DZone Refcardz

Drupal

Getting Started with **Drupal**

By Matt Vance

ABOUT DRUPAL

Drupal is an open source content management system and content management framework written in PHP. Drupal is used to power a variety of high-profile web sites, including sites for Popular Science (http://www.popsci.com/), Ubuntu (http://www.ubuntu.com/), and the Electronic Frontier Foundation (http://www.eff.org/).

Drupal is designed to be modular, with a number of optional modules included in the default "core" Drupal package. These modules serve to extend and enhance the capabilities of Drupal core. In addition, thousands of contributed modules are available for download from the central Drupal community website, Drupal.org.

Drupal is most often run on a LAMP server stack (an acronym for Linux, Apache, MySQL, and PHP). However, Drupal is not limited to a single architecture and can be run under Apple's OS X or Microsoft Windows. Alternative web servers supported by Drupal include Microsoft IIS and lighttpd. In addition to MySQL, Drupal can also be run using PostgreSQL for database storage.

> **Hot Tip**
>
> For a more thorough PHP reference, see the PHP Refcard (http://refcardz.dzone.com/refcardz/php). Drupal also includes the open source jQuery JavaScript library to simplify the creation of dynamic behavior within a user's browser. For help using jQuery, check out the jQuery Selectors Refcard. (http://refcardz.dzone.com/refcardz/jquery-selectors).

This Drupal Refcard will provide useful information to help you perform three of the common roles associated with setting up a new Drupal site: site building, theming, and development. This Refcard focuses on Drupal 6, the latest major release available as of this writing.

DRUPAL SITE BUILDING

Drupal can be used to create just about any type of website, from simple single-author blogs to multi-user social networking communities. Along with Drupal's flexibility comes a steep learning curve. First time site builders can quickly become overwhelmed by the thousands of available contributed modules.

While the Drupal community strives to work together to create user friendly modules that combine together well,

not all modules are created equal. Since developers can't be expected to test their modules in combination with every other available module, there are bound to be incompatibilities. It is important, especially for first time site builders, to evaluate modules on a test site before installing them on a production site.

Not all Drupal modules are ready for prime time, but some modules have been around long enough and are useful for enough applications to warrant being on a site builder's short list.

> **Hot Tip**
>
> Sign up for an account on Drupal.org. Some areas and tools on the site are only available to logged in users, plus it is an important first step toward getting involved in the "community" that develops, supports, and steers the Drupal project. Get involved on the forums; both asking and answering questions can help you get up-to-speed more quickly.

CCK

The Content Construction Kit module, also known as CCK (http://drupal.org/project/cck), allows site builders to extend beyond the default content types that are included with Drupal. Not only can the default page and story content types (sometimes called "node" types) be rearranged or customized with additional fields, but completely new content types can also be created. CCK includes several sub-modules for adding a variety of custom field types such as select boxes, check boxes, text boxes, and more. In addition, other contributed modules, such as Date (http://drupal.org/project/date), FileField (http://drupal.org/project/filefield), and ImageField
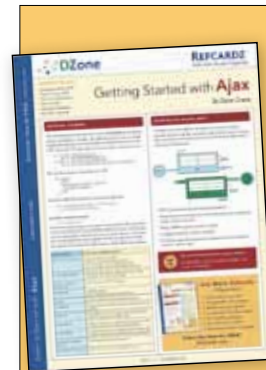
(http://drupal.org/project/imagefield), can be downloaded separately to add more complex fields to your custom content types.

