

# **Building a Sustainable Web Site**

by  
Peter Lavin

September 3, 2003

## Overview

Use PHP, CSS and MySQL to create dynamic web content. This will allow for the separation of presentation and content and the ability to create an economical website that is easily updated and maintained.

## Introduction

We've all seen those stale web pages out there – “Last updated: Feb 02 1999”. Not that you really need to see the date because the presentation itself speaks volumes about when the page was created. Perhaps the authors think that web pages are like wine – the older the vintage the better. More likely though, this is wine that has turned to vinegar. Unfortunately, there are far too many failed web sites. Not only do they clutter up the Internet but they seem to show that web sites are ineffective business tools.

Lack of proper maintenance is the major reason that web sites fail. Information becomes stale and interest and traffic drops off. The high costs and difficulty of updating and changing a site also contribute. Businesses change all the time and those changes need to be reflected in the web site on an ongoing basis. A web site that is stale and static detracts from rather than augments a business. What is needed is a “sustainable” site.

## A Sustainable Site

This article will discuss what we mean by the term “sustainable” and show some ways to achieve this goal. Sites need to be designed with maintenance in mind. With a bit more thought at start-up both the difficulty and the cost of updating can be reduced. The arguments in favour of Cascading Style Sheets (CSS) have been well presented elsewhere but it won't hurt to repeat them briefly here. Use of CSS is a quick and easy method of maintaining and changing the presentation of a site. CSS reduces labour because site-wide changes can be made in one location. The benefits of CSS far outweigh the argument that not all browsers support them. As each day passes, and more users upgrade their browsers, this argument becomes weaker and weaker.

But what about changes to content? We will argue that use of a database is essential to creating a sustainable site. By moving content and links out of HTML pages and into database tables a site becomes much more sustainable. Database examples will be implemented using *MySQL* and server-side scripting will use *PHP*. Some knowledge of these technologies and of HTML and CSS is assumed.

The choice of these technologies is not arbitrary. *PHP* and *MySQL* offer an inexpensive way for small and mid-sized business to create a professional web presence. Your web host does not have to pay licensing fees when using these technologies because they are open-source. For the same reason licensing fees are not paid when a web host runs Apache as the web server on the Linux operating system. Typically these hosts also offer *MySQL* and *PHP*. With these open-source tools annual web hosting costs can be exceptionally reasonable and a high level of functionality can be achieved.

## The Implementation

Every web site has hyperlinks, usually both internal and external ones and often the same hyperlinks appear on different pages. Links change regularly so need to be maintained. Sometimes that means changing the URL they point to or eliminating it altogether. If this information is stored in a database and retrieved whenever a page is loaded we will not need to make any changes to our HTML pages but just update a database table.

Find below a description of what a table containing hyperlinks might look like:

**Table “links”**

Field	Type	Key	Null	Extra
id	int(11)	PRI		auto_increment
url	varchar(255)			
target	varchar(35)		Yes	
description	varchar(255)			
typeid	int(11)			
whenadded	date			

The “id” field is a unique identifier and the primary key. This field is an “auto\_increment” type field. “url” will be the actual address that the link points to either a relative address on the site’s server or an external absolute address. “target” will determine whether a link replaces the current window or opens a separate window. Values allowed here will be the legitimate values for the target attribute, “\_self”, “\_top”, “\_blank”, “\_parent” an empty string or the name given to a frame. “description” will be the text that appears on your page as the hyperlink. “typeid” will allow you to sort your links or restrict selection of them. It may make sense for you to have a related table such as the following:

**Table “linktypes”**

Field	Type	Key	Null	Extra
id	int(11)	PRI		auto_increment

description            varchar(255)

If for example, you want to display external links only and the id for external links is "1" you could restrict selection to only those records where the typeid is 1. Finally, the field "whenadded" is the date on which your link was created in the database. There will be more said about this field shortly.

With your links stored in a database you never hard-code them into your page but rather select them from the database. We will examine the code to do this shortly but for the moment let it suffice to say that if an URL changes you can simply make the appropriate change to the database record and thereby update any reference made to it throughout your web site. There is no need to hunt through HTML pages, find the references and then change them.

