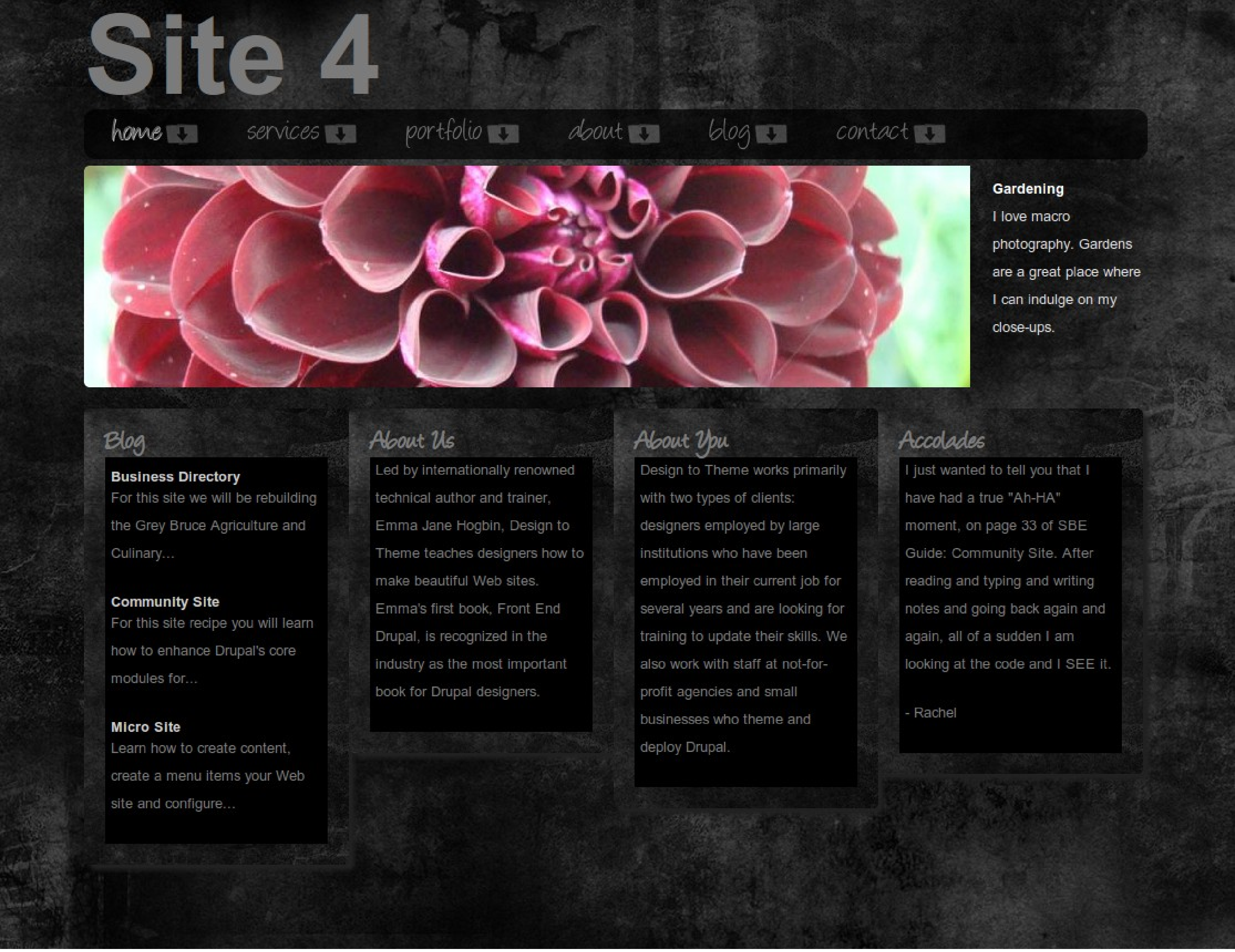


Site 4 Workbook: Portfolio
Site Building Extravaganza

By Emma Jane Hogbin

440-13th Street East

Owen Sound, Ontario

N4K 1W6

www.designtotheme.com

emma@designtotheme.com

License

About this Workbook

This workbook was created by Drupal expert Emma Jane Hogbin (www.designtotheme.com). It is the result of years of experience and many hours of work. It is dedicated to making this topic easier for you to quickly master—so that you can create a more profitable experience with Drupal. That said, mistakes sometimes sneak into workbooks. If you've found an error—please let me know by email.

This workbook was produced by Design to Theme as part of a year-long training program in 2011. Content was originally released to the participants of the program. If you were lucky to get this copy for free, we encourage you to consider how valuable it was to you and send that amount of money via PayPal to emma@designtotheme.com.

You are not allowed to sell this workbook or extract parts of it to create "derivative" works. (I've already nailed one person who tried selling my stuff on his site without permission. Don't be that guy. Ok?) You are encouraged to share the work digitally with others for free so long as the entire PDF (including all the pages) are included. You are also welcome to share print copies of the document with others—again only the *entire* document may be shared.

If you're a **Drupal trainer** and you're interested in using this workbook in your classes: get in touch to find out more about **licensing this workbook with your own logo for your next workshop**.

We hope you enjoy this workbook. If you do, tell the world. If you don't, tell Emma!

Table of Contents

Site Profile

As a Web designer, one of the obvious self-promotion tools you'll need is a portfolio Web site. This style of site is also appropriate for creative businesses such as photographers, hair dressers illustrators and wood workers.

This site is all about the visual display of lists. The functionality that we'll be building out in this site includes:

- Customized blog, including views for blog home page, RSS feed, monthly archive block, latest news block, all tags and most popular tags.
- Custom content type for portfolio items.
- Rotating banner of portfolio items.
- Custom display of portfolio items including carousel and light box.

Specific theming features that we'll be adding to the site include:

- Using the Drupal UI to customize jQuery effects.

Topics that won't be covered:

- Installing Drupal. Instructions are also available online at http://drupal.org/documentation/install
- Installing contributed modules (i.e. getting contrib modules onto your Web server). Instructions are available online at http://drupal.org/documentation/install/modules-themes/modules-7

The build for this site can be accomplished entirely with point-and-click configuration; however, if this is your first exposure to Views, you may choose to use the exports of the Views included in this package instead of configuring each View by hand.

Build the Site

Here are the basic elements we will be creating and configuring for this site:

1. Contact form.
2. Content pages. Includes the “About” and “Services” page.
3. Blog. Requires custom views.
4. Portfolio. Requires custom content type and custom views.
5. Rotating banner on the front page.

Note: PHP Errors, Warnings and Notices

Drupal 7 is still pretty new. There has been only one release since its initial launch, and contributed modules are still relatively "fresh." A very common problem is that developers turn off non-essential messages when doing their work. Many servers, however, have a more liberal error message policy. As a result we see a lot of warning messages that developers don't see.

If you get a big red "error" message, read it carefully. ***Warnings*** *and* ***Notices*** *can almost always be ignored (unless you want to submit a bug report on Drupal.org, which makes you awesome).* ***Errors****, on the other hand, are typically a real problem that should be investigated.*

These "notices" can be safely ignored if they occur within the administrative pages. Log out of your site and navigate to the pages where the errors appeared to see if the errors will be visible to the public.

Install Drupal

The first step in building a Drupal site is to ... install Drupal. Instructions are covered in the Installing Drupal mini guide. If you are new to Drupal go ahead and start with a fresh install (not a multisite installation). Make sure the folder `sites/default` is readable and writable by the Web server. You will get errors later on if you don't do this now. Refer to the Install Guide instructions for more information.

Once you've installed Drupal, log into your site as the administrative user you created when you installed Drupal.

Contributed Modules

Install the latest Drupal 7 version of each of the following contributed modules. In order of preference look first for a module that's Recommended (green). If there isn't a Recommended release available look for Other releases (yellow) and finally Development releases (pink):

- **CTools**. http://drupal.org/project/ctools (required by Views)
- **Views**. http://drupal.org/project/views
- **Advanced Help**. http://drupal.org/project/advanced_help
- **Token**. http://drupal.org/project/token (required by Pathauto)
- **Pathauto**. http://drupal.org/project/pathauto
- **Link**. http://drupal.org/project/link
- **Date**. http://drupal.org/project/date
- **Libraries**. http://drupal.org/project/libraries (required by Gallery Formatter)
- **Gallery Formatter**. http://drupal.org/project/galleryformatter
- **Colorbox**. http://drupal.org/project/colorbox
- **EPSA Crop**. http://drupal.org/project/epsacrop
- **Views Slideshow**. http://drupal.org/project/views_slideshow

If you don't have any content of your own to add to this month's site, but you want to see it fleshed out, download and install the module Devel (http://drupal.org/project/devel) as well. Instructions were included in Site 2 on how to use this module.

Tip: CTools and Views are related and are released in paired updates. If you have an older version of CTools (from a previous site) be sure to get an updated version of CTools.

Installing Contributed Modules

1. *Navigate to the project page on drupal.org for the module you wish to install.*
2. *Copy the download link for the latest Drupal 7 version of the module by right-clicking on the link and choosing "copy link location."*
3. *In your Drupal installation navigate to: Administration > Modules.*
4. *Click on the link "Install new module." If this link is not available, enable the core module **Update manager** from this page. The "Install new module" link should appear once Update manager is enabled.*
5. *Into the URL field, paste the download link you copied in Step 2 from the module's project page.*
6. *Scroll to the bottom of the screen and click, "Install."*
7. *Enter your authentication information for your Web server. This is your FTP username and password—not your Drupal administrative account information.*
8. *Click "Install module." Assuming there were no errors during this process, you should now be able to enable your module.*

Your module (and any installed dependencies) should now be available to be enabled at the appropriate time.

Contact Form

A portfolio of your work isn't much good for soliciting new business if potential customers don't know how to get in touch!

Task Summary

Configure the contact form as described in the sections Configuration, Permissions, Menu and Related Block.

Enable the Contact Module

1. Navigate to **`Administration`** > **`Modules`**.
2. Locate and enable the check box beside the core module Contact.
3. Scroll to the bottom of the screen and click **`Save configuration`**.

Configuration

Once the Contact module is enabled, it must be configured before it can be used. By default one category has been added to your contact form. You will need to add the recipient(s) email address(es) for this form.

1. From admin dashboard navigate to **`Structure`** > **`Contact form`**.
2. Next to the category **`Website feedback`** click the link labelled **`edit`** and configure the following settings:

- **`Category`**. This is only displayed if you have more than one category.
- **`Recipients.`** You may add one, or many, email addresses. These can be any address you like. For example: Hotmail or Gmail email addresses. Or even the email address for a discussion mailing list.
- **`Auto-reply`**. Enter the text you want to send to everyone who completes this form.
- **`Weight`**. If you have many categories on your Web form, you can choose the order in which they appear in the drop-down menu. By default all categories are given the same ranking and are sorted alphabetically.
- **`Selected`**. If you have more than one category, you may choose the default with this toggle.

Your contact form is now configured. You may add as many categories as you need. For example, you may wish to add categories for: Website feedback, Sales, Customer support.

Permissions

You must now allow people to *use* the contact form by adjusting the permissions for this module:

1. From the admin dashboard, navigate to **`People`**.
2. Click on the tab labelled **`Permissions`**.
3. Scroll to the module labelled **`Contact`**.
4. Next to the permission **`Use the site-wide contact form`** enable the check boxes for **`Anonymous user`** and **`Authenticated user`**.
5. Scroll to the bottom of the screen and click **`Save permissions`**.

Menu

Now you must create a link to your contact form so that people know where to find it.

1. Using the admin dashboard, navigate to **`Structure`** > **`Menus`**.
2. Next to the **`Navigation`** menu click the link labelled **`list links`**.
3. Next to the link labelled **`Contact`** click **`edit`**.
4. Enable (select) the check box labelled, **`enabled`**.
5. Change the **`Parent link`** to *`Main menu`*. You will need to scroll to the top of the select list within the select box.
6. Scroll to the bottom of the configuration screen and click **`Save`**.

A summary of all links in the main menu is now visible.

Order of List Items in Menus

When you add a new menu item it doesn't always appear in the correct position within a menu. Use the following instructions to adjust the position of your menu items.

1. Using the admin dashboard, navigate to `Structure` > `Menus` > `Main Menu`.
2. Adjust the order of these links by dragging and dropping them into the correct order.
3. Scroll to the bottom and click `Save configuration` to lock in your changes.

Related Content Block

Add your mailing address, phone number or a message to the contact form with a custom block:

1. Using the admin dashboard, navigate to `Structure` > `Blocks`.
2. Click on the link labelled `Add block`.
3. Enter a Block description (used in the admin section only) and the text that you want to appear on the Contact page (this goes into the Block body).
4. Scroll to the fieldset labelled `Region settings`. Place your block into the Sidebar first region.
5. Scroll to `Visibility settings`. Click on the vertical tab for `Pages`. Enable the option for `Only the listed pages`. In the large text area below the option you just enabled, type `contact`. This will limit the block to appearing only on pages with the URL *contact*.
6. Scroll to the bottom and click `Save block`.

Your new message should now appear on the Contact form. Close the overlay and navigate to your contact form to check your work.

Basic Content Pages

For this portfolio site you will use non-hierarchical / single pages for the About and Services page.

Basic pages have the following anti-features:

- Is not published to the front page of your Web site.
- Does not allow visitors to post comments.
- It does not have tagging enabled.
- It does not have an image upload widget.
- Is not date stamped.

In other words: it's just a basic page.

Automatic URL Aliases for SEO

We're going to make very search engine-friendly URLs with the module Pathauto (and its helper Token). Before proceeding with these instructions, make sure the module Pathauto is enabled.

1. Navigate to **`Administration`** > **`Configuration`** > **`URL aliases`** > **`Patterns`** (tab at the top).
2. Enter the following patterns:
   - **`Default path pattern`**: `[node:title]`
   - **`Pattern for all Basic page paths`**: `[node:title]`
3. Scroll to the bottom of the screen and click **`Save configuration`**.

Enabling a Module

1. Navigate to Administration > Modules.

2. Locate the module you want to enable.

3. Enable the check box beside the module name.

4. Scroll to the bottom of the screen and click "Save configuration."

Adding an Image Field

At this point you cannot attach images to your content. Use the following instructions to adapt the basic page content type:

1. Navigate to **`Administration`** > **`Structure`** > **`Content types`**.
2. Next to **`Basic page`** click on **`Manage fields`**.
3. Beneath the label **`Add existing field`**, enter the following values:
    1. **`Label`**: type in the label name `Image`
    2. **`Field to share`**: select *`Image: field_image (Image)`*.
    3. **`Form element to edit the data`**: leave as default (*`Image`*).
4. Click **`Save`**. The configuration window for this field will open.
5. The default values are acceptable for your new image field. Scroll to the bottom and click **`Save settings`**.

You will be returned to the list of fields for the basic page content type.

Task Summary

Alter the settings for the content type basic page to include one image field. Adjust the image style for the image field so that the medium-sized images are displayed.

Adjusting the Image Style

It's now time to adjust the display settings for your image so that you don't get huge pictures taking over your Web site display.

1. Navigate to **`Administration`** > **`Structure`** > **`Content types`** > **`Basic Page`** > **`Manage display`**.
2. Next to the field **`Image`**, adjust the following settings:
    1. **`Label`**: Set to *<`Hidden`>*.
    2. **`Format`**: Leave as the default value *`Image`*.
3. At the end of the row, click the cog icon to change the format settings for your image field. A new set of configuration options will appear. Adjust the settings as follows:

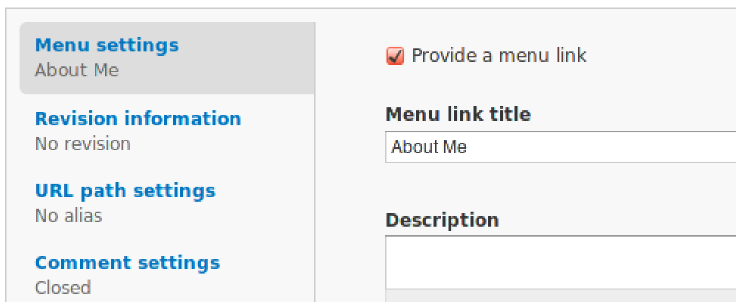
    1. **`Image style`**: Choose the image style *`medium`*.
    2. **`Link image to`**: Leave as the default value *`Nothing`*.
4. Click the button labelled **`Update`**.
5. Click the button labelled **`Save`**.

You may apply this image style to content types and views (and anything else that allows you to specify an image style).

Create a Basic Page

Use the following steps to add a new page to your Web site.

1. From the shortcut menu, click the link **`Add content`**. An overlay will appear.
2. Click on **`Basic page`**.
3. Enter a **`Title`** and **`Body`** for your page.
4. Click on the vertical tab for **`Menu settings`** and enable the check box beside, **`Provide a menu link`**.



5. Add a **`Menu link title`**.
6. Ensure the **`Parent item`** is set to *`Main menu`*.

   The **`Weight`** determines the order of menu items within the main menu. Leave the default setting.
7. Scroll to the bottom of the screen and click the button labelled **`Save`**.

Your **node** has been created and is listed in the main menu.

Task Summary

Use the instructions in this section to create two new pages:

1. About.

2. Services.

Ensure these two pages are available from the main menu and have one image each.

What is a node?

A node is a piece of content in Drupal, typically corresponding to a single page on the site, that has a title an optional body, and perhaps additional fields. Every node also belongs to a particular [content type](), and can additionally be classified using the [taxonomy]() system. Examples of nodes are polls, stories, book pages and images.

Source: http://drupal.org/glossary

Order of List Items in Menus

When you add a new menu item it doesn't always appear in the correct position within a menu. Use the following instructions to adjust the position of your menu items.

1. Using the admin dashboard, navigate to `Structure` > `Menus` > `Main Menu`.
2. Adjust the order of these links by dragging and dropping them into the correct order.
3. Scroll to the bottom and click, `Save configuration` to lock in your changes.

Blocks

In this site design blocks appear in the footer and in the first sidebar (on the left) of the design. Most of your work should already be finished from the site build section of this guide including blocks for portfolio navigation, block navigation and contact information.

Task Summary

Create three custom blocks and place them into the footer region of your new theme.

Custom Blocks

You will need three custom blocks for this design. The blocks in the footer are optional, but there's room for four if you want 'em. I made custom blocks for the following blocks:

- About Us block. Appears in the region Footer of every page.
- About You block. Appears in the region Footer of every page.
- Accolades block. Appears in the region Footer of every page.

In addition to these three blocks, a list of recent blog posts also appears in the footer. To add your custom blocks to the footer, complete the following instructions:

1. Using the admin dashboard, navigate to **`Structure`** > **`Blocks`**.
2. Click on the link labelled **`Add block`**.
3. Enter a Block description (used in the admin section only) and the text that you want to appear in the block (this goes into the Block body).
4. Scroll to the fieldset labelled **`Region settings`**. Place your block into *`Footer`* region.
5. Scroll to the bottom of the configuration screen and click **`Save block`**.

Your new block should now appear on every page. Close the overlay and navigate to your contact form to check your work.

Conditional Blocks

All of the blocks used for sub-navigation should already be built and placed in your site. For reference purposes only, here are the steps to complete if you only want to display a block on some pages:

1. Using the admin dashboard, navigate to **`Structure`** > **`Blocks`**.
2. Locate the block you want to alter and click the link labelled **`configure`**.
3. Scroll to the fieldset labelled **`Region settings`** and choose the region name for the theme you are working with. For example: Port, First sidebar *or* Bartik, Sidebar first.
4. Scroll to **`Visibility settings`**. Click on the vertical tab for **`Pages`**. Enable the option for **`Only the listed pages`**. In the large text area below the option you just enabled, enter the Drupal paths for the pages you want the block to appear on.
5. Scroll to the bottom and click **`Save block`**.

Disabling Blocks

This Web site is not a community site. You are the only person who will need to login. Discourage people from attempting to login by hiding the login block. You will still be able to login by navigating to http://yoursite.com/user. This login URL, `/user`, is standard for all Drupal installations.

1. Navigate to **`Administration`** > **`Structure`** > **`Blocks`**.
2. Locate the block User login and change the drop down menu for the region to *<none>*.
3. Scroll to the bottom of the page and click **`Save blocks`**.

You may also choose to disable the “Powered by Drupal” block. The sample site has disabled this block.

Note: If you are working from an early version of this workbook you may have created a view for “All tags.” We don't need it in the final design after all. Go ahead and disable the block using the steps outlined above.

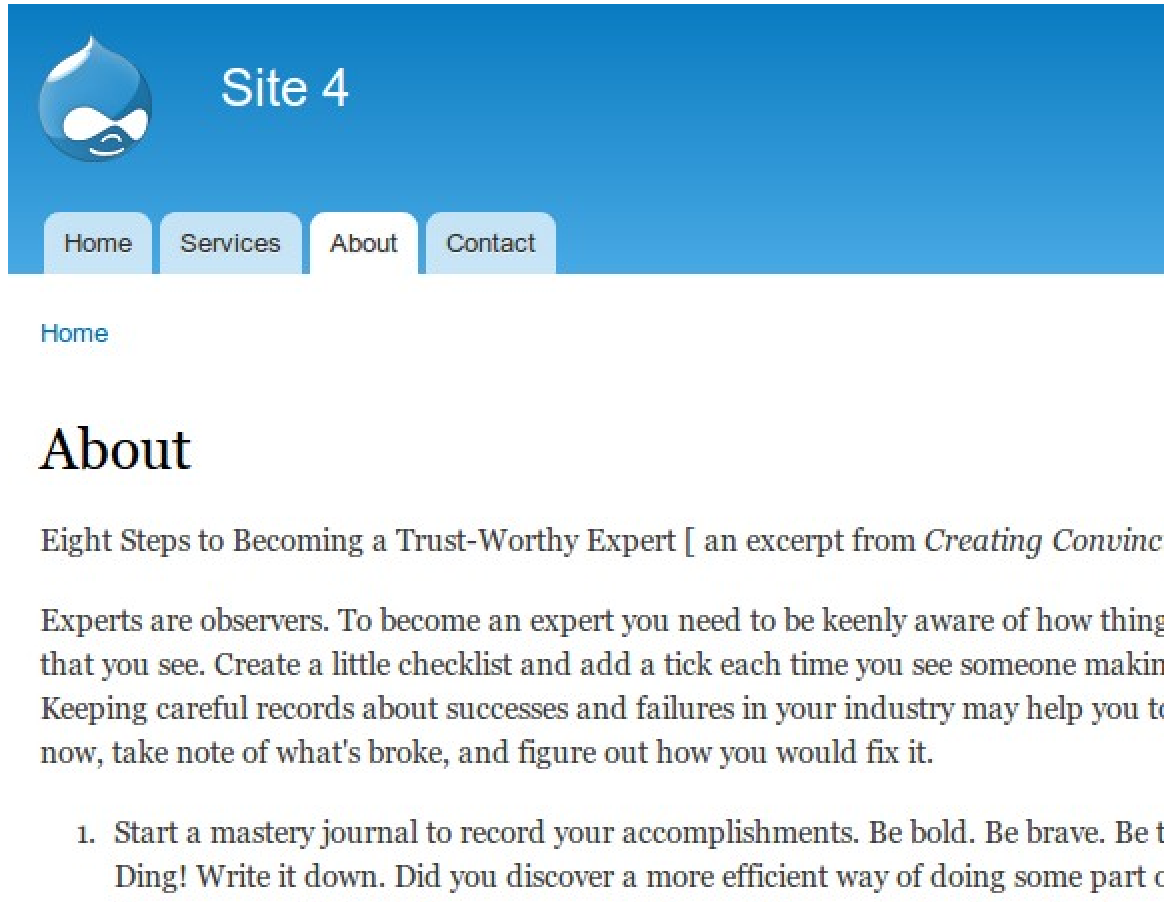
Task Summary

Disable the following blocks: Login and Powered by Drupal.