Views

The Views module (http://drupal.org/project/views) allows site builders a great deal of control and flexibility in presenting lists of content, also called "views", on a site. The Views module provides a graphical user interface that can output nodes, users, comments, and more. Views can be presented as pages, as RSS feeds, or "blocks" (the often small pieces of content that are typically reused on various pages in sidebars, headers, and footers). The individual items that make up a view can be customized in terms of sort order, fields displayed, and the format of the list (such as unordered lists or tables).
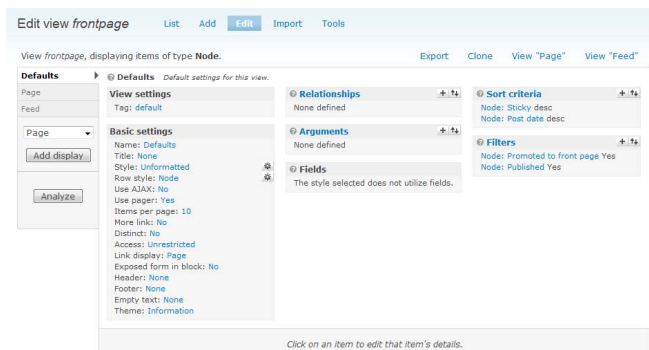
The Views module interface can take some time to master, but it is worth the investment. The flexibility of Views can be useful not just to site builders but also to developers, since views can be exported and bundled with modules.

Add on modules such as the Views Bonus Pack (http://drupal.org/project/view_bonus) can extend the Views module even more to present content as CSV files, Microsoft Word DOC files, or as XML.

> **Hot Tip**
> When installing new contributed modules and themes, resist the temptation to place the files in the default /modules or /themes directories. To make future upgrades of Drupal easier, create a /sites/all/modules and /sites/all/themes directory instead. Contributed and custom modules and themes should then be placed in their own respective subdirectories.

Administration menu

The Administration Menu module (http://drupal.org/project/admin_menu) provides administrators with a small drop-down menu at the top of every page to allow easy access to the various administration pages that must be configured when building a new Drupal site.

Pathauto

The Pathauto module (http://drupal.org/project/pathauto) extends Drupal's core Path module by automating the process of creating more user friendly and search engine friendly URL aliases for a site's content. Pathauto requires and works in conjunction with the Token module (http://drupal.org/project/token).

Wysiwyg API

Drupal does not include a WYSIWIG editor by default; however, a wide variety of WYSIWYG and other client-side editors can be made to work with Drupal. The WYSIWYG API module (http://drupal.org/project/wysiwyg) provides Drupal integration for a variety of add-on editors, including: TinyMCE, FCKeditor, the YUI Rich Text Editor. Other modules provide support for individual editors, but it is likely that more will opt to integrate with the WYSIWYG API module going forward, in order to avoid duplicating development effort.

> **Hot Tip**
> With thousands of contributed modules, there is inevitably overlap in the functionality provided by some modules. Rather than installing each overlapping module to compare functionality, first check the "Comparisons of contributed modules" section of the Drupal handbooks (http://drupal.org/node/266179) to see if someone else has already documented their comparison. Example comparisons include WYSIWYG modules, node access modules, and content rotator modules.

DRUPAL THEMING

Drupal provides a flexible theming layer to allow a site's design to be highly customized. While Drupal can be used in conjunction with a variety of "theme engines" such as Smarty or PHPTAL, most users will want to become familiar with Drupal's default theme engine format, PHPTemplate.

Depending on the needs of your site, you may choose to create a completely custom theme or use one of Drupal's "starter themes" as the basis for a new site design. Starter themes such as Zen, Blueprint, or Framework strive to provide a basic foundation that can be customized to provide the look and feel your site requires.

To create your own custom theme from scratch, start by creating a new directory for the theme under the /sites/all/themes directory. For example, to create a new theme called "Beautiful", start by creating a new /sites/all/themes/beautiful directory.

> **Hot Tip**
> Drupal's default theme, Garland, was not designed to be a "starter theme." When looking to transform an existing theme to match a specific design, avoid using Garland. Instead, start from scratch or look to themes such as Zen, Framework, or Basic. For a detailed comparison of starter themes, see http://adaptivethemes.com/starter-theme-comparison.html

.info file

Typically a theme directory contains a number of files that work together to make up the theme. At the very least, a [themename] .info file is required in order for Drupal to

recognize a theme. Building on the example above, a minimal beautiful.info file might start with the following:

```
name = Beautiful
description = A example theme created for educational purposes.
core = 6.x
engine = phptemplate
```

A number of optional keys such as regions, stylesheets, and scripts can also be defined in order to override the defaults Drupal provides. More details can be found in the Drupal handbooks(http://drupal.org/node/171205).

