

Drupal, the Community Framework

# DROPS ON THE WEB

The PHP-based Drupal framework makes dynamic websites simple. The modular system is extensible using free modules and the appearance is template-driven. In this article, we will be delving into Drupal's range of functions and discussing installation and basic configuration.

BY UWE HERMANN

Community sites account for some of the most heavily frequented sites on the Web. Users can register with the site and contribute articles or commentaries on a variety of topics. Drupal [1] was originally designed for this task and gives you everything you need to set up a convenient community site.

But on top of that, Drupal is a full-fledged Content Management System capable of supporting a wide range of websites, from simple weblogs to professional company sites. Drupal is written in PHP and uses a database to store content. The complete source code of the system has been placed under the GNU General Public License (GPL) and can thus be freely used, modified and distributed. The "The Birth of Drupal" box gives you a short overview of the history of Drupal and explains how it got its name.

Drupal is not difficult to install, however, it is not as simple as you might expect if you are familiar with other CMS systems. Debian users can simply type *apt-get update && apt-get install*

*drupal* (and answer the configuration prompts). If you do not have Debian, this luxury is not available to you, and you will need to go through the manual installation process.

## Easy Setup

The requirements are nothing unusual: you need a web server (say Apache, but IIS would work at a pinch) and PHP Version 4.1 or newer. PHP 5 is not fully supported at this time of writing, although the developers are working on solving the compatibility issues. You need to set the *session.save\_handler* user option for PHP, and the developers also recommend *session.cache\_limiter none*. Both options can be set in the PHP configuration file, *php.ini*. However, Drupal gives you a *.htaccess* file with these options and a selection of sane defaults.

Drupal additionally needs a database. The CMS supports both MySQL Version 3 (preferably 3.23.17 or newer) and 4, and PostgreSQL. For features that need XML, such as RSS or Blogger-API, PHP will need XML extensions, which are typically part of the default setup.

The "Clean URLs" feature additionally requires the Apache *mod\_rewrite* module and permission to use the *.htaccess* files. For this to work, you must avoid setting the Apache *AllowOverride* directive for the Drupal directory to *None*. This feature revamps URLs such as *http://www.example.com?q=node/34*, giving you *http://www.example.com/node/34* to improve readability and facilitate indexing of the page by search bots. If you want to optimize your site for search bots, you might like to enable the *path* module, which allows you to assign an arbitrary (and readable) URL to each node. This gives you URLs such as

## The Birth of Drupal

Dries Buytaert [2], a computer scientist from Belgium and the original Drupal author, started to develop a Content Management System for dynamic Web content in the year 2000. The system was intended to support a collaborative community weblog, which Dries intended to call *Dorp*. (“Dorp” is the Dutch word for village). While he was checking the availability of the *dorp.org* domain, Dries mistyped the name and entered “*drop.org*” instead. He liked the name so much that he decided to register the *drop.org* domain and rename the CMS software Drupal instead. (“Drupal” is the English pronunciation of the Dutch word for drop, “Druppel.”). Now, 5 years later, Drupal has grown to become a mature and flexible Content Management Framework with several hundred developers working on ongoing improvements and extensions.

<http://www.example.com/contacts> instead of <http://www.example.com/node/123>.

## Setting up the database

Even relatively inexperienced webmasters with a modicum of Linux experience should have no trouble installing Drupal via FTP and PhpMyAdmin. The following steps are required to install Drupal on a webhost with SSH access or on a local Linux machine. After downloading the current release (4.5.2 at time of writing) from <http://drupal.org>, give the following commands to unpack the archive and move the content to the document root of your web server, */var/www* in our example:

```
tar xfvz drupal-4.5.2.tar.gz
mv drupal-4.5.2/* >
drupal-4.5.2/.htaccess >
/var/www
```

The following MySQL commands, *mysqladmin* and *mysql*, set up a database and a database user:

```
$ mysqladmin -u root -p >
```

```
create drupal
Enter password:
$ mysql -u root -p
mysql> GRANT ALL PRIVILEGES ON >
drupal.* TO >
drupaluser@localhost >
IDENTIFIED BY 'secret';
mysql> FLUSH PRIVILEGES;
```

This gives you a database called *drupal*. The database user, *drupaluser*, who has been assigned a password of *secret*, has full privileges for all the tables in the database. You can press [Ctrl] + [D] to quit the MySQL shell session.