Check Your Work

At this point you should have the following in your new Web site:

- Four tabs in the main menu (Home, Services, About, Contact).
- Two basic pages with one image each (About and Services).
- A contact form.
- A contact block with your address and phone number visible only on the contact page.
- There is no content available from the home page.
- The login block should not be visible when you are logged out of the site.



About page on your new site

Blog

Out of the box, WordPress excels at creating the style of blog we expect to see on the Web today. Drupal requires more configuration to get the same widgets displayed. While it's a bit of a hassle, the advantage is that you can customize (almost) everything about your blog with a few clicks of the mouse.

Image Styles for Articles

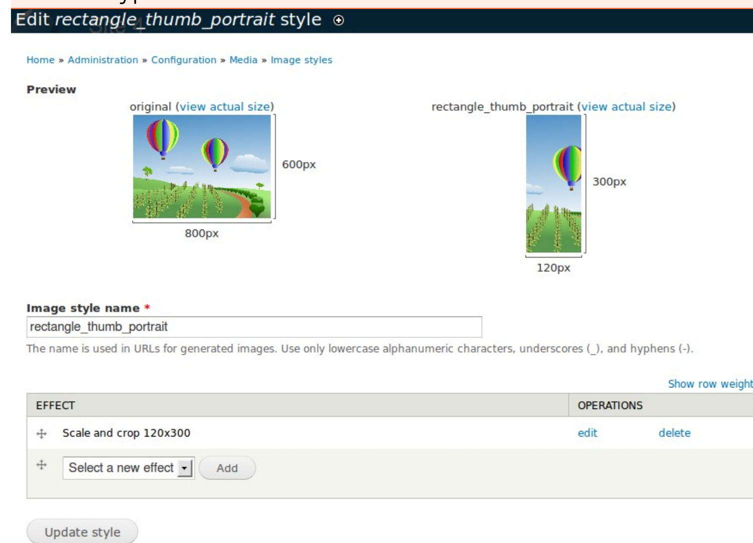
In the finished design the blog posts will have an image which stretches down the side of the entry. We'll go ahead and create that image style now and apply it to the content type articles. This step can be done at any time.

1. Navigate to **`Administration`** > **`Configuration`** > **`Image styles`** > **`Add style`**.
2. Enter the **`Style name`**: `rectangle_thumb_portrait`.
3. Click **`Create new style`**.
4. At the bottom of the screen in the effects table, change the drop down menu from *`Select a new effect`* to *`Scale and Crop`*.
5. Click **`Add`**.
6. For the **`Width`**, enter: `120`.
7. For the **`Height`**: enter `300`.
8. Click **`Add effect`**.

Your new image style has been created. You may close the overlay window.

Task Summary

Create a new portrait-shaped Image Style and apply it to the content type Article.



Display Settings for Articles

Use the newly created image style for the default display of blog entries (Articles).

1. Navigate to **`Administration`** > **`Structure`** > **`Content types`** > **`Article`** > **`manage display`**.
2. On the row for the **`Field Image`**, under the column **`Format`**, change the drop down

menu from *Image* to *Colorbox*.

3. On the row for the **Field Image**, click the cog at the end of the row. A new set of configuration options will open.
4. Change the **Node image style** to *rectangle_thumb_portrait*.
5. Click the button labelled **Update**.
6. Click the button labelled **Save**.

Your new image style has been applied to the content type article. Any nodes that have already been created will display the new image style.

Automatic URL Aliases for SEO

We're going to make very search engine-friendly URLs with the module Pathauto (and its helper Token). Make sure these two contributed modules (Pathauto and Token) are installed and enabled before proceed.

4. Navigate to **Administration** > **Configuration** > **URL aliases** > **Patterns** (tab at the top).
5. Enter the following patterns:
   - **Content paths** > **Pattern for all Article paths**: `blog/[node:created:custom:Y]/[node:created:custom:m]/[node:title]`
   - **Taxonomy Term paths** > **Default**: `[term:vocabulary]/[term:name]`
   - **Taxonomy Term paths** > **Pattern for all *Tags* paths**: `blog/[term:vocabulary]/[term:name]`
3. Scroll to the bottom of the page and click, **Save configuration**.

You now have shiny, search engine-friendly URLs!

URL Parameters

Pathauto takes advantage of the module Token to create variable URL patterns. As of Token 2.0 the module has streamlined some of its variable names. Where you used to be able to use the short hand [yyyy] to mean four-digit year, we now use [node:created:custom:Y]. This variable can be broken into:

`node:created` - the date the node was created on

`custom:Y` - display only the year.

It's a little bit more typing, but a lot faster for Drupal to process. Speed is good...even when it's a little bit complicated.

Create Blog Posts

Use the following steps to add at least 10 blog posts to your Web site.

1. From the shortcut menu, click the link **`Add content`**. An overlay will appear.
2. Click on **`Article`**.
3. Enter a **`Title`** and **`Body`** for your blog post.
4. Enter at least one **`Tag`** for your blog post. This is an auto-complete list of existing taxonomy terms. You can add any new tags, or re-use existing tags. Later we will create a view of popular tags, so be sure to re-use at least a few of your tags. For example: Design, Code, Drupal.
5. Add one **`Image`** to your blog post. This image will be automatically resized to match the image style you created earlier; however, the file stills need to be smaller than the maximum upload size (usually 2MB by default). You may click the button **`Upload`** now, or simply save the page to upload the image.
6. When you are finished creating this page, scroll to the bottom of the screen and click the button labelled **`Save`**.

Task Summary

Create at least 10 blog posts each with an image and at least one tag.

Custom Lists of Content

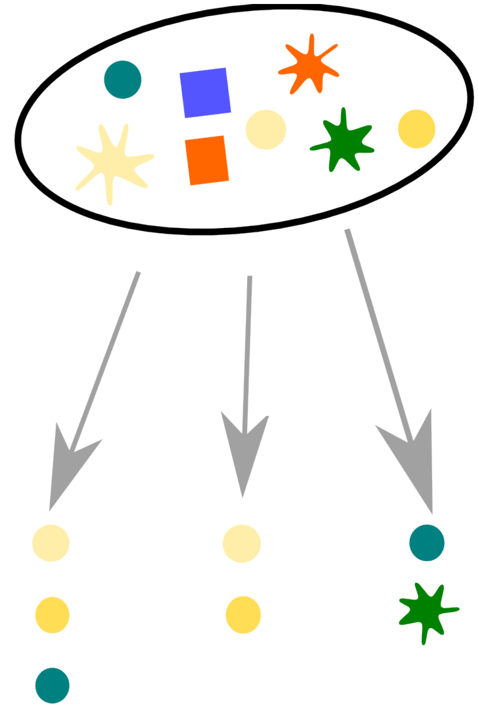
At its heart a blog is a list of nodes created from the content type Article with several custom lists used as entry points into the blog. We will use the Views module to create four custom lists of content:

1. Recent Blog Entries.
2. Monthly Archive.
3. Popular Tags.
4. All Tags.

Make sure the following contributed modules are installed and enabled before proceeding:

- Views, Views UI
- CTools
- Advanced Help

Information on installing these modules is available earlier in this workbook in the section labelled, “Contributed Modules.”



The Views module builds lists of content. In this example three custom groups have been created:
1. Only circles selected.
2. Only yellow circles selected.

Enabling a Module

1. *Navigate to Administration > Modules.*
2. *Locate the module you want to enable.*
3. *Enable the check box beside the module name.*
4. *Scroll to the bottom of the screen and click “Save configuration.”*

View: Recent Blog Entries

A custom view which lists the titles of the most recent 10 Articles, sorted in reverse chronological order.

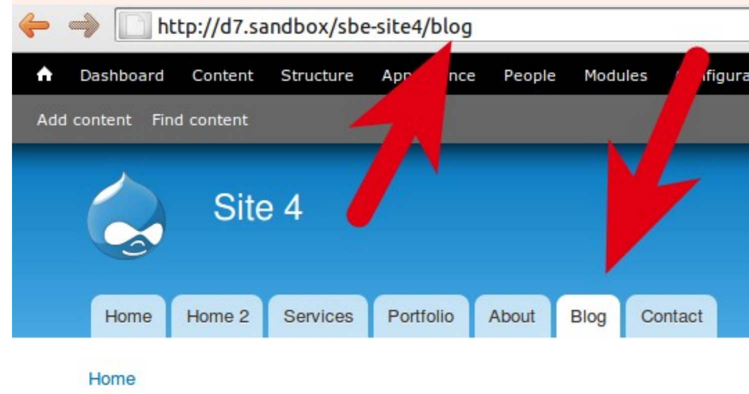
1. Navigate to **Administer** > **Structure** > **Views**.
2. At the top of the page, click on the link **Add new view**.
3. A new screen will appear where you can enter the basic settings for your new view. Use the following settings:
    1. **View name**: Blog
    2. In the mad-libs fill in the blank, alter the sentence to read: Show `Content` of type *Article* tagged with <leave blank> sorted by *Newest first*.
    3. Leave the box selected for **Create a page** and change the following settings.
        1. **Path**: blog
        2. **Display format**: *Unformatted list* of *[full posts] [with links] [without comments]*.
        3. **Items to display**: 10
    4. Enable the check box for **Create a menu link**. Leave the defaults of **main menu** and link text Blog.
    5. Enable the check box to **Include an RSS feed** and leave the URL as blog.xml.
    6. Enable the check box to **Create a Block**.
        1. Leave the **Block title** as Blog.
        2. Change the **Display format** to *HTML list* of *titles (linked)*.
        3. Change **number of items per page** to 10.
4. Click **Save and exit**.

You are now finished configuring this view. The blog is available from the main menu. The feed icon is activated. The block display, however, still needs to be assigned a region.

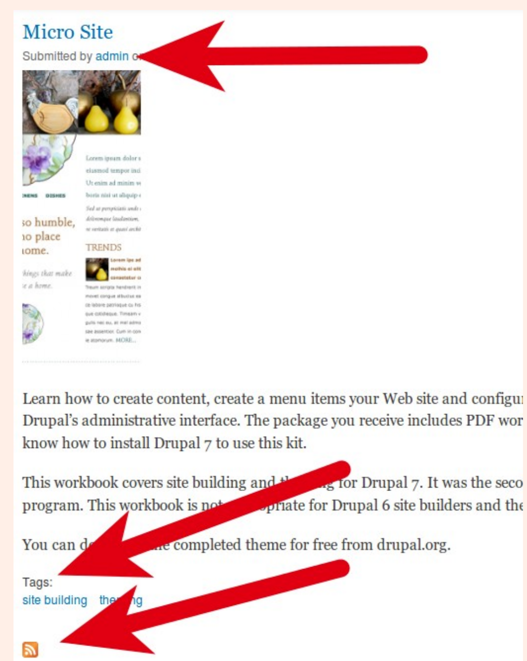
Task Summary

Create the recent blog entries view as described. When completed, it will look like this:

URL and Menu item:



Full post and the feed icon:



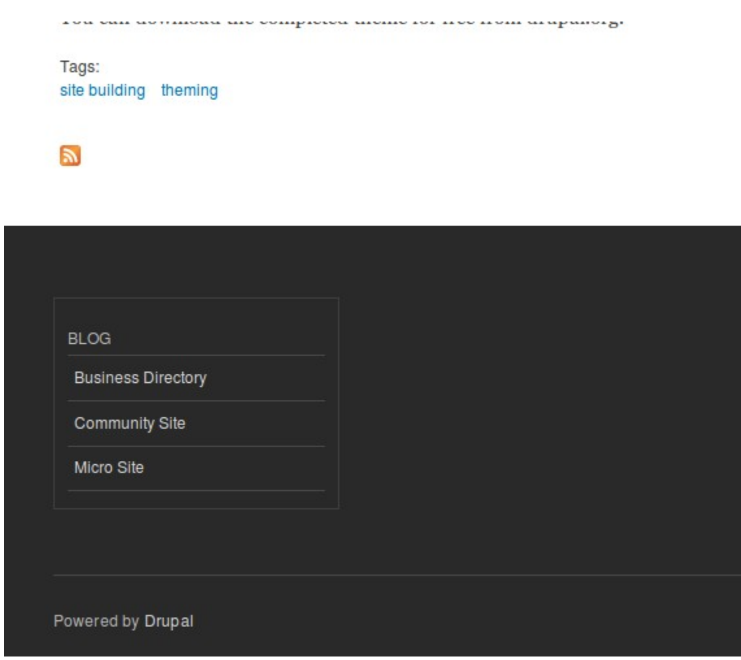
Assigning a Region to Your Views-Generated Block

1. Using the admin dashboard, navigate to **`Structure`** > **`Blocks`**.
2. Locate the block created by your view. It is named: **`View: Blog`**.
3. Change the drop down menu for the region from *`-None-`* to the name of the region you want the block to appear in: *`Footer first column`*.
4. Scroll to the bottom and click **`Save blocks`**.

The list of the ten most recent blogs titles will now appear in the footer of your site.

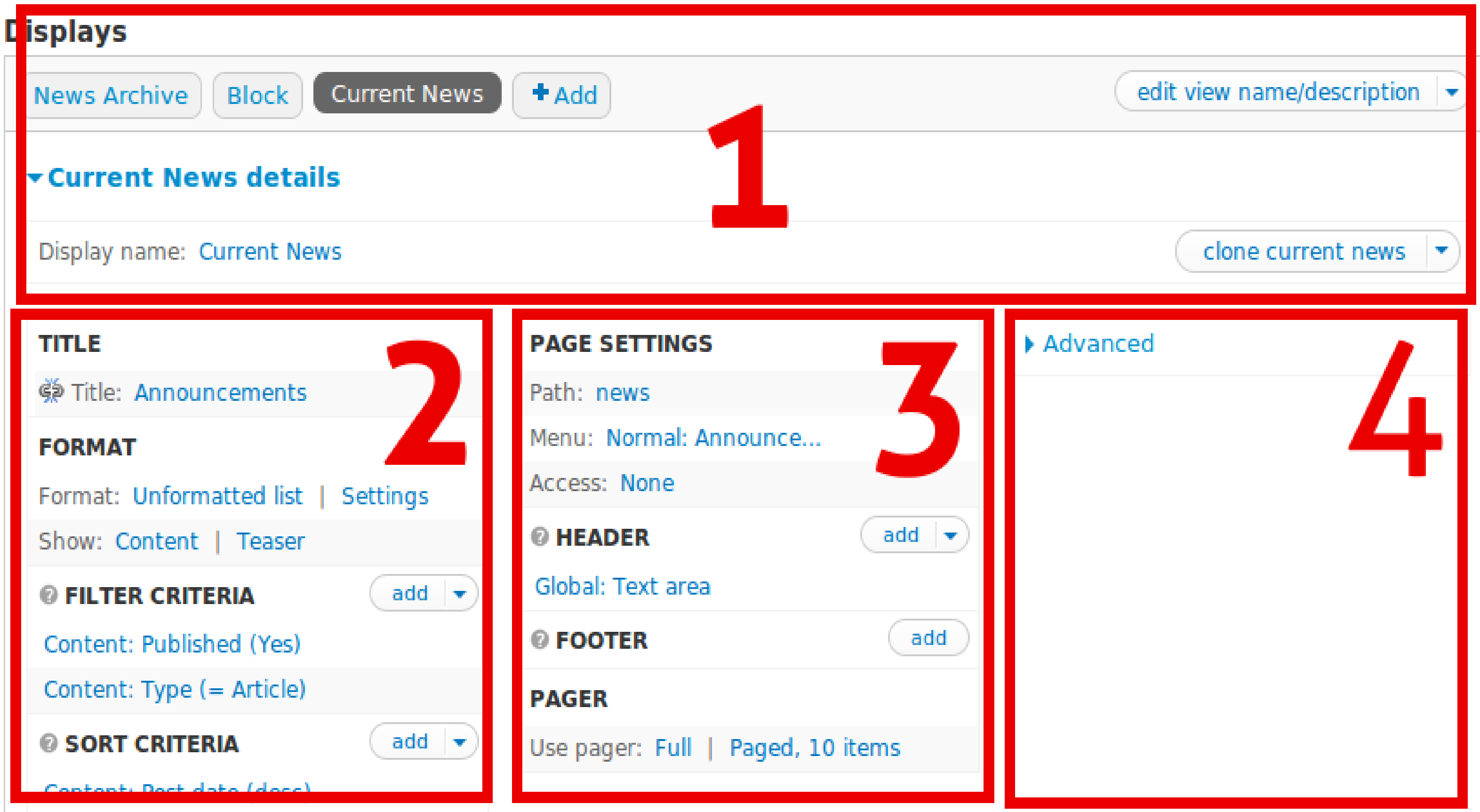
Task Summary

Place the block display into the footer of your Web site. When completed, it will look like this (although mine only has 3 blog entries listed):



Views Configuration Screen

Let's take a closer look at the configuration screen (below) you'll come to know and love.



The configuration screen has four basic areas:

1. **Meta Data** (Across the top): display selection and meta information for the view.
2. **Basic Settings**: Title, Format, Filter, Sort.
3. **Display-Specific Settings**: Path, Menu, Header, Footer, Pager
4. **Advanced Settings**. Contextual Filters, Relationships, No results behaviour, Exposed form, Other. This fieldset is closed by default.

View: Monthly Archive

Before creating your first view, make sure you have some Articles. If you don't have any, your view won't do anything and it will make you sad. Or frustrated. Or both. So please make sure you have some Articles in your site before proceeding.

We will use a preconfigured view this time:

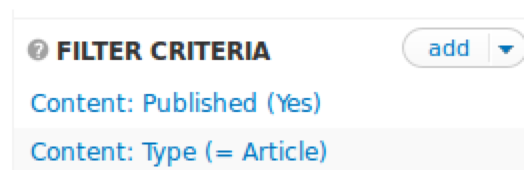
1. From the administrative toolbar, navigate to **`Structure`** > **`Views`**.
2. Scroll down to the greyed out view labelled **`Archive`**.
3. Under the column **`operations`**, select the link **`clone`** for the archive view (it may be part of a drop down menu). This will create a new copy of the view.

You will now be able to proceed through the Views wizard to create your first view.

1. Enter the title for your view: `Monthly Archive`.
2. Click **`continue`**.

You are now on the views configuration screen.

Add a Filter to Display Only Articles



1. Scroll to the section labelled **`Filter criteria`**. Click the drop-down menu labelled **`add`**. A new modal configuration window will appear.
2. In the search box type in the word `type`. The list will be automatically refreshed.
3. Enable the check box beside the option **`content: type`**.
4. Click the button labelled **`Add and configure filter criteria`**.
5. Under the list of content types enable the check box beside **`Article`**.
6. Click the button labelled **`apply`**. The modal window will close.
7. At the top of the configuration screen click **`Save`**.

Task Summary

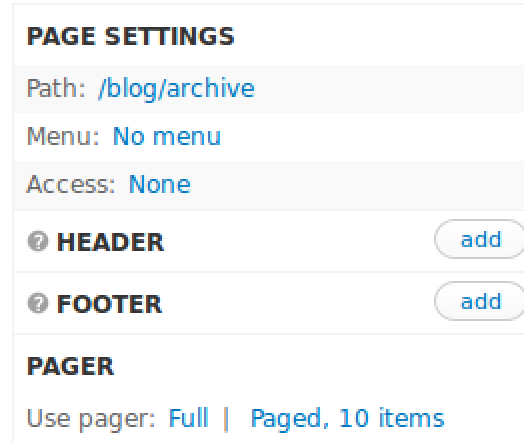
Create the monthly archive view block. Display the block in the first sidebar on all blog pages (including the summary view).

Why We Clone

If you make a mistake, or want to create multiple archives, the original will be preserved. This allows you to clone a new view and reuse the original settings over and over again.

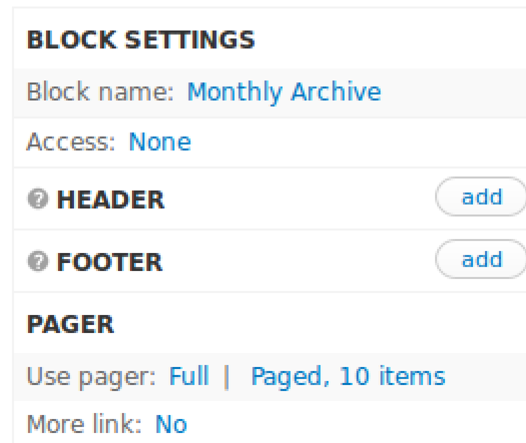
Configure the Page Display

1. Below the heading **`Displays`** at the top click on the tab labelled **`page`**. The configuration screen will refresh.
2. Below the heading **`Page settings`** click on **`Path: archive`**. A modal configuration window will open.
    1. Change the path to: `blog/archive`
    2. Click the button labelled **`apply`** (the modal window will close).
3. At the top of the configuration screen click **`Save`**.