Drupal themes typically contain a number of template files such as page.tpl.php or node.tpl.php. Each template file specifies how a particular portion of the site is to be presented. The various core modules provide default template files that can be copied into your theme directory and edited to customize them to suit the needs of a particular site.

Template files (.tpl.php)

The base template file found in most themes is the page.tpl. php file which acts as a wrapper around most of the other template files. When building a new theme, the default page. tpl.php theme file found in the /modules/system directory can provide a good starting point. You can copy that file into your theme's directory and customize it as needed.

> **Hot Tip**
> Drupal uses a theme registry to cache information about a theme. In order for Drupal to recognize changes to a theme such as adding files or changing the .info file, it is important to clear the theme registry. To clear the theme registry, go to the Performance page (Administer > Site configuration > Performance) and click on the "Clear cached data" button.

Additional template files can be used to customize the presentation of the various smaller elements of a Drupal site such as blocks, comments, polls, and more.

Template file	Purpose
page.tpl.php	the main container template file, which acts as a wrapper for the other template files below; see the following page for a list of available variables within the page.tpl.php file: http://api.drupal.org/api/file/modules/system/page.tpl.php
page-front.tpl.php	can be used to specify a custom front page
node.tpl.php	controls how the various content types or "nodes" are displayed; see the following page for a list of available variables within the node.tpl.php file: http://api.drupal.org/api/file/modules/node/node.tpl.php/6
node-[nodetype].tpl.php	optional template files used to target and customize nodes of a specific content type
block.tpl.php	contains the code used to display reuseable "block" content that may appear in multiple places within a site
block-[region].tpl.php	optional template files to customize how blocks in specific block "regions" will appear; regions are the areas specified in the theme's .info file where blocks can be designated to appear (such as headers, sidebars, etc)
comment.tpl.php	controls the formatting of user comments

Template files can be targeted to customize specific content of a site by naming the files according to specific "suggestions." For example, to customize the front page of a site, create a copy of the page.tpl.php file named page-front.tpl.php and

customize the new front page template as needed. See the Drupal handbooks (http://drupal.org/node/190815) for more details about the various core template files and how to use and customize the suggestion options.

> **Hot Tip**
> When creating your own theme from scratch, be sure to output the $closure variable, just before the closing </body> tag in the page.tpl.php file. The $closure variable is easy to overlook, but it is important because it outputs some final markup, which may cause some modules to fail, if it is missing.

template.php

The template.php file is an optional file used as a container for conditional processing and overriding of theme functions. In addition to overriding theme output using the template file method outlined above, Drupal also allows theme functions (any function beginning with "theme_") to be overridden inside the template.php file. Those more familiar with programming in PHP may prefer to use template functions to override Drupal's defaults in order to gain more flexibility and performance. A list of Drupal's default theme functions can be found on the Drupal API site (http://drupal.org/api/group/themeable).

> **Hot Tip**
> For help tracking down the particular function that controls the content you want to customize, install the Devel module (http://drupal.org/project/devel) and enable the Theme developer module that comes with it. The Theme developer module provides a graphical tool that can be used to select specific portions of a page and display the theme function or template file responsible for generating it.

style.css

Drupal will automatically recognize a style.css file as a theme's primary cascading style sheet file. Additional CSS files can be specified in the .info file using the "stylesheets" key.

screenshot.png

The screenshot.png file is a thumbnail sized image of the theme presented on Drupal's theme administration page to provide users with a preview of how a theme will look.

DRUPAL DEVELOPMENT

Drupal has a reputation as being created by developers for developers. While it has made great strides recently at becoming more user friendly, Drupal still has a lot to offer developers. Drupal has been designed with flexibility in mind. Developers can customize the vast majority of Drupal's functionality through Drupal's system of hook functions.

Novice users should not be dissuaded from creating their own custom modules. Even a simple single-function module can be useful for things such as customizing how Drupal presents a particular piece of content or form.