The *database/database.pgsql* file in the Drupal installation directory contains the schema definition for a PostgreSQL database. The MySQL counterpart is *database/database.mysql*. You can redirect the file in the shell to parse it:

```
$ mysql -u drupaluser -p drupal >
< /var/www/database/>
database.mysql
```

If *mysql* does not complain at this point, you can assume that everything worked as planned. To complete the installation, we now need an entry in the MySQL access data and the Drupal directory in the (amazingly readable) *includes/conf.php*. The variables we need here are *\$db\_url* and *\$base\_url*; we will assign the following values in our example:

```
$db_url = "mysql://drupaluser:>
secret@localhost/drupal";
$base_url = "http://localhost";
```

The first line lets Drupal know the database type (*mysql* in our example), the user, the password, the database host, and the database name. The second line specifies the visible path to Drupal for website visitors.

Our new Drupal installation is now browsable at <http://localhost>. The first thing you should do now is to set up a Drupal user. This user will be the administrator and will have full website configuration and administration rights, no matter what name you assign. If you

experience an error during the installation process, check out the “Troubleshooting” box for tips and a discussion of common issues and solutions.

Drupal terminology differs from that of other content management systems. The following sections explore the most important parts of a Drupal system and explain the relevant terms.

A node is the basic content building block on a Drupal website. A node can be an article, an image, or a forum entry. The following node types are enabled by default: *page* for static pages and *story* for articles. You can add more node

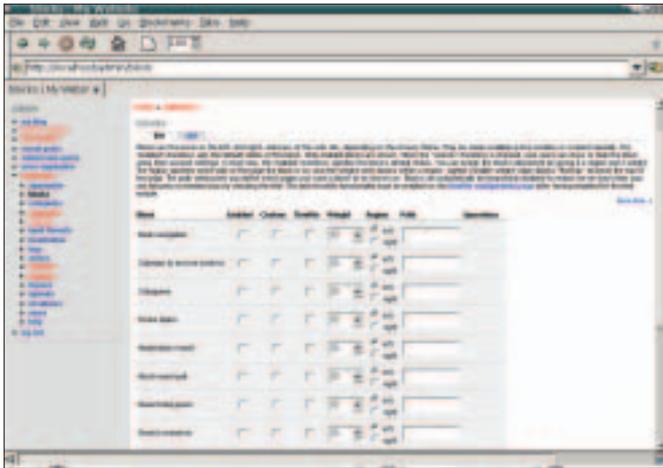
## Troubleshooting

Many common installation issues are easy to avoid.

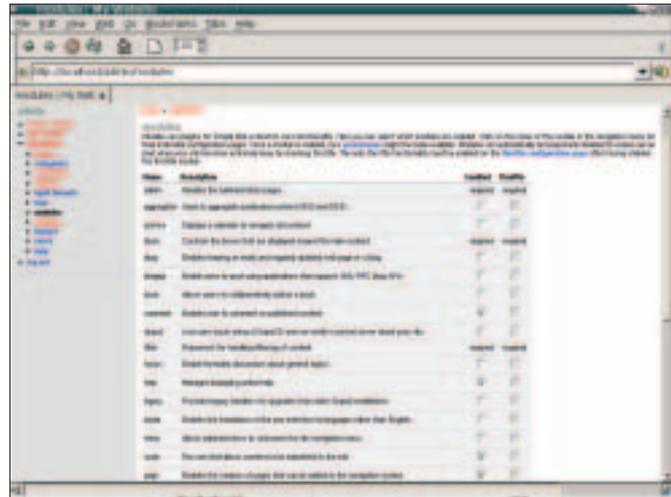
- One common error is assigning a value of *localhost* for the *\$base\_url* variable in the *includes/conf.php* file although Drupal is not running locally but in a connected domain. In this case, the line will need to be something like *\$base\_url = "http://www.example.com"*; If your DNS or hostnames are not working properly, you can specify an IP address here.
- Another common error is to specify an incorrect server path. If Drupal is not installed in */var/www*, but in */var/www/drupal*, for example, the entry must be *\$base\_url = "http://www.example.com/drupal"*.
- The *\$base\_url* variable must not end in a backslash */*; in other words, you need to enter *http://www.example.com*, rather than *http://www.example.com/*.
- If you are assigned an MySQL user, a password, and a database by your service provider, leave out the *GRANT* step and the database creation steps. Use the values given by your web hoster instead of *drupaluser*, *secret*, and *drupal*.

In case of installation problems, checking out the archives on the Drupal homepage [1] is generally a good idea. The Drupal forums [3] and mailing lists [4] are the domain of a group of helpful and competent developers who will be glad to assist.

