Introduction to PHP

Learning How and Where To Get Started

by Aaron Saray

Why trust this guy?

- Programming for more than 2 decades
- WROX Author
- MKE PUG Founder
- Zend Webinar Presenter
- Loves PHP
- Knows there was an 'i' before iPod

What is PHP

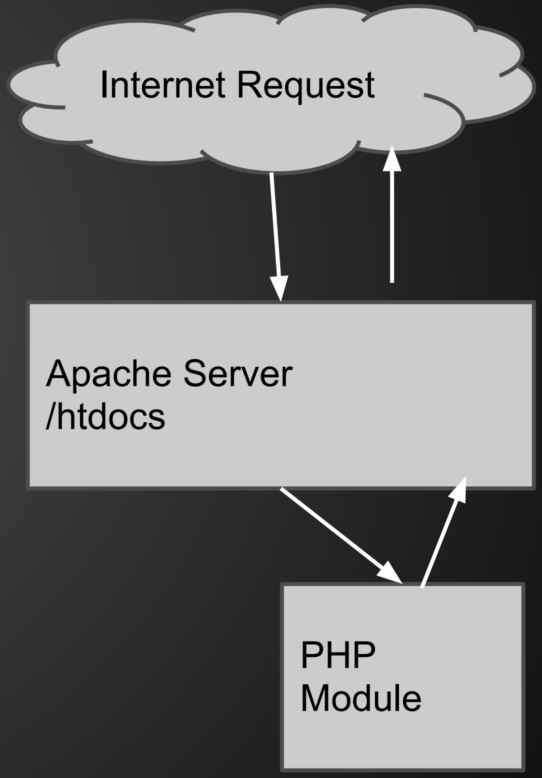
- What does PHP Stand for?
  - Personal Home Page
  - PHP: Hypertext Preprocessor
- What IS it?
  - Interactive portion of the back-end website
  - Programming (or scripting) language
  - Handles input/output, control structures, data access, persistent storage

Why is PHP Important?

- Easy learning curve
  - For the beginner stuff, that is
- The web is not persistent
  - PHP can help persist those single transactions
- No one JUST wants information anymore
  - An article without a comments section is 'dead'

How does it work? (The quick version)

- Web server loads PHP
- Server looks for PHP in source documents
- If found, executes PHP interpreter
- Final product is combined with source and sent via server



So...

You can serve web pages and documents without PHP.

But PHP puts the magic in it - the "interactiveness." :)

PHP is doing this...

- Commenting on a blog
- Writing a blog
- Uploading a file
- Processing a submitted contact form
- e-Commerce purchases
- Paypal integration / CC / as400

... why PHP?

- Open source
  - free
- Enterprise support
  - Zend
  - Many vendors
- Efficiencies
  - Yahoo, Facebook, etc.

Don't have a war!

Ok, I'm sold... now what?

- Installing / configuring web server and PHP
- What tools to write PHP Code
- Programming examples
- Where to go after this?
  - hint: it's gonna be a link on my website

Running/Installing PHP

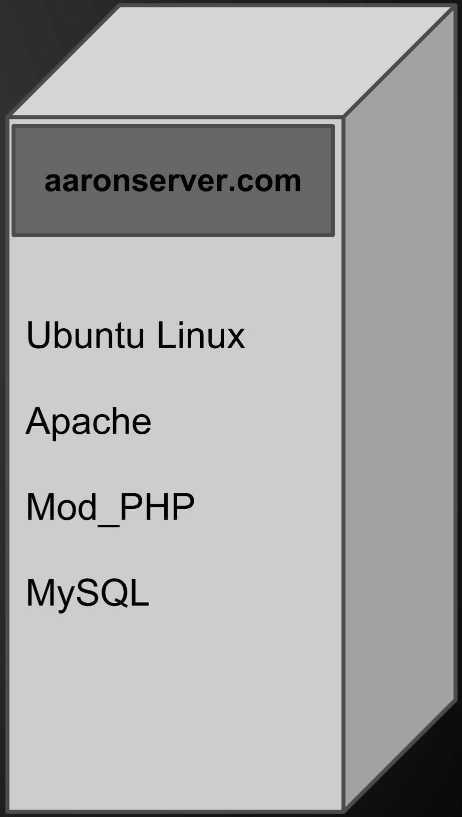
- Two ways* to run PHP
  - solely as a CL "program"
    - maintenance tasks, etc
  - as part of a web-page
    - most likely way to get introduced to it
- You can also install PHP using Zend tools on the system-i.
  - we're going to focus on local laptop work. Not everyone has a spare as/400 at home.

* PHP now supports running it's own webserver from the executable. I won't be covering this.

