

Drupal Modules You Should Know

This workbook is a recap of a five-part email series by Design to Theme's Emma Jane Hogbin.

You may not be on her mailing list (?!). If you'd like to sign-up, simply fill out the little form:

<http://designtotheme.com/signup>.

License

This workbook wants to be free! Please share it with others under the Creative Commons license CC-BY-NC (Attribution-Noncommercial 2.5 Canada).

Table of Contents

Introduction.....	2
Functionality.....	4
Content and Storage.....	6
People and Permissions.....	8
Work Flow.....	10
Display.....	13
About Design To Theme.....	16

Introduction

Over at Lullabot, my friend [Jeff Eaton](http://www.lullabot.com/about/team/jeff-eaton) (<http://www.lullabot.com/about/team/jeff-eaton>) has been writing a weekly-ish series called Module Monday. He likes to cover very useful, but slightly obscure modules. A few of my favourites that he's covered are [Fences](#), [Exclude Node Title](#), [Field Label Plurals](#), and [SlideBox](#).

Eaton's perfect module for this series is both obscure *and* highly useful. (He swears it has nothing to do with a pump-and-dump scheme he may or may not be running over on [Drupal Stocks](http://drupal.webstocks.ws/) (<http://drupal.webstocks.ws/>.)

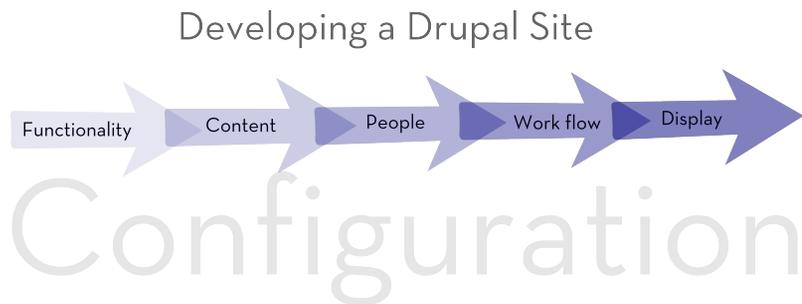
But what about all the modules that are so blaringly obvious that everyone forgot to tell you about them because they assumed you already knew? That's what this series of emails is going to cover. Over the next five weeks, I'll be covering some of my favourite modules under each of the following categories:

1. Functionality.
2. Content and storage.
3. People and permissions.
4. Work flow.
5. Display.

You may have noticed these categories have nothing to do with those over on [Drupal.org](http://drupal.org). Indeed, how will these categories be used for awesome modules like [SEO Checklist](#) or [String Overrides](#)? In short: any classification system is going to have limits, and this classification