## Advertisement



**Figure 1: Blocks give website users additional information. They are displayed to the left or right of all pages (or only the pages specified by regular expressions [5] in the path).**



**Figure 2: Drupal is extensible on the fly using a large collection of modules.**

types by installing and enabling modules. For example, the *forum* module defines a new node type called *forum topic*; the *blog* module gives you the *personal blog entry* node type. The administration menu gives you a list of node types in *create content*.

**Building blocks**

A block is a box with arbitrary content that appears to the left or right of web pages. Drupal gives you a number of preconfigured blocks, such as *User Login*, *Who's online* or *Recent Comments*. Additionally, any newly installed modules can provide blocks, which the administrator can then enable (see Figure 1). The administrator can also define new blocks that contain either HTML or text, or (for more complex content), PHP code and SQL database queries. The *Custom* field in each block specifies whether the user logged

on to the website is permitted to enable and/or disable blocks and thus modify the appearance of the website (in contrast to the default setting). The Custom Blocks Repository on the Drupal homepage has a useful collection of blocks.

Drupal is modular; the system itself provides only a framework of critical features, which is flexibly extensible using modules (see Figure 2). A module typically comprises a PHP file called *modulename.module* and optional files such as images or stylesheets. The download area of the Drupal homepage [6] has a large selection of modules that cover a wide range of new features (see Tables 1 and 2).

The Drupal term taxonomy needs some explanation; it uses vocabulary and terms to create what most people would refer to as a category. Each vocabulary has a name (such as *Topic*) and, typically, multiple terms which occur

within that vocabulary (for example *Politics*, *Sport* or *Technology*). You can define any number of vocabularies for a website and assign nodes to specific categories or terms from these vocabularies. This system is very flexible and applicable to many contexts. For example, hierarchies within forums or picture galleries are modeled on the taxonomy system in Drupal.

So-called *themes* allow and administrator to give users different views of the website (see Figure 3). A large number of ready-made themes are available for download at [6], and it is not difficult to modify a theme for your own use. The Drupal Theme Garden [7] lists a number of themes that you can try out online without needing to install first.

Drupal implements user and permission management based on users, roles, and permissions. A registered *user* can log on to a website. The user then has a

Table 1: Built-In Drupal Modules	
Module	Description
<i>aggregator</i>	Allows admins to read RSS feeds from other websites or weblogs and display these feeds in blocks.
<i>ping</i>	In case of content change, this module notifies interested services such as <i>technorati.com</i> , <i>weblogs.com</i> , <i>blo.gs</i> . Very useful for weblogs.
<i>search</i>	Supports searching in nodes for search keys. Needs cron to update the index.
<i>statistics</i>	Generates website visitor statistics. Logs IP addresses, referrers, the number of visits and so on.
<i>throttle</i>	This module disables some modules and blocks in high load scenarios to improve performance. Requires the <i>statistics</i> module.
<i>watchdog</i>	Logs all events on the website, for example, user login, who created content and when, error messages and more.

Table 2: Optional Drupal Modules	
Module	Description
<i>atom</i>	Generates an atom feed (Version 0.3), download available from <a href="http://localhost/atom/feed">http://localhost/atom/feed</a> .
<i>dba</i>	Minimal tool for database management in Drupal. Can often save you having to run PhpMyAdmin or similar tools.
<i>flexinode</i>	Supports the definition of more or less arbitrary content formats. For example, you could use this to define a recipe node type comprising a name, ingredients, quantities, tips on preparation and so on using point and click.
<i>print</i>	Creates a <i>Link printer friendly page</i> for each node, which provides a printer friendly view of the node.
<i>trackback</i>	A must for bloggers. Allows other bloggers to send so called trackbacks to this website, and for you to send trackbacks to other blogs.
<i>scheduler</i>	Supports content for release on a specific date or at a specific time.
<i>htmlarea</i>	Supports, easy, graphical, JavaScript based text editing.