Quick look at 'the stack'

- Server hardware running an OS
  - Win or Linux probably
- Web server daemon
  - Apache, Nginx, IIS
- PHP module
  - Apache Module, Fast-cgi, etc
- Database
  - MySQL, MSSQL, Postgres

My typical rig



Emulate the Server Locally

- Remember, things like HTML, CSS, Javascript can run locally

- PHP can't - it needs an interpreter and compiler

- Run little servers on our laptop
  - Using tools on next slide
  - or using VMWare!

Install Apache/PHP (and some goodies)

Windows

- WAMP

Mac

- MAMP

Linux

- apt-get/rpm
- taskel lamp-server

WAMPSERVER,

a Windows web development environment.

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your databases.

START WITH WAMPSERVER

WampServer installs automatically all you need to start developing web applications and is very intuitive to use. You will be able to tune your server without even touching the setting files.

INSTALLING

- Double click on the downloaded file and just follow the instructions.

FUNCTIONALITIES

WampServer's functionalities are very complete and easy to use so we

More WAMP Notes

- Pay attention to where you set your "www root" to
- WAMP icon in your systray gives options to start/stop services and short-cut to www directory

- Anything you put in www directory loads at http://localhost

  c:\users\aaron\www\myfile.php == http://localhost/myfile.php

Last WAMP Notes

- The install is pretty simple
- The forums really help a lot!
- Don't get discouraged
- The initial setup will work for 95% of the things you want to do
- It definitely will work for this talk! :)

How Do I Create PHP?

- PHP files are just plain text files

- You can edit them in Notepad!
  - Not MSWord or Wordpad though...

- You can use an IDE
  - Eclipse PDT
  - PHPStorm
  - Komodo
  - Websphere can do this
    - Note: your runtime is still WAMP, not websphere

How to create PHP (con't)

- Save the file as a plain text file with the name of the file followed by .php mypage.php, contact.php, index.php

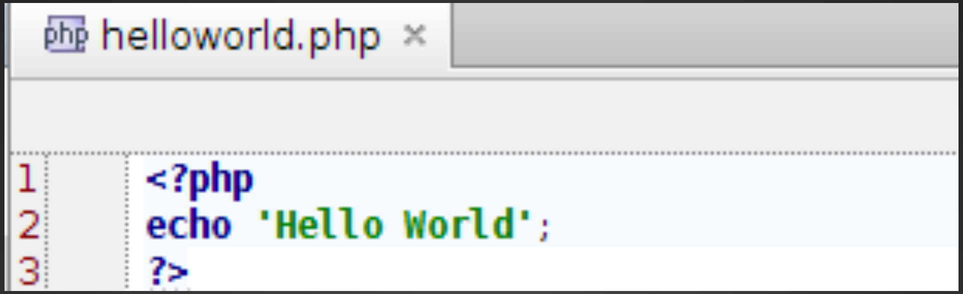
- Special note: when you have a file named index.php, that is loaded by default inside of any directory

```
http://localhost/cars  ==
c:\www\cars\index.php
```

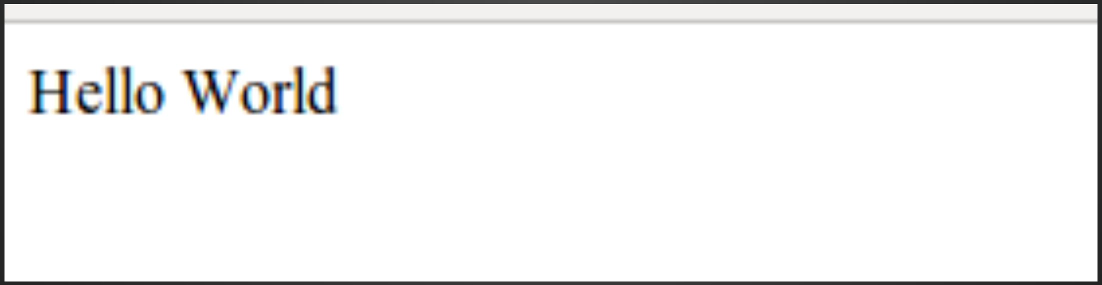
Ok, let's begin

Time to do Hello World.

