# PHP Lab 3 Troubleshooting PHP Scripts

The biggest problem my students have had is that they write a PHP script, they open it from their browser, and nothing happens. No errors, nothing!

When a user hits your PHP page, if there are errors in the PHP the user will see a blank page. You don’t want users to see error messages, as they could give valuable information to an attacker. It's good to be able to run the PHP script ahead of time to remove errors before the script is part of your site.

Note: This lab assumes you have read the module PHP Variables and Syntax. Please do that before continuing.

## Method 1: Command Line Interactive Mode

This method gives you a PHP prompt from the command line where you can enter PHP commands one at a time and see the result or errors. This is covered in slide 24 of the PHP Flow Control module.

1. Enter the PHP interactive mode.

[john@localhost ~]$ php -a

Interactive shell

php >

2. Enter the script from the Cyber Aces module PHP Syntax & Variables module, Variable Example, on slide 10. Type the commands one at a time. (You don't need to enter <?php because you're already in PHP.) Make some typos so you see what errors look like.

## Method 2 Command Line

This method runs a PHP script that you've typed into a file. In this case, the command is just

[john@localhost ~]$ php filename.php.

This is covered in slide 23 of the PHP Flow Control module.

This is the review question from slide 15 of PHP Syntax & Variables. Paste it into a file in your home directory, save it with the extension .php, and run it with php filename.php. Figure out what it will do before you run it.

<?php

$name = "Charles";

$quote = "Aces, $name";

echo "Chuck's dad always used to say \"$quote\" to him as a kid.";

Put an error into the file (removing a ";" will work) and run the file again. Note the error that you get. PHP error messages aren't very enlightening...

## Method 3 Running PHP in your web server

1. Copy the file you made in Method 2 (with the error) to the document root of your server, /var/www/html. Attempt to run the script by pointing the VM's browser to:

http://localhost/filename.php

If your script still has the error, your browser won't show anything. To see what happened, look at the error log in /var/log/apache2/error.log.

2. Fix the error in /var/www/html/filename.php and point your browser to it again. The browser should show output this time.

3. To see who has accessed your server, look at /etc/httpd/logs/access\_log. Can you tell which accesses had errors and which worked? (Hint: The HTTP status code is the second item in each line, after the URI that was accessed.)

## Practice

Test the examples in the PHP Variables & Syntax module using the three methods above.

# Hand in

Hand in a screenshot of browsing to your filename.php.