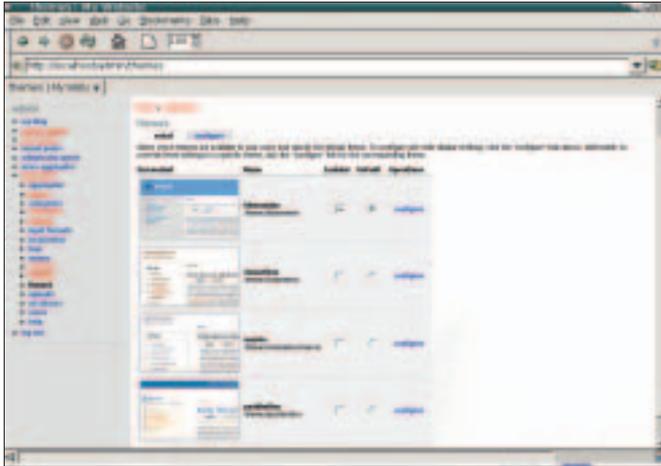


Figure 3: Themes allow the administrator (and even normal users if desired) to modify the appearance of the website.

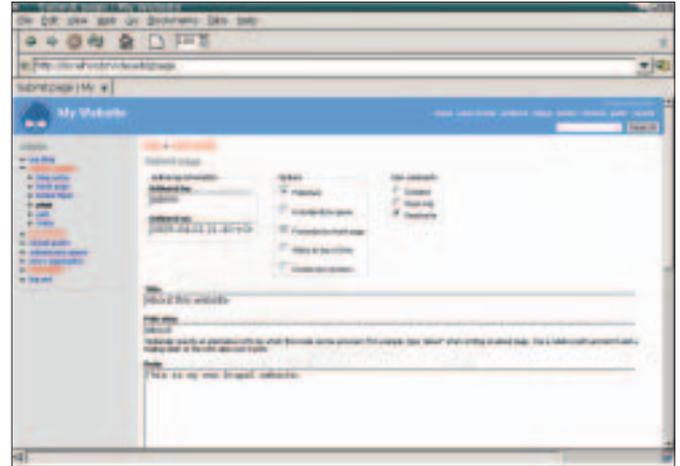


Figure 4: A number of options are possible when creating a node (the example shows a page type node).

number of options, depending on the *roles* assigned to that user, and the *permissions* assigned by the administrator to those roles. The section on user management later in this article discusses this in more detail.

Logged on users are allowed to *create content* or nodes for the website by selecting a menu item. Depending on the

active modules, content can be articles, blog entries, images, surveys, and many other things. Figure 4 shows how to create a *page* node. Each node can have at least a title and the content or *body*. If the *path* module is enabled, the user can assign an intuitive URL to the node.

There are many other settings for nodes, for example, whether or not com-

ments are permitted, if the node appears on the front page, and whether the node should be at the top of the node list (*sticky*). The *published* box gives the administrator the ability to remove individual nodes from the website at any time. The nodes stay in the database and can be re-enabled at any time. To delete a node, click *delete* on the *edit* page.

# Want to know what's up next?

## Subscribe to Linux Magazine Preview, our free monthly email newsletter!



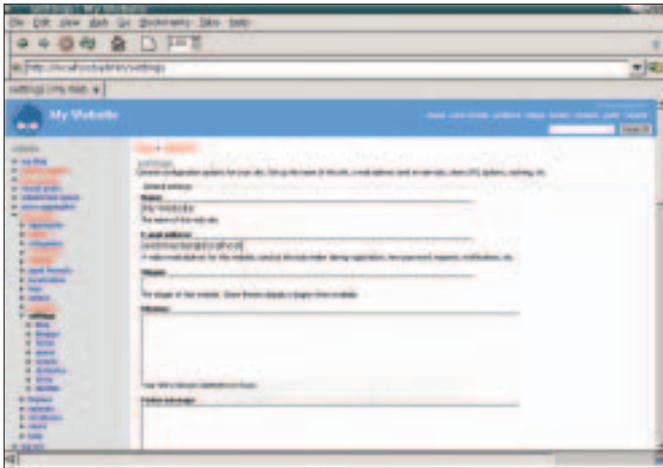


Figure 5: The administration menu: administer | settings.

Critical settings are available via the administration menu below *administer* | *settings* (see Figure 5). This is where you configure the website name, and optional slogan, the administrator email address, a footer (for example a copyright notice), and so on. The *Default front page* is the page that will be displayed first when a visitor accesses the website. This is *node* by default; in other words a list of the latest articles or nodes (but only those articles or nodes which have been *Promoted to front page*). You will not have any articles following the install; instead, Drupal displays a short help text for administrators.

Drupal's caching mechanism gives you better performance. It caches any pages designated for anonymous access (that is, users do not need to log on) in the database to avoid the need for generating the page multiple times on the fly. This menu is also where you enable the *Clean URLs* feature, which we discussed previously. Some modules introduce their own settings as subentries below *administer* | *settings*; for example, the statistics module adds an *administer* | *settings* | *statistics* menu item. Going into detail on this is far beyond the scope of this article, so check out the Drupal Manual [8] if you need more information.