Hello World



```
<?php
echo 'Hello World';
?>
```



```
Hello World
```

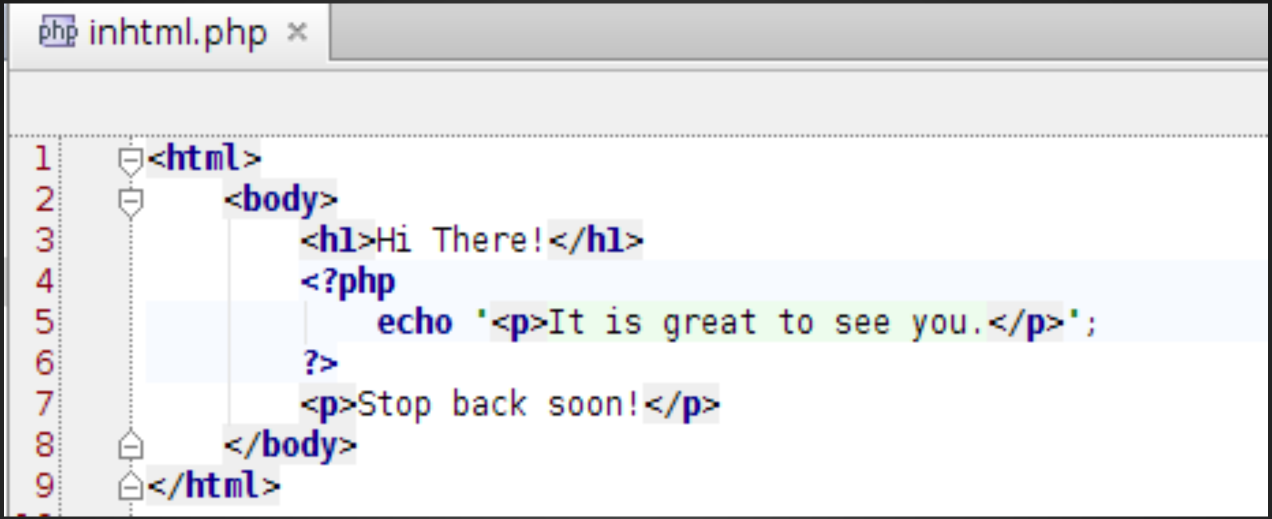
Using HTML and PHP

- The power of PHP comes with the fact that you can create individual PHP files - or embed it in your own HTML files

- This helps us create dynamic HTML

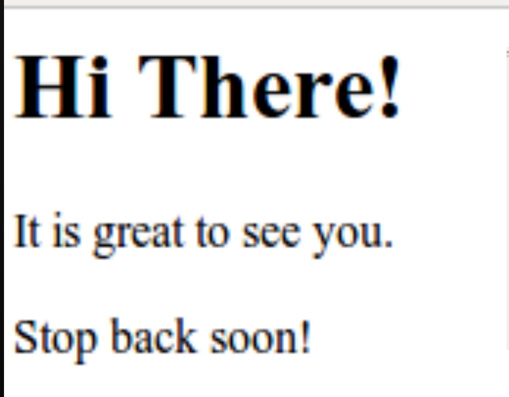
- Create as many instances of PHP code inside of an HTML file as you want.

Use PHP in HTML



inhtml.php

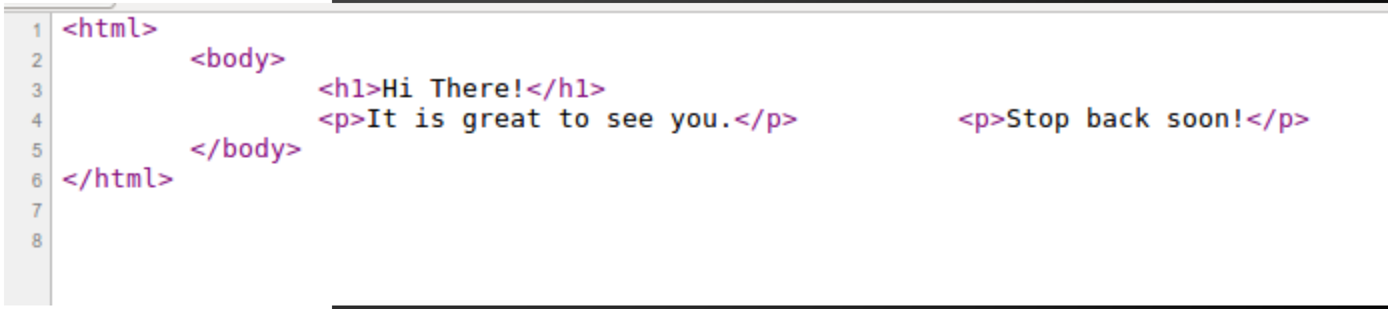
```
<html>
    <body>
        <h1>Hi There!</h1>
        <?php
          echo '<p>It is great to see you.</p>';
        ?>
        <p>Stop back soon!</p>
    </body>
</html>
```



Hi There!

It is great to see you.

Stop back soon!



