

The Drupal Cookbook (for beginners)

The Drupal Cookbook (for Beginners) helps Drupal "newbies" by providing a walkthrough of a common Drupal Setup.

Background

This handbook was originally written for Drupal 5. While the information is generally transferable to Drupal 6, some buttons, links, and menu items have been renamed or moved. Every attempt is made to keep these handbooks current.

The intent of the Cookbook is to help the new Drupal user create a typical site. At that point the user will be better equipped to diving deeper into more advanced features.

Terminology

This cookbook requires a basic understanding of the [General Concepts](#) of Drupal. Additional resources for understanding terms used in Drupal include:

- [Terminology \(Terms\)](#).
- [Drupal Jargon](#).

Conventions

The Cookbook and other documentation on drupal.org uses the following standard for indicating site navigation:

Administer >> Access control >> User management >> Roles.

The above example tells the user to click on "Administer" in the navigation menu, then "Access control," then "User management," and then "Roles".

Working with Drupal

Here are some general recommendations:

- Use a test site that uses the same Drupal version and modules as the target site. Use a copy of the live database. Avoid development on a live site.
- Don't try to make the "perfect site" on the first attempt. Muddle through for a while. Stressing over the perfect solution can lead to frustration.
- Start by learning the basic functionality of Drupal. Find out what Drupal *can* do before working towards a specific goal. Once comfortable with the "core" features and behavior, move on to more complex contributed modules such as Views, CCK, and Organic Groups. These modules and some others require a good bit of understanding to master. The power and flexibility of Drupal and its modules will become apparent over time.
- If you need a custom theme, customize one of the default themes before creating starting from scratch. Refer to the [Drupal 5 theme guide](#) or [Drupal 6 theme guide](#).
- Limit the number of blocks, images, and graphics that clutter the page.
- Participate in the [forums](#), the [Documentation Team](#), and [IRC](#).

Get support through Drupal.org

Before posting to the Drupal.org site:

- Search to see if the subject has already been covered to avoid having duplicate postings & issues. If Drupal's search fails, Google will often point to the right references. To use Google to limit the results to Drupal.org include in the search field *site:drupal.org*
- Ask one question per post. The issue tracking system can only handle one at a time.
- Don't hesitate to ask the question again and add "I'm a newbie, can you say this in easier to understand terms, please?"
- Before posting, [read the tips for posting in the Drupal Forums](#). Try to describe the situation/case completely. Explain what has been done so far and what you are trying to achieve.
- When requesting changes don't demand or threaten to abandon Drupal, and certainly don't resort to name calling or derogatory comments. Often times the best way is to jump in and ask how to get involved.
- In general you will see core versions written as **5.x** and **6.x** or a specific version as *5.18* or *6.12*. Contributed modules are listed with a *core compatibility* **and** a *version number*, such as **5.x-2.5** or **6.x-1.6**, which mean "Drupal 5 compatible, Version 2, release 5" or "Drupal 6 compatible, Version 1, release 6." This way, if you see "5.x-2.5" you can know that it means a module release rather than a core release.

For example, if a page is not showing up correctly, be prepared to provide the following information:

1. A descriptive title
2. Operating system and browser name. Version information and a list of potentially relevant plug-ins you are using are often important.
3. Version of Drupal (5.x, 6.x etc)
4. Any contributed modules
5. The versions of PHP and MySQL being used, as well as the name/URL of the hosting provider. To see which versions of PHP and MySQL (or other database) that the site is using, in Drupal 5 navigate to: **Admin >> Logs >> Status Report**, and in Drupal 6: **admin >> reports >> status**.

Things to keep in mind

A common mistake when approaching a solution to a problem in Drupal is to make an assumption about how Drupal works. Drupal is a unique and powerful platform that is probably quite different from other solutions you may have encountered.

Start simply by making something visible, then celebrate what you have accomplished. Taking one small step at a time will lead to a better site. Don't indiscriminately add many contributed modules right away. Get comfortable with modules one at a time. Modules that are not appropriate or used should be disabled and uninstalled.

As with most things worth mastering, there is a learning curve with Drupal. But there is lots of support available as well.

Welcome to your new Drupal website!

This cookbook assumes you have successfully installed Drupal and are looking at a "Welcome to

Drupal" screen. If you're not already to that point, the [Installation Guide](#) is for you!



Drupal's Welcome Screen (Drupal 6)

This guide will talk you through [adding content](#) to your site, setting up your homepage, [working with menus](#), [extending](#) your site with common [modules](#), configuring [permissions](#), setting up [blocks](#) and making your site look different via [themes](#).

Creating content

After your site is up and running, it is time to add content to your site.

At this point, you might want to read [The 11 Most Important Things To Do After You Install Drupal](#).

Content types

A content type defines the way in which content is collected and displayed. Content types are a container for all content with the same characteristics (e.g. a "Bio" might contain different kinds of information than an "Event"). Some fields are part of all content types (e.g., basic data such as a title, date, and author). Some modules create their own content types and some don't. Those that do can have unique fields defined in their content type (e.g., the Location module adds fields such as city, country, longitude, and latitude). Additional custom fields of many kinds can be added using the CCK module. Content types have settings that can be modified by editing the content type and managing the fields.

Examples of content types provided by core Drupal modules:

Page

A page is a simple method for creating and displaying information that rarely changes, such as an "About us" section of a website. By default, a page entry does not allow visitor comments and is not featured on the site's initial home page.

Story

A story, similar in form to a page, is ideal for creating and displaying content that informs or engages website visitors. Press releases, site announcements, and informal blog-like entries may all be created with a story entry. By default, a story entry is automatically featured on the site's initial home page, and provides the ability to post comments.

Examples of content types provided by non-core (contributed) Drupal modules:

Webform

Create a new form or questionnaire accessible to users. Submission results and statistics are recorded and accessible to privileged users.

Image

An image (with thumbnail or other sizes). This is good for photos, art work or screen shots.

Create your own content type

You can define a content type to exactly meet your information needs. The easiest way is to start with a page. As you create your content, think about the menu as well. Another handbook section you may find useful is [Creating new content](#).

Creating content

To create content: click *Content management* > *Create content*, then select the content type you want to create. There are usually one or more fields within each content type. We are providing an example of a page.

- **Title and Body** fields are found in many content types. The Title is used by Drupal to create the page URL. Keep it concise.
- **Body** is where you put the text for the page. "Input format" controls what code can go in the Body field. There are three (core) options: filtered HTML, PHP code, and full HTML. CCK allows you to change the name of this field; for example, you might call it "Product description."
- Use the **log message** to provide information that might be useful to other authors who may edit your document later, or provide your rationale for making edits to your own or other people's content. The log message is not visible to users without the appropriate content editing rights.
- **Menu settings** are used only if you are making this piece of content an item in one of your menus. It's the most efficient way to create menus. Another handbook section you may find useful is [Working with the Menu](#).
- **Comment settings** are probably best set at the site level (click *Content management* > *Comments* > *Settings*), but can be specified for an individual piece of content.
- You can **attach files** to many content types. The "Upload" module must be enabled, and then the content type has to be set to allow this.
- If you enabled the Path core module or added PathAuto, you'll have **URL path settings** next. You can enter a "normal" name here rather than being required to use "node/2" when you refer to it later on. PathAuto will automatically create a URL based on your page title.