Configure the Block Display

1. Below the heading **`Displays`** at the top click on the tab labelled **`block`**. The configuration screen will refresh.
2. Below the heading **`Block settings`**, click on the text beside **`Block name`**. A modal window will open.
3. In the **`Block admin description`** text input, enter: `Monthly Archive`.
4. Click the button labelled **`Apply`**.
5. At the top of the configuration screen click **`Save`**.



You have now created your view. To see it in action, go to http://yourwebsite.com/blog/archive. There is no link to the view, so you will have to manually type in the URL for it. The block for this view has not been placed into your site.

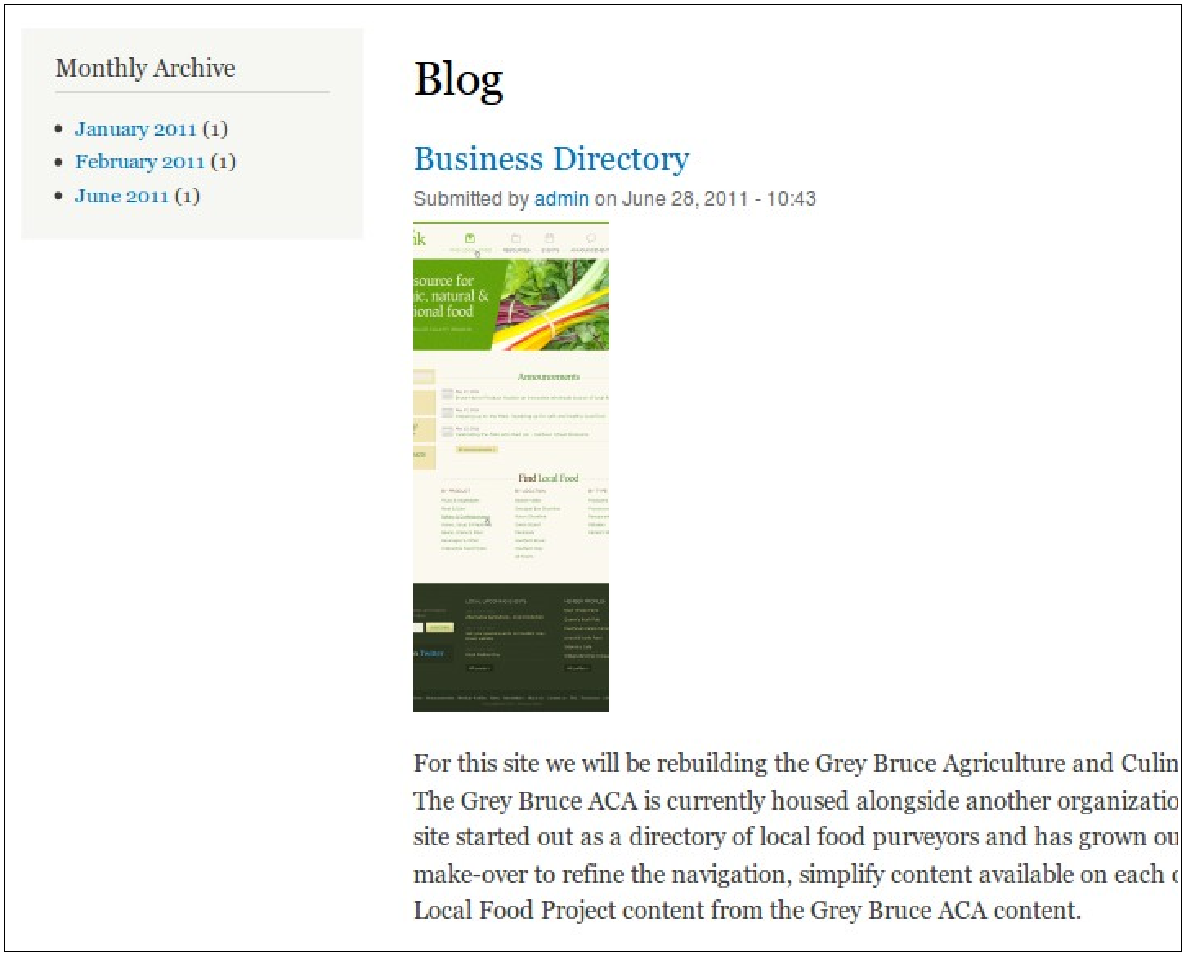
Monthly Archive Block

This block should appear only on blog-related pages.

1. Using the admin dashboard, navigate to **`Structure`** > **`Blocks`**.
2. Next to the block **`Monthly Archive`**, click the link labelled **`configure`**.
3. Scroll to the fieldset labelled **`Region settings`** and select from the Bartik drop down menu select *`Sidebar first`*.

4. Scroll to **`Visibility settings`**. Click on the vertical tab for **`Pages`**. Enable the option for **`Only the listed pages`**. In the large text area below the option you just enabled, enter the following two lines:
   ```
   blog
   blog/*
   ```
   This will limit the block to appearing only on pages with the URL *blog*.
5. Scroll to the bottom and click **`Save block`**.

Check Your Work: Monthly Archive



View: Popular Tags

This view ranks the taxonomy terms by the number of entries which are tagged by each term. Unlike the view which lists all tags, this view starts with the content and uses a summary to count how many nodes are tagged with each taxonomy term.

1. Navigate to **`Administration`** > **`Structure`** > **`Views`** > **`Add new view`**.
2. Complete each of the fields as follows (if the field is omitted, use the default):

Field	Value
View name	Popular Tags
Show	Content
Of type	Article
Sorted by	Title
Create a page	Enabled
Path	blog/popular-tags
Display format	HTML list [of] titles linked
Create a block	Enabled
Display format	HTML list [of] titles linked
Items per page	10

This table of values assumes you have Javascript enabled. If you do not, you will need to manually enter the titles for the block and page displays.

3. Proceed to the main configuration screen by clicking **`continue and edit`**. Your view is not saved at this point. You must save the view before navigating away from the main configuration screen!
4. At the top of the configuration screen click **`Save`**.

Connect Nodes to Their Taxonomy Terms

1. Click on the fieldset label **`Advanced`**. A column of advanced settings will appear.
2. Next to the label **`Relationships`** click **`add`**. A modal window will appear.
3. Enable the check box beside **`Content: Tags`**.
4. Click **`Add and configure relationship`**.

5. Enable **Require this relationship**.
6. Click **All (all displays)**.
7. Next to the label **Contextual Filters** click **add**. A modal configuration window will open.
8. Enable the check box beside **Taxonomy term: name**.
9. Click **Add and configure the contextual filter**.
10. Apply the following settings:
    1. When the **filter value *is not* in the URL** (use the following settings):
       Display a summary: Enable
       Sort order: *Descending*
       Sort by: *Number of records*
       Format: *list*
       Display record count with link: enable the check box
       This will print a list of all taxonomy terms, in order of popularity.
    2. **When the filter value *is* available** (use the following settings):
       Override title: enable the check box
       Override title value: `%1`
       This will print a the name of the taxonomy term and then a list of all Articles which have been tagged with this taxonomy term.
11. At the bottom of the configuration window click **Apply (all displays)**. The modal window will close.
12. Scroll to the top of the screen and click **Save**.

You are finished configuring this view.

Block for Popular Tags View

This block should appear only on blog-related pages.

1. Using the admin dashboard, navigate to **Structure** > **Blocks**.
2. Next to the block **View: Popular Tags**, click the link labelled **configure**.

3. Scroll to the fieldset labelled **`Region settings`** and select from the Bartik drop down menu select *`Sidebar first`*.
4. Scroll to **`Visibility settings`**. Click on the vertical tab for **`Pages`**. Enable the option for **`Only the listed pages`**. In the large text area below the option you just enabled, enter the following two lines:
   ```
   blog
   blog/*
   ```
   This will limit the block to appearing only on pages with the URL *blog*.
5. Scroll to the bottom and click **`Save block`**.

You are now finished configuring the block for the view Popular Tags.

Check Your Work: Popular Tags

A new block should now be visible below the Monthly Archive block in the first sidebar on blog-related pages which lists the top ten most popular taxonomy terms in the vocabulary Tags with the number of Articles in brackets after each category.

Spam Control

In the Drupal context spam refers to any content created by a computer program that has been designed to seek out Web forms, and publish unwanted content. Spam can be created by individual Web site visitors; however, it is more typically an automated process performed by a robot (“bot”).

To prevent spam bots from adding unwanted comments to your site you can add a "challenge" to the comment form which asks a skill testing question. These challenge questions are known as Completely Automated Public Turing tests to tell Computers and Humans Apart (CAPTCHA). Web CAPTCHA tests have been implemented in several different ways and include:

- graphic test (read a scrambled word and type it into the response box);
- logic test (simple math, or select a word from a list, and type it into the response box); and,
- audio test (listen to a word and then type it into the box).

The graphic test is one of the most common tests but it is also relatively inaccessible to Web site visitors with low vision and who are blind. The graphics-based CAPTCHA test is often paired with an audio test to allow low vision and blind users an alternative way to prove their humanity.

I use Mollom (www.mollom.com) on my Web sites—which is a spam filtering service.

A detailed tutorial on configuring Mollom for Drupal is available at http://mollom.com/tutorials/drupal. Although the tutorial is written for Drupal 6 the basic procedures are the same for Drupal 7.

Portfolio

The heart of this Web site is the portfolio of work. (Hard to believe given that there have already been more than 30 pages of non-heart instructions.) But here we go. It's an awesome combination of jQuery enhanced modules. Before proceeding, make sure you have the following modules enabled:

- **Link**. http://drupal.org/project/link
- **Date**. http://drupal.org/project/date. Enable Date, Date API and Date Popup.

Enabling a Module

1. *Navigate to Administration > Modules.*
2. *Locate the module you want to enable.*
3. *Enable the check box beside the module name.*
4. *Scroll to the bottom of the screen and click "Save configuration."*

Creating a New Content Type

This portfolio content type is pretty basic. It includes eight fields and the option to upload an unlimited number of images for each project.

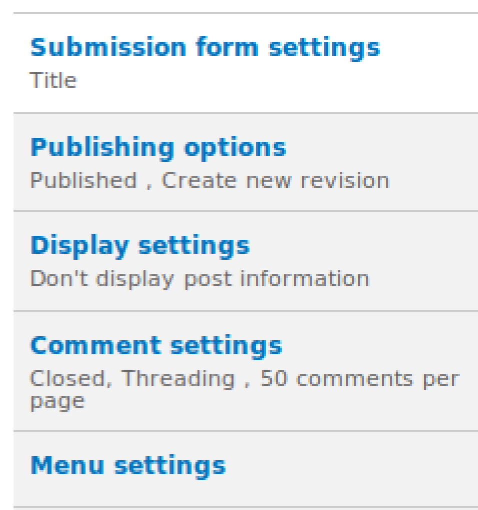
Task Summary

Create a new content type for your portfolio.

Let's start by creating the content type:

1. Navigate to **`Administration`** > **`Structure`** > **`Content types`** > **`Add content type`**.
2. Enter the following information for each of the fields (if there is no setting, leave the field as the default value):

Field	Value
Name	Portfolio
Publishing options	Published unselect promoted to front page select Create new revision
Display settings	Unselect display author and date
Comment settings	Closed
Menu settings	Unselect Available menus.



3. Scroll to the bottom of the configuration screen and click **`Save and add fields`**.

Time to customize it by adding some fields.

Field Summary

There are five types of fields that we're going to add. The following table outlines each of the new fields (**field_** is added automatically):

Field Label	Field Name	Field	Widget	Properties
Client Name	field_client_name	Text	Text field	Use default values.
Project Web site	field_url	Link	Link	Use default values
Screenshots	field_screenshots	Image	Image	Allow unlimited values allowed to be uploaded.
Start Date	field_start	Date	Text field with Date Pop-up calendar.	No end date. Use only Year, Month, Day for the granularity.
End Date	field_end	Datetime	Text field with Date Pop-up calendar.	No end date. Use only Year, Month, Day for the granularity.
Type of Work	field_type	Term reference	Check boxes/radio buttons.	Unlimited values allowed.

When each of the fields has been added, the tab `manage fields` will appear as follows:

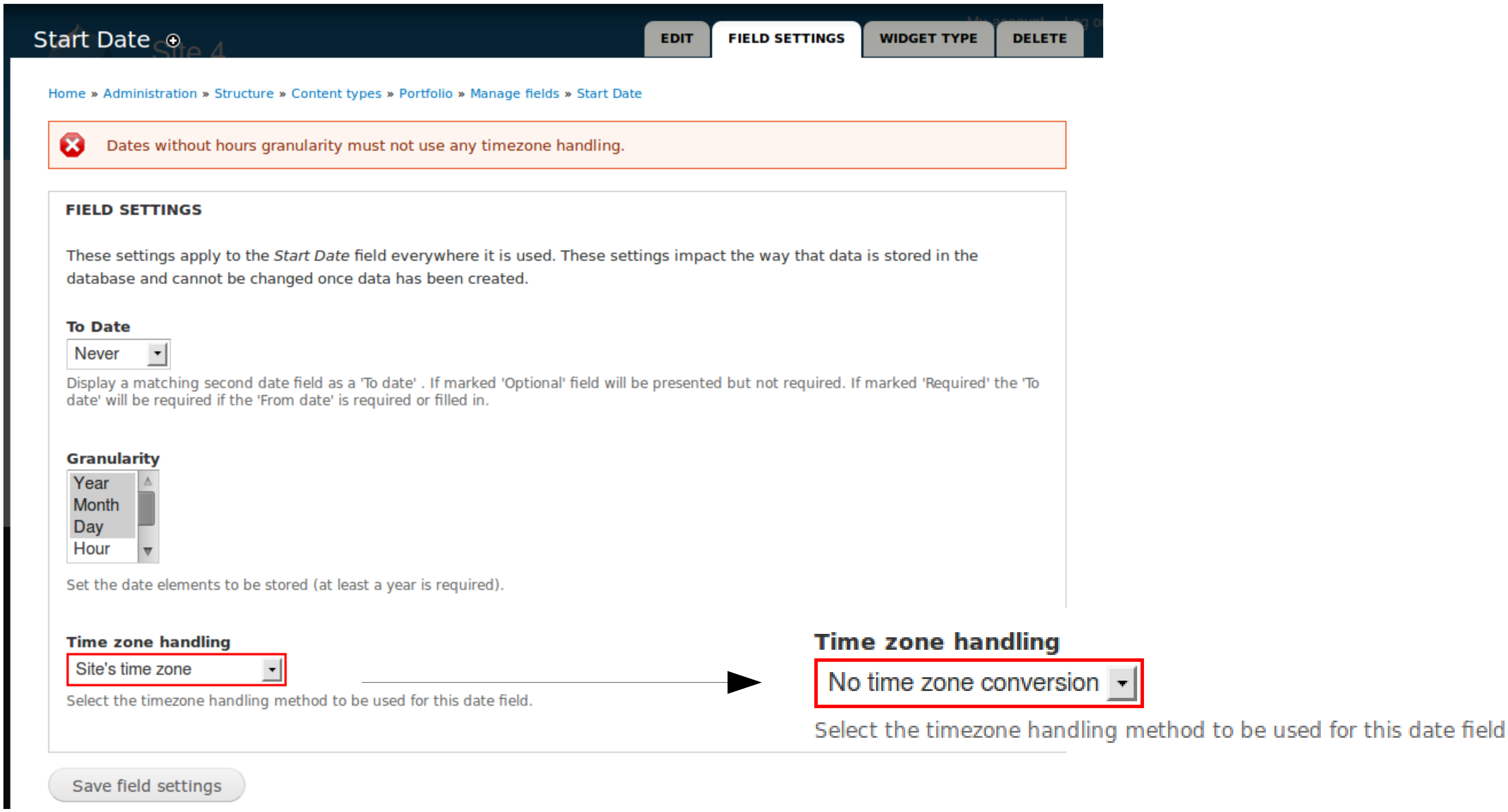
LABEL	NAME	FIELD	WIDGET	OPERATIONS	
Title	title	Node module element			
Body	body	Long text and summary	Text area with a summary	edit	delete
Client Name	field_client_name	Text	Text field	edit	delete
Project Web Site	field_url	Link	Link	edit	delete
Screenshots	field_screenshots	Image	Image	edit	delete
Start Date	field_start	Date	Text Field with Date Pop-up calendar	edit	delete
End Date	field_end	Datetime	Text Field with Date Pop-up calendar	edit	delete
Type of Work	field_type	Term reference	Check boxes/radio buttons	edit	delete

Adding Fields

In most cases the default values are appropriate for any new field that you will be adding. The following are a few things to watch out for.

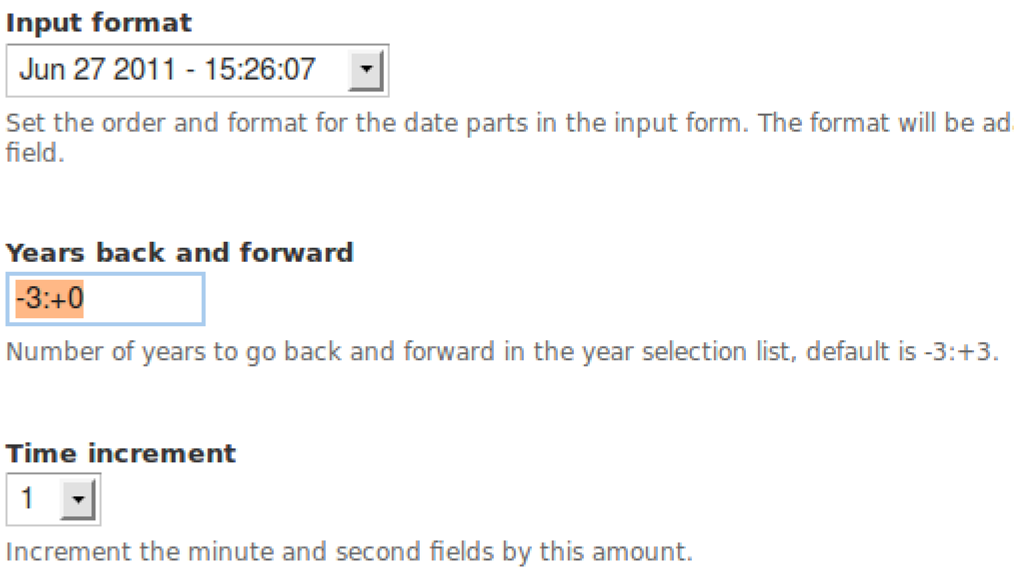
Date and Time Zone Handling

By default your site's time zone is used; however, if you are not using hours and minutes for your field you will get an error. Set the `Time zone handling` to *`No time zone conversion`*.



Date – Years Backwards and Forwards

You may time box the values for your date selection tools. By default the window of time is plus or minus three years. The format used is: `-years:+years`. If you want to set one of these to zero years, you must still use the – or + sign.



Date – Which Format to Use

There are long, heated debates on the Internet over whether it is most appropriate to use Date (this is ISO format YYYY-MM-DDTHH HH:MM:SS), Datetime (YYYY-MM-DD HH:MM:SS) or Datestamp. For most low-traffic sites it probably doesn't matter which option you use; however, datestamp is the slowest format. There is more information available from: http://drupal.org/node/269813. I have used both Date and Datetime to show there is virtually no difference between the two. When possible: avoid using Datestamp with the Date module as it adds an extra layer of program computation to convert back and forth between formats.

Creating a New Vocabulary and Adding Terms

Before we can add the taxonomy term field we must first create the vocabulary.

1. Navigate to **`Administration`** > **`Structure`** > **`Taxonomy`** > **`Add vocabulary`**.
2. Enter the name of the new vocabulary: `Development Tools`.

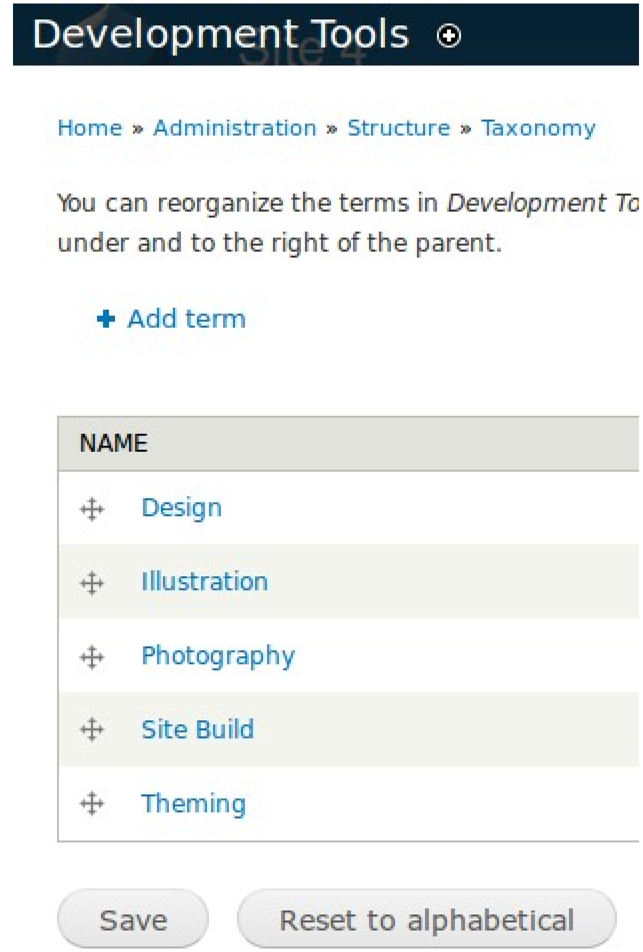
That's it.

Now you need to add some terms to your Vocabularies (use the link beside the vocabulary name). I added: Design, Illustration, Photography, Site Build and Theming. If you have different skills you should add your own taxonomy terms.

Adding Taxonomy Fields to Your Content Type

For each portfolio project you will need to be able to assign multiple taxonomy terms from your development tools vocabulary.