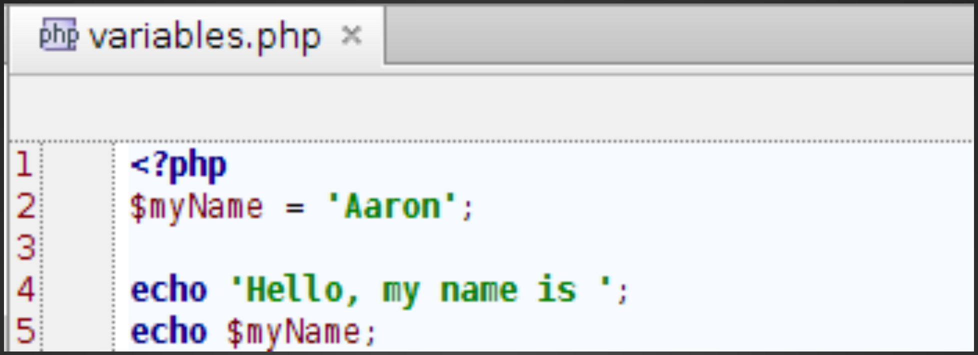
```
<html>
        <body>
                <h1>Hi There!</h1>
                <p>It is great to see you.</p>          <p>Stop back soon!</p>
        </body>
</html>
```

Variables make all the difference

- Variables in PHP are the first step in making a dynamic page
    - loosely typed
    - string, integer, float, object, class, null, resource

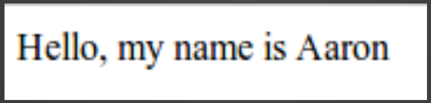
- Variable syntax:
    - Begins with $
    - Must not begin with a number, but then any letter, number, or underscore
    - Generally use camelCase
        - but you can use whatever you want, just stay consistent

Assigning/Displaying Variables



```php
<?php
$myName = 'Aaron';

echo 'Hello, my name is ';
echo $myName;
```



```
Hello, my name is Aaron
```

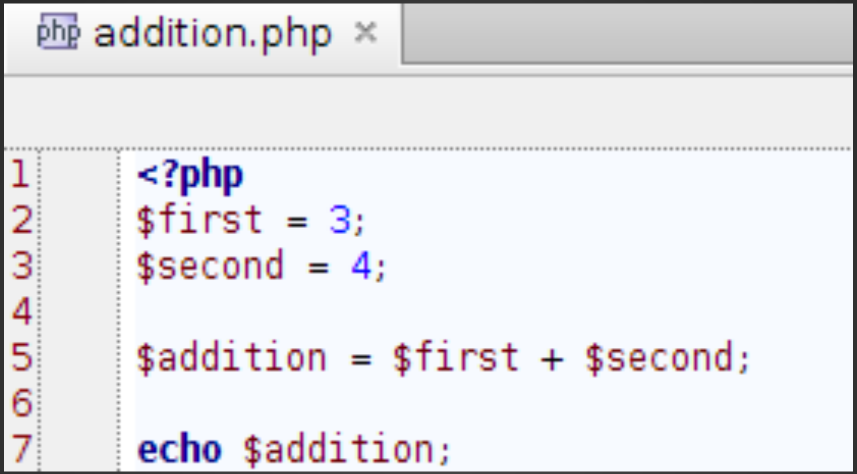
Addition

- Math!
  - We'll just cover addition

  - You can add variables or constants

- Strings
  - You can "add" strings to each other too
    - concatenation - more on this later

Math!



```php
<?php
$first = 3;
$second = 4;

$addition = $first + $second;

echo $addition;
```

