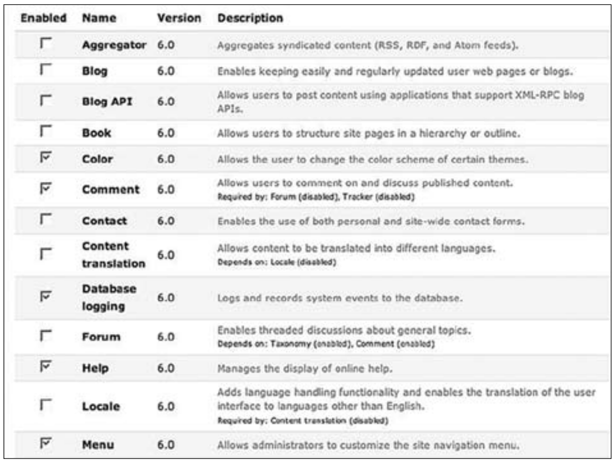
Modules

The other main customization for sites is the addition of modules. In Drupal, a module is an extension that provides a specific additional functionality for your site. For example the Poll module adds a new content type called “poll.” It also provides additional support for the use of polls on your site, including ways to display polls and security permissions for polls. One of the aspects that makes Drupal such a powerful content management framework is the ease of installing and using additional modules. There are a wealth of user-created modules to do all the extra things that you wish the core Drupal did.

When you visit the Modules area on the Drupal.org Web site, you will quickly realize that there are almost too many modules. This can make it very difficult to find the right modules to do what you want. Fortunately, there is a good sense of cooperation in the Drupal community, and it is considered good form to avoid duplicating functionality between modules. This means that you will only rarely have to decide between two modules that do the same thing. Oftentimes, so much effort is put into not duplicating work that modules are designed to be so flexible that you might not realize how useful they really can be for your particular problem (see figure 15).



Enabled	Name	Version	Description
□	Aggregator	6.0	Aggregates syndicated content (RSS, RDF, and Atom feeds).
□	Blog	6.0	Enables keeping easily and regularly updated user web pages or blogs.
□	Blog API	6.0	Allows users to post content using applications that support XML-RPC blog APIs.
□	Book	6.0	Allows users to structure site pages in a hierarchy or outline.
☑	Color	6.0	Allows the user to change the color scheme of certain themes.
☑	Comment	6.0	Allows users to comment on and discuss published content. Required by: Forum (disabled), Tracker (disabled)
□	Contact	6.0	Enables the use of both personal and site-wide contact forms.
□	Content translation	6.0	Allows content to be translated into different languages. Depends on: Locale (disabled)
☑	Database logging	6.0	Logs and records system events to the database.
□	Forum	6.0	Enables threaded discussions about general topics. Depends on: Taxonomy (enabled), Comment (enabled)
☑	Help	6.0	Manages the display of online help.
□	Locale	6.0	Adds language handling functionality and enables the translation of the user interface to languages other than English. Required by: Content translation (disabled)
☑	Menu	6.0	Allows administrators to customize the site navigation menu.

Figure 15
List of Drupal modules.

Building Modules

Modules are usually built and named specific to their functional use in context to Drupal and not necessarily in relation to the ultimate functionality that they offer the user. Many times, it will take a combination of modules to piece together what the user will think of as a single feature on the Web site. Drupal modules are like building blocks: you have all shapes and sizes, and your challenge is to piece them together in the right pattern to match your desired functionality. For example, if you wanted to create a social bookmarking site, you could go to the Drupal site only to discover that there is no module called “Social Bookmarking.” This does not mean that Drupal cannot provide this functionality; it just means that you need to understand which Drupal core functionality needs to be extended, then find the building blocks that can be used to construct the feature.

Drupal already lets users add content (nodes) to the site and gives the site administrator tools to manage that content. So to start creating social bookmarking, you would just need a new content type that lets users enter a URL to a node. Luckily, this has already been built: the Link module lets you add a link field to a node. Del.icio.us and other social bookmarking sites provide their users with a helpful bookmarklet that facilitates the quick addition of bookmarks from the original Web site. To replicate this, you can install the Prepopulate module, which lets administrators create a JavaScript bookmarklet that your users can use from the sites they're bookmarking to prepopulate the Link node. To more fully extend this example, you could also install the VotingAPI and Fivestar modules so your users can vote on each other's bookmarks. Finally, you can use Drupal's core taxonomy functions to let users tag their bookmarks. Now your Drupal site has all the basic functionality that you need for your social bookmarking site.

Almost all Drupal modules will extend Drupal core features like nodes, users, blocks, search, taxonomy in lieu of introducing something completely new. In the previous example, we could have coded a new custom module to handle all of the social bookmarking requirements. We could have also coded pages to display these links, as well as our own system for allowing users to tag these links. This, however, would have been a very inefficient use of Drupal. Instead of investing our limited development time in building a module from scratch, it is much better to use pre-existing modules that work to extend the core system. The more you understand the core set of Drupal tools, the easier it will be to understand the contributed modules and how they can be combined to create the kind of functionality that you and your users want from a library Web site.

On Using Modules

There has been a push on Drupal.org to get users to contribute the stories of building their sites. Included in these stories are the process users went through and the challenges they faced, as well as the modules they used and how they pieced them together. The best of these success stories have been compiled at Drupal.org's Success Stories page. More stories can be found in the Showcase forum, where users are encouraged to post their sites.

Drupal.org Success Stories
http://drupal.org/success-stories

Drupal.org Showcase Forum
http://drupal.org/forum/25

Don't Hack Core

Using Drupal, you can generally build a full-featured Web site to fit most of your needs without writing any code. There will be times, however, when you will find little things that you want to tweak or add to your site. This section is not meant to prepare you to become a Drupal coder, but is meant to give you a head start and point you in the right direction on some of the most basic ideas and uses of modules. There is, however, one rule of coding that must be shared at this time.

Don't Hack Core. Never, ever, not for any reason, even if it is the only way you can make something work, should you contemplate modifying the core code that makes up Drupal. We write this from having made this mistake on a regular basis in our early usage. Though it is fairly easy to find where Drupal is rendering the HTML for your page or making a query, you must resist modifying core code or contributed modules. If you change the code, you can no longer rely on the Drupal community's updates and bug fixes. While upgrading either modules or the whole core system is normally quite easy, any modifications made to that code will result in upgrade errors or a broken site after patches are applied. If you find a bug or need an extension to a module's capabilities, it is much better to submit a report or a request on the Drupal.org site.