## Scheduling Tasks

Recurring tasks in Drupal rely on an external cron daemon or on regular calls to the <http://www.example.com/cron.php> URL. This URL triggers regularly recurring actions in Drupal: for example, deletion of obsolete log entries, or updates of the Drupal search indexes.

The page can be accessed using any command line web client, for example, *wget* or *lynx*. The following line in */etc/crontab* would handle the job:

```
0 * * * * root \
/usr/bin/wget -o /dev/null \
-o /dev/null \
http://www.example.com/cron.php
```

This example runs *cron.php* once an hour on the hour (note that the URL must be the website domain and not localhost or 127.0.0.1 – even if the cron daemon is running on your local machine). If you do not have a cron daemon, don't worry; you can always resort to the *poormanscron* [9] module, which we will be looking at later.

## User Management

Below *administer* | *users* you will find a list of logged on users; administrators can check their settings and click on the *edit* link to modify them. For example, if a user misbehaves, you can disable that user's account. Administrators can use the submenus below *administer* | *users* | *configure* to create new users and define generic rules and privileges for user accounts:

- *settings*: Administrators can use this entry to specify who is allowed to create new users, whether users can register themselves, or if registered accounts need to be manually enabled. The administrator can also allow users to upload images, which will then be displayed in forum entries.
- *access rules*: Administrators can forbid certain user names or email addresses

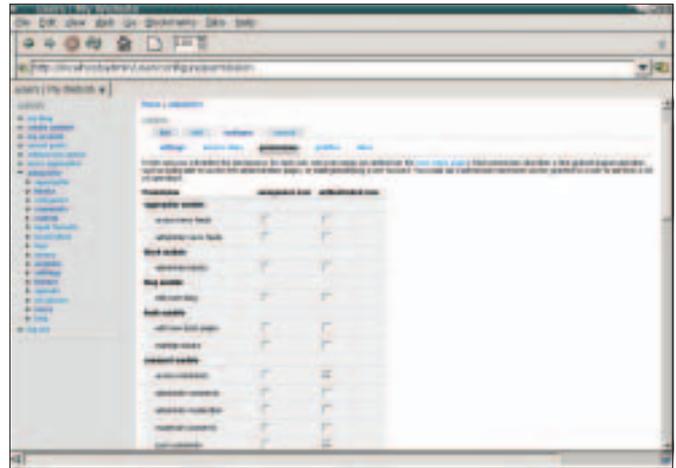


Figure 6: Each role is mapped to a ruleset.

(if they are insulting or obscene, for example), by defining one or multiple rules that can contain the “%” (no characters or more) and “\_” (exactly one character) wildcards.

- *roles*: Users can be assigned to one or multiple roles. The default roles are *anonymous user* for users who are not logged on (including search engine crawlers and bots) and *authenticated user* for users who are logged on. The administrator can define any number of additional roles and assign arbitrary names to them. For example, roles such as forum admin, editor, or customer are typical.
- *permissions*: Each role is linked to a set of permissions, which are defined by the administrator. For example, *post comments* allows users to do just that. Theoretically, any module can add new permissions to this list (see Figure 6). For example, the *poll* module adds the permission *create polls*, which allows users to create surveys, and *vote on polls* to allow voting. These options give the administrator extremely flexible user management tools geared to cope with a variety of requirements and scenarios, from a Slashdot-type page where most users are only allowed to add comments to an enterprise Intranet with different permissions for normal staff, heads of department, suppliers, and so on.

## Installing New Modules

Drupal is not a monolithic system but is extensible at runtime using modules. The default installation gives you a good selection of modules, although not all of them are enabled. Additionally, a collec-

tion of more than 100 modules is available from the Drupal homepage, all of which were created by the developer community and stored in a CVS repository [10], where development work on them continues. Of course, all of these modules are GPLed.

Installing modules is less convenient than one might have hoped, just like the Drupal installation itself. Additionally, the installation steps for each module may be different, although a de facto standard is starting to establish itself and many modules now honor it. We will be installing the *poormanscron* and *image* modules as examples.