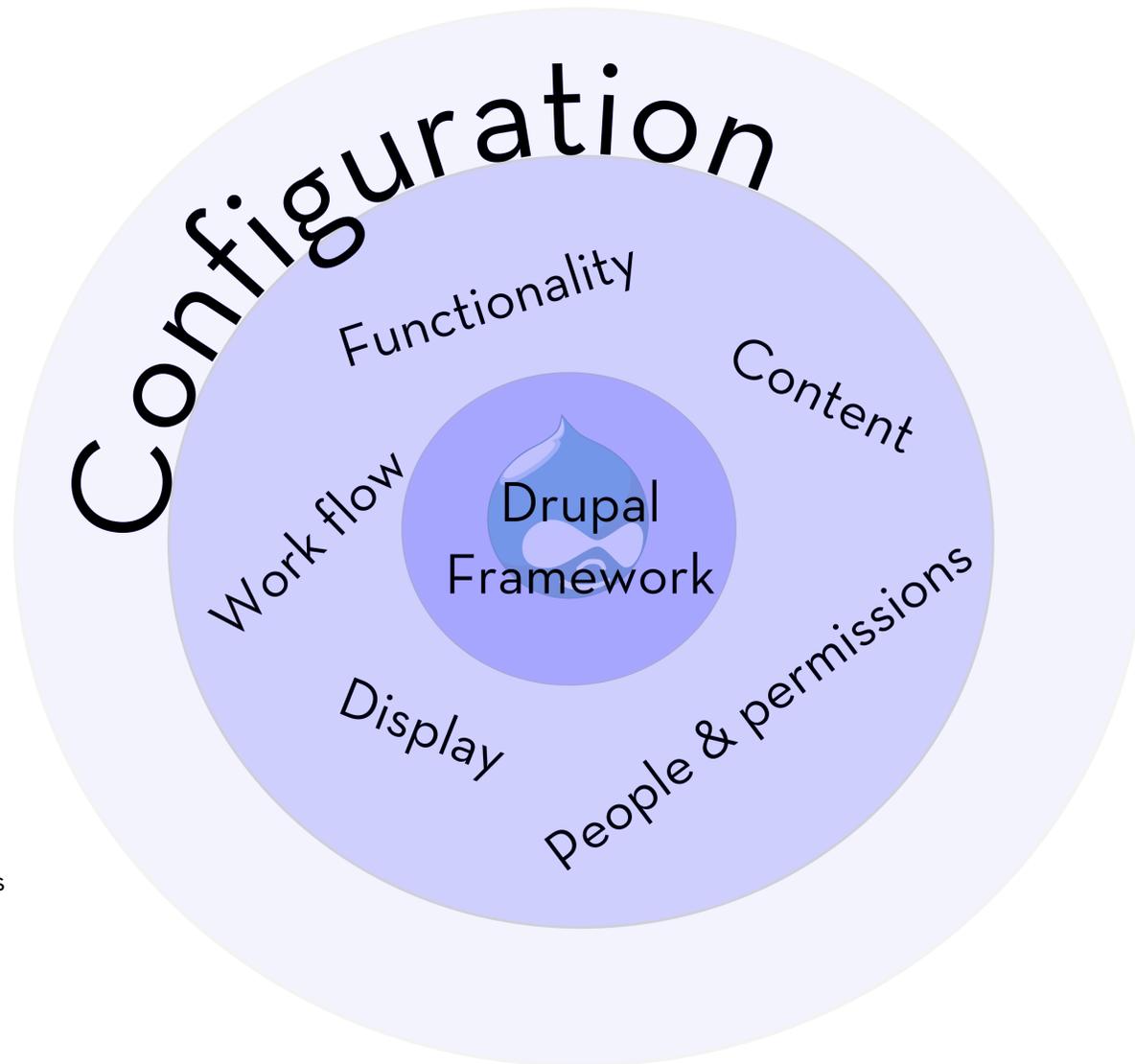
system was designed for my students who are just learning Drupal. From a site building perspective, Drupal is the configuration of these five components. They don't act in isolation. If you're in the evaluation stages, the five components might be contained within a larger configuration piece (diagram on the right).

But if you're in the development stages, the components might be more linear.



This series will look at some of the key modules that help you at each of these stages (and will be honest about how much work flow really does suck). If you're new to Drupal, this series is going to be perfect for you (please do share these emails with people on your team who are new to Drupal...it's especially helpful for project managers too).

PS If you'd like to see some of my favourite modules with more "traditional" categories, you can check out my slide deck [There's a Module for That](#).



Functionality

In last week's email I set up a framework for talking about Drupal. There were five components you needed to configure in order to make a successful Drupal site (bonus points for those of you who figured out this framework can apply to *any* CMS).

1. Functionality.
2. Content and storage.
3. People and permissions.
4. Work flow.
5. Display.

Today we're going to focus on Functionality. These are the *verbs* of your site. The modules you install so that you can *do things*. Here are my top picks for this category:

1. [Mailchimp](http://drupal.org/project/mailchimp) (http://drupal.org/project/mailchimp). I use this module to manage the subscribers for this mailing list. I don't store any of the newsletters on my site, although the responsive web design workshop uses a Mailchimp RSS feed to email new blog posts to the students in the course.
2. [Organic Groups](http://drupal.org/project/og) (http://drupal.org/project/og). I use this module to create clusters of activity on my class sites. g.d.o uses it to create clusters of people around common interests. No matter what you're clustering, this module is great for in-site groups (this is different from having micro-sites at different domain names).
3. [Commerce Kickstart](http://drupal.org/project/commerce_kickstart) (http://drupal.org/project/commerce_kickstart). Thinking about putting together an ecommerce store? Start with this distro which pre-configures

[Drupal Commerce](#) for you. (The Design to Theme store is still on Drupal 6 and it's running [Ubercart](http://drupal.org/project/ubercart) (<http://drupal.org/project/ubercart>.)

