

Getting Started with PHP

Grant Root
grant@rootcentral.org



This Presentation

- ... is posted on my site, at <http://www.rootcentral.org>.
- Look for a “Site News” entry with a link to the presentation.

What is PHP?

- A general purpose programming / scripting language available as Free Software
- Syntax is borrowed from C, Java and Perl... with a few twists
- Oriented toward web development
- Suitable as a template engine
 - Can be embedded in HTML pages

Modes of Operation

- Web server integration
 - Via SAPI or CGI
 - Nothing needed on client side
- Batch mode
 - Via CLI (command-line interface)
- GUI
 - Via PHP-GTK extension

Where did PHP Come From?

- 1995: Rasmus Lerdof creates PHP/FI (Personal Home Page / Forms Interpreter)
- 1997: PHP/FI 2.0 (Rasmus and a few others)
- 1997-1998: Complete rewrite by Andi Gutmans and Zeev Suraski as PHP 3.0 (PHP: Hypertext Preprocessor)

Where did PHP Come From?

- 1998-1999: Rewrite by Andi and Zeev for performance and modularity. Result was PHP 4.0 based on the Zend Engine.
- 2004: PHP 5.0 with Zend 2.0, new object model and many new features
- Current versions are 4.4.0 and 5.0.5, with 5.1.0 in Release Candidate status.

Popularity

- PHP/FI: 50,000 Internet domains
- PHP 3: Hundreds of thousands of domains
- Today: 22 million domains
- Most popular Apache module, installed on 46% of Apache servers

Extensions Archives

- Similar to CPAN for Perl
- PEAR (PHP Extension and Application Repository)
 - Reusable PHP components
- PECL (PHP Extension Community Library)
 - Extensions to the PHP engine

Environment

- OS: Linux / Unix, MS Windows, NetWare, OS2, AS/400, etc.
- Web server: Apache, MS IIS, Netscape/iPlanet and others via SAPI, all others via CGI
- Databases: ODBC, MySQL, mSQL, MS-SQL / Sybase, PostgreSQL, Firebird / Interbase, DB2, dBase, etc.

Getting PHP

- Provided by most web hosting services
- Available in most Linux distributions
- From PHP site
(<http://www.php.net/downloads.php>)
- Bundles and installers
 - e.g. EasyPHP
(<http://www.easyphp.org/?lang=en>)

Installing PHP

- Establish your environment first; install OS, web server and database
- Help available at PHP web site for installing on Unix, MacOS and Windows
 - <http://www.php.net/manual/en/install.php>
- See the user comments for tips on integration w/ uncommon web servers

Documentation

- Extensive documentation, with user comments and code examples, at <http://www.php.net/docs.php>
- Zillions of web sites (many linked from the php.net resource page)
- Huge numbers of books available; search Amazon.com for “php”
- Magazines, e.g. PHP Architect (<http://www.phparch.com>)

Recommended Books

- PHP and MySQL Web Development – Welling & Thomson
- Learning PHP 5 – David Sklar
- PHP Cookbook – Sklar & Trachtenberg
- PHP 5 Objects, Patterns, and Practice - Matt Zandstra

Editing PHP Files

- Plain text files - text editors such as vi, Notepad, etc. will work fine
- Often integrated with HTML
- Files distinguished by extension: php, php3, phtml
- Editors / IDEs are available w/ useful features such as syntax highlighting, function completion, code tidying, class explorers, debuggers, etc.

PHP Code Delimiters

- Separate, or escape, PHP from HTML
- Four kinds:
 - `<?php foo(bar); ?>` (preferred)
 - `<script language="php">foo
(bar);</script>` (makes some editors happier)
 - `<? foo(bar); ?>` (short form, not supported by all servers)
 - `<% foo(bar); %>` (ASP-style)

Finally, a program!

```
<?php echo "Hello, world!"; ?>
```

or...

```
<?php  
echo "Hello, world!";  
?>
```


Embedded PHP Code

```
<?php $name = "Grant"; ?>
<html>
<head><title>PHP Page</title></head>
<body>
<h1>My PHP Page</h1>
<p>Hi, my name is <?php echo $name; ?>,
  and I program in PHP!</p>
</body>
</html>
```