1. Navigate to **`Administration`** > **`Structure`** > **`Content types`** > **`Portfolio`** > **`Manage fields`**.
2. In the field **`Add new field`** enter the following:
    1. **`Label`**: `Type of Work`
    2. **`Field name`**: `type`
    3. **`Type of data to store`**: *`Term reference`*
    4. **`Form element to edit the data`**: *`check boxes / radio buttons`*
3. Scroll to the bottom of the configuration screen and click **`Save`**.
4. On the **`Field settings`** page choose *`Development Tools`* for the Vocabulary.
5. Proceed through the wizard changing only the number of allowed values from *`1`* to *`unlimited`*.



Task Summary

Add fields for the portfolio content type—use default values unless the field summary table indicates otherwise.

Display Settings for Portfolio Content Type

Although we will be theming the site, you can do a bit to clean up your display now. Once you've added each of the fields proceed with the following:

1. Navigate to **`Administration`** > **`Structure`** > **`Content types`** > **`Portfolio`** > **`Manage display`**.
2. Drag the fields into the following order and set the label as follows:

Field	Label	Format
Screenshots	Hidden	Image. Use the Medium size. Review the Image Style instructions from the Basic page content type earlier in this document for more information.
Body	Hidden	Default
Type of Work	Above	Link
Client Name	Inline	Default
Start Date	Inline	Default. Use the Medium format type.
End Date	Inline	Default. Use the Medium format type.
Project Web Site	Hidden	Title, as link (default)

3. Scroll to the bottom of the configuration screen and click **`Save`**.

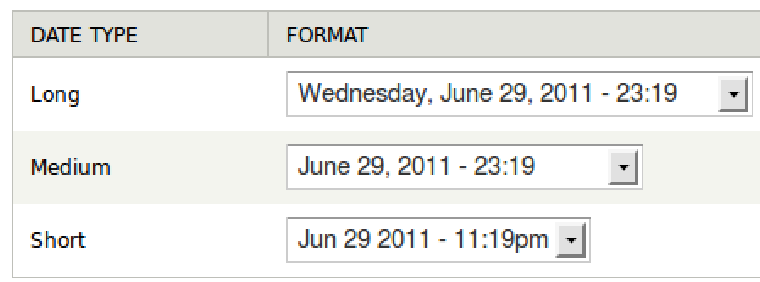
Task Summary

Update the display settings for the portfolio content type as described.

Adjust the Date Format

I'm picking about my dates. I like them to be in English or ISO. It's probably because I'm Canadian and I never know if 2011/06/12 is June or December. To configure how your dates are displayed follow these instructions:

1. Navigate to **`Administration`** > **`Configuration`** > **`Date and time`**.
2. Choose the appropriate date format for **`Long`**, **`Medium`** and **`Short`** formats.
3. Click **`Save configuration`**.



Your new date formats will be applied throughout the site.

Automatic URL Aliases for SEO

We're going to make very search engine-friendly URLs with the module Pathauto (and its helper Token). Before proceeding with these instructions, make sure the module Pathauto is enabled.

1. Navigate to **`Administration`** > **`Configuration`** > **`URL aliases`** > **`Patterns`** (tab at the top).
2. Enter the following patterns:
    - **`Pattern for all Portfolio paths`**: `portfolio/[node:title]`
3. Scroll to the bottom of the screen and click **`Save configuration`**.

Add New Content Using Portfolio Content Type

To be able to test the display of our site build you will need to add some content. I added four portfolio projects with at least four images each. This was sufficient to test the display. Ideally you would add more. Right now the display will look frustratingly ugly. That's okay.

Task Summary

Add at least four new nodes using the content type Portfolio.

Make sure you have images as well.

View: Portfolio Terms

For the navigation block within the portfolio section, we'll use a view of terms and limit their visibility to paths that contain “portfolio.”

1. Navigate to **`Administration`** > **`Structure`** > **`Views`** > **`Add new view`**.
2. Complete the following fields:
    - **`View name`**: `Portfolio Terms`
    - **`Show`**: *`Taxonomy terms`*
    - **`of type`**: *`Development Tools`*
    - **`Create a page`**: disable
    - **`Create a block`**: enable

- **`Block title`**: `Portfolio Terms` (should be auto completed)
- **`Display format`**: *`HTML List`* of *`titles (linked)`*
- **`Items per page`** `5`

3. Click **`Continue and edit`**.
4. Beside the heading **`Filter Criteria`**, click on the button **`add`**. A modal window will open.
    1. Enable the option *`Taxonomy term: Vocabulary`*.
    2. Click **`Add and configure fields`**.
    3. Under the heading **`Options`** enable *`Development Tools`*.
    4. Click **`Apply (all displays)`**. The modal window will close.
5. Beside the heading **`Sort Criteria`**, click on the button **`add`**. A modal window will open.
    1. Enable the option *`Taxonomy term: Name`*.
    2. Click **`Add and configure sort criteria`**.
    3. Leave the default selection of *`Sort ascending`* and click **`Apply (all displays)`**. The modal window will close.
6. Beside the heading **`Use pager`**, click on **`Full`**. A modal window will open.
    1. Change the enabled option to *`Display all items`*.
    2. Click **`Apply (all displays)`**. Leaving the offset set to `0`, click **`Apply (all displays)`** again. The modal window will close.
7. Scroll to the top of the configuration screen and click **`Save`**.

Block for Portfolio Tags View

This block should appear only on portfolio-related pages.

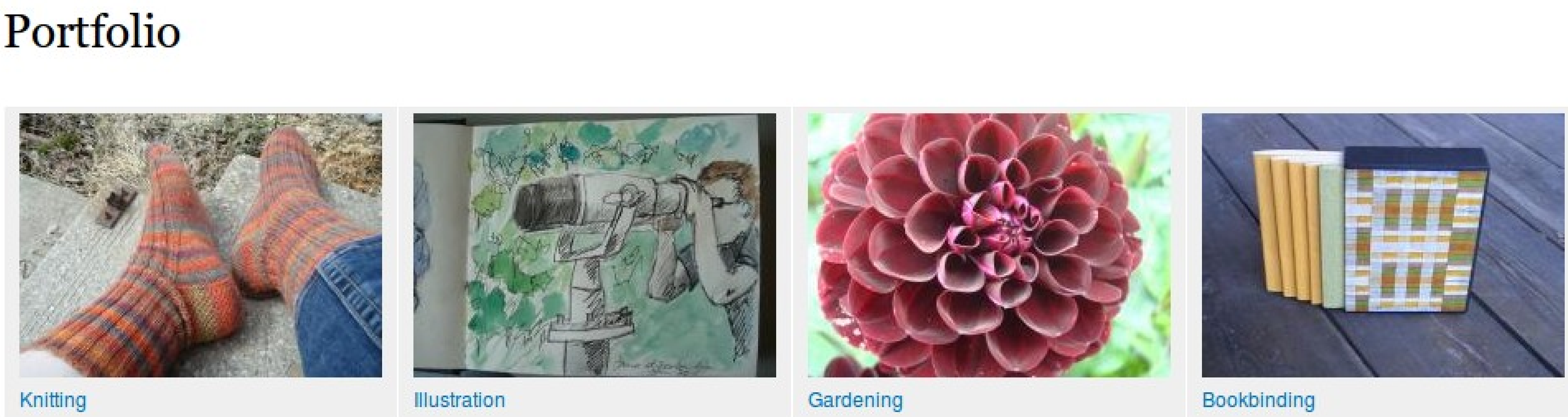
1. Using the admin dashboard, navigate to **`Structure`** > **`Blocks`**.

2. Next to the block **`View: Portfolio`**, click the link labelled **`configure`**.
3. Scroll to the fieldset labelled **`Region settings`** and select from the Bartik drop down menu select *`Sidebar first`*.
4. Scroll to **`Visibility settings`**. Click on the vertical tab for **`Pages`**. Enable the option for **`Only the listed pages`**. In the large text area below the option you just enabled, enter the following two lines:
   ```
   portfolio
   portfolio/*
   ```
   This will limit the block to appearing only on pages with the URL *portfolio*.
5. Scroll to the bottom and click **`Save block`**.

You are now finished configuring the block for the view Portfolio.

View: Lists of Icons

Grids of pictures are very common in Web portfolios. Using only the Views module you will create a grid-based (table-based) layout for your images. It will be three columns wide and use a custom image style to ensure all photos are landscape format. The default is four columns wide—this works well for Bartik, but it won't fit in our custom theme later in the workbook.



Make sure you have at least four portfolio items with at least one photo each before you proceed.

Create a Landscape Image Style

1. Navigate to **`Administration`** > **`Configuration`** > **`Image styles`** > **`Add style`**.
2. Enter the **`Style name`**: `rectangle_thumb`.
3. Click **`Create new style`**.
4. At the bottom of the screen in the effects table, change the drop down menu from *`Select a new effect`* to *`Scale and Crop`*.

5. Click `Add`.
6. For the `Width`, enter: `220`.
7. For the `Height`: enter `160`.
8. Click `Add effect`.

Your new image style has been created. You may close the overlay window.

Create a Grid-Based View

1. Navigate to `Administer` > `Structure` > `Views` > `Add new view`.
2. A new screen will appear where you can enter the basic settings for your new view. Use the following settings:
    1. `View name`: `Portfolio Gallery`
    2. In the mad-libs fill in the blank, alter the sentence to read: Show *`Content`* of type *`Portfolio`* tagged with `<leave blank>` sorted by *`Newest first`*.
    3. Leave the box selected for `Create a page` and change the following settings.
        1. `Path`: `portfolio`
        2. `Display format`: *`Grid`* of *`fields`*.
        3. `Items to display`: 20
    4. Enable the check box for `Create a menu link`. Leave the defaults of `main menu` and `link text` as `Portfolio`.
3. Click `Continue and edit`.
4. Add the image field for portfolio items. Locate the heading `Fields`. Click the button labelled `add`. A modal window will open.
    1. In the search box type: `screenshot`.
    2. Enable check box beside the field `Content: Screenshots`.
    3. Click `Add and configure fields`.
    4. Disable the setting `Create a label`.

    5. Change the **Image style** to *rectangle_thumb*.
    6. Change the **Link image to** field to *Content*.
    7. Click on the fieldset label **Multiple field settings** to reveal the configuration options.
    8. Change the number of displayed values from `all` to `1`.
    9. Click **Apply (all displays)**. The modal window will close.

5. Put the title at the bottom of the stack of fields. Locate the heading **Fields**. Next to the bottom labelled **add**, click the arrow and choose the option **rearrange**. A modal window will open.
    1. Drag the field **Content: Title** so that it appears below **Content: Screenshots**.
    2. Click the button **Apply (all displays)**. The modal window will close.
6. Update the grid to show 3 items per row. Locate the heading **Format**. Next to the link **Grid**, click on the link labelled **Settings**. A modal window will open.
    1. Change the **Number of fields** to `3`.
    2. Click **Apply (all displays)**. The modal window will close.
7. Scroll to the top of the screen and click **Save**.

Your view is now configured and available from the main menu.

Task Summary

Create a new view for your portfolio with a display for page which has a menu item. Update the view to include a icons for each item displayed.

Override Taxonomy Term Listings

I hate the default taxonomy listing page that shows teasers of nodes. Although there are limited configuration options at **Structure** > **Taxonomy** > **Manage Display**, I want to make sure the category listings are identical to the summary views I've created. Views allows you to easily override these pages:

1. Navigate to **Administration** > **Structure** > **Views**.
2. Locate the view labelled **Taxonomy term**. From the dropdown menu, select *Clone*.
3. Enter the view name `Formatted Taxonomy Terms`.
4. Click the button labelled **Continue**. The views configuration screen will open.
5. Locate the heading **Format.** And beneath it the heading **Format**. Click on **Unformatted list**. A modal window will open.
6. Enable the option *Grid*.
7. Click **Apply (all displays)**.
8. The default Page style options are fine, click **Apply (all displays)** again. The modal window will close.
9. Locate the heading **Format.** And beneath it the heading **Show**. Click on **Content**. A modal window will open.
    1. Change the setting to *Fields*.
    2. Click the button labelled **Apply (all displays).**
    3. The default settings for the field options are acceptable. Click **Apply (all displays)**. The modal window will close.
10. Locate the heading **Fields**. If there is no field listed:
    1. Click the link **add**. A modal window will open.
    2. Search for: `title`.
    3. Beside ***Content*: Title**, enable the check box. (This is usually the 2nd option. Read carefully as you do not want to accidentally enable ***Comment*: Title**.)

4. Click the button labelled **`Add and configure fields`**.
5. Beside the field **`Create a label`**, disable the check box.
6. Click the button labelled **`Apply (all displays)`**. The modal window will close.

11. Locate the heading **`Fields`**. Click the button labelled **`add`**. A modal window will open.
    1. In the search box type: `image`.
    2. Enable check box beside the field **`Content: Image`**.
    3. Click **`Add and configure fields`**.
    4. Disable the setting **`Create a label`**.
    5. Change the **`Image style`** to *`rectangle_thumb`*.
    6. Change the **`Link image to`** field to *`Content`*.
    7. Click **`Apply (all displays)`**. The modal window will close.
12. Locate the heading **`Fields`**. Click the button labelled **`add`**. A modal window will open.
    1. In the search box type: `screenshot`.
    2. Enable check box beside the field **`Content: Screenshots`**.
    3. Click **`Add and configure fields`**.
    4. Disable the setting **`Create a label`**.
    5. Change the **`Image style`** to *`rectangle_thumb`*.
    6. Change the **`Link image to`** field to *`Content`*.
    7. Click on the fieldset label **`Multiple field settings`** to reveal the configuration options.
    8. Change the number of displayed values from `all` to `1`.
    9. Click **`Apply (all displays)`**. The modal window will close.
13. Locate the heading **`Sort Criteria`**.

1. Click on **add**.
2. Search for: `title`.
3. Beside ***Content*: Title**, enable the check box. (This is usually the 2nd option. Read carefully as you do not want to accidentally enable ***Comment*: Title**.)
4. Click the button labelled **Add and configure sort criteria**.
5. Enable the option *Sort ascending*.
6. Click the button labelled **Apply (all displays)**. The modal window will close.
7. Click on **Content: Post date (desc)**. A modal window will open.
8. Click the button **Remove**. The modal window will close.

14. Scroll to the top and click Save.

You are finished configuring this view. It will be automatically enabled and will be in use whenever you click on a link to a taxonomy term.

Funk It Up With jQuery

From Facebook to Flickr, Web site visitors expect to have on-page interactions with the sites they visit. The scripting language Javascript makes it possible. Javascript has the same cross-browser problems we know well from CSS. It's also “just” a language. There are no short cuts. In CSS, a suite of grid frameworks has emerged allowing us to develop faster, cross-browser-friendly layouts. The same has happened for Javascript as well, front end Web developers have created tool kits of Javascript, which give us the shortcuts to the most common techniques we use today.

There are many different Javascript libraries. Some of the more popular ones are:

- **jQuery**. http://jquery.com
- **Dojo**. http://dojotoolkit.org
- **YUI** (Yahoo! Interface Library). http://developer.yahoo.com/yui
- **Prototype**. http://www.prototypejs.org
- **MochiKit**. http://mochi.github.com/mochikit

You can use any of them with Drupal, but the Drupal community has decided that jQuery is the best library for the kinds of things we do. Drupal 7 includes jQuery 1.4.4 and jQuery UI 1.8.7. These are not the latest versions of the Javascript libraries, so there may be times when you need to overwrite the Drupal jQuery library and put a newer one in place.

jQuery (http://jquery.com) is good at finding and manipulating objects on your page, its partner library, jQuery UI (http://jqueryui.com) is good at creating visual widgets and effects. A few examples of how jQuery and jQuery UI work together in Drupal include:

- administrative overlay
- drag-and-drop ordering for menu items, taxonomy terms and block display
- date picker pop-up calendar (used in the portfolio content type)
- progress bar used for file uploads and Drupal installations

In Site 3 (Business Directory) we used jQuery to recolour the first word of headings. In this site we will add several different types of functionality to our site using Drupal modules to configure jQuery plugins.

- Rotating banners (sequence of rotating images)
- Carousels (row of images you can scroll through)
- Light box (pop-up overlay image)
- Image cropping

The Drupal module for each of these features allows you to set the parameters for the jQuery plugin through Drupal's administrative interface.

Adding jQuery Libraries

At the beginning of the workbook you downloaded and installed the following Drupal modules (if you haven't done that yet, do it now):

- **Libraries**. http://drupal.org/project/libraries (required by Gallery Formatter)
- **Gallery Formatter**. http://drupal.org/project/galleryformatter
- **Colorbox**. http://drupal.org/project/colorbox
- **EPSA Crop**. http://drupal.org/project/epsacrop
- **Views Slideshow**. http://drupal.org/project/views_slideshow

Each of these modules comes with a very useful README file. You should read it. If you are using the Web-based module installation you will need to download the module and unzip it on your local computer to be able to find the README file. Although you should do this for all modules, it's especially useful for modules where you need to connect to another service, or download a the jQuery plugin.

Libraries is the Drupal module which allows us to put the Javascript libraries into a common directory, instead of the module's directory. It should be enabled first. Once enabled you should have a new libraries directory you can put your jQuery plugins into. The new folder should be located at `sites/all/libraries`. If it doesn't exist, create it now.

Task Summary

Enable each of the six modules listed. Install the Javascript libraries using the specified folder names.

Enabling a Module

1. *Navigate to Administration > Modules.*
2. *Locate the module you want to enable.*
3. *Enable the check box beside the module name.*
4. *Scroll to the bottom of the screen and click "Save configuration."*

When you enable each of the Drupal modules for the jQuery plugins you will get error messages warning you that you have not downloaded the relevant libraries:

- Could be find the json2.js file, check your install
- Could be find the json2.js file, check your install
- The Jcrop plugin is missing. Download and extract it in your *sites/all/libraries* directory. (Currently using EPSA Crop : Jcrop Not found)
- The json2.js file is missing, you can download it here and put it in your *sites/all/libraries* directory. Don't forget rename the downloaded folder into json2. At the end, you should have something like this sites/all/libraries/json2 (Currently using EPSA Crop : json2.js Not found)

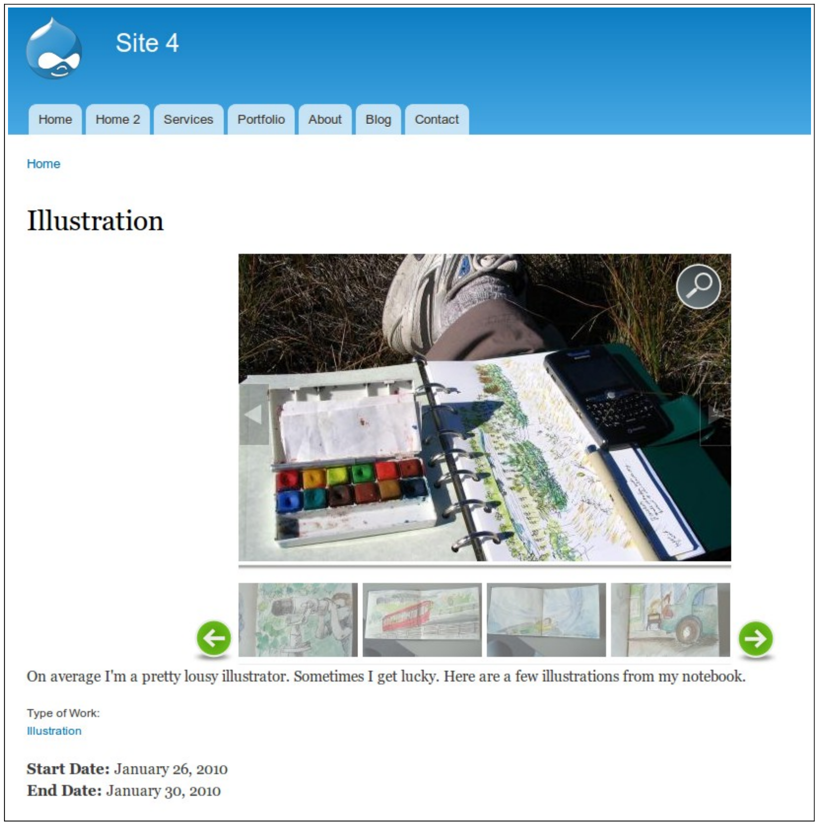
You will need to download and unpack *four* separate libraries. In most cases you will also need to rename the unpacked library folder so that it matches what the Drupal module is expecting to find.

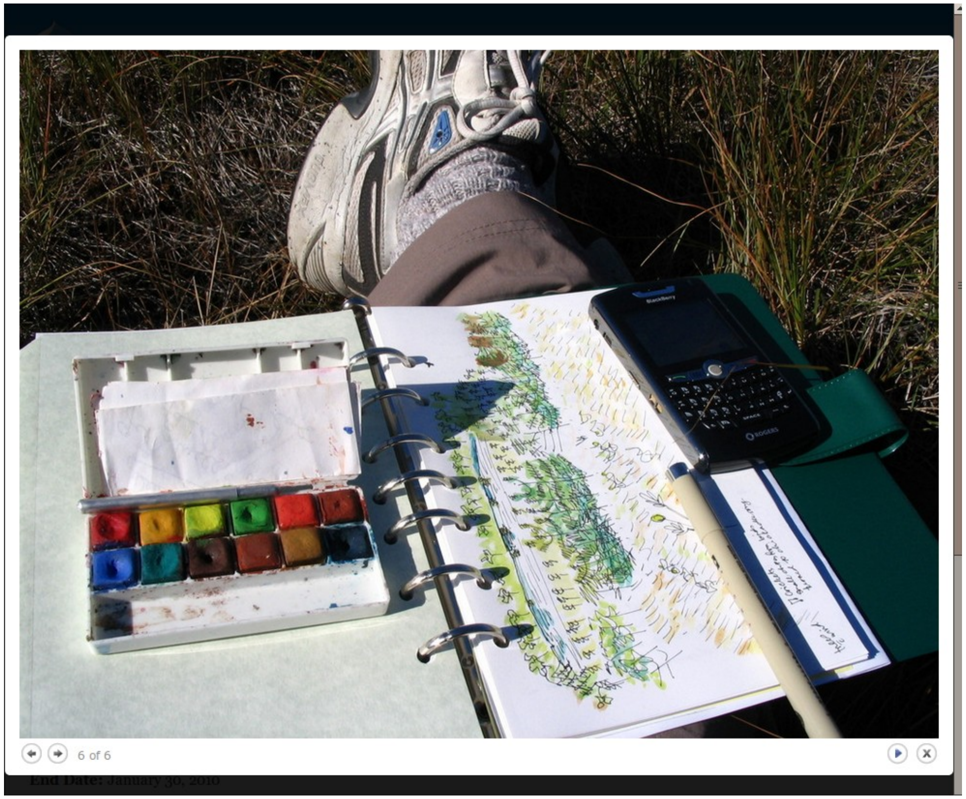
- Colorbox. http://colorpowered.com/colorbox/
  Unpack and upload to `sites/all/libraries/`**`colorbox`**
- Jcrop. http://deepliquid.com/content/Jcrop.html
  Unpack and upload to `sites/all/libraries/`**`Jcrop`** (capital J)
- jquery.cycle. http://jquery.malsup.com/cycle/
  You want the .zip package. If the link isn't obvious, use the direct link:
  http://malsup.com/jquery/cycle/release/jquery.cycle.zip
  Unpack and upload to `sites/all/libraries/`**`jquery.cycle`** (with a period)
- json2. https://github.com/douglascrockford/JSON-js
  Click the download button and choose the `.zip` option.
  Unpack and upload to `sites/all/libraries/`**`json2`**

Once the modules are enabled and the libraries are uploaded, you are ready to proceed.