> **Hot Tip**
>
> Resist the temptation to modify the core of Drupal. "Hacking core" is discouraged, in favor of creating modules that modify Drupal's behavior instead. Creating modules avoids having to maintain and apply patches to Drupal as updates get released.

To develop a new custom module, start by creating a new directory for the module under the /sites/all/modules directory. For example, to create a new module called "Functional", start by creating a new /sites/all/modules/functional directory.

> **Hot Tip**
>
> Before sitting down to develop a new custom module, take the time to browse or search the contributed modules available on Drupal.org. The thousands of modules available there are typically developed to be reusable in a variety of environments.
> You may not find a module that does exactly what you are looking for, you may find one that can be adapted to suit your needs. If you make improvements to an existing module, consider contributing your code back in the module's issue queue, in the form of a patch. In addition to helping others, if the patch gets accepted, it will make future upgrades of the module on your own site easier.

.info file

As with themes, each Drupal module must include at least a .info file in order to be recognized. An example functional.info file might consist of the following:

```
; $Id$
name = Functional module
description = "A simple example module created for educational purposes."
core = 6.x
```

See the Drupal handbooks (http://drupal.org/node/231036) for details on the other keys that can be specified in the module's .info file.

Once the .info file has been created, a module can be enabled on the Module administration page (Site building > Modules). However, until the .module file has been created, the module won't provide any new functionality.

.module file

The .module file typically contains the majority of the code for a module. Modules interact with the core of Drupal by implementing various hook functions that get called as Drupal carries out various tasks. Certain specific hook functions will need to be implemented depending on what functionality or customization a particular module needs to provide. A list of available hook functions can be found on the Drupal API Reference site (http://api.drupal.org/api/group/hooks).

For our example functional.module file, we will implement the hook_nodeapi function. By implementing hook_nodeapi, a module can act on the node object used as a container for the majority of content within Drupal.

To implement a particular hook function within a module, replace the word hook within the functions name with the name of the module. For example, in a module called functional, the hook_nodeapi function is implemented as functional_nodeapi:

```php
<?php

function functional_nodeapi(&$node, $op, $a3 = NULL, $a4 = NULL) {
  switch ($op) {
    case 'view':
      $node->content['body']['#value'] .= "<br><br>Brought to you by RefCardz";
      break;
  }
}
```

Once the "functional" module is enabled, all nodes within the site will appear with a "Brought to you by RefCardz" notice at the bottom.

> **Hot Tip**
>
> To quickly dump variables to the screen for debugging purposes, you can take advantage of the message area typically found at the top of the page in most themes, by using the following function: drupal_set_message('<pre>' . print_r($variable, TRUE) . '</pre>');
>
> If you have the Devel module enabled, you can take advantage of more cleanly formatted output by using the following instead: dpm($variable);

By implementing hook_form_alter, modules can use Drupal's Form API to add, remove, or change forms. For example, some sites use the Views module or Panels module to customize the site's front page, so the "Promoted to front page" publishing option may need to be removed to avoid confusion. The following code will remove the checkbox used to promote content to the front page.

```php
function functional_form_alter(&$form, $form_state, $form_id) {
  if ($form_id == 'page_node_form') {
    unset($form['options']['promote']);
  }
}
```

For a more detailed introduction to the Forms API, see the "Forms API Quickstart Guide" (http://api.drupal.org/api/file/developer/topics/forums_api.html)

Useful Hook Functions (for a complete listing of hook functions, see http://api.drupal.org/api/group/hooks/)	Description
hook_block http://api.drupal.org/api/function/hook_block/	Declare a block or set of blocks
hook_cron http://api.drupal.org/api/function/hook_cron/	Perform periodic actions
hook_db_rewrite_sql http://api.drupal.org/api/function/hook_db_rewrite_sql/	Rewrite database queries, usually for access control
hook_enable http://api.drupal.org/api/function/hook_enable/	Perform necessary actions after module is enabled
hook_form_alter http://api.drupal.org/api/function/hook_form_alter/	Perform alterations before a form is rendered
hook_mail_alter http://api.drupal.org/api/function/hook_mail_alter/	Alter any aspect of the emails sent by Drupal
hook_mail http://api.drupal.org/api/function/hook_mail/	Prepare an email message based on parameters
hook_menu http://api.drupal.org/api/function/hook_menu/	Define menu items and page callbacks
hook_perm http://api.drupal.org/api/function/hook_perm/	Define user permissions
hook_user http://api.drupal.org/api/function/hook_user/	Act on user account actions
hook_watchdog http://api.drupal.org/api/function/hook_watchdog/	Log an event message