4. [Mollom](http://drupal.org/project/mollom) (<http://drupal.org/project/mollom>). My favourite spam catcher.
5. [VotingAPI](http://drupal.org/project/votingapi) (<http://drupal.org/project/votingapi>) on its own doesn't do much. But when added to [FiveStar](http://drupal.org/project/fivestar) (<http://drupal.org/project/fivestar>) you've got a great little social ranking tool.
6. [Google Analytics](http://drupal.org/project/google_analytics) (http://drupal.org/project/google_analytics). Track your visitors as they do things on your site.
7. [Apache Solr Search](http://drupal.org/project/apachesolr) (<http://drupal.org/project/apachesolr>) Drupal core's search is very, very limited. If you need a very, very powerful search interface for your site visitors, this might be the right option for you. (This is also available as a service from Acquia.)
8. [Persistent Login](http://drupal.org/project/persistent_login) (http://drupal.org/project/persistent_login). Adds a "remember me" checkbox for long-term access to a site.
9. [Case Tracker](http://drupal.org/project/casetracker) (<http://drupal.org/project/casetracker>). Convert your Drupal site into a ticketing system for project management. Alternately, you may want to start with the [OpenAtrium](http://drupal.org/project/openatrium) (<http://drupal.org/project/openatrium>) project, or hook Drupal into a robust third party project management tool.
10. [Invoice](http://drupal.org/project/invoice) (<http://drupal.org/project/invoice>). For making invoices you can send to clients. After doing some research recently, I also found out there are QuickBooks-to-Drupal integration tools. I wasn't expecting that! I don't have any experience with the services, so I'll let you do that Googling for yourself.

This is actually a remarkably difficult category for me as most of my favourite modules fall into one of the other four components.

Content and Storage

These are the baskets of stuff you put into your site. The modules you install so that you can *store things*. Here are my top picks for this category:

1. [Backup and Migrate](#). I've been using [NodeSquirrel](http://nodesquirrel.com/) (http://nodesquirrel.com/) as well to store my Drupal backups offsite.
2. [Date](#), [link](#), [location](#), [references](#), [media](#) and all those other field modules.
3. Collecting and grouping fields with [Field group](#) and [Field collection](#).
4. [Webform](#). For collating data (all submissions are attached to a single node and can be exported as a spreadsheet). This is also available as a [hosted service](#).
5. [Markdown](#), [GeSHi](#) (for code syntax highlighting), [WYSIWYG](#) (Jen has some [comments here](#) on which editor she likes best) are all a bunch of handy text editors.
6. [Meta tags](#) provides meta data for each of your nodes.
7. [i18n](#) and [i10n client](#) for multilingual sites. (If you're making a multilingual site for the first time PLEASE read [Gabor's introduction to multilingual content in D7](#) first, or [buy my handy guide](#).) Just in case I forget on the "display" category, you'll want to know about [i18n Views](#) too.
8. [Node Clone](#). Great for making copies of webforms if you don't want your data sets to be mixed up, but you want to ask the same questions.
9. [page title](#) is a handy SEO module that allows you to change the titles that appear in <title> and <h1> for each node. Or if you'd like to get rid of the default node title field, you can use this handy little module: [Auto Node Title](#). For more granular control, try [Exclude Node Title](#). (Technically we're now probably getting into "display" modules

instead of "content storage modules".)

10. [Pathauto](#). Automatically generate awesome URLs. Nuff said.

It was difficult for me to stop at only 10 modules. What would your top ten list have been?

