

Quick Reference Guide to PHP

Beginner's Code Tips

- PHP code starts `<?php`
- PHP code ends `?>`

- Always put a semi-colon at the end of a statement, as in:
`Print "Hello world!";`

- Always comment your code; you'll forget what you meant by it if you don't.
`//Purpose: to create a form for patrons' questions`
`#Purpose: to create a form for patrons' questions`
`/* Purpose: to create a form for patrons' questions`
`Author: Lisa Bartle */`

- Use `=` to assign value and `==` to compare values. In other words, Always use `==` after If and Elseif statements.

Connect to Servers and Databases

Connect to the database server at an address with your username and password using this code:

```
$connection=mysql_connect("hostaddress", "myusername",  
"mypassword") or die ("message");
```

Connect to a particular databasewith the database name.

```
$result=mysql_select_db($database) or die ("Database could  
not be selected");
```

Close the connection to the server.

```
mysql_close($connection);
```

Alter Input from User

<code>\$keyword=trim(\$keyword);</code>	Removes white space at front and back of input
<code>\$keyword=strip_tags(\$keyword);</code>	Removes HTML tages embedded in input
<code>\$keyword=strtolower(\$keyword);</code>	Converts all input to lower case
<code>\$anyname=str_replace(" and ", " +", \$anyname);</code>	Replaces the Boolean AND with a + [plus sign] in variable \$anyname
<code>\$anyname=str_replace(" or ", " " ", \$anyname);</code>	Replaces the Boolean OR with a blank space in variable \$anyname
<code>\$anyname=str_replace(" not ", " -", \$anyname);</code>	Replaces the Boolean NOT with a - [minus sign] in variable \$anyname
<code>\$anyname=str_replace(" +not ", " " -", \$anyname);</code>	Replaces the +NOT with a -[minus sign] in variable \$anyname

Creating the SQL Query

An SQL query looks like this:

```
SELECT * from books where title LIKE "%Great%" ORDER BY
author ASC, title ASC;
```

Using a PHP variable, the query looks like those below.

This first uses a keyword in the query, while the second uses a variable in the query.

- `$query="SELECT * from books where title LIKE \"%Great%\" ORDER BY author ASC, title ASC";`
- `$query="SELECT * from " . $tablename. " where title LIKE \"% " . $keyword. "%\" ORDER BY author ASC, title ASC";`

Patterns for punctuation in an SQL query using PHP look like these:

- `[like] \"%\" . $variable . \"%\" (for text)`
- `[=, <>] \"%\" . $variable . \"%\" (for text)`
- `[=, <, >, >. <>] " . $variable . " (for numbers)`

Creating a True Boolean Query Using SQL Command Match Against in Boolean Mode

```
$query="SELECT * from " . $tablename . " WHERE
MATCH(firstname, middlename, lastname, maidenname) AGAINST
('$anyname' IN BOOLEAN MODE) AND newsyear>= ".$newsyear1." AND
newsyear<=".$newsyear2." ORDER BY " . $sortby . $order;
```

Submitting the Query

```
$result=mysql_query($query) or die("Couldn't execute query.");
```

Displaying the Results

```
while ($rows=mysql_fetch_array($result)) {Print
$rows["fieldname"]." ". $rows["firstname"]."
".$rows["fieldname"]."<br>";}
```