Content Type Carousel With Light Box

For this effect we will add a carousel and light box effect to the portfolio content type.





There are a number of different elements to configure, but it's surprisingly simple to snap together.

Configure the Light Box

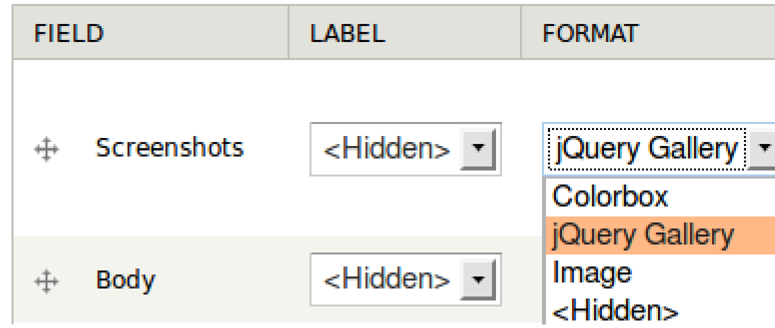
1. Navigate to **`Administration`** > **`Configuration`** > **`Colorbox`**.
2. Under **`Image Field Options`**, enable *`per field gallery`*.
3. Scroll to **`Styles and Options`**. From here you can alter the transition type,

transition speed, opacity of the background and words used on the pop-up overlay.

4. Scroll to **Slideshow Settings**. Choose whether you would like the slideshow enabled or disabled. With the full carousel enabled for the content type, I think it makes sense to disable the slide show. Choose whichever option you like best.
5. Scroll to the bottom of the screen and click **Save configuration**.

Configure Carousel

All of the options for the carousel are controlled from within the content type display settings. The carousel is controlled by the module Gallery Formatter.



1. Navigate to **Administration** > **Structure** > **Content types** > **Portfolio** > **Manage display**.
2. For the field Screenshots, change **Format** from *Image* to *jQuery Gallery*.
3. At the end of the Screenshots row, click the configuration cog. The configuration options for the jQuery Gallery will open.
    1. **Select the slide style** (big image). The image style *galleryformatter_slide* appropriate resizes both vertical and horizontal images. I like using this style.
    2. **Select the thumbnail style** (small icons across the bottom). Leave the default image style of *galleryformatter_thumb* for now. We will change this to the cropped thumbnail style later.
    3. **Style**. The option *Greenarrows* is used for the sample screenshots.
    4. Enable **Link slides to the full image**.
    5. **Select full image style**. In the screen shots above I used *None (original)*, but if you don't want the image extending to the full size of the browser, use a smaller image style, such as *Large*.
    6. **Use jQuery modal for full image link**. Set this option to *colorbox*.
    7. Click the button **Update**.
4. Click the button **Save**.

Your new carousel has been applied to portfolio nodes.

Task Summary

Modify the portfolio content type field Screenshots by replacing the format from Image to jQuery Gallery. Configure the gallery settings as described.

Cropping Images

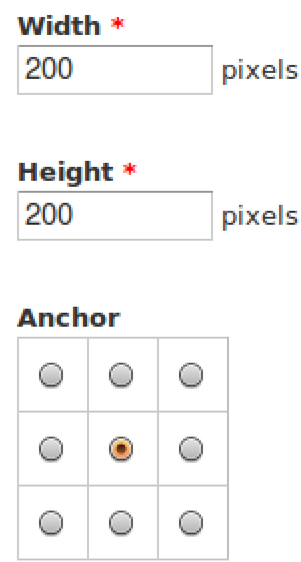
The EPSA Crop module (7.x-2.1) was a little bit finicky for me if I worked too quickly and closed the cropping window too soon. This may have been because of my server set-up. Avoid hassles and take your time. There is also an identified bug for this version of the module whereby any image field which does *not* use a cropped image style reports an error. The development snapshot already has the fix and it should be available in the next release of the module. With all of those caveats out of the way, let's configure and start using custom cropped images!

There are no configuration options for the module EPSA Crop; however, there are permissions. If you are running a multiuser site where some users are not administrators, you will need to adjust the permissions (`Administration` > `People` > `Permissions`).

Add the Crop Dialog to an Image Style

EPSA Crop works with image styles. By default it provides one image style, but does not add the cropping dialog to it. (This probably makes sense to someone...)

1. Navigate to `Administration` > `Configuration` > `Image styles` > `Add style`.
2. For the `Style name`, enter `cropped_thumb`.
3. Click the button `Create new style`.
4. At the bottom of the configuration screen click change the drop down menu from *`Select new effect`* to *`Crop Dialog`*.
5. Click the button `Add`.
6. In the `Width` and `Height` fields enter the final dimensions for your cropped image.
7. Adjust the `Anchor` location for the which part of the image you want to retain if there is no custom cropping. By default this is the middle of the image.
8. Scroll to the bottom of the screen and click `Add effect`.



If your site uses many different image styles you can add the crop dialog to each and get custom images for each image style by repeating these steps per image style.

Update Your Image Field to Use the New Image Style

1. Navigate to Administration > Structure > Content types > Portfolio > Manage fields > Screenshots > edit.
2. Scroll to the fieldset **EPSACrop settings**. If there are no image styles with the crop dialog enabled, this fieldset will be missing. Enable the image style *cropped_thumb*.
3. Scroll to the setting **Preview image style**. Change the selected style to *cropped_thumb*.
4. For custom titles on your images (and higher accessibility and better search engine rankings) enable the **Alt** and **Title** fields.
5. Scroll to the bottom of the configuration screen and click **Save settings**.

Update the Display to Use the Cropped Image

1. Navigate to **Administration** > **Structure** > **Content types** > **Portfolio** > **Manage display**.
2. At the end of the Screenshots row, click the configuration cog. The configuration options for the jQuery Gallery will open.
    1. **Select the thumbnail style** (small icons across the bottom). Change this image style to *cropped_thumb*.
    2. Click the button labelled **Update**.
3. Click the button labelled **Save**.

Crop Your Images

You're now ready to use the new cropping tool.

1. Navigate to portfolio content type page and click the **edit** tab.
2. Click the link labelled **manage image crops**. A modal window will open.
3. Give it a few seconds and a small square will appear on the top left of the image. Adjust the size and position of the square to isolate the cropped image.

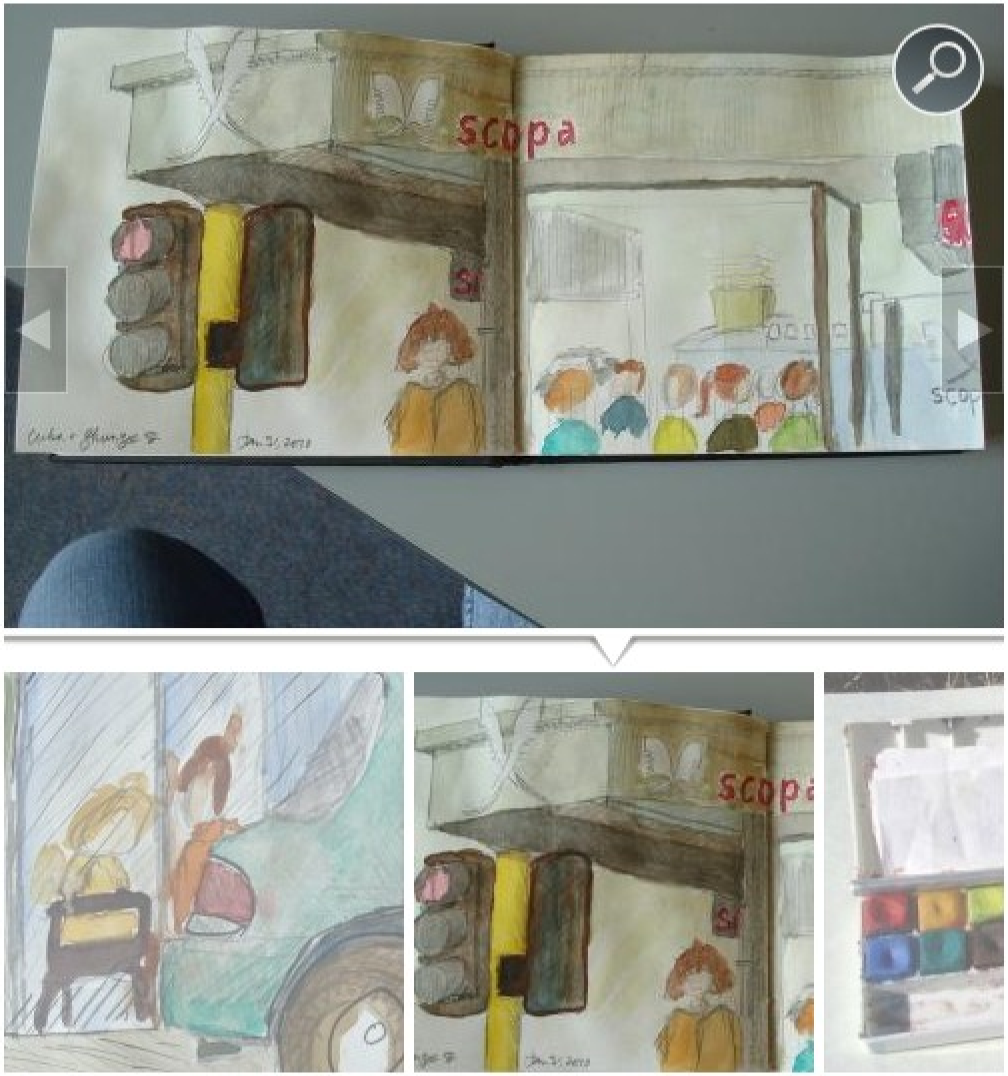
4. When you're finished, click **Save**. The modal window will close, but the new image will *not* appear in the file information for this node until after you've saved the node.
5. Crop all images attached to the node using steps 2 to 4.
6. Scroll to the bottom of the node editing screen and click **Save**.

Your cropped image will now appear on the site. The first image in the jQuery Gallery doesn't seem to show the cropped image. Disable the gallery, crop your images, and enable the gallery if you are having this problem too.

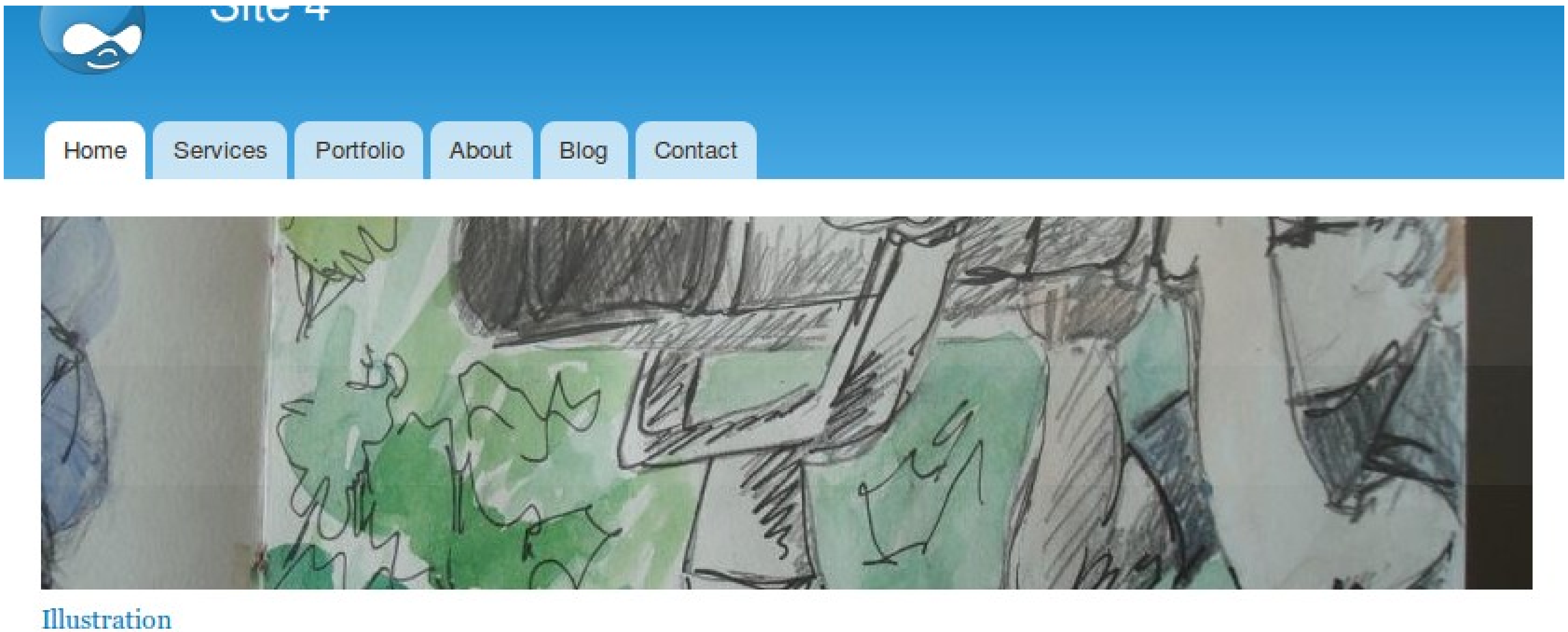
The image on the right shows the new jQuery Gallery with the cropped thumbs along the bottom.

Task Summary

Create a new image style which uses the crop dialog. Enable the image style in your portfolio content type, and add it to the display of your portfolio content type. Crop your portfolio images.



View: Front Page Rotating Banner



Create a Banner Image Style

1. Navigate to **`Administration`** > **`Configuration`** > **`Image styles`** > **`Add style`**.
2. Enter the **`Style name`**: `banner`.
3. Click **`Create new style`**.
4. At the bottom of the screen in the effects table, change the drop down menu from *`Select a new effect`* to *`Scale and Crop`*.
5. Click **`Add`**.
6. For the **`Width`**, enter: `800`.
7. For the **`Height`**: enter `200`.

8. Click **Add effect**.

Your new image style has been created. You may close the overlay window.

Create a Grid-Based View

1. Navigate to **Administer** > **Structure** > **Views** > **Add new view**.
2. A new screen will appear where you can enter the basic settings for your new view. Use the following settings:
   1. **View name**: `Home Banner`
   2. In the mad-libs fill in the blank, alter the sentence to read: Show *Content* of type *Portfolio* tagged with (`leave blank)` sorted by *Newest first*.
   3. Leave the box selected for **Create a page** and change the following settings.
      1. **Page title:** `(leave blank)`
      2. **Path**: `home`
      3. **Display format**: *Slideshow* of *fields*.
      4. **Items to display**: 4
   4. Enable the check box for **Create a menu link**. Leave the defaults of **main menu** and change the **link text** to `Home`.
3. Click **Continue and edit**.
4. Locate the heading **Fields**. Click the button labelled **add**. A modal window will open.
   1. In the search box type: `body`.
   2. Enable check box beside the field **Content: Body**.
   3. Click **Add and configure fields**.
   4. Disable the setting **Create a label**.
   5. Click the fieldset heading **Rewrite results**. A new set of configuration options will be revealed.

6. Beside **`Trim this field to a maximum length`**, enable the check box.
7. In the field **`Maximum length`**, set the number of characters to `80`.
8. Enable the options **`Trim only on a word boundary`**, **`Add an ellipse`** and **`Strip HTML tags`**.
9. Click **`Apply (all displays)`**. The modal window will close.

5. Locate the heading **`Fields`**. Click the button labelled **`add`**. A modal window will open.
   1. In the search box type: `screenshot`.
   2. Enable check box beside the field **`Content: Screenshots`**.
   3. Click **`Add and configure fields`**.
   4. Disable the setting **`Create a label`**.
   5. Change the **`Image style`** to *`banner`*.
   6. Change the **`Link image to`** field to *`Content`*.
   7. Click on the fieldset label **`Multiple field settings`** to reveal the configuration options.
   8. Change the number of displayed values from `all` to `1`.
   9. Click **`All (all displays)`**. The modal window will close.
6. Locate the heading **`Fields`**. Next to the button labelled **`add`**, click the arrow and choose the option **`rearrange`**. A modal window will open.
   1. Drag the field **`Content: Title`** so that it appears below **`Content: Screenshots`**.
   2. Click the button **`Apply (all displays)`**. The modal window will close.
7. Locate the heading **`Pager`**. Click on the link **`Paged output, full pager`** (or whatever is right beside the label **`Use pager`**). A modal window will open.
   1. Change the setting to *`Display all items`*.
   2. Click the button **`Apply (all displays)`**. The modal window will close.

Task Summary

Create a new view for your portfolio with a display for page which has a menu item. Update the view to include a icons for each item displayed.

8. Scroll to the top of the screen and click **Save**.

Your view is now configured and available from the main menu.

Configuring the Slide Show

At any time you can adjust the settings for the slide show for your view:

1. Navigate to the page where the view is displayed. Hover over the view and look for a cog icon in the top right corner. Click the cog and choose the option **edit view**. The view configuration screen will open.
2. Locate the heading **Format**. Click the link beside **Format** labelled **Settings**. (There is a heading Format and a Label format.) A modal window will open.
3. Configure your heart away.
4. Click **Apply (all displays)**. The modal window will close.
5. At the top of the views configuration screen, click **Save**.

Close, But No Cigar

Not all transitions available created by the jquery.cycle library are working in Drupal 7. Update your settings and then test the resulting slideshow. If you get "close" to perfect you may want to clone your view before making further changes. Sometimes it can be really hard to get exactly the right set of configuration options after changing it to something "wrong."

Choose the Front Page

You have to tell Drupal to use this new banner as the home page.

1. Navigate to Administration > Configuration > Site information.
2. Locate the setting for **Default front page**. Update the setting to `home` (the path for the view you just created).
3. Scroll to the bottom and click **Save configuration**.

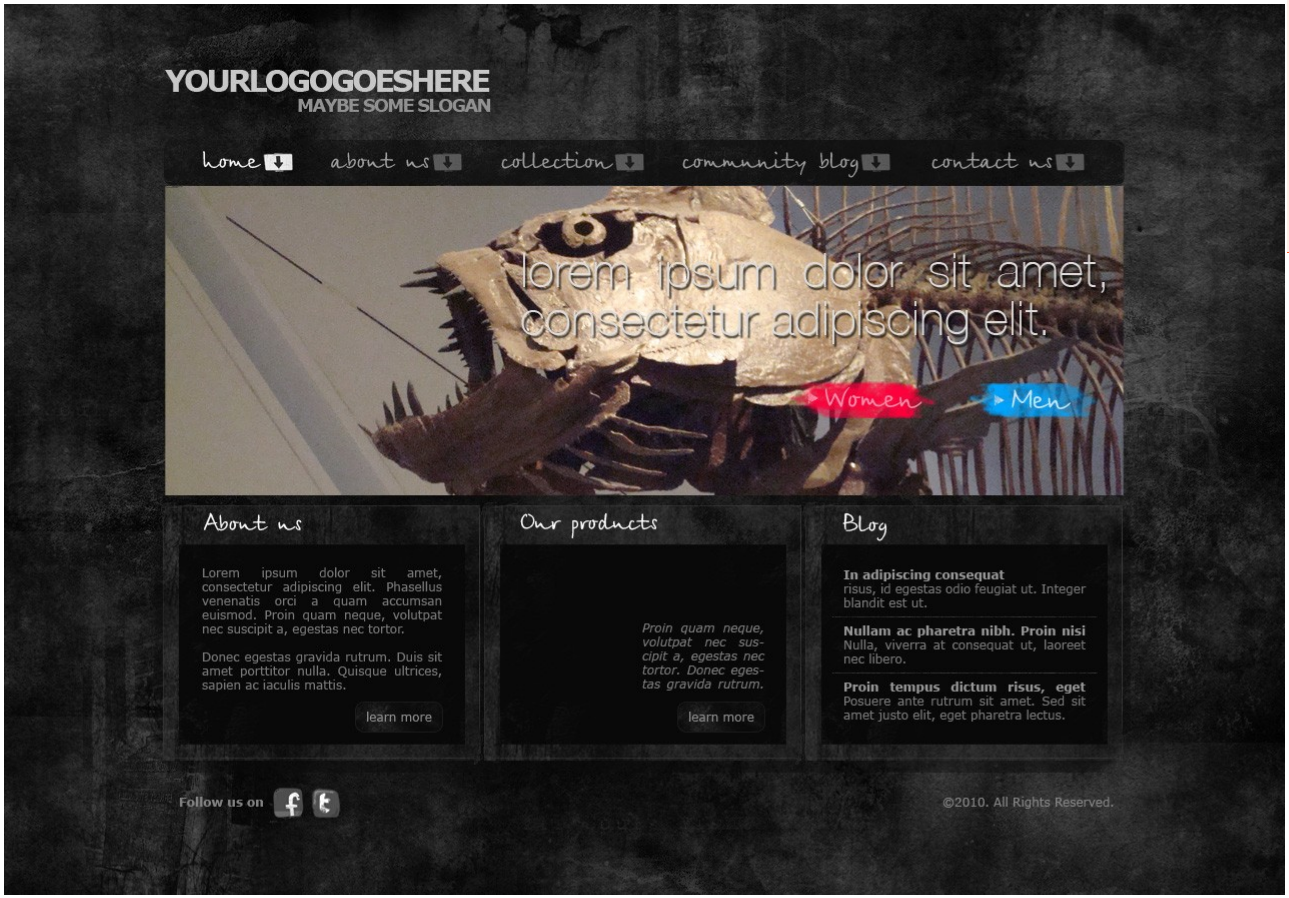
Learning More jQuery

If you decide jQuery is *totally* your thing and you want to learn more, I recommend the following resources:

- jQuery Tutorials
  http://docs.jquery.com/How_jQuery_Works
- jQuery in Action
  http://www.amazon.com/gp/product/1935182323
- jQuery Novice to Ninja
  http://www.amazon.com/jQuery-Novice-Ninja-Earle-Castledine/dp/0980576857/
- jQuery Fundamentals
  http://jqfundamentals.com/book/index.html
- jQuery for Absolute Beginners Video Series
  http://blog.themeforest.net/tutorials/jquery-for-absolute-beginners-video-series/
- Drupalize.Me Videos
  http://drupalize.me/videos/hands-on-videos/theming

Build the Theme

Portfolio themes are a dime-a-dozen in the WordPress world of themes. Most are simple and elegant and in muted, if not grayscale, colours. Like the WordPress portfolio themes, the design for this site relies on a dramatic dark background, white highlights and a hand writing-styled font for its appeal. In the original design a three column footer region was used.



