PHP: Hot Pages

http://hoboes.com/NetLife/PHP/

August 16, 2012

What is PHP? 1
Why use PHP? 1
More information 2
The php extension 2
Basic Code 3
Dates and Times 3
Functions 4
Variables 4
Custom functions 4
Comments 6
Isset 6
If 6
Time 6
Semicolons and line-endings 6
Finding errors 7
Line endings 7
Avoiding errors 7
The Final Code 8
Your Favorite Color 9
Classes 9
Simple form 10
Include file security 12
Braces vs. colons 12
Cleaning your data 13
Verify an answer 13
Triple equals (===) 14
Else 14
Different shades of White 15
HTML code injection 17
Email 18
Add a name 18
Single vs. double quotes 20
Notification 21
No untrusted recipients 21
Post and Get 22
The final code 23
fields.phpi 23
color.php 24
A Poll 25
Save data to a file 26
Create the folder 26
Save the first ballot 27
Hide the form 27
Arrays 28
Accessing array data 30
Sorting 31
Keep your powder D-R-Y 32
Don't trust anyone 33
Visitor Sessions 34
Display the results when they've already voted 35
Cookie errors 36
Sessions and security 37
Extra credit 38
Templating and programming 38
Multiple questions 38
Graphical results 42
Poll colors 45
SQLite 47
Create storage 47
Create the file 47
Create the table 47
Tables 48
Manage your data 49
Queries 49
Save records 49
Display records 50
Other database servers 51
More Information 52
Programming is Easy 52
GNU Free Documentation License 53

WHAT IS PHP?

PHP allows you to create *dynamic* web pages: web pages that do things differently depending on when or how someone visits your page, or that change depending on some other outside data. In this tutorial, I'm going to cover some of the very basics of using PHP to handle forms in your web pages. I'm going to assume that you know a little bit about HTML, but nothing about programming or PHP. If you aren't familiar with HTML, you'll want to go over the Web Writing Guide at *http://hoboes.com/NetLife/Web_Writing/* first.

You'll need a text editor of some kind; many web page editors have them built in; Dreamweaver does, for example. As I write this, you cannot use the text editor built-in to SeaMonkey. It will throw out your PHP code when you switch between WYSIWYG editing and text editing. If you want a dedicated text editor, NoteTab on Windows comes recommended by people I trust, and Fraise or Smultron on Mac OS X are great choices.

WHY USE PHP?

If you need to embed dynamic text into otherwise unchanging web pages, you'll find PHP extremely useful. It was designed for this and it excels at it. You can easily create your web page as normal, with boilerplate text for the parts that need to be changed per visitor. Then, once you've designed the page, you can replace the boilerplate with PHP code to display the changing parts of the page.

You'll find that PHP is very portable: if it works on one server, it will usually work on any other server that has PHP installed. Most ISPs that provide server-side scripting provide PHP automatically.

PHP is also very useful for integrating web pages with databases. It has support for SQLite built in, and it often has support for MySQL, one of the most popular SQL database servers, pre-installed.

If you've already done some programming, the PHP scripting language resembles JavaScript, Java, and Perl. These languages all share a common ancestor, the C programming language. Of those, PHP is most different from JavaScript. PHP is a server-side scripting language. All of the "work" is done on the server. JavaScript usually runs on the client. It has little access to the information that the server has, and mediated access to information on the client. It can do lots of things on the client that PHP cannot. PHP, however, has full access to information that the server has. It only has information about the client that the client tells the server.

Because it is on the server, PHP cannot be modified by the client. While you cannot necessarily trust the information that the client gives to PHP, you can trust that your PHP is doing what you told it to do. Because PHP is on the server end, your PHP scripts can affect your server—such as by keeping an activity log or updating a database.

PHP can work with JavaScript to provide a combination of server-side and client-side functionality that will make your web pages more useful and your visitors happier.

MORE INFORMATION

You can get more information about PHP, as well as the full PHP manual, at the PHP web site, *http://php.net/*. The PHP on-line manual is extremely useful: not only does it let you quickly look up any part of PHP, it also includes notes from people who use PHP about problems you might run into and how to fix them. When you're looking at the official instructions for a function you're having trouble with, do a search through the comments at the bottom of the page. Chances are, someone else has run into the same problem and has already posted a solution.

If you want more information about PHP in printed form, there is *Programming PHP* by Rasmus Lerdorf and Kevin Tatroe. Rasmus Lerdorf is one of the authors of the PHP language.

If you are interested in using PHP with the MySQL database there is also *Web Database Applications with PHP & MySQL* by Hugh E. Williams and David Lane, and *MySQL* by Paul DuBois. I've found the latter to be an extremely useful MySQL reference. You might also find my own MySQL tutorial at *http://hoboes.com/NetLife/MySQL/* useful.

THE PHP EXTENSION

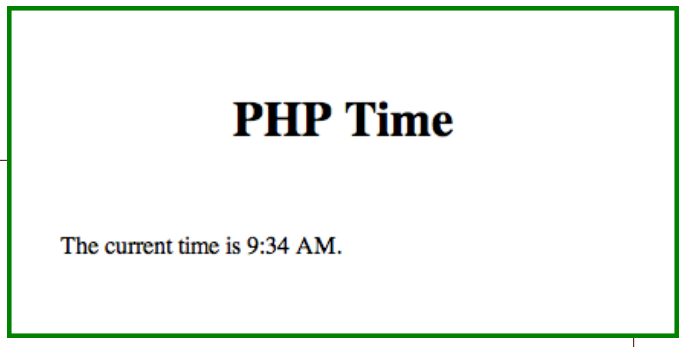
This depends on how your server is set up, but your PHP web pages usually need to end in the extension "php". This lets the server know that it needs to hand this page off to the "php" module before letting the visitor see the web page.

The choice of extension is completely up to whoever sets up your server. While it will usually be "php", it can be anything. Contact your system administrators to be sure.

BASIC CODE

DATES AND TIMES

PHP code starts looking like HTML code and ends looking nothing like it. If you've looked at HTML code, you've seen things that look like "" or "<h2>". Here's a simple web page:




```
<!doctype html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <title>Time for PHP</title>
      <style>
         body {
            margin: 10em;
            padding: 2em;
            border: solid .2em green;
         }
         h1 {
            text-align: center;
         }
      </style>
   </head>
   <body>
      <h1>PHP Time</h1>
      <p>The current time is 9:34 AM.</p>
   </body>
</html>
```

You can see this file in the resources archive as "time.php". Go ahead and grab it, and save it on your web site. Of course, even by the time you first view this page the current time is probably going to be wrong, and for your visitors it will be right only one minute out of every day. This is a great example of web content that would be better served with PHP. You will put your PHP code between "<?php" and "?>". Here's how to display the current time:

```
<p>The current time is <?php echo date('h:i A')?>.</p>
```

If you save this file as "time.php" on your web site, and then view it, you will see the current time every time you reload your page.