Authoring sets the time stamp and creator information. The other use for this section is to control the page or story order when they are based on the time and date it was created.

- **Publishing options:** only "Published" items show up on the website. You can also promote content to the front page and make the content item sticky to the top of lists.

The final step is to Submit your page.

"Teasers" - short summaries of node content

Throughout Drupal you will find the concept of a [Teaser](#). A teaser is a short piece of text, usually the first paragraph or two of an article. This text is then displayed in most lists, including the default home page (/node). Many modules (including Views) are designed to work with node teasers, and the concept of a "teaser" vs the node "body" is integral to the workings of Drupal core and the Drupal UI.

The length of the default teaser can be modified by visiting **Administer > Content management > Post settings** and will attempt to break cleanly at paragraphs or words in the text. When the length of the content exceeds the set limit, a link to "Read More" will automatically be displayed.

Some modules help you to display different content, or different versions of that content in 'teaser' view vs full 'body' view. image.module for example inserts a thumbnail of an image into the teaser, while you see a page-sized version when viewing the node.

CCK provides many features under **Administer > Content management > Content types** allowing you to fine-tune what does and doesn't show as a teaser.

Teaser break

The length of the teaser can be overridden when editing the node content itself and allows you to choose a logical length for the introduction. In Drupal 6 the edit interface gives you the option to "Split Summary at Cursor". Using that button will visually divide your edit area into [teaser] and [the rest of the content].

Some of the Drupal contributed WYSIWYG editors also provide that feature in their own way, when configured to do so.

Drupal 5 offers the ability to manually enter a special tag to indicate where to split the content.

Show summary in full view

When you first 'split' a teaser from the body content, it is still treated as part of the page when the full page is displayed. This is the easiest, and most intuitive way for automatically-generated teaser text to work. A reader always sees [teaser] and [the rest of the content] without seeing any break.

For more control, better grammar, or to make the teaser text more promotional in nature when the first paragraphs on their own don't adequately describe the node, you can choose to **not** "Show summary in full view".

If you choose this option, the custom description will **not** be displayed as part of the full-text. Your full text should be complete, and make sense alone, and your teaser, which is used to entice readers to click "Read more" can be entirely different from the text that appears in the actual item.

Some more reading: [Using teasers for effective SEO](#)

Setting up your home page

Now that you have created some content, you can make one of those items your home page. Go to that page and note the URL, e.g., `node/1`. The URL must be the system URL (eg "`node/1`"), not an alias such as that generated by another module or one you've assigned to a node yourself ("`pretty-url-1.html`"). If you try to use an alias url to set the front page, the results will be quirky, such as some blocks not showing up.

Go to **Site Configuration > Site Information**. At the bottom of this page where it says "Default front page" you will see **Default front page**. Type in the URL of the page you want to be your home page, e.g., `node/1`.

Save this and test it out by clicking on the home page link.

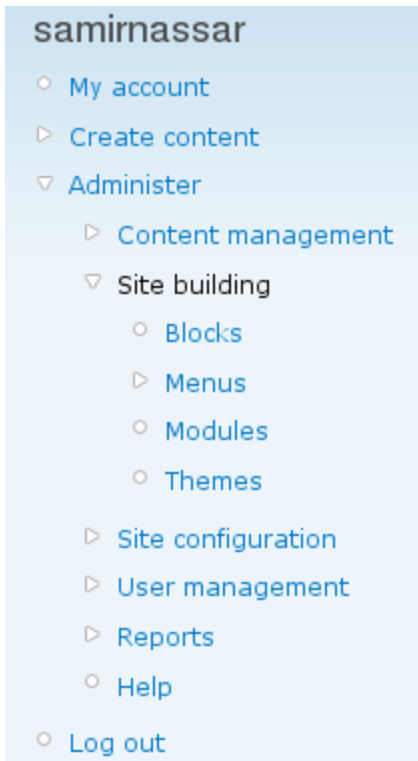
Working with menus

Menus are how users move around a website; by clicking on words like 'Services' or 'Contact Us', they will get to your content. As such, it is important to think about what the best terms are to describe your site to someone who isn't familiar with what you or your organization offers. Happily, the time it takes to think through and decide on the best options probably takes less time than it takes to actually build them.

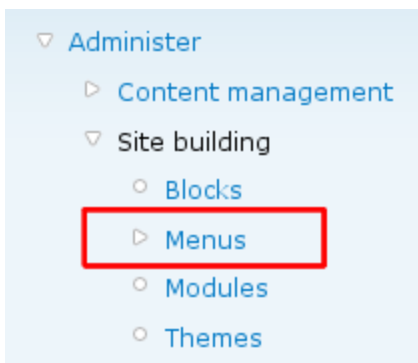
Drupal 6 comes out-of-the-box with three menus. The first is the 'Navigation' menu, which is the one used to administer the site (it has the 'My Account' and 'Administer' links in it). There are also the 'Primary' and 'Secondary' menus, which show up in different places on the page (and, by default, are empty in a new Drupal installation). You can also add new menus, like 'Footer Links' or 'Foo Bar' or anything else you might like.

The next few steps show you how to add a new menu as well as how to add new items (links) to an existing menu.

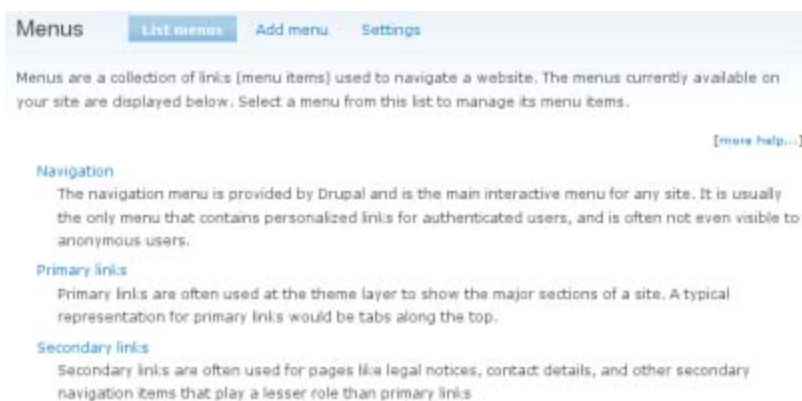
Creating a new menu



Start by going to the navigation menu and follow the links: **Administer > Site building > Menu**



You'll end up at the **Menus** page





There are three fields: menu name, title, and description

Menu name

The menu name has to be 'machine readable', which means that it can't have spaces, punctuation, and has to be lower case (like the help text says). This is for Drupal's own internal use.

Title

The title is for humans and may be visible to site visitors as well. It can be anything you want.