Build a Design Library

I have a collection of screenshots that I've taken of various portfolio themes and sites which I keep around for inspiration. If you don't have a library of design ideas, I highly recommend starting one. It doesn't have to be complicated, just a folder with screenshots. If you include the URL in the screenshot you have an easy reference so that you know where you got the screen shot. Another neat tool is the Firefox extension Zotero. This plugin allows you to save an entire Web page to your local computer. This is especially useful if you want to be able to refer to how a page "worked" later.

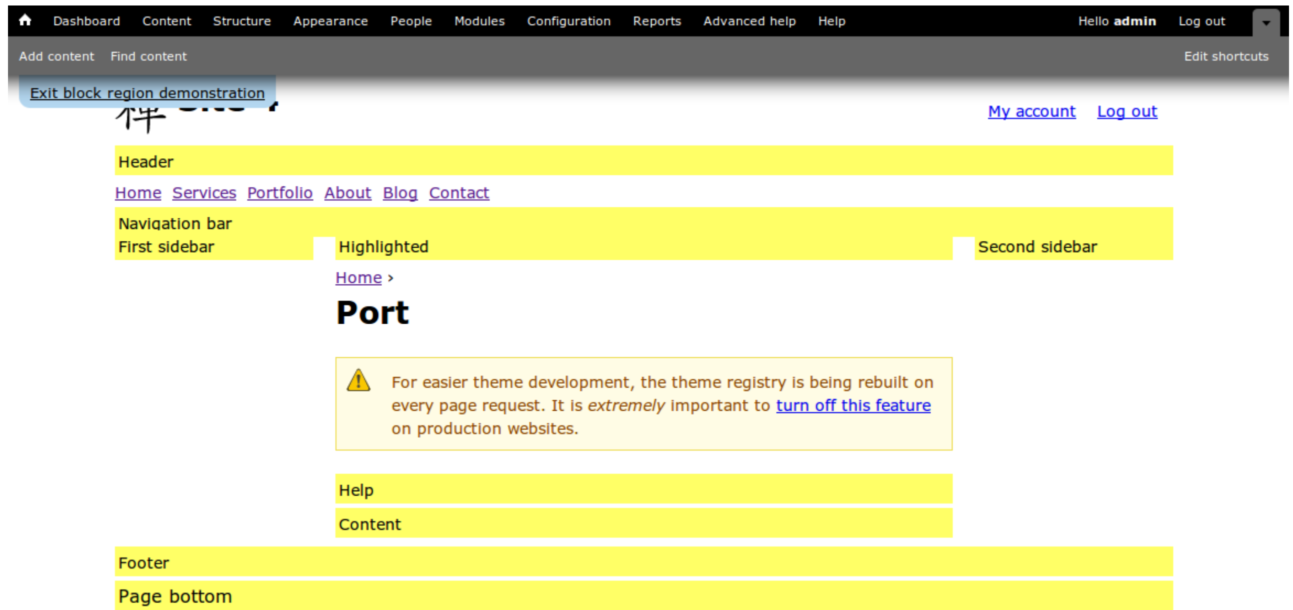
The final theme makes a few changes against the original design. (No, the original design did not have a dinosaur fish. I added that because it's freaking awesome.) The hand writing font (Birch CT) is replaced by a free font (Angelia). On the home page banner I've placed the text beside the banner instead of on top of it. The technique of how to place text on top of images is included in the portfolio gallery. The footer has four columns instead of three. The rest is more or less the same and the final home page is as follows:

Page-by-Page

The layout of this site is really straight forward: banner across the top, two columns for content (the narrower column on the left, the wider column on the right) and a footer. I haven't included the copyright notice at the bottom of the page, but you can add this into the page template yourself.

For this theme we will be using the base theme, Zen. It comes with nine basic regions stacked as follows:

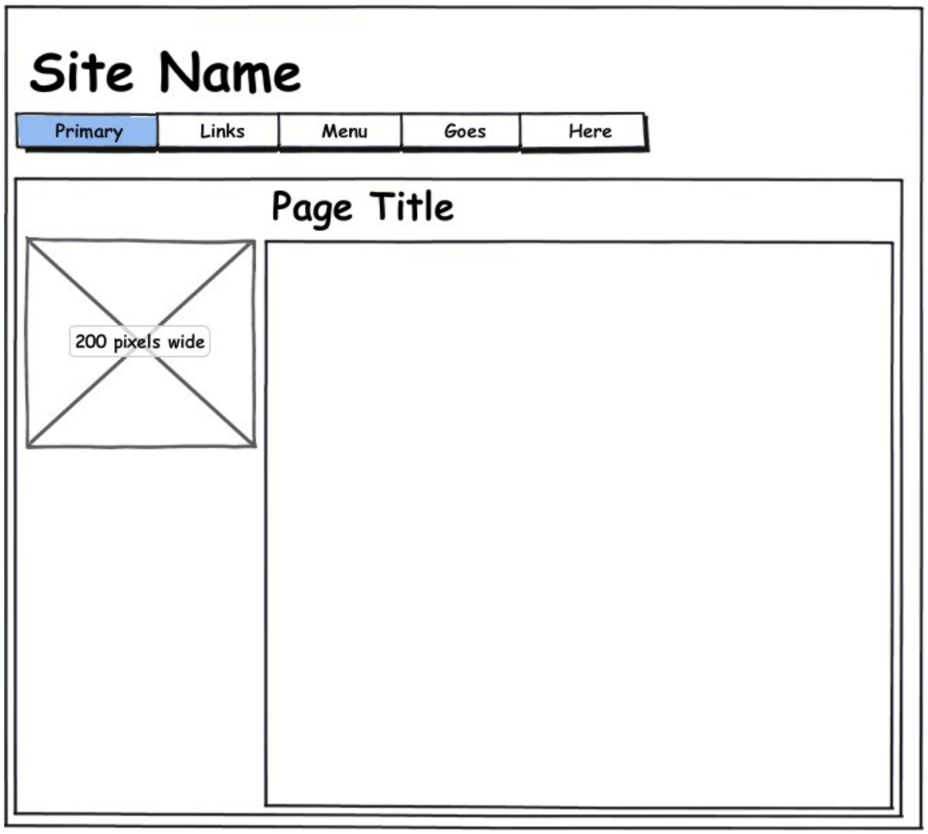
- Header – Full width (960 pixels by default).
- Navigation bar – Full width. By default the primary links appear above the navigation bar.
- Content – In columns.
    - First sidebar – 200 pixels.
    - Content Column – width varies based on visible sidebars
        - Highlighted – full width of the content column
        - Help – full width of the content column
        - Content – full width of the content column
    - Second sidebar – 200 pixels
- Footer – Full width.
- Page bottom – Full width.

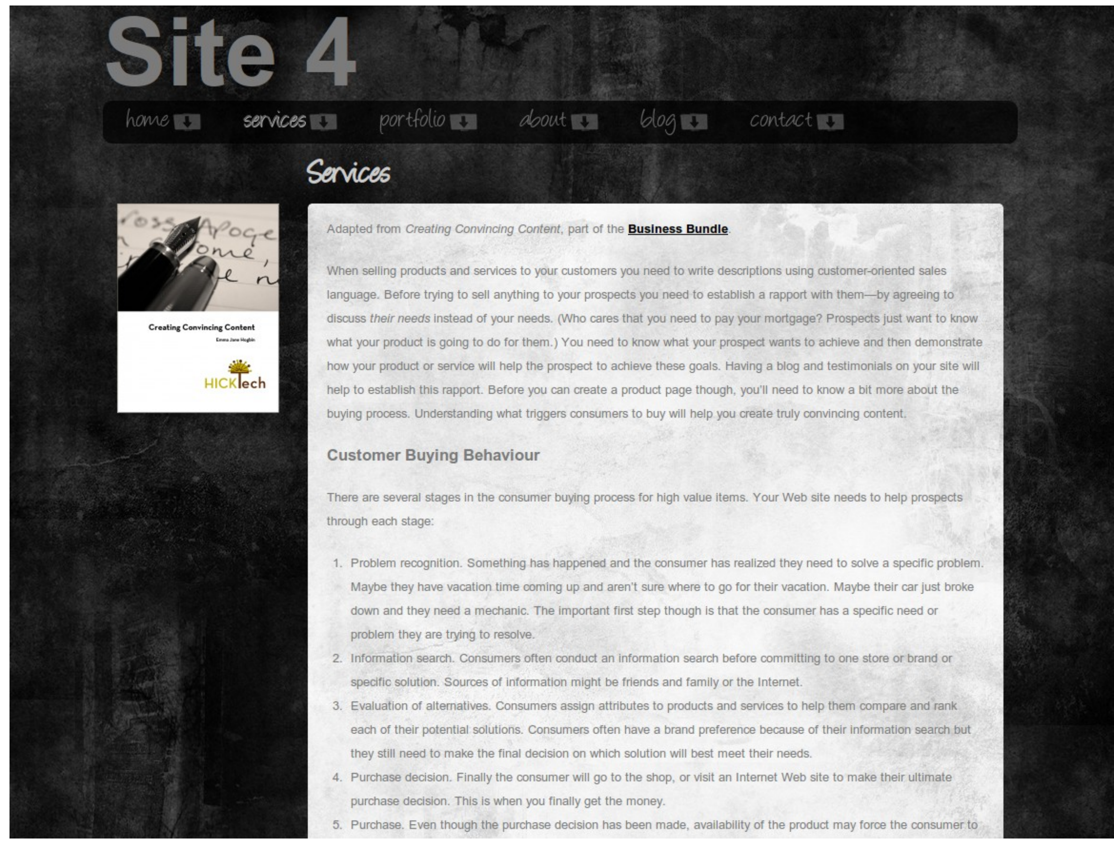


For our portfolio site these regions suit us just fine—which means (almost) no editing of the page template file!

Basic Page

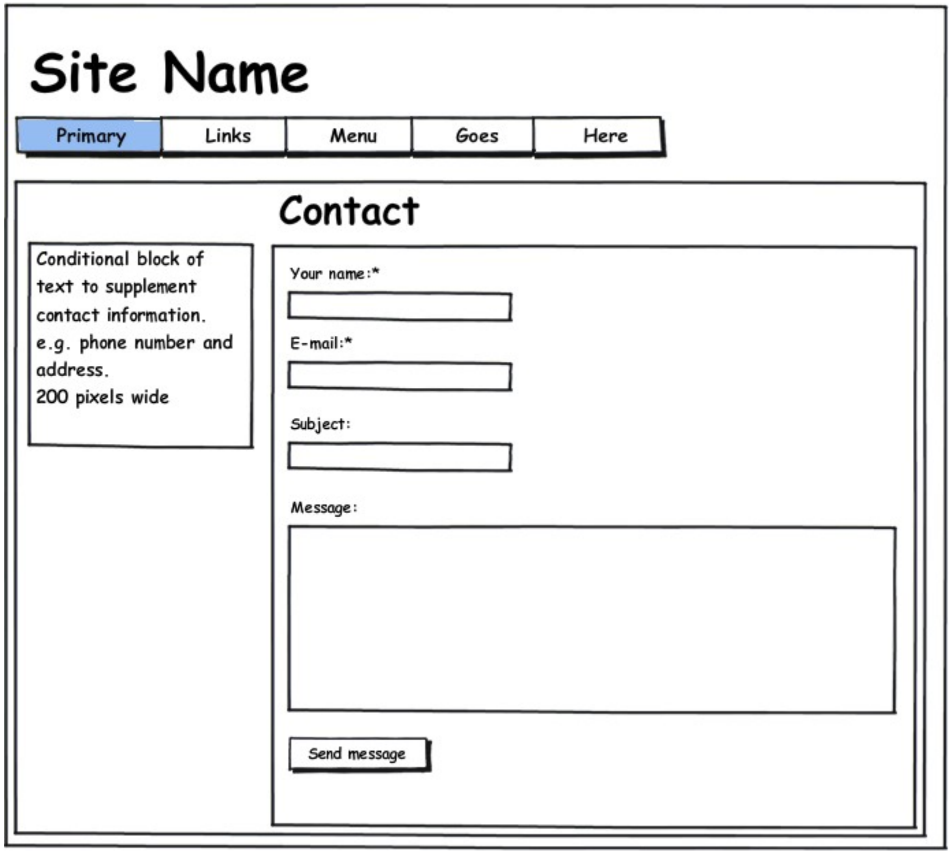
This site does not include sub-navigation for basic pages such as About and Services. To keep the width of the content region consistent from page-to-page the left-most column of basic pages displays an image. The heading is forced to the right so that it sits on top of the content column—this is the only type of page where the heading has to be forced over. All other page types have a block in the first sidebar which does the work for us.

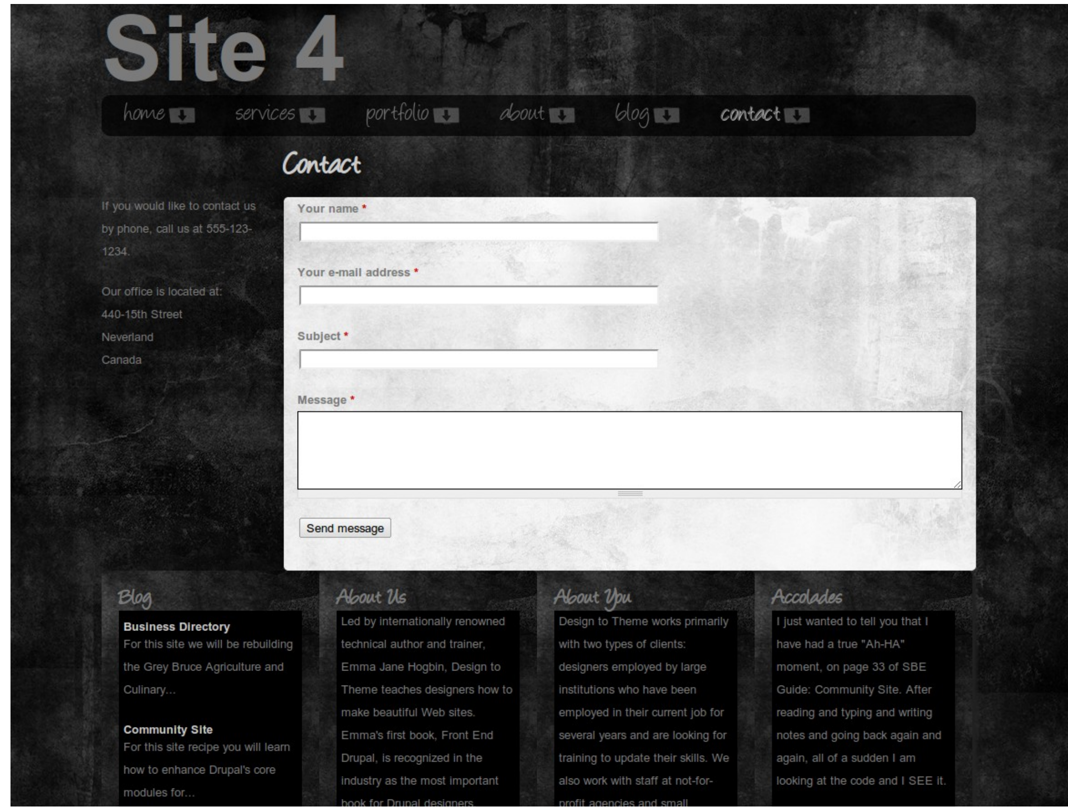




Contact

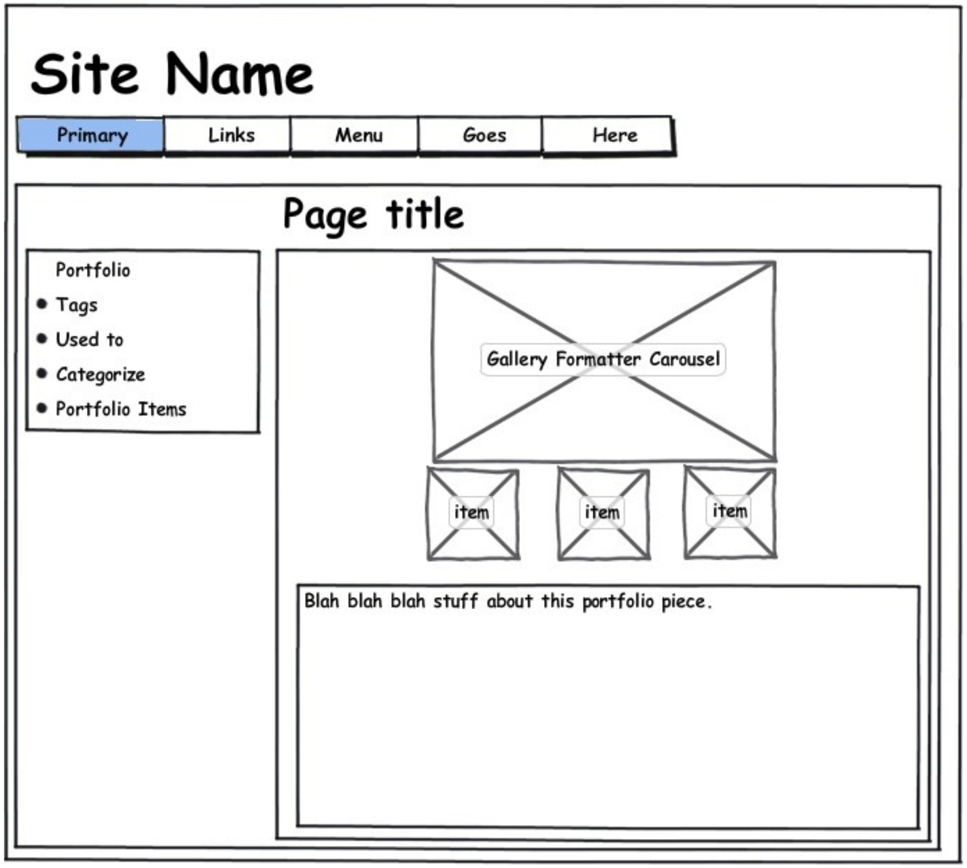
This page is visually similar to the basic page; however, it uses a custom block in the first sidebar. This conditional block includes a phone number and street address. The heading does not need to be pushed over manually for this page—the block in the first sidebar does the work for us.