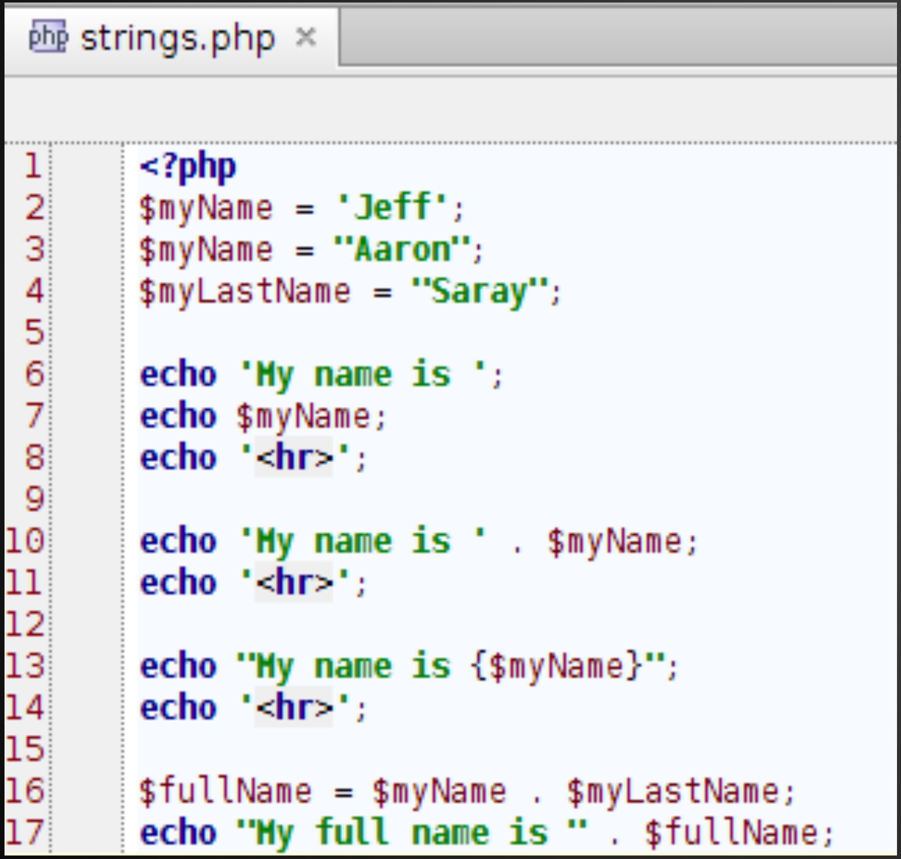
Strings

- Strings are contained in single or double quotes
  - Double quotes will interpret variables as part of the string

- Strings are combined with a period

- Strings can be combined into another variable or on-the-fly with echo

String Examples



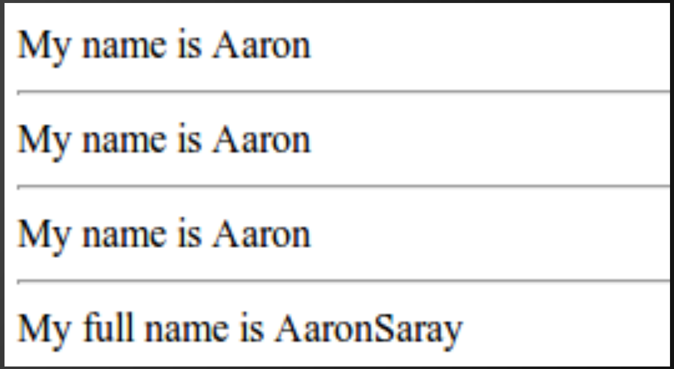
```php
<?php
$myName = 'Jeff';
$myName = "Aaron";
$myLastName = "Saray";

echo 'My name is ';
echo $myName;
echo '<hr>';

echo 'My name is ' . $myName;
echo '<hr>';

echo "My name is {$myName}";
echo '<hr>';

$fullName = $myName . $myLastName;
echo "My full name is " . $fullName;
```

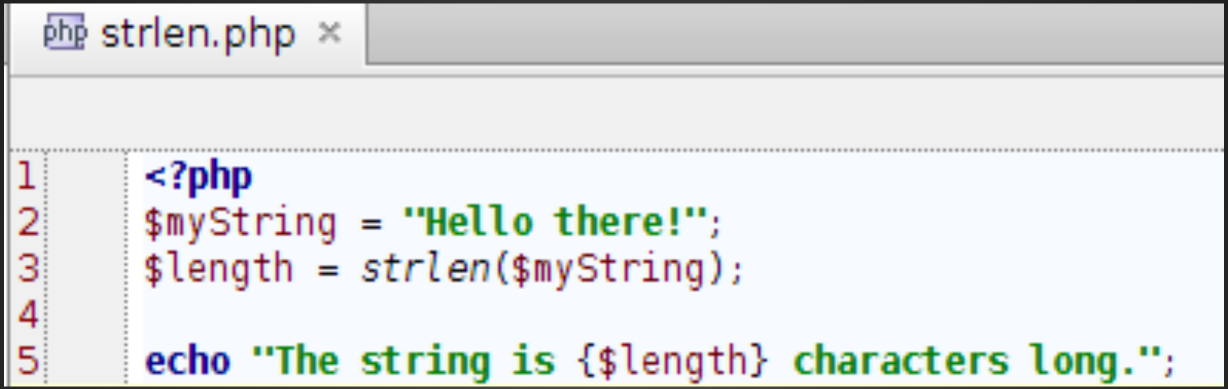


```
My name is Aaron
My name is Aaron
My name is Aaron
My full name is AaronSaray
```

Functions

- Built in
  - over 500 built in functions and growing
- Create your own
  - create your own functions
    - advanced: create methods in classes using same syntax
- Functions accept 0-n parameters, return mixed or void
- Save code - use functions!