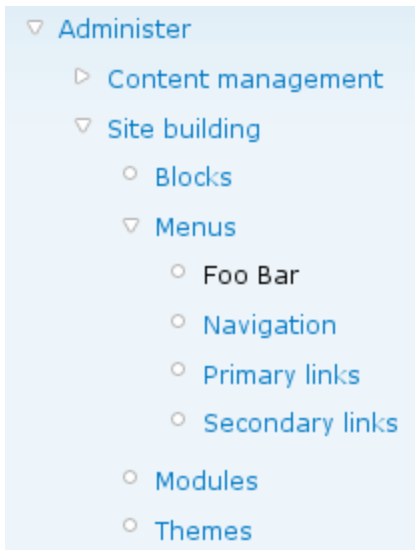
Description

Writing a description for your menu can be useful to describe why the menu was created.

Once you have entered your information save the menu.

Adding a menu item to an existing menu

Click on the name of the menu you would like to edit. (If you created an entirely new menu per the above, the navigation menu will now show that new menu, e.g. 'Foo Bar'.)



To add a new menu item, click the **Add item** link

Foo Bar List items **Add item** Edit menu

Menu settings

Path: *

The path this menu item links to. This can be an internal Drupal path such as *node/add* or an external URL such as *http://drupal.org*. Enter *<front>* to link to the front page.

Menu link title: *

The link text corresponding to this item that should appear in the menu.

Description:

The description displayed when hovering over a menu item.

Enabled
Menu items that are not enabled will not be listed in any menu.

Expanded
If selected and this menu item has children, the menu will always appear expanded.

Parent item:

The maximum depth for an item and all its children is fixed at 9. Some menu items may not be available as parents if selecting them would exceed this limit.

Weight:

Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.

Path

The path is where the menu entry will take your visitor. This can be external, as it is in this

case <http://drupal.org>, or internal such as **node/1** or **admin/reports**.

Menu link title

Give your menu entry a title. In this example, **Drupal**

Description

A description can be useful to tell yourself and others what the menu entry is for

Enabled

This checkbox allows you to make the menu entry active or inactive (show or not show) without having to delete it.

Expanded

If this menu entry has sub-entries, known as children, this allow the menu entry to stay expanded so that the children are shown by default.

Parent item

Parent item selection allows you to choose where in the menu hierarchy your menu entry belongs.

Weight

Weight defines the relative rank of your menu entry. A smaller number means it is placed higher up.

Now save your menu entry.

Display your new menu

To display a new menu you will need add a *Block* display to your Drupal installation. To do so, go to **Administer > Site building > Blocks** and scroll down to the 'Disabled' section (where blocks that aren't currently being displayed are managed). You can activate the Foo Bar block by dragging it to the left sidebar (or any other region). When complete, you should see:

Block	Region	Operations
Left sidebar		
+ Navigation	Left sidebar ▼	configure
+ User login	Left sidebar ▼	configure
+ Foo Bar	Left sidebar ▼	configure

Your left sidebar should now display **Foo Bar** followed by the link you specified



You should now be able to create and edit menus. You can read more about blocks at [Blocks - Adding extra information](#)

How to display sub menus (children) of primary links in Drupal

This page describes how to display sub menus (children) of primary links in Drupal.

Step 1: Since we'll be using php codes in a block to display menu, the first thing to do is to enable the PHP Filter under modules.

Step 2: Go to administer->block->add block and create a new block. Add a **description** (for instance submenu). Choose **PHP Code as the filter** and paste the code below in the body textarea.

```
<?php
$menuItems = menu_tree_page_data('primary-links');

foreach($menuItems as $key => $m) {
    //check current active menu
    if ($m['link']['in_active_trail'] && $menuItems [$key]['below']) {
        $menu = menu_tree_output($menuItems [$key]['below']);
    }
}

//print the menu
print $menu;
?>
```

Make sure you include the php delimiters in your code. Click Save.

Step 3: Place your block in a region you want. and that's it. next time you select a menu item in the primary links, the children menu items will show in this block.

You can use active-trail to style the selected menu item.

Users, roles and permissions

Your site requires different permissions to be set per user type. For example, you probably won't want casual visitors to edit your homepage. However, the site owner should be able to do so. To learn more about the term "user", learn about [Differentiating the Four Different Kinds of "Users" Encountered When Installing Drupal](#).

Drupal allows you to setup any number of different kinds of users or 'Roles'. Many websites have editor and site administrator roles; editors to make content updates and site admins to install new modules and make larger configuration changes.

Out of the box, Drupal recognizes two types of site visitors - those who are logged in (or 'Authenticated' users) and those who are not (or 'Anonymous' users). The exception is the first user created (user/1) -see [here](#). Although it is not necessary, many sites have additional levels of users.

To create or edit a role, click **Administration > User management > Roles**.

To create or edit a user, click **Administration > User management > Users**.

To specify the permissions for a role, click **Administration > User management > Permissions**.

To add editors to your site, you will first need to create an editor role. Click **Administration >**

User management > Roles. Type in the name of your new role (e.g. 'editor') and click 'Add role'.

To add a new 'editor' user, go to **Administration > User management > Users** and click the 'Add User' tab. After typing the username and email address, enable the 'editor' checkbox and click 'Create new account'.

Finally, you can configure permissions for editors at **Administration > User management > Permissions.** To give editors the ability to edit any page within the site, scroll down the permissions page and click the checkbox next to 'edit any page content'.

Adding modules and themes

Just a note: Before installing extra modules, you should consider the required functionality for your site, and only add a module if your site actually needs it. Unnecessary modules add to the memory and other resource needs of your site, and will make pages load slower. Because a module is popular or someone tells you that 'every site needs it' doesn't mean that yours does. There are good sites that have been built with absolutely no contributed modules at all.

Also, you should focus on content, functionality, and configuration of your site before you consider how it should look (the theme), since that decision can be made later and may detract from your thinking about functionality and content.

Themes are largely a matter of taste. For example, many people do not like to use a "fixed width" theme, but lots of people do. One nice thing about themes is that they are mostly independent of your content so you can switch them when you want. Caution, though, there are parts of your site (blocks, for example) that get built based on the theme, so switching themes can leave junk lying around to slow down your site.

