns in the result set.
<code>fetch()</code>	Returns the next row from the result set.
<code>fetchAll()</code>	Returns all of the rows in an array of arrays representing each row from the set.
<code>fetchColumn(number)</code>	Returns the next column from the result set.
<code>rowCount()</code>	Returns the number of rows in the result set.

PHP Regex Functions

Function	Description
<code>preg_match(regex, str)</code>	Returns whether <code>str</code> matches <code>regex</code>
<code>preg_replace(regex, repl, str)</code>	Returns a new string with all substrings of <code>str</code> that match <code>regex</code> replaced by <code>repl</code>
<code>preg_split(regex, str)</code>	Returns an array of strings from given <code>str</code> split apart using given <code>regex</code> as delimiter

Regex Reference

<code>[abc]</code>	A single character of: a, b, or c	<code>.</code>	Any single character	<code>(...)</code>	Capture everything enclosed
<code>[^abc]</code>	Any single character except: a, b, or c	<code>\s</code>	Any whitespace character	<code>(a b)</code>	a or b
<code>[a-z]</code>	Any single character in the range a-z	<code>\S</code>	Any non-whitespace character	<code>a?</code>	Zero or one of a
<code>[a-zA-Z]</code>	Any single character in the range a-z or A-Z	<code>\d</code>	Any digit	<code>a*</code>	Zero or more of a
<code>^</code>	Start of line	<code>\D</code>	Any non-digit	<code>a+</code>	One or more of a
<code>\$</code>	End of line	<code>\w</code>	Any word character (letter, number, underscore)	<code>a{3}</code>	Exactly 3 of a
<code>\A</code>	Start of string	<code>\W</code>	Any non-word character	<code>a{3,}</code>	3 or more of a
<code>\Z</code>	End of string	<code>\b</code>	Any word boundary	<code>a{3,6}</code>	Between 3 and 6 of a
options: <code>i</code> case insensitive <code>m</code> make dot match newlines <code>x</code> ignore whitespace in regex <code>o</code> perform <code>#{...}</code> substitutions only once					

Special characters that need to be escaped to match as literals: `[] ^ $. | ? * + () { } \`