Built in Function Example

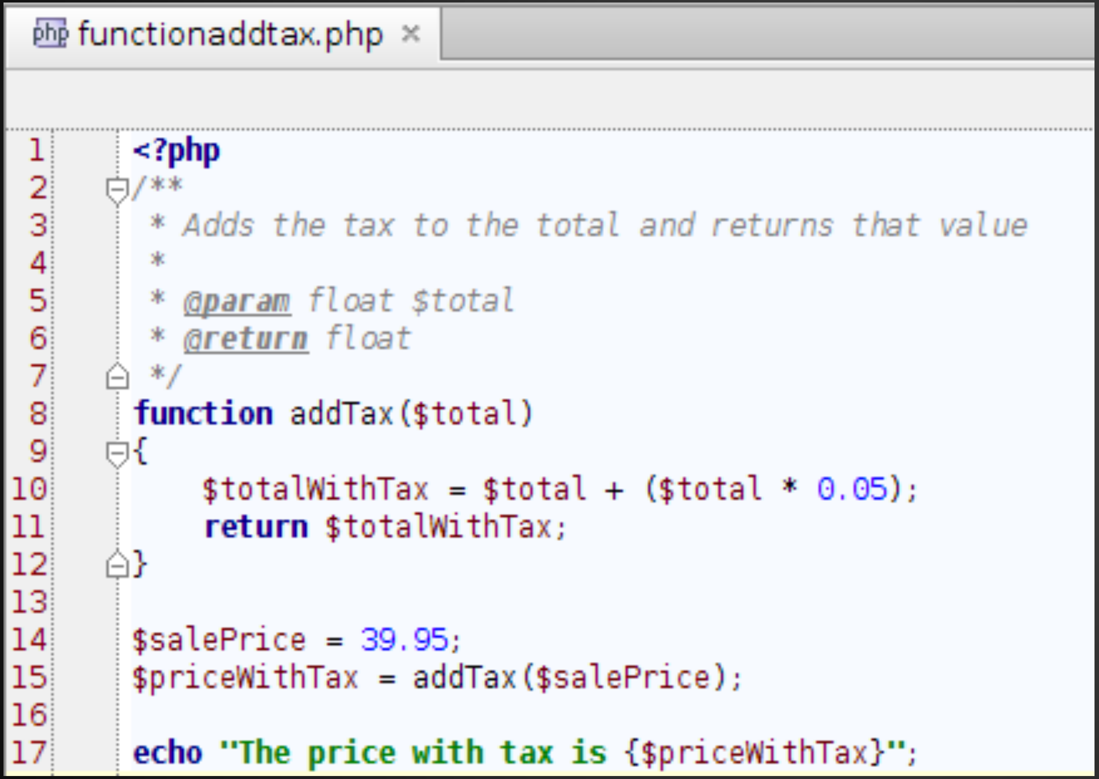


```php
<?php
$myString = "Hello there!";
$length = strlen($myString);

echo "The string is {$length} characters long.";
```

Other built in functions include options to format dates, see when the sun rises, parse XML, advanced math calculations, string manipulation, object creation, array methods, and more!

Our first function



functionaddtax.php

```php
<?php
/**
 * Adds the tax to the total and returns that value
 *
 * @param float $total
 * @return float
 */
function addTax($total)
{
    $totalWithTax = $total + ($total * 0.05);
    return $totalWithTax;
}

$salePrice = 39.95;
$priceWithTax = addTax($salePrice);

echo "The price with tax is {$priceWithTax}";
```

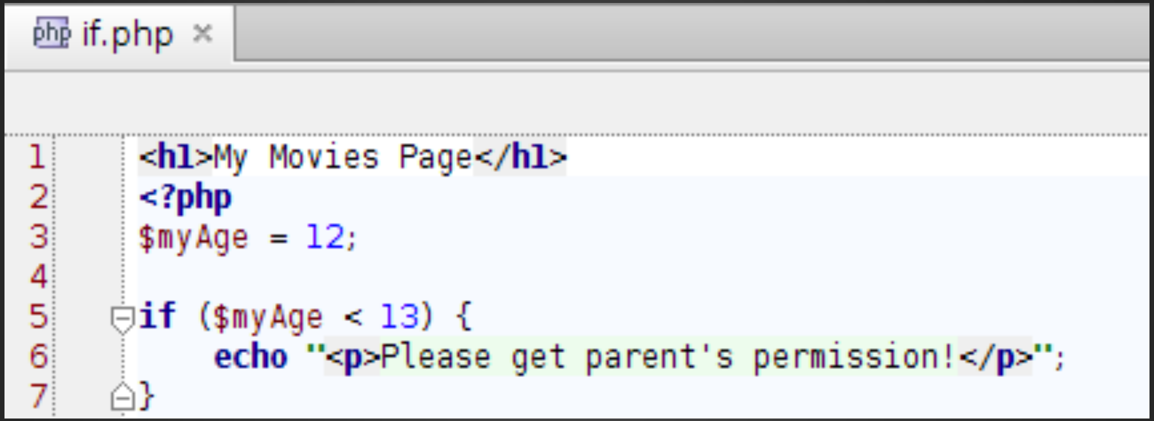
Quick notes:

- Comments
  - //
  - /* */
- Document your code
- We could run addTax() as many times as we want in this page

Logic and Control

- If statements
  - ==, <, >, <=, >=, etc
  - switch/case statements
- For loops
  - do/while loops

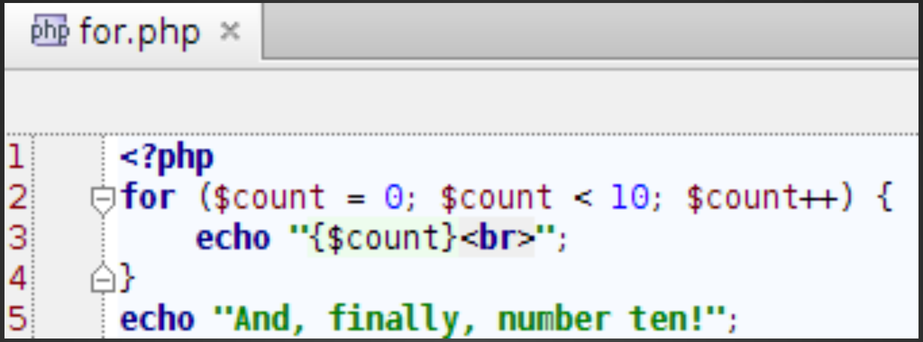
If statement



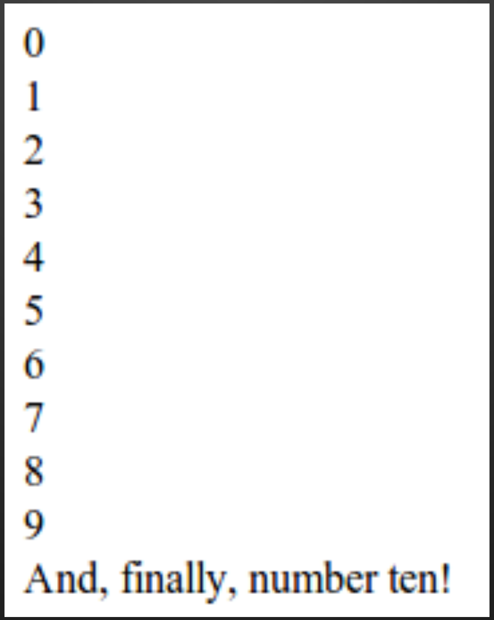
```
<h1>My Movies Page</h1>
<?php
$myAge = 12;

if ($myAge < 13) {
    echo "<p>Please get parent's permission!</p>";
}
```

For Loop



```php
<?php
for ($count = 0; $count < 10; $count++) {
    echo "{$count}<br>";
}
echo "And, finally, number ten!";
```



```
0
1
2
3
4
5
6
7
8
9
And, finally, number ten!
```

