

Power to the Website!

by Lou Hornberger
lou_h@msn.com

According to a recent *Barna* survey, nearly 60% of Protestant churches have a website. If your experience is anything like mine, you will find that nearly 90% of those stink. Sorry, they just do. How long do we need the 2002 calendar of events online? How many animated GIF files *can* you fit on one webpage?

One way to avoid the problem of having “stale” content on your website is to

create dynamic content. Dynamic content (LINK: http://en.wikipedia.org/wiki/Dynamic_web_content) can be as simple as a JavaScript that puts today’s date on the page,

or dynamic content can generate an entire page every time it is requested by a user.

This can be done, basically, either on your



computer (client-side) when the actual program is sent to your machine for execution (like JavaScript) or on the website to which you are connected (server-side) and sent to you completed (like PHP and ASP).

While JavaScript allows you to do some neat things, you are limited. To create some great pages easily and cheaply, PHP is the way to go. PHP is free and widely supported.

PHP is essentially a programming language for web pages. The code that you entered is parsed (or run) on the computer where the web page resides and then the results of the PHP code are sent to your browser as a completed page.

According to PHP Headquarters (LINK: www.php.net), “PHP (which is a recursive acronym for “PHP: Hypertext Preprocessor”) is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.”

Here’s some helpful information on what you just read: *Hypertext* is the first two letters of the abbreviation “HTML.” It indicates that the text is more than just text, that there are ways to

interact with the text, like clicking on certain words and opening other web pages.

A “*Preprocessor*” is another way of saying “it does server-side processing.” The script is processed on the server before being sent to the client, i.e. your browser.

Open Source, some call it “free software” (LINK: www.fsf.org), is software that is distributed for anyone to use for no cost, and if you are into programming you can also download the “source code” (the original text the programmers used to create the program) and make changes for your own work.

A “*scripting language*” is like a light-weight programming language. Other scripting languages include JavaScript, Visual Basic Script (another client-side processor) and CGI (a server-side processor).

The idea that PHP can be “*embedded into HTML*” means that you can have both static and dynamic elements of your page when you use PHP.

If you want to get started using PHP you’ll need several things. The first and most important thing you’ll need is a web host that supports PHP. If you already have a hosting company for your website, check to see if you already have PHP support. Many do, but some require that you notify your provider that you want it and they will make it available to you.

If you just want to get started for absolutely nothing you can create an account for free with the United Kingdom version of Tripod (<http://www.tripod.lycos.co.uk/>). For some reason the U.S. version of Lycos does not support PHP.

You can edit your PHP pages easily in Notepad or any other plain text editor. Make sure you save every PHP file with the three letter extension of php, for example:

hello.php tells the server to interpret this as a PHP file and to preprocess it, then send the resulting HTML to the client browser.

Then, like with other web pages, you’ll need the ability to upload the files you create from your computer to the hosting server.

We will start like any good programming tutorial, the “Hello World” script. Type this into your text editor:

```
<html>
<head>
<title>My First PHP Script</title>
</head>
<body>
<h1>Here is my first PHP page,
ever!</h1>
<?php echo “Hello World!”; ?>
<center><h2>I like Christian A/V
Magazine!</h2></center>
</body>
</html>
```

Now, after you have typed these lines into Notepad (or your preferred text editor), save it as hello.php, then upload it to your PHP enabled server and then point your browser to it to view it.

We start off just like a regular HTML page, turn on HTML, turn on the heading, turn on the title and name the page, then turn off the title and heading sections. Now we move into the body where we have some large text that declares what is on this page.

Up to now, we have what appears to be a standard HTML page—that is until you type: `<?php` This is the tag that tells the server to start parsing in PHP. We use a simple PHP feature (technically it is a “language construct”) called *echo* (for tons of more info on the echo “feature” click here (LINK: <http://www.php.net/manual/en/function.echo.php>) to send text to the browser that text is within the

quotation marks, and every PHP command ends with a semi-colon (;). This is called using proper “syntax.” The syntax is how things are supposed to be laid out in order to work properly; usually if you don’t follow the syntax rules, the script won’t work. The semi-colon tells it that that command has ended and to be ready for the next one.

The next one in this case is the command to turn off PHP parsing. We end each PHP section with `?>`. Then we see we have some more plain HTML. If I wanted to I could add another PHP section after the next line. Let’s try that and get an idea of what our PHP server can do:

```
<html>
<head>
<title>My First PHP Script</title>
</head>
<body>
<h1>Here is my first PHP page,
ever!</h1>
<?php echo “Hello World!”; ?>
<center><h2>I like Christian A/V
Magazine!</h2></center>
<?php phpinfo(); ?>
</body>
</html>
```

That `phpinfo()` is a built in function of PHP. Functions are basically preset commands that are designed to accomplish particular things, often repetitive stuff that can be used with ease instead of having to type all the code again. This function is a simple way to see a whole mess of info on how your hosting company has setup your PHP service.

PHP can do some amazing stuff. But we will start with something simple. Next month we will learn how to randomly show images on our webpage. See you next time...