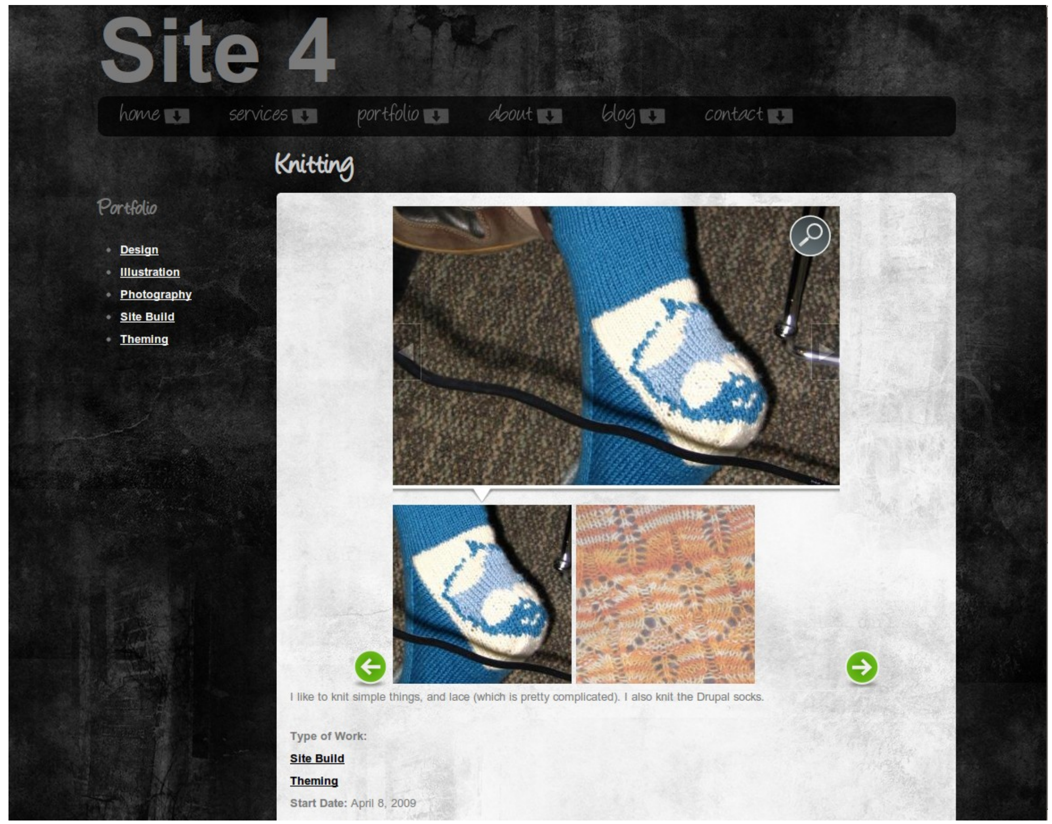




Portfolio Item

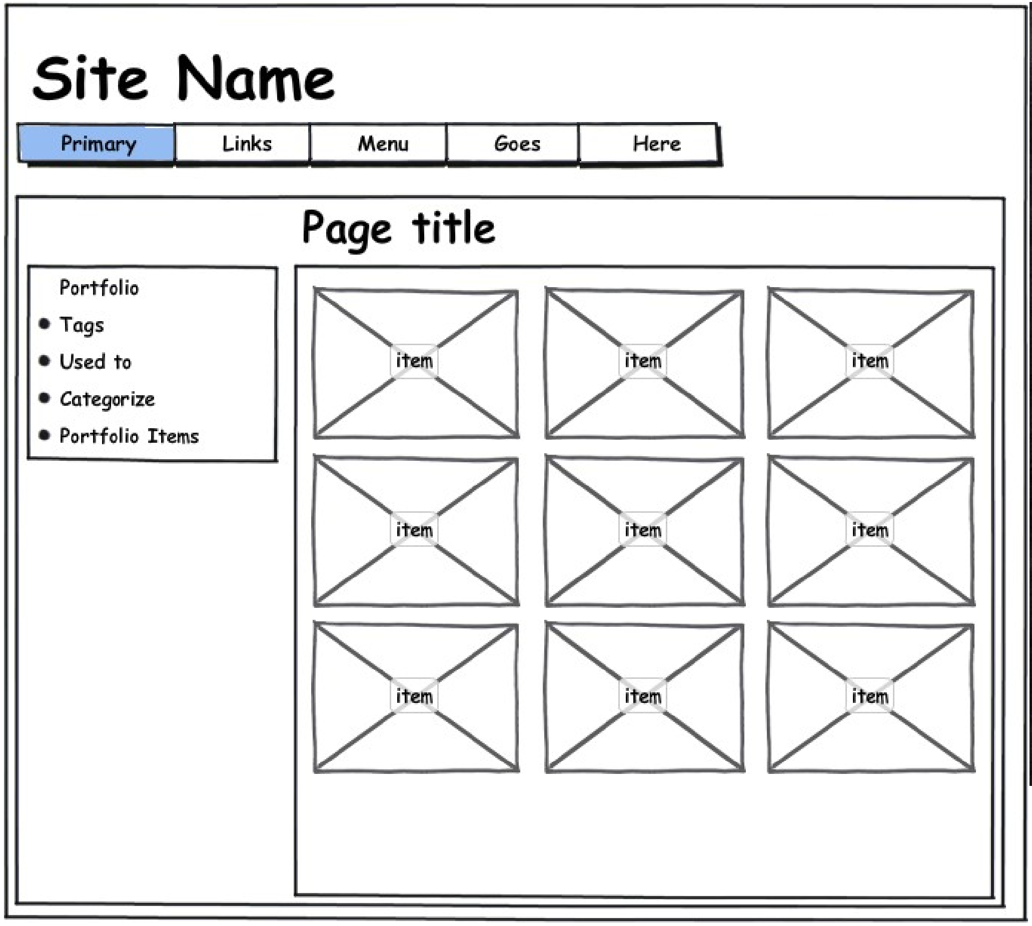
Like the Contact page, the portfolio item is a two-column layout with the first column held open by a conditional block. The block this time lists all tags used on the portfolio content type. The remainder of the display was configured as part of the site build.

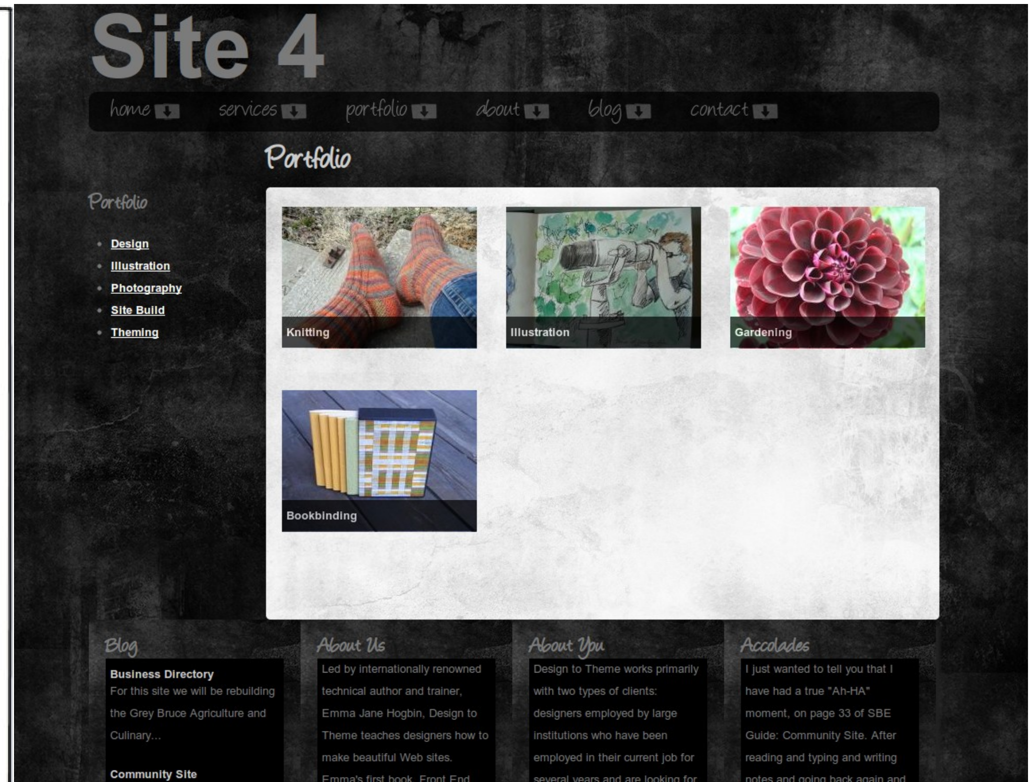




Portfolio Summary (Gallery)

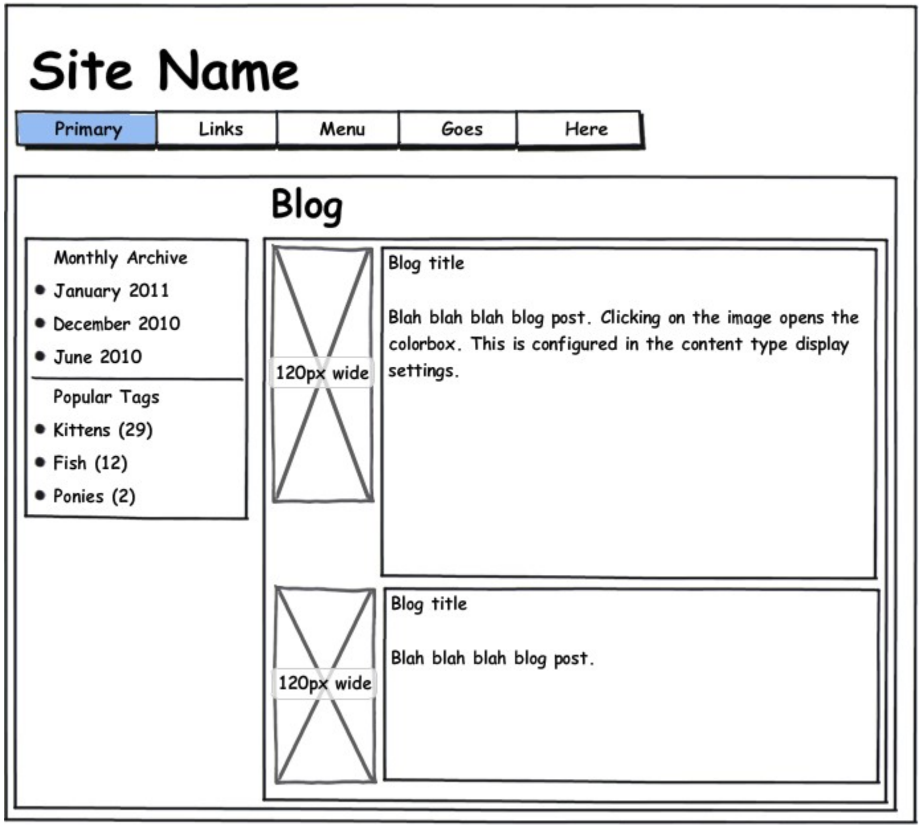
The gallery is a view including fancy CSS to stack text on top of the images. The first sidebar is held open by a conditional block which displays a view of all portfolio-related tags. The same display is used for taxonomy terms so make sure your blog posts each have a picture!





Blog

This two-column layout actually features a sneaky third column of images. The first sidebar is filled with the monthly archive and popular tag blocks—which are views created during the site build. A custom node template breaks articles into the left-right display with skinny images, the blog post's date and tags appearing on the left and content appearing on the right. There's a bit of additional CSS finessing but most of the heavy lifting was completed during the creation of views in the site build.



Assembling Design Assets

Most of the work for this theme is done with CSS. There were only a few assets that were extracted from the design file. I don't ever aim for pixel-perfect design across all browsers. That is the path to madness. Rather, I am for a pleasing (but not necessarily identical) experience in common, modern browsers.

From the Photoshop file I extracted the following images:

- Background image. Saved as a JPG.
- Dark primary navigation icon (down arrow) dark. Saved as a transparent PNG.
- Light primary navigation. Saved as a transparent PNG.

To create the content background I simply inverted the background image so that I had a light textured background instead of a dark one.

The font used in the design file isn't a freely licensed font, so I picked another from Font Squirrel which already had a font-face kit associated with it (http://www.fontsquirrel.com/fontface).

Extracting Images

Photoshop can be overwhelming for non-designers. Especially if there are a lot of layers and effects. To extract an image you want to use in your design, disable all layers you don't want in your final graphic. Then simply crop the image to only the portion you want to keep. In the case of the primary navigation icon I disabled the background image as well, making a slightly see-through button that could be applied to any background pattern or colour.

Task Summary

Extract the background image and two buttons from the design file. Create an inverted version of the background image for use as a light background.

Zen: Starter Kit and Base Theme

For this theme you'll be using the base theme Zen (www.drupal.org/project/zen). This base theme is one of the best documented themes available for Drupal. (It has an overwhelming amount of helpful documentation.) In previous workbooks I've covered the base theme NineSixty. The great thing about NineSixty is that it allows you to prepare a page template in HTML and then add Drupalisms via PHP.

The Zen theme uses a very different approach to theming—instead of building up a page template, we're going to modify a few CSS files which already have all of the CSS selectors typed out. You're also going rely more on what the Zen developers think is a good theme for our own theme—making very few changes to the template files.

If you like really really pared down HTML you may find Zen is "bloated" compared to what you're used to. Take a deep breath and get Zen. Learn to love all of the CSS selectors and the work that's already done for you. (And if you can't, that's okay. There are loads of other base themes that might be more appropriate for your style of coding.)

Create Your Theme

Working on your local computer to start, complete the following steps to create your minimum viable theme:

1. Download a copy of the base theme Zen (http://drupal.org/project/zen).
2. Unzip the base theme, `zen`. You will not edit, or place any additional files into this folder. It is for reference purposes only.
3. Locate the file `README-FIRST.txt` in the `zen` theme folder. Open the file and read it. You should already know everything in that file. Confirm that you do by reading it.
4. Create a new folder for your theme named `port` in the same parent directory as `zen`. For example: these two folders may both be on your desktop to start.
5. Open the folder `zen` and locate the folder labelled `STARTERKIT`.
6. Copy the contents of the folder `STARTERKIT` into your theme's folder. There are seven files and five sub-directories.

Task Summary

Download Zen from Drupal.org. Use the STARTERKIT to create your own theme.

Preparing the Starter Kit

Before you can enable this theme you need to make a few minor changes to the files so that your theme will work. A longer version of this set of instructions is included in the `README` file that you copied over from the `STARTERKIT`. Each of these steps is completed within your new theme folder, not the original zen folder.

1. Rename the file `STARTERKIT.info.txt` to `port.info`.
2. Using a text editor, open the file `port.info` and update the name and description for your new theme.

```
name = Port
description = A simple two column site in monochrome colors with a front page
banner, appropriate for portfolio and image-rich sites.
```

3. Save the file and exit. No other changes are needed at this time.
4. Copy the two background images and two button files you extracted from the design file into the folder `port/images`. (The four images are also available in the design folder with the PSD if you don't want to extract them yourself.)

That's it! Your starter kit is ready to use as a minimum viable theme.

Task Summary

In your new theme's folder, rename the info file and then update the info file to use a relevant name and description for your theme. Copy the images from your design file into your theme's images folder.

Upload Your Theme Files

You must now upload your theme to your Web server. Once you've connected to your Web server you will need to find the Drupal base installation. Depending on your server's configuration it may be in a directory labelled, "`public_html`" or something to that effect.

1. The base theme, `zen`, and all of its contents should be uploaded to the folder `sites/all/themes`.
2. Your new theme, `port`, and all of its contents should be uploaded to the folder `sites/default/themes`. If you have more than one Drupal site running on this server you may need to change "`default`" to a more specific site name.

Tip: Check the permissions of sites/default.

By design Drupal removes the ability for you to add files to any folder containing the settings file (`settings.php`). This means you can't create a sub-folder for your themes without adjusting the permissions. If you are having difficulty adding files with your FTP program to sites/default, check the permissions. Chances are good you will need to allow the owner to "write" to the folder. To be most secure, change the permission back to non-writable after you have created the sub-folder for your theme.

See: http://drupal.org/node/1077888 for more information.

Task Summary

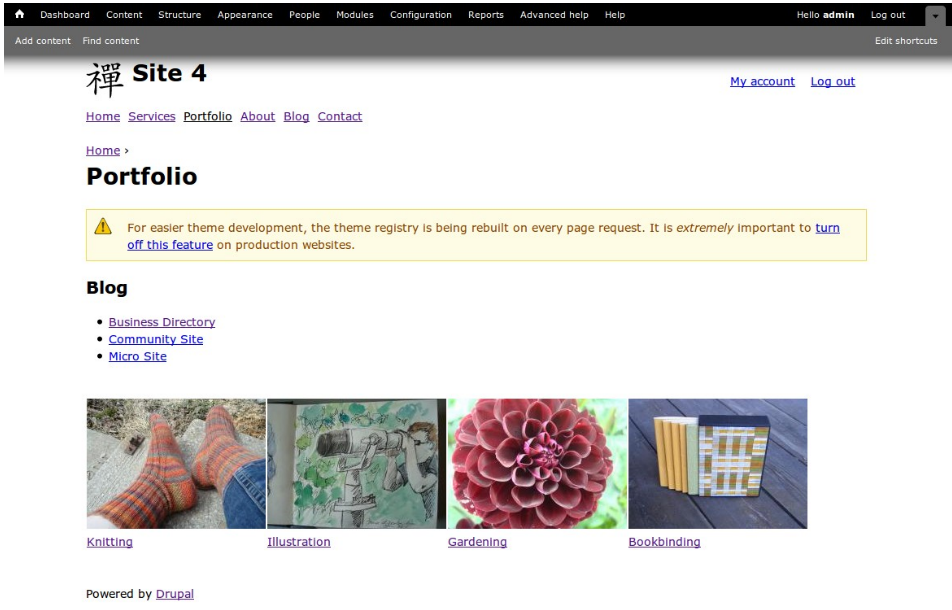
Upload your new theme, and its base theme, to your Web server.

Enable Your New Theme

Log in to your Drupal site as an administrative user and complete the following steps:

1. Using the admin dashboard, click on the tab labelled **`Appearance`**.
2. Scroll down to the bottom of the screen (where all the disabled themes live), beneath the Port theme, click the link **`Enable and set default`**.

The new theme has been applied. When you close the overlay you will see a stark site with blue and purple links and not a lot else. It's okay if your site isn't a perfect match for content that's displayed—at this point you just need to make sure that it's no longer the default theme being used on your site.



Theme Settings

We'll make a few adjustments to your theme to start:

1. Navigate to **`Administration`** > **`Appearance`**.
2. Below the theme Port, click on the link **`Settings`**.
3. In this fieldset **`Toggle display`**, disable the **`Logo`** page element. (The Port design doesn't use one, of course if *want* a logo, leave this enabled.)
4. In the fieldset **`Breadcrumb settings`**, change the drop down menu for **`Display breadcrumb`** to *`Only in admin section`*.
5. In the fieldset **`Theme development settings`** enable the option **`Rebuild theme registry and every page`** and change the **`Layout method`** to use *`Fixed layout`*.
6. Scroll to the bottom of the screen and click **`Save configuration`**.

The theme is now configured and ready for refinement and enhancement.

Task Summary

Enable your theme and adjust its core settings according to the instructions provided.

Blocks Into Regions

Sometimes things shift around a bit when you enable your new theme. Check to make sure your blocks are appearing in the correct region.

1. Navigate to **`Administration`** > **`Structure`** > **`Blocks`**.
2. Locate each of your blocks and make sure they are placed in the correct region and in the correct order.
3. Scroll to the bottom of the screen and click **`Save blocks`**.

Node Templates

This theme has a custom node template for basic pages and articles.

1. Copy the default node template from `zen/templates/node.tpl.php` to `port/templates/node.tpl.php`.
2. Create a second copy of the file and name it `node--article.tpl.php`. That's a double dash, not an em-dash.
3. Create a third copy of the file and name it `node--page.tpl.php`. That's a double dash, not an em-dash.

For these two templates you will be isolating the image field and printing it before the content. The CSS enhancements will then float the image and the content so that they are side-by-side. This emulates a sidebar region for basic pages which do not need a sidebar.

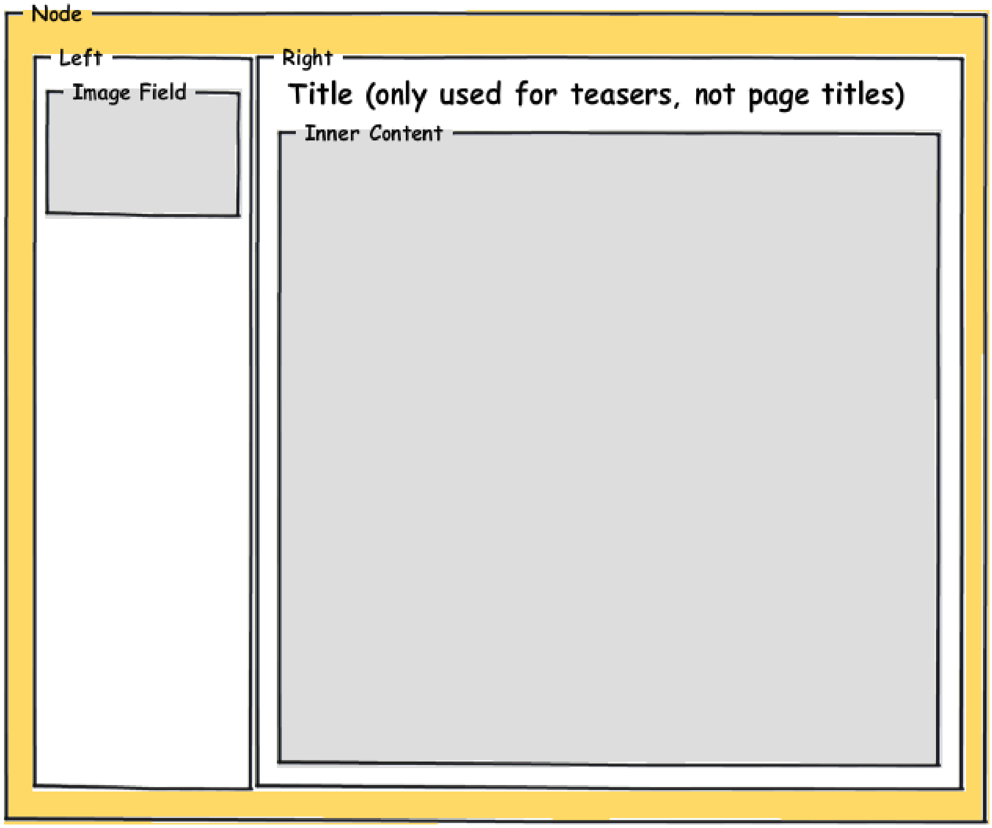
A node is a simple container with all content rendered and printed inside a second `<div>`. We are going to further split the template so that there is a left and right separation to divide the fields.

In the article template file we will also update the date display to omit the time. Some of my blogs are so neglected that I feel I ought to omit the year as well.

Task Summary

Copy the node template from the master Zen folder into your theme's templates folder. You do not have to put it in this sub-directory...it just keeps things neat and tidy.

