

Re: CSS + xhtml static site – going dynamic?

Re: CSS + xhtml static site – going dynamic?

Source: <http://newsgroups.derkeiler.com/Archive/Uk/uk.net.web.authoring/2008-02/msg00043.html>

- *From:* "Roger Moss" <roger.moss(BINTHISBIT)@tiscali.fr>
 - *Date:* Tue, 12 Feb 2008 23:58:48 +0100
-

"Saul" <saul.dobney@xxxxxxxxxx> wrote in message
<news:76fdf1e8-9d8d-48f7-8959-8257948e1055@xx>

On 8 Feb, 11:48, "Roger Moss" <roger.moss(BINTHISBIT)@tiscali.fr>
wrote:

Having got a site up and running against a deadline, using the tried and tested CSS + xhtml static route, we're now growing fast and, as we always knew, about to outgrow this approach quite soon. We plan to go with PHP/MySQL. So far, so good. Our hosting provider already has the apps in place.

My question, then, is this: Since we're very much a content-driven site – specifically an online magazine – can anyone suggest some relevant reading to help us make the transition, please? Here's our URL: www.mountainpassions.com

We have a PHP/MySQL/Apache package set up and running locally, along with a basic grounding book, but what we haven't come across so far is anything which deals with what for us are the key issues of setting up a data-driven/dynamic environment for editorial pages. I guess we're 'untypical'.

Hi Roger,

Very nice site – really like the photographs from one of my favourite areas of France. The answer to your last point about being 'untypical' actually isn't true. More and more sites are content managed even for relatively simple page-driven sites. Essentially it gives the site owner the opportunity to edit and add to their site without having to keep going back to a web-designer – even small businesses need to keep their websites up-to-date and 'fresh', or to tweak them to improve search engine rankings.

I know that you're looking to do this yourself, but as an experiment to see what would be needed I copied your site content over to our CMS

Re: CSS + xhtml static site – going dynamic?

system (which uses PHP/MySQL) to see if there was anything special that would be difficult to replicate and the answer was in pretty much no (it only took about an hour to transfer excluding fancy twiddles and the page skin). There's nothing particular special and lots of different CMS systems would be able to build your site, but going to a full CMS would depend on what compromises you are prepared to make with regards to fitting in with someone else's user-interface, CMS page structure and display constraints. However, you would get something much more powerful than if you build by hand.

To do this yourself, start with the HTML page structure, thinking as if the page is made up of a series of areas, then to fill elements of the structure with the content. So the base page might be structured into header, middle, footer, then in the middle area you would have left-column, mainarea, right-column. You build each area as if it is a series of DIV layers so you can target them with CSS and place the content appropriately.

PHP then fills each content area. This can be simple – for instance using PHP 'include' command to include individual files for each area. However, for this content still has to be created by hand in separate files and does need some management as to what file is included with what page (your sidebars in the main area change regularly with the main content, so this would be hard work to keep organised, but would allow fixed elements to remain fixed across all pages with a single header include for instance).

A better alternative is to have PHP pull the content off a database. There would be a simple HTML form with a textarea for each area that content can go in on the page, this is then passed back to PHP which then saves it to the database. Content can be pulled off as say "select right_column from pages where page=12 or page_name='seasons.html'" in MySQL and put in the correct place on the page using a PHP display_page script. You have to write the PHP editor code to present the author with a form for a page, showing any existing data so it can be edited and which then writes the new/changed form back to the database, and the second page displayer in PHP.

So far this is simple and easy, the downside of this is link management – menus and page linking need to be done by hand. Most people flounder at this point because sites quickly run into link spaghetti – that is as the site grows, creating, maintaining and editing links by hand for pages starts to become harder, particularly adding links to new pages into old content. There are two parts to an initial solution to this. Firstly is to separate page specific content from global site content. So one editable form for site structure which controls site-wide page structure and menus at a global level, then a separate page editor which deals with the page-specific content. When displaying the page you bring the two together – for instance header from the site structure table, content from the page

Re: CSS + xhtml static site – going dynamic?

content table.

However, there will still be link spaghetti because the menu links for each particular page content needs to be different too – not just global 'section' menus.

So the second part is to formally build the linking structure into the database (a linking table) so the database itself knows which page is linked with which (eg which page is a child or parent of what). This then allows dynamic menuing which can be created on the fly so the menu for each page changes appropriately. You just have to pull the list of child or linked pages from the database and place them as a list on the page into an appropriate content place. When you add a new page, you just need to link it appropriate in the linking table and menus will adapt automatically.

In broad terms you would be finished here for something simple – a global page structure table, a content table for each page's specific content and a linking table between pages in MySQL with an PHP editor/creator form for each of these three and a PHP display page script bringing the elements together to make a page.

This is pretty rudimentary. You don't have much in the way of bells and whistles to make site building that bit easier and more flexible – no image uploading control for instance, dynamic database driven URL rewriting, wysiwyg editors etc and you will find maddening constraints that you'd want more control over. For instance it's quite possible you'll need multiple menuing structures built dynamically, or would like the summary of article features pulled through automatically. You would have to take decisions as to where to draw the line between the flexibility and development.

In addition, looking at your current site I wasn't sure if it was designed to be that expandable. Even with simple content managed systems, the number of pages in the site increases rapidly. For instance our users typically start thinking they can manage with 10–12 pages then quickly find adding stuff so easy that they quickly run up to 100+ pages. So what happens when you have 30 or 40 articles on properties, or need an extra section on food because this then has enough content for a section all of it's own? How will the user navigate the site structure or menus with these extra pages? As an on–line magazine are you expecting to be in editions (Q1 2008, Q2 2008, Q3 2008 etc), or just regularly adding articles which then appear on the front page – or do you use a combination so it can be seen by edition and by section? Some of these structural aspects can be built into the database design if you start to think about them now.

Saul
www.notanant.com

Re: CSS + xhtml static site – going dynamic?

communities of websites
websites for communities

Now that's what I call a considered response – just what I need to get a clearer picture of where we need to think about going from here.

And an object lesson in the value of newsgroups...

Thanks for your efforts, and your kind comments on our work; you can see why it's a labour of love!

RM

.