PHP CLI Tutorial

- HOWTO make you first PHP CLI script
- HOWTO read arguments for command line
- HOWTO read from STDIN with PHP CLI

This tutorial is intended for anybody who wants to write PHP Scripts that don't require a web server (neither Apache Web Server nor Microsoft IIS). PHP CLI can be used for a wide range of things. After you read this tutorial, you able to write your own PHP CLI scripts.
There are two ways you can execute your PHP CLI scripts:

- The first way is to use php -f /path/to/yourfile.php. This calls the PHP binary and passes your script as a parameter. This is ugly and ineffective.
- The preferred manner is to chmod +x PHP files that you wish to call from a command line and add an appropriate shebang line at the top of your PHP script (#!/usr/local/bin/php - make sure that you supply here correct path to your PHP binary).

- ## HOWTO make you first PHP CLI script

  You need to create the following file and save it under myfile.php name for this tutorial. The simplest "Hello world" CLI script should look like this:
  #!/usr/bin/php -q
  <?php
  echo "Hello world of PHP CLI!";
  ?>
  Do not forget to set **executable** permissions to the file (e.g. 755):
  $ chmod 755 myfile.php
  Now run it simply by typing the following:
  $ ./myfile.php
  In the case if you are using PHP with Windows then you do not need to set permissions. You will run your PHP script like this:
  Microsoft(R) Windows DOS
  (C)Copyright Microsoft Corp. 1990-2001.

C:>cd Desktop

DESKTOP> php.exe myfile.php
Obviously, your first line of the script will look like this:
#!C:\Program Files\php\php.exe -q
Voila! Congratulations! You created and run your first PHP CLI script!

-

HOWTO read arguments for command line

Like most scripting languages you are able to pass in command line arguments. Create a file named testargs.php

```
#!/usr/bin/php -q
<?php
echo "Test Arguments:\n";
echo $_SERVER["argc"]."\n";
echo $_SERVER["argv"][0]."\n";
?>
```

$_SERVER["argc"] will give the integer number arguments enter including. Note the script itself is an argument. That means that you always will have 1 or more.
$_SERVER["argv"] is an array of arguments. To access the first argument it will be at index 1, $_SERVER["argv"][1]. The script's file name itself is at index 0, $_SERVER[argv"[0].

-

HOWTO read from STDIN with PHP CLI

Here is very simple, but powerful example demonstrating how to read from STDIN:

```
#!/usr/bin/php -q
<?php
/* Define STDIN in case if it is not already defined by PHP for some reason */
if(!defined("STDIN")) {
define("STDIN", fopen('php://stdin','r'))
}

echo "Hello! What is your name (enter below):\n";
$strName = fread(STDIN, 80); // Read up to 80 characters or a newline
echo 'Hello ' , $strName , "\n";
?>
```

Now you can do everything you like with PHP CLI!