## The Code

Find below the PHP code that will retrieve data from your table and dynamically create HTML code in the following format:

```
<a class="new" href="westnile.html" target="_self" >West Nile</a><br />
```

```
<?php
//dynamically create links
//include database password information etc.
$hostname = "myhost.com";
$username = "user";
$password = "password";
$databasename = "mydatabase";
$strsql="SELECT description, url, target, ".
        "IF(whenadded > (SUBDATE(CURDATE(), ".
        "INTERVAL '14' DAY)), 'new', 'info') as cssclass FROM links ".
        "ORDER BY cssclass, UCASE(description)";
$connection = @ mysql_connect($hostname, $username, $password)
or die("Cannot connect to database");
if (! mysql_selectdb($databasename, $connection))
    die("Cannot open table");
if (!$result = @ mysql_query($strsql, $connection))
    die("Cannot select record");
$strlinks="";
//tbllinks url, description, type
while($row = mysql_fetch_array($result)){
//should output in the following format
//<a class="new" href="westnile.html" target=_self >West Nile</a><br />
    $strlinks.="<a class=\"";
    $strlinks.=$row["cssclass"];
    $strlinks.="\" href=\"";
    $strlinks.=$row["url"]."\" ";
    if($row["target"]!= null)
        $strlinks.="target=\"". $row["target"]."\"";
    $strlinks.=">". $row["description"]."\". "</a><br />\n";
}
}
```

```
echo $strlinks;  
?>
```

Save this code in a file called “links.inc” and you can include it wherever appropriate. If you look closely at the code you’ll see that it does more than just retrieve information from the database. The “whenadded” field is used to return a value of “new” if this link was added less than 14 days ago. This value is used to set the “class” attribute of the anchor tag. We can dynamically change the class of an anchor based on when it was added to the database. By using CSS we can give different display attributes to newer tags. Let’s look at how we might define the style for an anchor tag of the class “new”.

```
A.new:link {  
    color: Red;  
    text-decoration: none;  
    background-color : transparent;  
}  
A.new:visited {  
    color: Red;  
    text-decoration: none;  
    background-color : transparent;  
}  
A.new:hover {  
    color: Orange;  
    text-decoration: none;  
    background-color : transparent;  
}
```

Any new links will appear in red and will be first in our list of links because of the “ORDER BY” clause. Additionally, the class name will automatically change when the link is more than two weeks old. This will happen without the intervention of the user or designer. Not only do we get improved maintenance by putting our links into a database table we also get improved presentation with no extra overhead.

## Other Dynamic Content

Just as links may be stored in a database so may web page content. A table such as the following would be useful for organising articles on a web site. The “contents” field could store plain text or HTML.

**Table “contents”**

Field	Type	Key	Null	Extra
id	int(11)	PRI		auto_increment
title	varchar(35)			
description	varchar(255)			

topic	varchar(35)	YES
author	varchar(35)	
contents	text	
whenadded	timestamp(14)	YES

As you might imagine the various fields give us many ways of presenting our material – by author, date, topic etc. Again, in conjunction with CSS, there are even more possibilities.

## How are Records Changed

The question may remain about just how easy it is to change information in the database on your web server. If you separate content into a database and that database is difficult to get at then you're not very far ahead. Fortunately, many web hosts use an open-source tool called "phpMyAdmin". This makes updating your database a fairly easy task. A very user-friendly interface allows easy addition, editing or deletion of records. Tables may be created and changed without any knowledge of SQL.

If your web host doesn't offer phpMyAdmin there are many other tools out there to help you out. At this point I will only mention one such example, *MySQLGraphic*. This is an open-source tool developed by the author of this article. It provides a graphical interface for MySQL and may be used under Linux or Windows. It may be downloaded for free– no strings attached, no trial period, from the URL, <http://www.peterlavin.com/~mysqlgraphic/>.

## Conclusion

Dynamic web content can be created inexpensively using open-source tools and technologies. For a site to be an effective business tool, dynamic content is not just a nice add-on but a requirement. It is a requirement because it is the most effective means of creating a sustainable site.

## About the Author



Peter Lavin runs a Web Design/Development firm in Toronto, Canada. For more information visit <http://www.softcoded.com/>. He may be reached at [peterlavin@sympatico.ca](mailto:peterlavin@sympatico.ca).