

LIS-5364

PHP Basics

PHP

- First weirdness – it's not really run or compiled like most languages. It lives inside HTML...

More specifically...

The process

Webserver parses/converts the HTML that it knows

If it sees `<?php`, it passes that to the PHP “converter”

PHP does its thing, and spits text **back into the HTML** (as opposed to “directly on the page”)

Basic PHP

```
<html>
<head>
<title>My First PHP Page</title>
</head>
<body>
<?php
echo "Hello World!";
?>
</body>
</html>
```

Whitespace?

- Like in HTML, mostly ignored
- Hence, the importance of a semicolon (;) at the end of every line.

Variables

- Start with \$
- Case sensitive
- Must begin with alpha or underscore
- Must have only alphanumeric or underscore

PHP Data Types

- 3 classes:

Scalar – One thing

Compound – Multiple things

Special – umm, special

Scalar

String – words and stuff

Integer – whole numbers

Float – Decimals

Boolean – true or false

Compound

Arrays

Objects

Special

Resource – Pulling data from an external resource

Null – Generally, unset

“echo”

Printing words to the screen.

Single Quotes = literal

Double Quotes = will resolve variable (you'll probably use double waaay more)

More echo:

- But, what if you want to actually print quotes and such? Use Backslash for escaped characters, e.g.

`\'` = literal single quote

`\"` = literal double quote

`\$` = dollar sign

`\n` = newline

`\t` = tab

Big chunks of text?

Use “heredoc” to resolve variables, and the new “nowdoc” to do literals

A little on comments

PHP supports C and Shell comments, so:

// .. to the end of the line

to the end of the line

/* to */

Operators

String operators? Really the period, and even then, you'd probably not use it much.

Math? They basically work as expected. Remember, PHP is weakly typed, so with numbers, they're integers until they're not. Be careful.

(ps, remember modulus and what it's good for?)

The magic increment

If x is your integer variable then,

$x++$ (without the equals assignment) will increment it by one.

Test operators

We use ONE equal sign to assign a value to a variable...

.. what do we use TWO for?

Conditionals (and conditional loops)

The venerable if. So remember, like in most languages (including English) an if (and the other conditionals) never stands alone, so it doesn't need a semicolon (until it does).

“If I see you around here one more time..”

(if what, mofo?)

Conditionals and Loops

Multiline conditionals require brackets

}

(Singles don't)

C and L

(quick, why don't we use unconditional loops?)

If (with or without else or elseif)

While (and *do...while*)

For

..and in a minute, *foreach*

Arrays

- Basic arrays:

0 indexed

Can declare, but don't need to

```
$myarray = array("these","are","elements");
```

```
Add to end with $myarray[] = "yo";
```

Associative Arrays

Key-Value relationship

Think like “dictionaries”

Remember, they are one-way functions; A value might be the same for multiple keys, but each key has **ONE AND ONLY ONE** value.

Assign like: `array (“word” => “the definition”)`

foreach

Do something on each element of the array.

foreach (\$thisarrayelement as \$thing)

or

- foreach (\$currentelement as \$key => \$value)

Array functions

array_intersect

array_diff

array_merge

array_unique

sort

shuffle