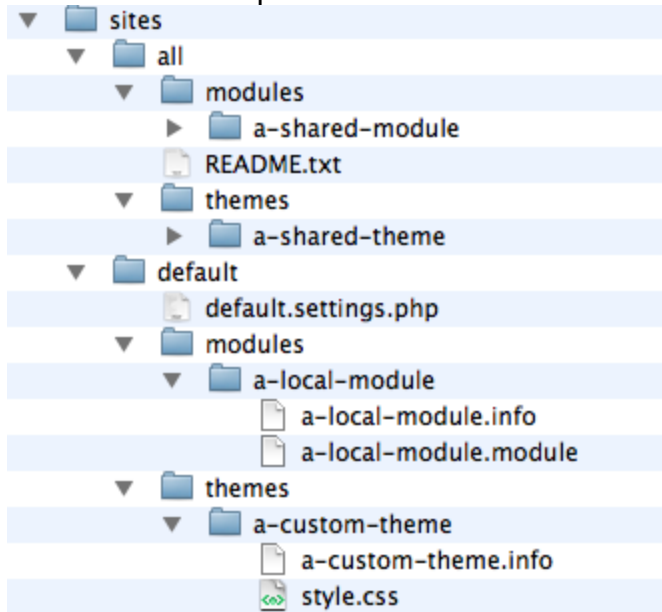
Now, if you experiment with different themes and modules, you should also look at the [Update Status](#) (core in D6) and [Site Documentation](#) modules to make sure you are current and to document and clean up the mess your experimentation will make. Here are some suggestions on choosing the release: [Strong stomach?](#)

For examples of different opinions on what are "must have" modules, [here is one](#), and [here is another](#).

Installation

1. Go to the Drupal site and click on the "Downloads" tab. Then select either "Modules" or "Themes" depending on what you're after.
2. Locate the module or theme you want.
3. Make sure there is a version for the version of Drupal that you are using. Drupal 5 modules and themes will not work on Drupal 6.
4. Click "Find out more" and read the notes. This gives you the chance to see if there is support for your release of Drupal. You can also look at pending bugs and feature requests - it might change your mind.
5. Download the release for your Drupal version.
6. Unzip the downloaded file (on Windows you can use a program like WinZip).
7. Extract the code to to your `/sites/{sitename}/modules/{modulename}` or `sites/{sitename}/themes/{themename}` folder. If you are not running multiple sites, this would be `/sites/all/modules/` or `/sites/all/themes/`.
8. That's it! Now you need to enable the module or theme.

You should end up with a folder structure something like:



For advanced users

If you have access to a Unix shell (e.g. SSH), the following series of commands will download and extract the files directly to your server.

Before you start, use the `pwd` (print working directory) command to confirm that you are in the correct directory (e.g. `modules`). If not, use the `ls` (list directory) and `cd` (change directory) commands to navigate to the correct directory.

Steps:

1. Download archive:
`wget http://copy_and_paste_archive_url/sitedoc-5.x-1.x-dev.tar.gz`
2. Uncompress archive: `tar zxvf sitedoc-5.x-1.x-dev.tar.gz`
3. Delete/Remove archive: `rm sitedoc-5.x-1.x-dev.tar.gz`

Note: the module **sitedoc-5.x-1.x-dev.tar.gz** is taken here as example please check the current version and or substitute with your preferred module.

Modules

To enable a module, click `Administer >> Site building >> Modules`. The non-core modules are listed farther down. With 5.x, they now show you some of the inter-module dependencies. You can turn them on and "Save configuration" in order of the dependencies. For example, "Views UI" requires "Views", so you can turn on "Views" first, save the configuration, then turn on "Views UI." and save again.

Most modules introduce some kind of menu items. Those will generally appear automatically when the modules are enabled. A few menu items will not show up until the permissions are set (the next step). And even fewer require you to take action to add the menu items, but the modules will have instructions on how to do that.

Now the real work begins. Click `Administer >> User management >> Access control` (or `Administer >> User management >> Permissions` in D6) to select who can use the features of the new module.

If the module introduced new content types, go to `Administer >> Content management >> Content types` and configure them. Don't forget this may also affect your "Input formats" (`Administer >> Site configuration >> Input formats`) and "Categories" (or taxonomy, `Administer >> Content management >> Categories`); you'll have to check those too.

Okay, now you can start using the new module.

This site

[Nancy's documentation site](#) is a relatively "vanilla" implementation of Drupal.

Core Modules Enabled	Contributed Modules in Use
----------------------	----------------------------

- | | |
|--|---|
| <ul style="list-style-type: none"> • Blog • Book • Comment • Contact • Help • Menu • Path | <ul style="list-style-type: none"> • Codefilter - Provides tags for automatically escaping and formatting large pieces of code; used for formatting code snippets. • Meta Tags (Nodewords) - Allows users to add meta tags, e.g. keywords or description. • Site Documentation - Documents and cleans up your configuration. • Taxonomy List - a demonstration for one of my modules. • Web Links - a demonstration for one of my modules. |
|--|---|

To get some idea of what modules are available, check these links: [module handbook](#) and [contributed modules handbook](#).

Themes

To enable a theme, click `Administer>>Site building>>Themes`

If the theme has never been enabled on this site, you will have to check the "enable" box and then click the "Save configuration" button at the bottom.

To set up how the theme works, click on the "Configure" **link** (not the tab at the top).

Fill in the fields.

Save the configuration.

Don't leave the page yet.

Logo and favorite icon

Now use that "Configure" tab

Some people prefer to do this part under the "Global Settings" but it can be done theme-by-theme.

The "Default Logo" is that little picture (usually) in the upper left corner of the page. For example, on the "Bluemarine" theme, it's the Drupal logo.

If you want to change it, here's how:

1. First, find out what size it is because you'll want your own logo to be about the same. (If you are comfortable with HTML and CSS, you can also edit the theme's code to accommodate your logo rather than resizing the image. How to edit theme code is not covered in this guide.)

<u>Theme Name</u>	<u>Width</u>	<u>Height</u>
Bluemarine	48	55
Chameleon	49	57
Garland	64	73
Minelli	64	73
Pushbutton	144	63
Fancy	80	80

2. Under "Logo image settings" either type in the path to your logo, or upload it from your PC.
3. **Note:** Neither one of those options turns off the "Use default logo" check box. You must select the correct check box yourself.
4. The "Shortcut icon" (a.k.a. the favorite icon, or "favicon.ico") is the same way. If you want to change this, you must specifically say, "Hey, Drupal, I'm changing this."

Now you click the "Save configuration" button. If you did this in "Global settings" it affects all themes (assuming they behave properly); if you did it for a single theme, then only that theme is changed.

For a list of all available themes, check [Themes](#).

HINT: Going to make a few (or a lot) of changes to a standard theme? Think about copying it over to your `/sites/sitename/themes/` folder and renaming it. Then you can do anything you want and still be able to undo it easily by recopying. If the changes are a bit bigger, think about contributing it back to the community (with your name, of course).

Adding a contact form

While a contact form is a simple concept, it can be a bit complex to get set up and operating. Read this whole page before doing it the first time. Don't give up; it really does work.

Setup

The easiest way to add a contact form to a basic Drupal site is to enable the contact module. Go to **Administer > Site building > Modules** and locate it in the list of core modules. Enable the check box and go to the bottom to save the change.

Click **Administer > Site building > Contact form**. Here you can set up the "Categories" - or recipient name/office.

For example, email for the Sales Department might be given a category of "Sales." The email address that the form is sent to may be `sales@mycompany.com`. If you want a reply automatically sent to the person sending the contact email, you can specify that here. Don't worry if you don't know them all right away, you can come back and change this at any time. Click the "Submit button."