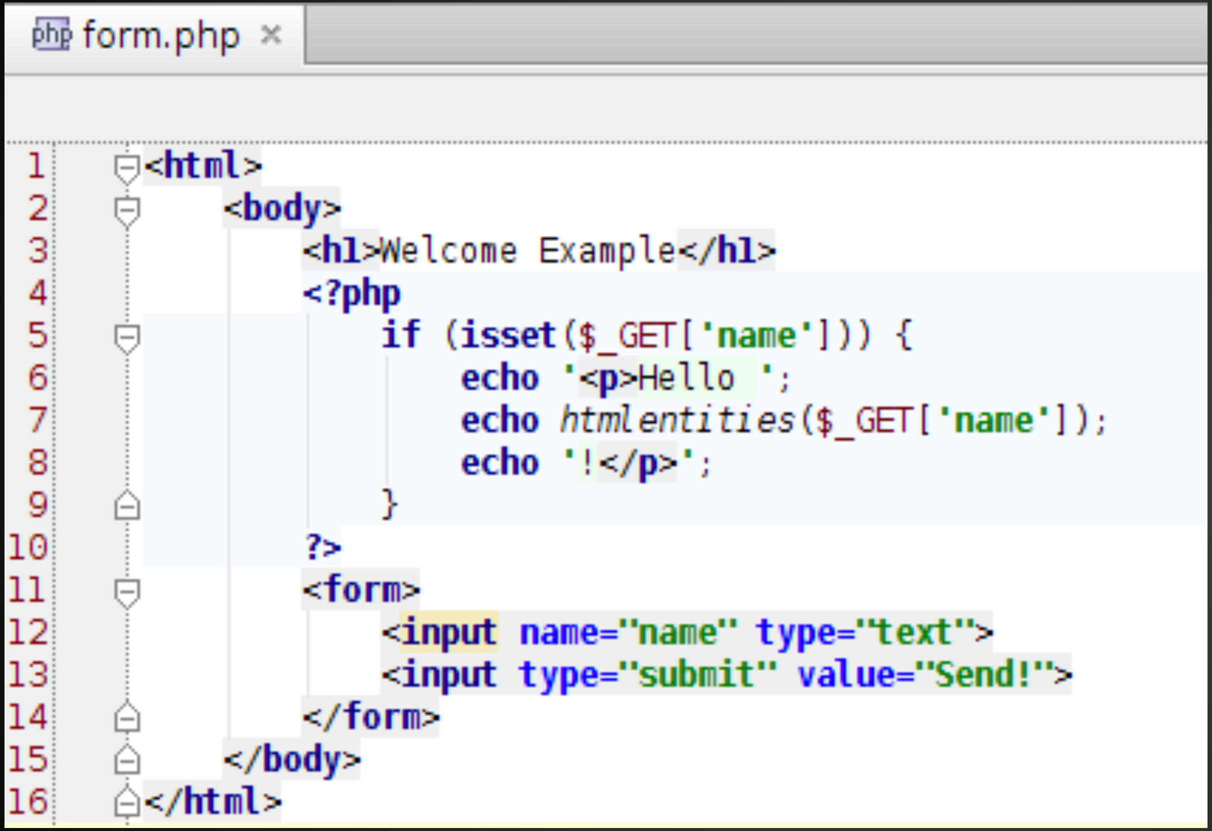
Special Bonus: Forms!

- This is where it gets dynamic

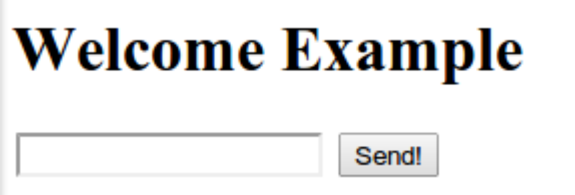
- Forms are how you interact with most webpage users

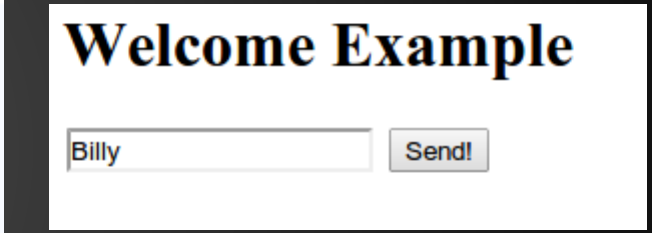
- PHP makes accessing form data easy!

Form Example

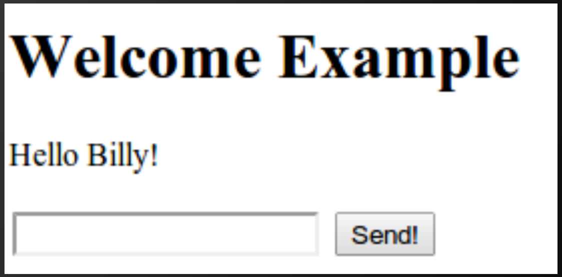


```
<html>
    <body>
        <h1>Welcome Example</h1>
        <?php
            if (isset($_GET['name'])) {
                echo '<p>Hello ';
                echo htmlentities($_GET['name']);
                echo '!</p>';
            }
        ?>
        <form>
            <input name="name" type="text">
            <input type="submit" value="Send!">
        </form>
    </body>
</html>
```


Welcome Example


Welcome Example

Billy


Welcome Example

Hello Billy!

Closing thoughts

- I haven't even grazed the surface
  - but I hope you're interested
- There are many ways to do PHP
  - That was the hardest part of this presentation - which way should I show you??
- Keep learning!

Where to go from here?

- Things to learn:
    - Read some tutorials
    - Try making a contact form that sends a mail message
        - validate all fields are filled in
        - send a message to the recipient
        - thank the user

- Check out the companion information:
    - http://aaronsaray.com/wmcpa2013

The End

Any questions?

:)

http://aaronsaray.com/wmcpa2013