Create two copies of the node template file and alter their contents according to the directions in the sections node--page.tpl.php and node--article.tpl.php



node--page.tpl.php

The complete template file is below, with the new lines marked in bold and highlighted. There are only five new lines. The yellow-highlighted lines (first and last) show the overall node container. The white-highlighted lines correspond to the new HTML divs.

```
<div id="node-<?php print $node->nid; ?>" class="<?php print $classes; ?> clearfix"<?php print $attributes; ?>>

<div class="left">
  <?php print render($content['field_image']); ?>
</div>

<div class="right">
  <?php print render($title_prefix); ?>
  <?php if (!$page && $title): ?>
    <h2<?php print $title_attributes; ?>><a href="<?php print $node_url; ?>"><?php print $title; ?></a></h2>
  <?php endif; ?>
  <?php print render($title_suffix); ?>

  <?php if ($unpublished): ?>
    <div class="unpublished"><?php print t('Unpublished'); ?></div>
  <?php endif; ?>

  <?php if ($display_submitted): ?>
    <div class="submitted">
      <?php print $submitted; ?>
    </div>
  <?php endif; ?>

  <div class="content"<?php print $content_attributes; ?>>
    <?php
      // We hide the comments and links now so that we can render them later.
      hide($content['links']);
      print render($content);
    ?>
  </div>
```

```
  <?php print render($content['links']); ?>
 </div>
</div><!-- /.node -->
```

With these very simple changes, you are finished with this template file. The CSS enhancements added later will snap the new HTML into visual position.

node--article.tpl.php

This template is very similar to the previous one, except there are more fields in the first column (image, date, links and taxonomy terms). The date has also been formatted to omit the time and display only month (F) day (j) and four-digit year (Y).

```
<div id="node-<?php print $node->nid; ?>" class="<?php print $classes; ?>
clearfix"<?php print $attributes; ?>>

 <div class="left">
  <?php print render($content['field_image']); ?>
  <?php print date("F j, Y", $created); ?>

  <?php if ($view_mode == "teaser") {
    print render($content['links']);
   } else {
    print render($content['field_tags']);
  } ?>
 </div>

 <div class="right">
  <?php print render($title_prefix); ?>
  <?php if (!$page && $title): ?>
    <h2<?php print $title_attributes; ?>><a href="<?php print $node_url; ?>"><?php
print $title; ?></a></h2>
  <?php endif; ?>
  <?php print render($title_suffix); ?>

  <?php if ($unpublished): ?>
    <div class="unpublished"><?php print t('Unpublished'); ?></div>
  <?php endif; ?>

  <div class="content"<?php print $content_attributes; ?>>

    <?php
      // We hide the comments and links now so that we can render them later.
      hide($content['comments']);
```

```
    print render($content);
  ?>
</div>

<?php print render($content['comments']); ?>
</div>

</div><!-- /.node -->
```

With these very simple changes, you are finished with this template file. The CSS enhancements added later will snap the new HTML into visual position, and with the site building you did earlier, the image will open with a lightbox effect when clicked on.

Now you're ready to crack open a text editor to add the CSS enhancements and custom node templates used by this design.

CSS Enhancements

There are a LOT of CSS files in your theme's folder. The documentation in the CSS folder is amazing. The README file is included as an Appendix at the end of this workbook. You should read it. It's useful. We will be making changes in the following CSS files:

- `css/blocks.css`
- `css/html-reset.css`
- `css/layout-fixed.css`
- `css/navigation.css`
- `css/nodes.css`
- `css/pages.css`
- `css/views-styles.css`

In addition to these files we'll be adding our own style sheet for fonts.

The advantage of using these style sheets is that the CSS selectors are already written for you. Want to change the colour of the links? Just open up the appropriate CSS file and add your colour. You don't need to remember if it's `:hover` before `:active` or whatever. All the thinking is done. I think this is *awesome*. But I also find it can get in the way sometimes because there's so much stuff. So it depends on the theme I'm building—sometimes I like NineSixty because it doesn't get in my way; and sometimes I like Zen because it does all of the thinking for me.

The following sections show you the CSS that was added to each of the Zen style sheets.

Task Summary

Apply the CSS enhancements outlined throughout this section. There are seven CSS files that will be modified, and one CSS file that will be created.

General

```
FILE: html-reset.css

/* Headings */
h1 {
  color: #ccc;
  font-size: 2em;
  line-height: 1.5em;
  margin: 0.75em 0;
}

/* Default link colours */
a:link, a:visited {
  color: white;
  font-weight: bold;
}

/* Link colours for the content region (light background) */
.region-content a:link, .region-content a:visited {
  color: black;
}

/* don't change the weight of the administrative toolbar links */
#toolbar a:link, #toolbar a:visited {
  font-weight: normal;
}

/* pink links when activated */
a:hover, a:focus, a:active,
.region-content a:hover, .region-content a:focus, .region-content a:active {
  color: #ef2b51;
}

/* removes the extra space from initial paragraphs */
/* resets the top margin that Drupal adds */
p:first-child {
```

```
  margin-top: -.5em;
}


FILE: pages.css

#page {
  color: #7b7b7b; /* default text colour is medium gray */
}

h2 a {
  text-decoration: none;
}

div.messages a {
  color: inherit;
}
```

Site Name

```
FILE: pages.css

h1#site-name,
div#site-name /* The name of the website */ {
  font-size: 100px;
  line-height: 1em;
}

#site-name a:link,
#site-name a:visited {
  color: #7b7b7b;
  text-decoration: none;
}

#site-name a:hover,
#site-name a:focus {
  color: white;
  text-decoration: none;
}

/* The title of the page != site heading */
/* reset the page title here on inner pages */
.not-front h1.title {
  margin-left: 215px;
  font-size: 3em;
}

/* when there is a side bar don't shift the page title over */
.sidebar-first h1.title {
  margin-left: 0;
}
```

Footer Blocks

By default the blocks in the footer region will stack on top of each other. This CSS snippet makes the blocks one quarter of the width of the footer region, and then floats the blocks so they line up side-by-side. Additional styling was added to give the blocks a background image, a drop shadow, and pink links for the highlights.

```
FILE: blocks.css

.region-footer .block {
/* background image */
  background: url(../images/background.jpg);

/* drop shadow */
  -moz-box-shadow: 5px 5px 5px #222;
  -webkit-box-shadow: 5px 5px 5px #222;
  box-shadow: 5px 5px 5px #222;

/* rounded border */
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  border-radius: 5px;

/* quarter width, floated left */
  float: left;
  padding: 2%;
  width: 21%;
}

/* black inner background */
.region-footer .block .content {
  background: black;
  padding: 5px;
}

/* gray links */
.region-footer .block a {
```

```
  color: #ccc;
  text-decoration: none;
}

/* pink activated links */
.region-footer .block a:hover,
.region-footer .block a:active,
.region-footer .block a:focus {
  color: #ef2b51;
}
```

Navigation

The main navigation bar is basically just a decorated version of the Zen navigation bar.

```
FILE: navigation.css

li a.active {
  color: #ccc; /* make current pages light gray in the navigation */
}

/*
 * Navigation bar
 */
#navigation {
 font-family: 'AngelinaRegular'; /* you'll create this font-face later */
 font-size: 30px;
 text-transform: lowercase;
}

/* the main nav bar is transparent black with rounded corners */
#navigation #main-menu {
  background: black;
  -webkit-border-radius: 20px;
  -moz-border-radius: 20px;
  border-radius: 10px;
  opacity: 0.7;
  padding: 10px;
}

/* primary links have an icon off to the right and are medium gray */
#navigation ul.links li a {
  background: transparent url(../images/button-default.png) no-repeat center right;
  color: #868686;
  font-weight: normal;
  padding: .5em 1.1em .5em .5em;
  margin-right: 20px;
  text-decoration: none;
```

```
}

/* when activated, the links become white and the light icon is swapped in */
#navigation ul.links li a:hover,
#navigation ul.links li a:focus,
#navigation ul.links li a:active {
  background: transparent url(../images/button-active.png) no-repeat center right;
  color: white;
}
```

We also need to adjust the height a bit to deal with the small hand writing font.

```
FILE: layout-fixed.css

#navigation {
  float: left; /* LTR */
  width: 100%;
  margin-left: 0; /*LTR */
  margin-right: -100%; /* Negative value of #navigation's width + left margin. */
  padding: 0; /* DO NOT CHANGE. Add padding or margin to #navigation .section. */
  height: 4em; /* The navigation can have any arbitrary height. We picked one
                    that is the line-height plus 1em: 1.3 + 1 = 2.3
                    Set this to the same value as the margin-top below. */
}

.with-navigation #content,
.with-navigation .region-sidebar-first,
.with-navigation .region-sidebar-second {
  margin-top: 4em; /* Set this to the same value as the navigation height above. */
}

.region-sidebar-first .section {
  margin: 5em 20px 0 0; /* LTR */
  padding: 0;
}
```

Nodes With Fake Sidebar

The basic pages throw their image field off to the left to simulate a sidebar. When there is actually a sidebar in place the content area shrinks.

```
FILE: nodes.css

/* fake sidebars for basic pages */
/* appears only on content type: page where there are no sidebars present */
.node-type-page.no-sidebars .left {
  float: left;
  width: 200px;
}
.node-type-page.no-sidebars .right {
  float: left;
  margin-left: 0;
  padding: 0;
}


/* blog articles already have a sidebar and the image is narrower */
.node-type-article .left, .page-blog .left {
  float: left;
  width: 150px;
}
.node-type-article .right, .page-blog .right {
  float: left;
  width: 560px;
}

/* the gallery formatter used on portfolio pages
   does stupid things in some browsers and
   needs its own padding across the top */
.node .galleryformatter {
  padding-top: 15px;
}
```

```
FILE: pages.css
.not-front .region-content, .node-type-page .region-content .right {
  background: url("../images/background-light.jpg") repeat top left;
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  border-radius: 5px;
  padding: 0 15px 15px 15px;
}

.page-contact .region-content {
  margin-top: -15px;
}

.node-type-page.no-sidebars .region-content .right {
  padding: 20px;
  width: 690px;
}

.node-type-page .region-content {
  background: none; /* because of the fake side bar */
}
```

Stretchy Background Images

This is a two-part effect. Start by modifying the CSS file `pages.css` as follows:

```
FILE: pages.css

html, body {
  padding: 0;
  margin: 0;
}

#page-bg-img { /* an IMG needs to be added to the page template too */
  width: 100%;
  overflow: hidden;
  z-index: -999;
  min-height: 100%;
  min-width: 1024px;
  height: auto;
  position: fixed;
  top: 0;
  left: 0;
}
```

Next you will need to update the template file `page.tpl.php` to include an image that will get stretched. Except your `templates` folder is EMPTY! Oh no! Don't panic.

1. In the main zen folder find the file `page.tpl.php`.
2. ***Copy*** it into your theme's `templates` folder.
3. Add the following line as the very first line.

```
<img src="<?php print $base_path . $directory;?>/images/background.jpg" id="page-bg-img" alt=""/>
```

4. Save and close the file.

Task Summary

Copy the default page template from the master Zen folder and place it in your theme's templates folder. Update it according to the instructions in Stretchy Background Images.

Views

This file is empty by default.

```
FILE: views-styles.css

/* Padding to content for Views */
.view-blog.view-display-id-page .views-row,
.view-monthly-archive .views-row {
  margin-bottom: 35px;
  padding-top: 15px;
}
```

View: Banner – side-by-side fields

```
FILE: views-styles.css

/* Banner Slideshow on the Home Page */
#views_slideshow_cycle_main_home_banner-page .views-field-field-screenshots img {
  -webkit-border-top-left-radius: 5px;
  -webkit-border-bottom-left-radius: 5px;
  -moz-border-radius-topleft: 5px;
  -moz-border-radius-bottomleft: 5px;
  border-top-left-radius: 5px;
  border-bottom-left-radius: 5px;
}

#views_slideshow_cycle_teaser_section_home_banner-page {
 height: 200px !important;
 overflow: hidden;
 position: relative;
}

#views_slideshow_cycle_main_home_banner-page .views-field-title,
#views_slideshow_cycle_main_home_banner-page .views-field-body {
  position: relative;
  top: -200px;
```

```
  margin-left: 820px;
  width: 140px;
}

#views_slideshow_cycle_main_home_banner-page .views-field-body {
  color: #ddd;
}

#views_slideshow_cycle_main_home_banner-page .views-field-title a {
  text-decoration: none;
}

#views_slideshow_cycle_main_home_banner-page .views-field-title a:hover,
#views_slideshow_cycle_main_home_banner-page .views-field-title a:active,
#views_slideshow_cycle_main_home_banner-page .views-field-title a:focus {
  color: #ef2b51;
}
```

View: Image Gallery – stacked fields

```
FILE: views-styles.css

.sidebar-first table.views-view-grid.cols-3 {
  /* chrome collapses the table margin which is propping up the top padding */
  border-collapse: separate;
  margin: 0; /* chrome collapses the table margin which is propping up the top
padding */
  padding: 1.5em 0;
  position: relative;
}

.sidebar-first .views-view-grid.cols-3 .views-field-title {
  background: black;
  color: white;
  left: 0;
  padding: 5px;
  position: relative;
```

```
  opacity: .7;
  top: -44px;
}

.sidebar-first .views-view-grid.cols-3 .views-field-title a {
  text-decoration: none;
}

.sidebar-first .views-view-grid.cols-3 .col-2 {
  padding: 0 30px;
}

.sidebar-first .views-view-grid.cols-3 .views-field-title a,
#views_slideshow_cycle_main_home_banner-page .views-field-title a {
  color: white;
  display: block;
}

.sidebar-first .views-view-grid.cols-3 .views-field-title:hover,
.sidebar-first .views-view-grid.cols-3 .views-field-title:active,
.sidebar-first .views-view-grid.cols-3 .views-field-title:focus {
  color: #ef2b51;
  opacity: 1;
}
```

Fonts

The @font-face definition is usually done in the file `html-reset.css`. I put it in its own file because once the font is declared, you never really need to look at it again.

```
FILE: port.info
stylesheets[all][]        = css/fonts.css

FILE: fonts.css
@font-face {
  font-family: 'AngelinaRegular';
  src: url('../fonts/angelina-webfont.eot');
  src: url('../fonts/angelina-webfont.eot#iefix') format('embedded-opentype'),
    url('../fonts/angelina-webfont.woff') format('woff'),
    url('../fonts/angelina-webfont.ttf') format('truetype'),
    url('../fonts/angelina-webfont.svg#AngelinaRegular') format('svg');
  font-weight: normal;
  font-style: normal;
}

FILE: html-reset.css
#page {
  font-size: 0.8em; /* 16px x .75 = 12px */
  line-height: 2em; /* 12px x 1.5 = 18px */
}
body {
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
}

FILE: pages.css
h2.title, /* Comment section heading */
h2.comment-form, /* Comment form heading */
h3.title /* Comment title */ {
  font-family: 'AngelinaRegular';
  font-size: 2em;
  margin: 0;
}
```

With the styles in place you're ready to drop in the font kit from Font Squirrel.

1. Create a new sub-folder in your theme named `fonts`.
2. Unzip the Font Squirrel font package and dump all of its contents into the `fonts` folder.

That should be it. If your font doesn't appear for some reason it's probably because you've got the naming wrong.

1. Open the CSS file that comes with the font kit.
2. Copy all of the styles.
3. Open your theme's file `fonts.css`.
4. Replace the font declaration you have with the styles from the font kit.
5. Fix the directory location in the CSS by adding `../fonts` in front of the font file names.
6. Save and close the file `fonts.css`.
7. Open the file `pages.css` and make sure the `font-family` is the same name that's listed in the revised `fonts.css` file. If it is not identical (case and all), your font won't work. Update the name if you need to and save the `pages.css` file.

Last Step

There's one last thing you need to do before launching your site: turn off the theme setting that keeps rebuilding your theme registry.

1. Navigate to **`Administration`** > **`Appearance`**.
2. Below the theme Port click on the link **`Settings`**.
3. Locate the heading **`Theme development settings`** and disable the setting **`Rebuild theme registry on every page`**.
4. Scroll to the bottom of the page and click **`Save`**.

Summary

Throughout this site building workbook you learned how to create a portfolio Web site with the following features:

- Custom content type for portfolio pieces.
- Views, views and more views.
- Light box overlay.
- Rotating banner.
- Cropped images.

With the theme you were also able to see how little Drupal “code” is involved in building a theme with the Zen starterkit. By planning ahead with the site build, and using CSS enhancements there was virtually no PHP required to build this theme. Yes, it was a pretty simple layout, but a lot of sites are a two or three column layout that would be a perfect for fit for Zen.

I look forward to seeing the Web sites you make with this package!

About

Emma Jane Hogbin (drupal.org user emmajane) is an internationally renowned technical author and trainer who specializes in teaching people, just like you, to make beautiful Web sites. Emma's first book, *Front End Drupal*, is recognized in the industry as the most important book for Drupal designers. She has been teaching internet technologies since 2002 and has been building Web sites for over ten years.

A frequent speaker at technical conferences in Canada, the US and Europe and Emma also likes single malt whiskey. These two things are probably not related. Her presentations have taken her to France, Belgium, Hungary, Canada, New Zealand, England and the United States. In addition to her engaging conference presentations, Emma has also worked as a technical college instructor at Humber College and Seneca College, and has worked on curriculum development for Humber College and the Ubuntu Linux distribution.

Emma encourages non-traditional participation in technology through craft and believes that everyone is capable of mastering the tools that surround them. To help engage new ways of participating in technology, she open sourced one of her knitting patterns so that you can make your very own Drupal Socks (as featured in CRAFTzine). She is the recipient of the Google Diversity Award for helping to encourage women in technology and is the sponsor of Creative Use of Technology which is awarded every year to a female student at West Hill Secondary School in Owen Sound, Canada.

Accolades

“Emma is an amazing teacher.” — Betty

“Thank you for sharing your experience through e-books in addition to seminars and presentations. It's really helpful to have short, **easy-to-use examples to learn from as well as refer back to** while trying to develop good Drupal theming skills.” — Spence

“Emma Jane worked with me on a dramatically ambitious Drupal project a couple of years ago. She managed to keep my overactive imagination in check so we focused on realistic goals and milestones, all the while **making me feel like I could get my hands dirty** in the project. She was timely, proficient, and a joy to work with.” — Kim Werker, founder of CrochetMe.com

“Emma Jane combines the perfect amount of predictability and spontaneity...her technique has **sparked my excitement** about developments in technology and has inspired me to engage in new projects. She presents herself as very approachable and always answers questions thoroughly, making sure that the user feels comfortable and at ease.” — Jorge Castro, External Developer Relations, Canonical Ltd.

“Taking your course is one of the best investments I have made.” — Louise

Services

Emma Jane Hogbin is the founder of a great little Drupal consulting and training agency, Design to Theme. She makes theming Drupal easier, faster and more profitable.

- Drupal Site Building Consulting: There are thousands of Drupal modules out there that will get you from idea to finished Web site faster and with higher profits. With a Site Building Consultation we can help you choose (install and configure) the best modules for your next project. If you're tired of handing over all your profits to your programmer we need to talk.
- Support for Small Businesses and Designers: Did you get in over your head a little bit with a project? We can help you get unstuck with gentle technical support that will make you feel smart and wonderful and capable of taking back control of your Drupal project.
- Drupal Training: Drupal site building and theming training sessions are available on-line. Check the web for a list of upcoming workshops. Custom training is also available.

Appendix A: Zen's CSS README File

```
ZEN'S STYLESHEETS
-----------------

Don't panic!

There are 25 CSS files in this sub-theme, but its not as bad as it first seems:
- The drupal7-reference.css is just a reference file and isn't used directly by
  your sub-theme. See below.
- There are 7 CSS files whose names end in "-rtl.css". Those are CSS files
  needed to style content written in Right-to-Left languages, such as Arabic and
  Hebrew. If your website doesn't use such languages, you can safely delete all
  of those CSS files.
- If you aren't using this theme while doing wireframes of the functionality of
  your sub-theme, you can remove wireframes.css from your sub-theme's .info file
  and delete the file as well.

That leaves just 16 CSS files. (Okay, still quite a few, but better than 25.)

- Instead of one monolithic stylesheet, your sub-theme's CSS files are organized
  into several smaller stylesheets that are grouped to allow cascading across
  common Drupal template files.
- The order of the stylesheets is designed to allow CSS authors to use the
  lowest specificity possible to achieve the required styling.


ORDER AND PURPOSE OF DEFAULT STYLESHEETS
----------------------------------------

First off, if you find you don't like this organization of stylesheets, you are
free to change it; simply edit the stylesheet declarations in your sub-theme's
.info file. This structure was crafted based on several years of experience
theming Drupal websites.

- html-reset.css:
```

```
  This is the place where you should set the default styling for all HTML
  elements and standardize the styling across browsers. If you prefer a specific
  reset method, feel free to add it.

- layout-fixed.css:
- layout-liquid.css:
  Zen's default layout is based on the Zen Columns layout method. The
  layout-fixed.css file is used by default and can be swapped with the
  layout-liquid.css file. These files are designed to be easily replaced. If you
  are more familiar with a different CSS layout method, such as Blueprint or
  960.gs, you can replace these files with your choice of layout CSS file.

- page-backgrounds.css:
  Layered backgrounds across scattered divs can be easier to manage if they are
  centralized in one location.

- tabs.css:
  While most of the CSS rulesets in your sub-theme are guidelines without any
  actual properties, the tabs stylesheet contains actual styling for Drupal
  tabs, a common Drupal element that is often neglected by site desiners. Zen
  provides some basic styling which you are free to use or to rip out and
  replace.

- pages.css:
  Page styling for the markup in the page.tpl.php template.

- blocks.css:
  Block styling for the markup in the block.tpl.php template.

- navigation.css:
  The styling for your site's menus can get quite bulky and its easier to see
  all the styles if they are grouped together rather then across the
  header/footer sections of pages.css and in blocks.css.

- views-styles.css:
  Views styling for the markup in various views templates. You'll notice this
  stylesheet isn't called "views.css" as that would override (remove) the Views
  module's stylesheet.
```

```
- nodes.css:
  Node styling for the markkup in the node.tpl.php template.

- comments.css:
  Comment styling for the markup in the comment-wrapper.tpl.php and
  comments.tpl.php templates.

- forms.css:
  Form styling for the markup in various Drupal forms.

- fields.css:
  Field styling for the markup produced by theme_field().

- print.css:
  The print styles for all markup.

- ie7.css:
- ie6.css:
  The Internet Explorer stylesheets are added via conditional comments. Many CSS
  authors find using IE "conditional stylesheets" much easier then writing
  rulesets with CSS hacks that are known to only apply to various versions of
  IE. The full conditional comment syntax can be found on Microsoft's website:
  http://msdn.microsoft.com/en-us/library/ms537512.aspx
  An alternative method presented by Paul Irish can be found at
  http://paulirish.com/2008/conditional-stylesheets-vs-css-hacks-answer-neither/

In these stylesheets, we have included all of the classes and IDs from this
theme's tpl.php files. We have also included many of the useful Drupal core
styles to make it easier for theme developers to see them.


DRUPAL CORE'S STYLESHEETS
-------------------------
Many of Zen's styles are overriding Drupal's core stylesheets, so if you remove
a declaration from them, the styles may still not be what you want since
Drupal's core stylesheets are still styling the element. See the
```

```
drupal7-reference.css file for a complete list of all Drupal 7.x core styles.
```