.install file

Another file commonly found in Drupal modules is the .install file. The .install file is used to store code that is run when a module is installed or uninstalled, such as the hook_schema and hook_install functions.

Common .install File Functions	Description
hook_install http://api.drupal.org/api/function/hook_install/	Install the current version of the database schema, and any other setup tasks.
hook_schema http://api.drupal.org/api/function/hook_schema/	Define the current version of the database schema.
hook_uninstall http://api.drupal.org/api/function/hook_uninstall/	Remove any tables or variables that the module sets.

MODULES FOR DEVELOPERS

Modules for Developers

Several of the contributed modules for Drupal provide tools useful to module developers.

Deadwood

As new versions of Drupal get released, it can take a while for existing modules to get upgraded. The Deadwood module (http://drupal.org/project/deadwood) helps speed up the process of upgrading Drupal modules by automating some of the code updates.

Devel module

The Devel module (http://drupal.org/project/devel) provides tools for clearing Drupal's cache, for generating dummy content, for debugging access issues, for performance logging, and much more.


The Devel module also includes an open source debugging tool called Krumo that can display a structured representation of any PHP variable. Devel adds "Dev load" and "Dev render" tabs to nodes which output the Node object through Krumo. With Devel enabled, you can output your own variables through Krumo using Devel's print message function, dpm().

Coder module

Since the Deadwood module doesn't completely automate the process of upgrading modules from one version of Drupal to the next, there is usually some manual coding to be performed after running Deadwood. That is where the Coder module (http://drupal.org/project/coder) comes in handy. The Coder module does not modify code, but it provides an automated code review that can pinpoint areas of code that need to be updated.
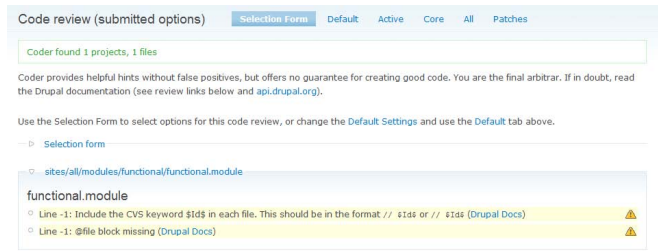
The Coder module can also help enforce coding standards by pinpointing sloppy coding and suggesting ways to clean it up. The Coder module can be especially helpful to run before contributing code back to the community at Drupal.org.

Schema

The Schema module (http://drupal.org/project/schema) provides tools for working with Drupal's database Schema API. If you've already manually created the tables required by a module, the schema module can provide the corresponding schema data structure to be added to your module's .install file, making it easier to replicate the same tables on another Drupal installation.

Cache Disable

Drupal's caching system can be a source of confusion and productivity loss, if you are not careful. Rather than having to remember each of the various tasks that require the cache to be flushed, install and enable the Cache Disable module (http://drupal.org/project/cache_disable) when doing development on a new site. Just remember to turn the module off, once you have completed your development work.

API

The API module (http://drupal.org/project/api) is used to generate the hyperlinked documentation found at api.drupal.org. The same module can be used to provide a local version of Drupal's API documentation. In addition to providing documentation on Drupal's core modules, the API module can also generate documentation pages for installed modules that follow Drupal's coding standards. If you prefer not to set up your own API site, you can still browse the documentation for many of the contributed modules, courtesy of Drupal consulting company Lullabot (http://api.lullabot.com/)

Other API Modules

Though Drupal does provide a full-fledged development framework, it doesn't include everything one might need for every possible coding task. Thankfully, the repository of contibuted modules on drupal.org includes quite a few API modules. Many of the API modules provide no functionality on their own, but provide API code that other modules and module developers can build on. Example API modules include the Voting API (http://drupal.org/project/feedapi), Chart API (http://drupal.org/project/chart), and the Import/Export API (http://drupal.org/project/importexportapi).