Now click on the "Settings" tab. Here you can limit how many contact emails an individual may send in an hour -- this helps limit spamming. You may also turn on personal contact forms here; this allows users to contact each other. Click on the "Save configuration" button.

Grant access permissions

In order to allow people to use this contact form, you will need to give them permissions to do so.

Click **Administer > User management > Access control** (in D6 **Administer > User management > Permissions**), locate the "Contact module" entry and enable it for the roles that will use Contact. Save your changes. The menu link (next step) will not be visible to anyone not having access.

Add "Contact" to the menu

Go to **Administer > Site building > Menus**. You can add a 'Contact' menu item as described in the [Working with Menus](#) section.

Using it in content

To add a link to a content page use `Contact Us`.

Need a customized contact form? Check the [WebForm](#), [Contact Forms](#) or [CCK](#) modules.

Contact form spam

If you find you're getting spam emails through your contact form, check out the [Gotcha](#) module or use the [Captcha](#) module.

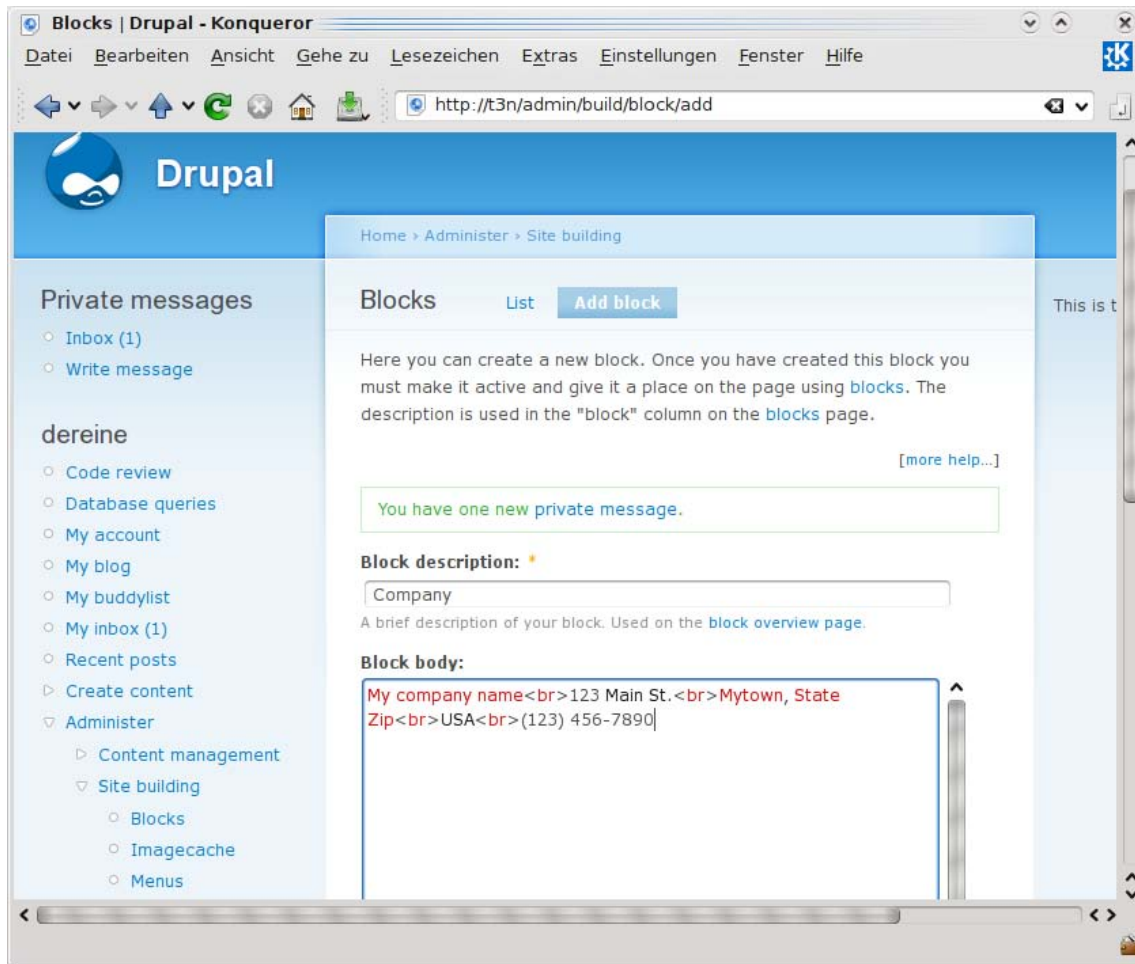
Blocks - adding extra information

Blocks are used to show extra information on pages, as is often seen in the left or right columns of a page.

Adding a 'Contact Information' block

Many websites include contact information in the sidebars. This is a very straightforward task to accomplish with Drupal.