The *poormanscron* module is a cron replacement. Drupal launches the module whenever a page is accessed to check if a specific (configurable) interval has elapsed. If so, it performs the actions that *cron.php* would perform and resets the interval counter to zero. This module is useful if your web hoster does not give you cron-based access. The installation steps are quite simple:

```
$ wget http://drupal.org
/files/projects/
poormanscron-4.5.0.tar.gz
$ tar xfvz
poormanscron-4.5.0.tar.gz
poormanscron/
poormanscron/LICENSE.txt
poormanscron/README.txt
poormanscron/poormanscron.module
$ cp poormanscron/
poormanscron.module
/var/www/modules
```

The module comprises a single file, *poormanscron.module*, which you simply drop into the *modules* directory of your Drupal installation and then enable below *administer* | *modules*. The settings for the module are located in *administer* | *settings* | *poormanscron*.

The *image* module is interesting for a majority of websites, as it supports image uploading and management. The module requires ImageMagick, GD, or ImLib2. At least one of these tools should be available from the web hoster. After downloading and unpacking the tarball, the administrator needs to modify the database by parsing the *image.sql* SQL file. Then the files need to be copied to the *modules* directory for your Drupal installation:

```
$ wget http://drupal.org
/files/projects/
image-4.5.0.tar.gz
$ tar xfvz image-4.5.0.tar.gz
[...]
$ mysql -u drupaluser -p
drupal < image/image.sql
$ mkdir /var/www/modules/image
$ cp image/image.module
image/image.inc
/var/www/modules/image
```

The next step is to enable the module in *administer* | *modules* and access *administer* | *settings* | *image* to configure the module. For example, it needs directories for images and thumbnails, possibly the path to the ImageMagick *convert* program, typically */usr/bin/convert*.

You can also create image galleries in *administer* | *categories*. To do so, create a vocabulary with one or multiple terms. Then enter the vocabulary as *Gallery Navigation Vocabulary* (again in *administer* | *settings* | *image*). Finally, in *administer* | *users* | *configure* | *permissions*, select the users or roles permitted to upload and manage images. This allows users with appropriate permissions to create new images by selecting the item *create content* in the administration menu.

**Outlook**

The next Drupal release, Version 4.6, is imminent, and we can look forward to a number of interesting changes. The new version will have a vastly improved search function, which uses UTF-8 encoding to provide multiple language support. Many usability enhancements in the configuration menus make the administrator’s job easier. The new *contact* module gives logged on users a form-based email feature for sending mails to other users. Multi-site configuration gives administrators the ability to operate multiple Drupal websites from a single Drupal installation. There are also a number of performance boosting features, which should help Drupal run more efficiently and help it handle a lot more simultaneous page hits. The number, quality, and completeness of the admin interface translations continues to make rapid progress. Translations in more than 20 languages are now available. Thanks to the commitment of some developers, new themes will also be

available. These include popular blog themes such as Kubrick, Persian, or Manji, which were ported from the Wordpress blogging software [11].

**Conclusions**

Drupal is a free, compact, and highly flexible, configurable Content Management System with a large and active developer and user community. It is easily adapted to support a wide variety of application scenarios. Developers will appreciate the solid framework with its well documented API [12], which is extensible using PHP modules. The modular structure, the caching mechanism, and the throttle module make Drupal highly scalable and powerful – for example, *drupal.org* has over 18,000 registered users and more than 17,000 nodes right now. ■

**INFO**

- [1] Official Drupal homepage: <http://www.drupal.org>
- [2] The Drupal author Dries Buytaert’s website: <http://www.buytaert.net>
- [3] Drupal forums: <http://drupal.org/forum>
- [4] Drupal mailing lists: <http://drupal.org/mailling-lists>
- [5] Regular expressions in Wikipedia: [http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression)
- [6] Drupal downloads (Releases, Module, Themes, Translations): <http://drupal.org/project/releases>
- [7] Drupal Theme Garden: <http://webschuur.drupaldevs.org>
- [8] Drupal Manual: <http://drupal.org/handbook>
- [9] Poormanscron module: <http://drupal.org/project/poormanscron>
- [10] Drupal “contributions” CVS repository: <http://cvs.drupal.org/viewcvs/contributions/>
- [11] Wordpress, free weblog software: <http://wordpress.org>
- [12] Drupal 4.5.x API documentation: <http://drupaldocs.org/api/4.5>
- [13] Crazy Hacks, ailing computer projects: <http://www.crazy-hacks.org>
- [14] Unmaintained Free Software, free projects which need new maintainers: <http://www.unmaintained-free-software.org>
- [15] Uwe Hermann’s blog and homepage: <http://www.hermann-uwe.de>