Advanced Escaping

```
<?php
if ($expression) {
    ?>
    <strong>This is true.</strong>
    <?php
} else {
    ?>
    <strong>This is false.</strong>
    <?php
}
?>
```

Variable Typing

- Scalar types:
 - Boolean, integer, float (aka 'double'), string
- Compound types:
 - Array, object
- Special types:
 - Resource, NULL

Variable Typing

- Weakly typed variables
- Decided at runtime depending on context
- Type can be specified via type casting or `settype()` function
- Values for comparison purposes can change based on context
 - Become familiar with `==` (equal) vs. `===` (identical) comparison operators

Control Structures – if

```
if ($name == "Fred") {  
    echo "Fred's here!";  
}  
else {  
    echo "who are you?";  
}
```

Control Structures – elseif

```
if ($name == "Fred") {  
    echo "Fred's here!";  
}  
elseif ($name == "Tom") {  
    echo "Tom's here!";  
}  
else {  
    echo "who are you?";  
}
```

Control Structures – switch

```
switch ($name) {  
  case "Fred":  
    echo "Fred's here!";  
    break;  
}  
  
case "Tom":  
  echo "Tom's here!";  
  break;  
}  
  
default:  
  echo "who are you?";  
}
```

Control Structures – Loops

- while (test precedes execution)
- do... while (test follows execution)
- for
- foreach (iterate over arrays)
 - PHP 5 adds iteration over objects (in customizable ways)

Control Structures – Alternative Syntax

```
<?php
if ($a == 5):
    echo "a equals 5";
else:
    echo "a is not 5";
endif;
?>
```

Ternary Comparison Operator

```
<?php
$quantity = (is_numeric($qty)) ? $qty : 0;

// The above is identical to this if/else:
if (is_numeric($qty)) {
    $quantity = $qty;
} else {
    $quantity = 0;
}
?>
```

Including Code

- include, require
 - Vary in failure handling
- include_once, require_once
 - Avoids duplicate definitions
- Often used for function or class libraries
- Be very careful of variables in include statements!

Functions

- Thousands of functions in 162 different categories
- Special emphasis on...
 - Database interface
 - HTTP and URL
 - XML and web services
- Complete list at
<http://www.php.net/manual/en/funcref.php>

Objects

- Completely overhauled object model in PHP5
- Robust set of features including constructors and destructors, abstraction, interfaces, visibility control, method overloading and “magic” methods, iteration, autoloading, etc.
- True multiple inheritance is not supported.

Accessing Web Data (Old Way)

- register_globals directive must be On
- GET and POST variables are automagically registered as global variables in your script's namespace:
- Deprecated because of security concerns
 - Just where did that variable come from anyway? GET? POST? Cookie?

Accessing Web Data (Preferred)

- Use the superglobal arrays
 - `$_GET`, `$_POST`, `$_COOKIE`, `$_FILES`
 - `$_SERVER`, `$_ENV`
 - `$_SESSION`
- Allows you to know where the values are from
- Little likelihood of an uninitialized variable being exploited

Handling External Input Safely

- Stay alert, trust no one, keep your regex handy!
- Be suspicious of any external data source, even the web server itself.
- Filter all input.
- Escape all output.

Filter Input

- Make sure each field has exactly the kind of data that you expect.
- Use type checking and regular expressions.
 - `gettype()`, `is_numeric()`, `intval()`, `ereg()`, `preg_match()`, etc.
- Functions like `strip_tags()` are useful for free-form fields.

Escape Output

- HTML output needs to have special characters and replaced with character entities using `htmlspecialchars()`.
- Variables used in database queries must be sanitized using functions like `mysql_real_escape_string` or (at least) `add_slashes`

Resources

- The PHP Related Links page (<http://www.php.net/links.php>) contains links to support companies, professional associations, news sites, FAQ sites, tutorials, scripts and programs, magazines, multimedia, authoring tools, commercial tools, accelerators, merchandise, job opportunities, ISPs, and... other collections of PHP links!

Resources

- PHP Security Consortium
 - <http://phpsec.org/>
- PHP Security Guide
 - <http://phpsec.org/projects/>
- DMA Web Development mailing list
 - <http://www.dma.org/mailman/listinfo/web-development>

Questions?