1. Click *Administer >> Site building >> Blocks*.
2. Click the "Add block" tab.

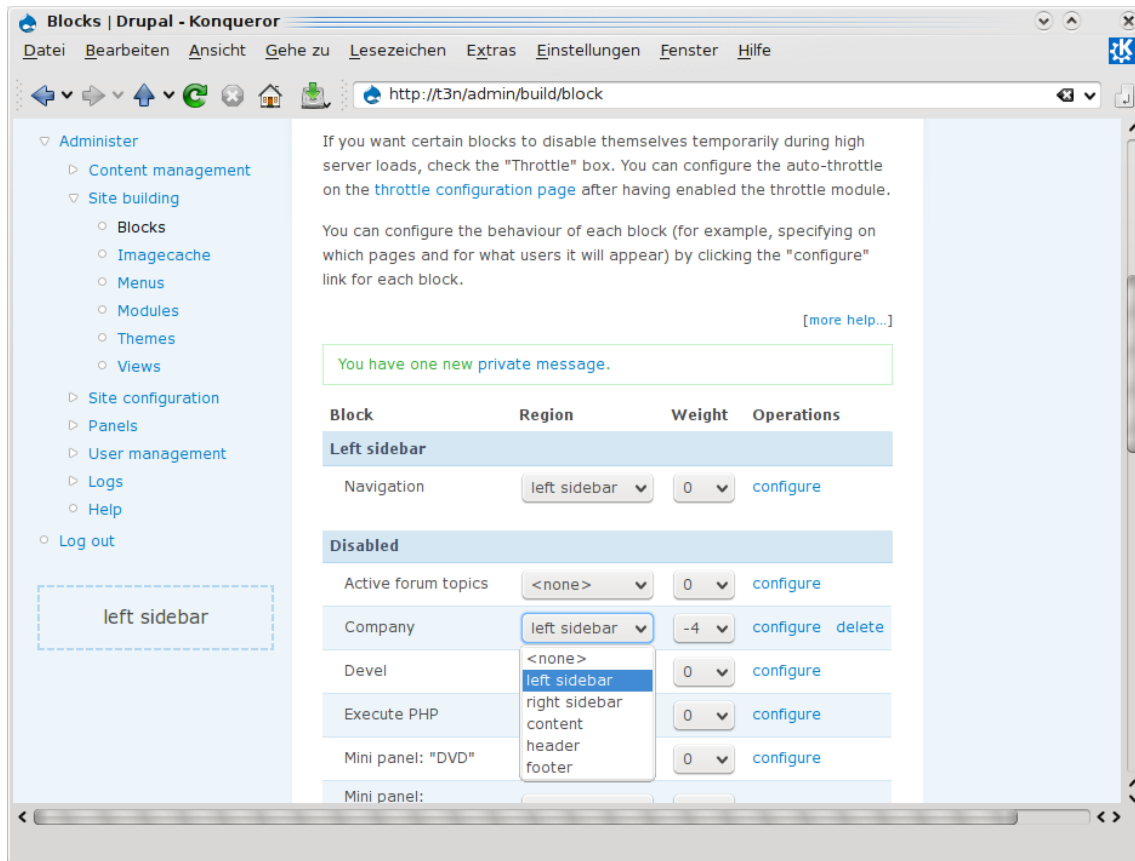


3. Fill in the "Description" and "Body." Here's a sample body:

```
Example Organization<br>
123 Main St.<br>
Mytown, State Zip<br>
USA<br>
(123) 456-7890
```

4. Save the block.

5. Now you can "Configure" the block to add the block's title and define its "Visibility".



6. Follow the Configure link next to the block and enter "Contact Information" as the block's title.
7. Decide if you want to allow users to turn the block on or off, and, if so, which roles should have that ability. Leave this as is for now. The next section allows you to choose which pages the block will be shown on; Leave this empty to show the block on all pages.
8. Save the block.
9. Now you're back on the block list. Find the block you just created in the list and choose a "Region". You can use the "Weight" parameter to set its position within the selected area. For example, if you like the address at the bottom, you can use a heavier weight.
10. Click the "Save blocks" button.

Migrating content from another site

Once your new Drupal site is up and running, your next challenge may be to port possibly years worth of content from a previous site.

There are probably as many ways to do this as there are sites on the web. Therefore no one method works for everyone, but you usually do not need to start from scratch. Various migration methods have been contributed to the community. There are even Drupal modules to convert an entire site from other standard CMSs into Drupal.

See the page, Migrating to Drupal in the Installation Guide for more information.
<http://drupal.org/handbook/migrating>

Moving stuff to your site

Occasionally, you will need to move files to your website (e.g. new modules or themes).

FTP stands for File Transfer Protocol. It is a common way to move files, modules, themes, etc. to a remote website. FTP is not used to edit the contents of your website.

In many web browsers, you can type: `ftp:username:password@ftp.example.com` (where *username* is your user name at your site, *password* is your password), *example.com* is your site's domain name), which will allow you to drag and drop files from your computer to the remote server. For more information on tools that you can use to transfer files to your website, see [Development tools](#)

It is unlikely (and inadvisable) for your FTP username and password to be identical to your Drupal username and password! They are probably different. FTP account login happens at a network level below the Drupal CMS. Your FTP credentials are probably supplied directly from your web host, or managed through your web hosts "Account Control Panel" somehow. Check your web hosts help documentation (or read the emails they sent you when you set it up) for more information.

Keeping these two types of accounts separate is more secure, even though it's more to remember.

Setting up cron

This page is for beginners; more complete information can be found on the main [cron page](#). If you can understand that page, then use it first.

Cron is a UNIX command that allows you to run tasks on a schedule. There are things in Drupal that will run on a timed basis, if cron is available.

Access to the shell means you can get to the operating system to issue commands. Many hosts won't allow this. But they do provide the more-or-less-standard cPanel function. On some versions, the "cron" entry is in the lower left.

For Cron jobs another possibility is <http://drupal.org/project/poormanscron>

For every page view, this module checks to see if the last Cron run was more than one hour ago (this period is configurable). If so, the Cron hooks are executed, and Drupal is happy. These Cron hooks fire after all HTML is returned to the browser, so the user who kicks off the Cron jobs should not notice any delay.

Additional tips and tricks

This page contains some common settings and describes the background of each to make your site easier to configure and maintain.

- Go to the **Administration > Site configuration > Site information** page. Many people don't like, when they visit a site, being called "Anonymous" so you might want to change the designation to "Visitor."
- If there are any modules (core or contributed) that you want to use, go ahead and enable them. Many people do recommend turning on (enabling) the "Path" core module so you can use "normal" names for your pages.

There are a few things that are commonly recommended for almost all sites, so this is a good time to do it:

- Turn on "clean URLs" to make your site more user friendly. Go to **Administration > Site configuration > Clean URLs**. At the bottom there is a link to run the "Clean URLs Test." If it passes, then the "Enable" radio button will be available. Click on that. (If the URLs stop working for some reason here are [instructions to unset clean URLs](#).)
- While preparing the site's initial content, it might be useful to go to **Administration > Site configuration > Input formats** and set "Full HTML" as the default, *but only until the site is ready to go live* (then it needs to be set back to "Filtered HTML" for security reasons). Then you may still allow administrators to use that format. Do this now and you will avoid a very [common problem](#) with building your site.
- Many people don't like having "Promoted to front page" as a default for content, so go to **Administration > Content management > Content types** and turn that off - in each content type.
- While you're there, decide on your default comment mode. Go to **Administration > Content management > Comments > Settings** (in D6: **Administration > Content management > Post settings**) and set the comments to be entered on a "separate page" and make sure that "Preview comment" is set to "Required."

If you want to use the contact form to email anyone from the site, be sure to enable the "Contact" module (see [Adding a Contact Form](#)).

- Now, let's turn on the Contact form so your users can send you a message. Go to **Administration > Site building > Menus** and locate the "Contact" item, it should listed be under the "Navigation" menu. Click on the "enable" link. Remember that later on you will want to go to **Administration > Site building > Contact form** and finish setting that up.

URL aliases

URL is an abbreviation of "Uniform Resource Locator" and is the page's address on the web. It is the "name" by which a browser identifies a page to display. In the example "Check us out at [mysite.com](#)." the part *mysite.com* is the URL for the home page of your web site. Users use URLs implicitly to locate content on the web.

Drupal uses "nodes" to identify pages by their position in your database. So your page on "The History of the Macadamia Nut - Part 1" might be known as "node/167." to Drupal. By maintaining the list of nodes in your site, the URLs are maintained as a result, which in turn enables your visitors to find your site's content.

Drupal has a feature called "URL Alias" that allows you to provide a more understandable name to the content. As far as browsers, servers, and search engines go, it is totally unnecessary. But for humans, it is nearly mandatory. This is why most consultants tell people to always turn on the Path core module, which supports URL aliasing.

So, just before you submit that treatise on macadamia history, and if you have the Path module enabled (and you will want it), then you'll see a section on the edit page that says "URL path settings." So let's say you want your visitors to see it as

`http://www.example.com/MacadamiaHistory`. In the URL field, enter `MacadamiaHistory`.