On slideshare.net the following content-related slide decks may also be of interest to you:

- [Taxonomy](#) (includes a sequence of how to categorize cheese. .. Drupal AND cheese? what's not to love?!)
- [Content Strategy FTW](#)
- more [Content Strategy](#)
- [How to build linked data sites with Drupal7 and RDFa](#)

People and Permissions

This is what people are allowed to *do* on your site (and sometimes how to *prevent* people from doing things they shouldn't be allowed to do). Here are my top picks for this category:

1. [Field Permissions](#). Role-based control of individual fields, instead of whole content types.
2. [Forum Access](#), [Content Access](#) and [Taxonomy Access Control](#). In this same vein, I also use [Organic Groups](#) to segregate content and make it available to only some people at a time.
3. [Guest Pass](#). Allow your friends to login once (or for a very short period of time) with your permissions.
4. [Masquerade](#). Allows administrators to "masquerade" as someone else. Great for testing!
5. [Feeds](#). Allows you to create a series of user accounts from a CSV file. (Unfortunately [User Import](#) still doesn't have a Drupal 7 version and [Migrate](#) is ... well .. just too code-y for non-coders.)
6. [Terms of Use](#). For when your site needs a little more legaleze.
7. [Login Destination](#). Redirect your users to a special page when they log in.
8. [Narcissist](#) and [Misery](#) are more lighthearted modules...with a sprinkling of use cases where they could be quite useful.
9. In Drupal 7 you can add fields to users to extend their profile. Sometimes users need to have multiple profiles (e.g. someone is both a student and a teacher). In cases like this [Profile 2](#) comes in handy. Another useful profile module is [Real Name](#). It allows

users to have both a "username" and a "Real Name".

10. [User Relationships](#). I've only just found out about this module. It looks like it could be quite useful for developing a social site that has user-defined "circles" of friends (as opposed to administrator-defined groups, such as [Organic Groups](#)).

It was difficult for me to stop at only 10 modules. (To be fair: I didn't really stop at 10, did I?)

There are several relevant categories of modules you should check out on Drupal.org:

Community, Content Access Control, User Access & Authentication, and User Management ([click here to view all categories](#)). (And before you jump into the deep end and install all the modules, read [Jody's basic rules for node access control](#). It's an older post, but the rules still apply.)

Work Flow

Today we're going to focus on **developer work flow** and **content management work flow**.

This is Drupal's weakest area. There are huge gains being made in Drupal 8, but for now...well...some things are just exceptionally difficult.

Here are my top picks for developer workflow. This list isn't in order, instead it's a series of components that will make your job as a site builder, or module developer easier.

1. Non-coding site builders should know about [Module Filter](#) and possibly [Admin Menu](#) (as of D7, I just use the one provided by core) or [Admin](#).
2. [Drush](#). "Drupal Shell" Very, very useful for all site administrators and developers.
3. [Devel](#). Helpful tools for module developers (like printing out arrays of doom).
4. [Coder](#). Give yourself a mini (and automated) code review. Fail to pass? You may want to go back and check out the [Examples](#) project as well.
5. [Hacked](#). Not sure if you're running the original version of a module, or if it's been hacked by your team (or something more nefarious)?
6. [Features](#) and [Ctools](#). Export your settings. Commit them into source control. Share your settings with other developers and servers. etc.
7. Source control. Drupal uses [Git](#). There's a great article about [building a Drupal site with Git here](#).
8. [Drupal for Firebug](#). I'll admit that I've never had great luck with this module. .. Let me know how you make out though.
9. [Acquia Dev Desktop](#). For when you need a quick and dirty local installation of Drupal.

Comes pre-installed with a bunch of non-core stuff. I keep an export of a Drupal core site kicking around so that I can reset the database.

10. [Install profiles](#) and [Drush Make files](#). walkah outlines a great reason to use install profiles [in this article](#).

Want more? There's a whole slew of [useful developer tools listed here](#) (includes IDEs, browser testing tools, etc) and a [slidedeck version here](#).

