

### connection to mysql database

```
Connect to the database
$dbc = mysqli_connect(HOST, USER,
PW, DB);

Query the database
$result = mysqli_query($dbc, $query);

Close the connection
mysqli_close($dbc);
```

### PHP functions

isset()	test for variable exists
empty()	test for empty variable
mail(\$to, \$subject, \$msg, 'From: ' . \$email)	mail function
mysqli_fetch_array(\$result)	fetch each row of a query (in \$result)
header()	send a header from the server
is_numeric()	test to see if a value is number
exit()	causes script to stop immediately
trim(\$string)	trims leading and trailing spaces
mysqli_real_escape_string(\$string)	escapes special characters
str_replace('a', 'b', \$string)	replace a with b in a string
explode(' ', \$string)	make string into array
implode(' ', \$string)	make array into string
substr(\$string, start, len)	grabs a substring
preg_match('regex', \$string)	matches regular expressions
preg_replace('regex', \$replace, \$string)	replaces characters in a string by regex

### IF syntax

```
if (condition) {
... }
elseif (condition) {
... }
else {
... }
```

### Loops

```
FOR loop    for (initialize; condition;
update) { ... }

WHILE loop  while (condition) { ... }

FOREACH     foreach ($array as $value) {
loop        ... }

DO WHILE    do { ... ;} while (condition)

FOR (loop until a condition is met)
WHILE (loop through query results)
FOREACH (loop through an array)
```

### CONTINUE

```
for ($i = 0; $i < 5; ++$i) {
if ($i == 2)
continue;
print "$i , ";
}
```

produces the following output:  
0 , 1 , 3 , 4

### SWITCH syntax

```
SWITCH ($s) {
case 1:
...
break;
case 2:
...
break;
default:
...
}
```

can be used with numbers, or strings

### Regular Expressions

^	start of string
\d	number from 0 to 9
\s	whitespace
.	any letter or digit or whitespace
\w	any alphanumeric [0-9A-Za-z]
\$	end of string
()	group
[]	character class
{x}{x,}	x of   x or more of   x to y of {x,y}
	or
*	none or more
?	none or one
+	one or more
\	escape

### Arrays

Create	\$myArray = array();
Push into	\$myArray[] = "Something";
Push to associative	\$myArray["key"] = "Value";
Create numeric	\$myArray = array('value', 'value2');
Create associative	\$a = array('key'=>'val');
Print from numeric	echo \$myArray[0];
Print from associative	echo \$myArray['key'];
Associative arrays	Keys are strings
Numeric arrays	Keys are numbers: 0,1,2,3,4