You can administer the URL Aliases directly on the node edit or add forms, or by doing the following:

First, visit the page you created. In your browser's address field, you'll see its URL. On the end it will probably say "node/xxx" where xxx is some number. Write down that number. Now go to **Administer > Site building > URL Aliases**. There's an "Add Alias" tab at the top. In the top

box, enter "node/xxx" from above. In the second box, enter "MacadamiaHistory".

Automatic Alias Generation

If your site is going to have lots of content, particularly user-submitted content, you might want to look at the [PathAuto module](#). Not only will this module automatically generate URL aliases for new content (according to rules you can set up), but can even go back and change aliases in bulk.

Path and clean URLs

Note that the Path module does not require Clean URLs. As an example:

With clean URLs:

- <http://www.example.com/MacadamiaHistory>

Without clean URLs:

- <http://www.example.com/?q=MacadamiaHistory>

Permissions

After you enable the Path module, you need to give the appropriate roles permission to use it in `admin/user/permissions`. Scroll down to the Path module to enable "administer url aliases" and/or "create url aliases" permissions.

User login FAQs

How do I disable "Create New Account?"

It's in `Administer » User management >> User settings`. Click the radio button that says: Only site administrators can create new user accounts.

How do I disable User Log In entirely, and how would I get in if I do?

First, you can go to `Administer » Site building >> Blocks` and change the "Region" setting for the "User Login" block to `<none>`. Now your user login block will disappear.

In order for you to login, here are two techniques:

- Enter `http://www.myexample.com/user` (or, if you don't use Clean URLs, `http://www.myexample.com/?q=user`). Yes, it is the word "user" -- not your user ID.
- Go to your Site information settings and stick a "login" link in the footer. `login` Many people do this on their sites.

Browser tips for accessing your site maintenance tools

A lot of people don't realize that their browser home page does **not** have to be out on the web somewhere. It can be right on your computer; just create an HTML file which will reside on your computer and specify your browser homepage as that file.

Earth 2025	First Citizens
Project Management Institute	PMI - Midlands (SC)
Activskin Performance Wear	Powerball
National Tartan Day	Scottish Festivals
College Football TV Schedule	Sooner Football News
Yahoo Mail	Web MD
BellSouth Home	FTP to BellSouth

Here's part of one admin's home page, with links to her Drupal sites' maintenance tasks (cPanel, FTP, etc.) included, so she doesn't have to keep looking them up.

She also included links to her test sites, so they are easy to get to as well.

Configure the browser Startup to "Show my homepage" not to "Show my windows and tabs from the last session". That way, it should always start up by displaying the home page you specify.

Tracking module status

A "site" you might want to build on your PC is a "catch-all" site for monitoring module and theme status with the [Update Status](#) module.

This module will check with the Drupal site and let you know if newer releases of your modules are available.

Many people eventually find that they need (want) to patch, tweak, or otherwise modify modules or themes. If you create a "catch-all" site, you can place unmodified code here so that you always have a "clean" copy to fall back on. You can then copy it to the correct site before changing it. Now, not only do you have the fall-back plan, but you get the added advantage of knowing if it needs to be updated.

Error pages

Occasionally, a user may do something that confuses Drupal, such as typing a wrong page name or trying to access content they shouldn't. These will generate 404 and 403 errors, respectively.

A recent SEO newsletter, they mentioned the value of letting Drupal handle these errors:

Your unique 404 error page should look like a regular page of your site. It should include your site's header, footer and navigation bar so that the site visitor can easily click on another area of your site. The content of this unique 404 error page should contain text explaining that the page selected is no longer available along with contact information so the site visitor has the option of emailing or calling your company.

The same more or less goes for the "access denied" (403) error message. Let them know they did

a no-no and try to explain why.

Just go to "Create content" and select "Page." Some might title them "Access Denied" and "Page Not Found" but you can call them whatever makes sense to you and your users. When you submit them, note the node IDs (the last number in the link displayed in your browser e.g. <http://www.example.com/node/3> -->nid=3). Then click Administration >> Site configuration >> Error handling (Or Administration >> Site configuration >> Error reporting for Drupal 6) and enter "node/nnn" in the appropriate boxes.

Page not found

Here's the HTML for one site's 404 page:

```
<p>Sorry! The page you were looking for no longer exists. We redesigned our site and many
of the pages have changed.</p>
<p>&nbsp;</p>
<p>If you are unable to find something on our new site or have a question about our site
or services feel free to <a href="/contact">contact us</a>.</p>
<p>&nbsp;</p>
<p>--Webmistress</p>
```

Access denied

Here's the HTML for one site's 403 page:

```
<p>We're sorry, but you must have permission to view the page you requested.</p>
<p>&nbsp;</p>
<p>If you are already a registered member of this site, please try <a href="/user">logging
in</a>.</p>
<p>&nbsp;</p>
<p>If you are not a member, you need to <a href="/join_us">join us</a>.</p>
<p>&nbsp;</p>
<p>If you have any questions about our site or group, please feel free to <a
href="/contact">contact us</a>.</p>
<p>&nbsp;</p>
<p>--Webmistress</p>
```

Don't worry that you haven't created the "join_us" page yet. This is an advantage to having URL Alias support (the Path module) enabled. Just add to your to-do list to create this page when you get to the "Creating Content" chapter in this book.

Links and IMG

Yes, you can link between pages in Drupal. It's the same as not having Drupal, except the name may be weird if you're not using the Path module (URL Aliasing).

The biggest mistake people make is not knowing that there needs to be a leading slash ("/"). Omitting this will probably cause a "404:Page not found" error and, depending on which browser you're using, additional problems, like being logged out.

For example, if your home page on a site is node #4, a link to it would look like this: `Home`, but with URL aliasing turned on, you can also code it like this: `Home`

A picture could be embedded with this tag: ``

Common problems

There are some problems we all seem to "find;" this section documents a few of them. [Remember to always search before posting on the forums.]

Tables messed up, images not showing, other "strange" problems with HTML.

Drupal defaults to filtered HTML; that is, only certain tags are allowed. Further, that input format also breaks long lines of text. The fix is easy: switch to the "Full HTML" input format. You can make that the default for administrators (like yourself). Note that you may still want the URL Filter and Code Filter modules turned on for this input format; they are not defaults.

Help, I turned on Site Maintenance, now I can't login!

About three out of four Drupal users have done this to themselves. You can still log in with `http://www.example.com/?q=user`. Note that "user" is exactly that - do not put your username there.

I don't want anonymous users to see "Create content."

"Create content" is actually a child menu item of "Content" which is usually disabled. Go to your menus administration screen and enable the "Content" parent above "Create Content." Then you will see a "Reset" link appear. Click on that. Once again, "Content" will be disabled, but "Create content" will not be shown to anyone who does not have the access permissions to do so (especially anonymous users).

I enabled the Contact module, but if I click on it as an anonymous user, I get the Access denied page.