These modules are useful for content management (the process of managing content), and content managers (the people creating/updating the content):

1. [WYSIWYG](#). Jen likes [TinyMCE best](#).
2. [Diff](#). See what's different between the current version of content and a previous version of the content.
3. [Revisioning](#) allows you to have new versions of content unpublished while you work on the updates.
4. [Flag](#). A simple toggle for content. I use this to allow students to 'flag' when they've completed a lesson.
5. [Rules](#). If (condition) do (action). Very useful. Johan has a [great video series on using the Rules module](#). See also: [Workflow](#) (and a [tutorial on how to use it](#)).
6. [Translation management](#). For multilingual sites.
7. [Workbench](#). And if Workbench is a little too much for you, check out Jen's admin dashboard, [Total Control](#). It's an inspiration to how we can make things easier for our content administrators. [Maestro](#). Created by a local-to-me development firm. This is essentially a competitor to Workbench (which is considered The Way for Drupal 7). [Nextide](#) has given some demos at Toronto DrupalCamps that I've been impressed with,

so don't skip reviewing this module if you're looking for an out-of-the-box solution for workflow management.

8. [File Stage Proxy](#) helps you get images from here to there.
9. [Deploy](#). Allows you to stage content. Sort of. Works with very limited (simple) content types. This problem will be solved (hopefully!) in Drupal 8. Follow the [config management initiative](#) for progress updates. In the mean time, you can also check out: For content, consider using [Feeds](#), [Services](#), or [Migrate](#).
10. [Domain Access](#). Allows you to publish content to multiple domains from a single "controller" site. This means you could publish to a staging server, when the content is approved, publish it to the live server (the staging server cannot be inside a DMZ).

If content workflow is a headache for you too, you should consider joining the [workflow group](#). (it's free!)

Display

Today we're going to focus on **the display** of content in your Drupal site ... aka "theming". (I still read this as them-ing, but the grammar people said you drop the 'e' and add 'ing' to verb the noun.) There will be big changes in Drupal 8. At every theming sprint the D8 team moves us forward by leaps and bounds. This list of top picks focuses on the modules and themes that I think will best position you for theming today, **and** for theming Drupal 8.

1. I build my rotating galleries with [Views Slideshow](#), but [Dynamic Display Block is also popular](#).
2. [Lightbox2](#) is great for pop-up/overlay images and galleries. I use [Colorbox](#) only because it was first to ship a D7 version (and [there are others](#) too).
3. If I want to use custom, non-free fonts on my site, I use [TypeKit](#). For free fonts, I simply use the @font-face rules. I trust [Squirrel Font](#) to always give me the best, cross-browser CSS vendor prefixes.
4. [Block class](#) allows you to inject a class name into any block without having to write code. (Although if you prefer doing things "The Right Way", check out [Jesper's talk from DrupalCon Munich](#).)
5. Need more than two tiers for your navigation? Check out [Menu block](#).
6. Fly-out (or "drop down") menus are available through the [Nice Menus](#) and [Superfish](#) projects. You might also want to check out [Mega Menus](#) (or [Megamenu Minipanel](#)), although this style doesn't seem to be as prevalent with the switch to responsive sites.
7. Breadcrumbs are really hard to get right. They're almost always wrong, and as web

users we rarely trust them to be right (and therefore never use them). If you can, omit them from your site; however, if you must use them, check out: [Menu Breadcrumb](#), [Taxonomy Breadcrumb](#) and [Custom breadcrumbs](#).

8. For layout: [Display Suite](#) and [Panels](#) are worth the time investment now and for D8 theming.
9. [Nodes in Block](#) and [Node Block](#) (for i18n sites) are pretty handy if you want to manage your blocks from the "content" administration screen.
10. Finally there's the question of base themes. You can watch [my video from DrupalCon Munich](#). Don't forget to [download the handout](#). Careful observers will note that I do have strong favourites even though I don't declare a "winner".

The spring issue of Drupal Watchdog was all about design for Drupal. You can [read \(most of it\) online for free](#).

Wish you could put all this stuff into action with a step-by-step workshop? I thought you might. There are THREE different options to choose from!

1. Responsive web design for Drupal [with support](#).
2. Responsive web design for Drupal [without support](#).
3. Theming and basic site building for [new-to-Drupal designers and themers](#) (includes support).

I hope you've enjoyed this five-part series!

This workbook is a recap of a five-part email series by Design to Theme's Emma Jane Hogbin. You may not be on her mailing list (?!). If you'd like to sign-up, simply fill out the little form:

<http://designtotheme.com/signup>.

About Design To Theme

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

- **Drupal Site Building Consulting:** We're great at saving you money. 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.

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