RESOURCES

URL	Reference
http://api.drupal.org/	Documentation of Drupal's API
http://drupal.org/handbooks	The Drupal community maintains several handbooks including documentation for beginners, themers, and developers
http://drupalcodesearch.com/	Drupal Code Search allows users to search much of Drupal's contributed code, though the index may not reflect the latest up-to-the-minute changes
http://cvs.drupal.org/	A web interface to browse the code in the Drupal.org CVS repository, from the start of the project to the latest in contributed modules
http://drupalmodules.com/	Drupal Modules allows users to rate modules. Ratings can be handy when choosing modules to use on a site
http://drupal.org/planet	Drupal Planet aggregates a number of blogs that post regularly about Drupal
http://groups.drupal.org/	Groups.Drupal is a gathering place for niche groups within the Drupal community. Try searching for a user group near you or a group focused on your specific area of interest
http://drupal.org/irc	Learn how to connect to Drupals IRC channels and connect with a world-wide community of developers at all hours of the day
http://drupal.org/mailing-list	Drupal.org provides a number of mailing lists to keep the community up-to-date. Anyone running a public Drupal site should subscribe to the security announcements mailing list
http://association.drupal.org/	The Drupal Association is a not-for-profit association charged with providing support to the Drupal project
http://drupal.org/getting-involved	The Drupal project thrives largely due to a strong community of contributors. Read the "Getting Involved" handbook to learn how you too can contribute
http://ww.lullabot.com/podcast	Several Drupal podcasts exist, both in audio and video formats, but the Lullabot podcasts are longest-running and arguably the most in-depth

ABOUT THE AUTHOR

Matt Vance is a Web developer, technology consultant, and freelance writer living in Austin, Texas. He has written for Macworld magazine and Lifehacker.com. He can be contacted through his website, http://minezone.org/

RECOMMENDED BOOK

Newcomers will find a thorough introduction to the framework, while experienced Drupal developers will learn best practices for building powerful websites. With Using Drupal, you'll find concrete and creative solutions for developing the exact community website you have in mind.

BUY NOW
books.dzone.com/books/using-drupal

Professional Cheat Sheets You Can Trust
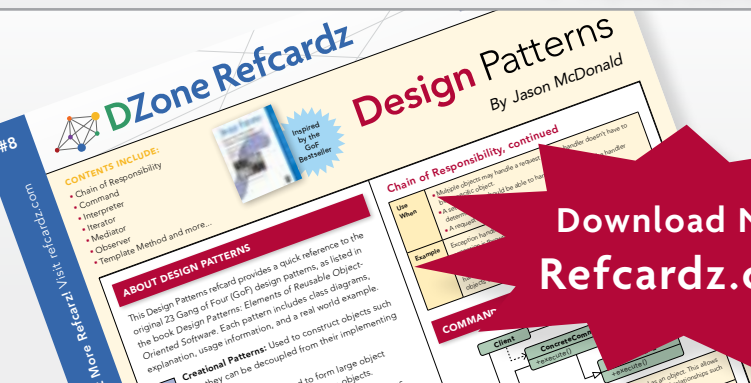
"Exactly what busy developers need: simple, short, and to the point."

James Ward, Adobe Systems

Upcoming Titles	Most Popular
RichFaces	Spring Configuration
Agile Software Development	jQuery Selectors
BIRT	Windows Powershell
JSF 2.0	Dependency Injection with EJB 3
Adobe AIR	Netbeans IDE JavaEditor
BPM&BPMN	Getting Started with Eclipse
Flex 3 Components	Very First Steps in Flex

DZone

DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheatsheets, blogs, feature articles, source code and more. **"DZone is a developer's dream,"** says PC Magazine.

Version 1.0