You've missed a step. Go to Administration >> User management >> Access control (or Administration >> User management >> Permissions in Drupal 6) and scroll down to the contact module section. There you need to click on the check box for "access site-wide contact form" for "anonymous users" (and, presumably, "authenticated users").

Also read the handbook section [Troubleshooting FAQ](#)

More reading

Now that you're a Drupal expert, there are some additional topics you might find useful:

- [Adding Hidden Site Design Notes](#) - how to put design or maintenance notes on your site that only admins can see.
- [Core modules](#) - what the many core modules do.
- [Contributed modules](#) - a starter list to get you to the right modules, if you need them.
- [Site recipes](#) - collection of tricks and techniques.
- [Localhost or testing environment setup](#)
- [Multi-site Installation](#)
- [Site configuration challenge: corporate brochure](#) - many ways to get to a "Corporate Brochure" type site.
- [Best practices guidelines](#) - a guide to doing the right things.
- [My favorite module or theme is outdated. What next?](#) -- it happens!
- [PHP and Javascript snippets](#) -- useful code to use as is or adapt to your own needs.
- [SQL snippets](#) -- database stuff.
- [CSS Tips, Tricks, and Techniques](#)

- [Theme developer's guide](#)
- [How to write automated tests](#)
- [The Road to Drupal Hell](#) -- this should be required reading for anyone who wants to do something not in the core Drupal.

Glossary

A **node** is a container for stuff (sorry for the technical term). Some of that stuff is the content you create. Drupal itself creates a few nodes for its own stuff.

A **module** is a way to extend the functionality of Drupal. It is usually a lot of programmed code (usually in php) and, usually, a style sheet (CSS). For example, if you want to include meta tags to describe your content, you would add on the "Nodewords" module (also known as "Meta Tags"). *[Note: Similar concepts exist in other software and are sometimes called plugins, addons or extensions]*

A **teaser** is a short enticing phrase about the asset (page) to encourage readers to visit the full story. By default, the first paragraph or two of the page content is used (there is a setting for how much), usually displayed above a "read more..." link.

A **theme** is a means of manipulating and describing how you want your content displayed to your visitors. This includes elements such as your header, icons, block layout, etc. It also includes programming and style sheets.

A **server** is (generally) a computer that provides services to the Internet. These services may be things like running the database or managing the gathering and dissemination of information.

A **browser** is the "program" that you use to display content from the Internet. In reality, it is usually a set of programs, not a single one; it is also a set of tables (e.g. settings) that are used to control its display. Examples are Internet Explorer, Netscape, and Firefox. This operates on the **client**, or user, side of the presentation.

A **URL** (Universal Resource Locator) is the "address" of a resource (such as a page of content) on the web. It is the way the web browser locates your content or site. You will see the URL listed in the address bar on your browser.

A **path** is generally site-specific and refers to the means by which a resource is located. This could be a full URL (see above), or a relative location (such as "files/xyz/image.jpg" - where "files/xyz" would be the path to the file "image.jpg").

HyperText Markup Language (**HTML**) is the standardized language of the web. It has its own "vocabulary," consisting of tags, elements, and descriptors. A **tag** is the basic component and is used to say, "The following content is to be displayed according to these rules." An example of a tag is a level one heading (<H1>). Most tags can have additional information to tell the browser more specifically how you want it to render the content. This specification is called an **element**. Most elements require more information to make them work, this is the **descriptor**, which really should be called "value." For example, if you want that heading centered, you would use the "align" element and give it a descriptor (value) of "center." So , completely constructed it would look like this: <H1 ALIGN="CENTER">.

A **Taxonomy** is a way of characterizing stuff. It can be used for grouping, selecting, and protecting stuff. Many people who are new to Drupal think this is a very difficult subject (admittedly, we can make it so), however, virtually all of us had an introduction to taxonomy in

school: classifying living creatures (i.e. the Linnaean taxonomy). In that taxonomy, we classified living things according to kingdoms (plant or animal), phylum, class, and so on, down to genus and species. In reality there is an additional classification below species; sub-species (animals) or varietal (plants). [Oh, yeah, I vaguely remember that! That's a taxonomy?]

In Drupal, the highest level of taxonomy description is the "**vocabulary**;" it is used for defining the terms, or tags, that actually end up on your stuff to be used for the various purposes. In the above example, think of "Living things" as the vocabulary. Each vocabulary has one or more "**terms**" that are used to tag (i.e. define, or describe) your stuff. Terms may be hierarchical; that is they may exist in levels. Genus and species would be hierarchical terms. The vocabulary is assigned to input types (e.g. stories, recipes); terms are assigned to a given piece of content (e.g. "Groundbreaking Research on Macadamia Nut Yields" or "My Fabulous Macadamia Brittle"). Notice that I said "terms" - plural - because an individual node may have more than one term associated with it; for example, the "Research" news article may be assigned to "Nuts," "Trees," and "Harvesting." It could then be viewed through any of those terms (or keywords).

Breadcrumbs is a term borrowed from Hansel and Gretel, who left crumbs of bread along their path so they could find their way back out of the forest. In current computer parlance, it refers to the section, usually near the top of the page, that shows the path you followed to locate the current page. For example, it might show Home > Macadamia Nuts > Current Events > News Articles, meaning that you started at the home page, clicked on "Macadamia Nuts" in the menu, then selected "Current Events" in the sub-menu, and finally selected, "News Articles."

A **database** is a container for containers. It is a collection of related "tables" that are generally used for a single application (such as Drupal). A **table** is a collection of data used for a specific purpose within that application, such as identifying users. Within a table, each individual grouping of data is referred to as a **row** (or in traditional terms, a "record"). Each row is identified by one or more **keys** that allow easy retrieving of the row. Each row is then broken down into **columns** (often called fields, although this is more appropriate for forms on which the data is displayed). A column holds a specific piece of information for the row, such as a user name or country. (You can also think of this much like a spreadsheet.)

Now, just to complicate things a bit, some times we describe the collection of **item**-related table rows with a collective term. One such term is "node." For example, information about a page on your site may exist in several tables; yet we describe all of this as a "node."

Still confused? Let's try to relate this to an example you're probably familiar with. Let's relate this to your windows computer.

Your [hard] disk (or disc) is sort of like a database; it is a collection of your data. On that disk, you have folders; they are analogous to tables within a database. Inside those folders, you have documents or programs; these relate to rows. Within the document (e.g. a Word document), you have paragraphs; these are much like columns.

Okay, let's add to the analogy a little. Word or Lotus 1-2-3 would be your theme, as they describe and manipulate the content before it is displayed to you. It's a bit of a stretch for several reasons, but you can then think of Windows itself as your browser, since it is responsible for the final rendering of the content to you.

Does that help a little?

The Structured Query Language (**SQL**) is a standard specification for how database engines locate data that you want. An example might be `SELECT country FROM user_profile WHERE username = "Nancy"`; this would get the value from the column "country" in the "user_profile" table using column "username" as the key.

See also:

- [Terminology](#) for a more complete list of Drupal terms
- [Common expressions & acronyms](#)