2

# About nodes

**All content on a Drupal website is stored and treated as "nodes".** A node is any piece of individual content, such as a page, poll, article, forum topic, or a blog entry. **Comments are not stored as nodes but are always connected to one. Treating all content as nodes allows the flexibility to create new types of content**. It also allows you to painlessly apply new features or changes to all content of one type.

Behind the scenes, the Node module manages these nodes. This module lets you:

1. List, sort through, and manage all the content on your site.
2. Set defaults for how all posts are displayed.
3. List and configure the "content types" for your site, and create new ones.

**The Node module manages the creation, editing, deletion, settings, and display of the main site content**. Content items managed by the Node module are typically displayed as pages on your site, and include a title, some meta-data (author, creation time, content type, etc.), and optional fields containing text or other data. (Fields are managed by the [Field module](https://www.drupal.org/documentation/modules/field) in Drupal 7.)

**Each node has an** [**unique ID**](https://www.drupal.org/node/624494)**.**

## **Uses**

### **Creating content**

When new content is created, the Node module records basic information about the content, including the **author**, **date of creation**, and the content type. It also manages the ***publishing options***, which define whether or not the content is published, **promoted to the front page** of the site, and/or **sticky** at the top of content lists.

### **Creating custom content types**

The **Node module** gives users with the ***Administer content types* permission** the ability to create new content types in addition to the default ones already configured. Creating custom content types allows you the flexibility to add and delete fields and configure default settings that suit the differing needs of various site content.

### **Administering content**

The Content administration page allows you to review and bulk manage your site content.

### **Creating revisions**

The Node module also enables you to create multiple versions of any piece of content, and revert to older versions using the *Revision information* settings. You can review and revert revisions through the "Revisions" tab on any content, and add new ones in the "Revision information" settings when editing a given node.

### **User permissions**

The Node module makes a number of permissions available for each content type, which can be set by role on the permissions page.

# 

# 

# Demonstrate ability to create and configure Content Types with appropriate fields and field settings for building basic data structures

## Working with content types and fields (Drupal 7 and later)

In Drupal, a Content Type is a pre-defined collection of data types (Fields) which relate to one another by an informational context. In this sense, "context" means "parts that should be considered as a correlated whole".

Content Types define default fields for editors to add content on a Drupal site and are the building blocks for structured authoring and content.

**Field terminology**

* **Field.** A category of data that can be added to an element, for example: Title, Body, Comment body, Tags, Image.
* **Field type.** The datatype of a field, for example, text, integer, image.
* **Field instance.** A field added to an element (either an [entity type](https://www.drupal.org/node/1261744) (such as a node or taxonomy term) or a [bundle](https://www.drupal.org/node/1261744) (such as an article or page)).

An element (i.e., an entity type or bundle) can have only one instance of each field. However, a field can have a setting called "Number of values," which determines how many times a field can appear in an element. For example, if an instance of the field Image is added to the content type Basic page with "Number of values" set to 1, 3, or unlimited.

The primary (but not all) information about **fields** and **field instances** is stored in, respectively, the database tables "**field\_config**" and "**field\_config\_instance**". The **field type of a field** is stored in the **column "type"** of the **table "field\_config".**

## 

## 

## **The Field UI**

The Field UI module provides an administrative user interface (UI) for attaching and managing fields. Fields can be defined at the content-type level for content items and comments, at the vocabulary level for taxonomy terms, and at the site level for user accounts.

The Field UI is a form used for managing fields. It comes up when you click on the *Manage Fields* tab of an editing form, for example, when editing a content type (such as Basic page) (/admin/structure/types/manage/page) or a taxonomy term (such as Tags) (/admin/structure/taxonomy/tags/edit). It can also be brought up by clicking on "manage fields" in the list of content types (/admin/structure/types).

The Field UI has two parts:

* A list of field instances of the element being edited.
* A pair of subforms for adding new field instances.   
  These subforms are:
  + **Add new field**. This is used to both create a new field and to add an instance of that field to the element being edited.
  + **Add existing field**. This is used to add an instance of a field that already exists.

### Planning fields

There are several decisions you will need to make before defining a field for content, comments, etc.:

* **What the field will be called**A field has a label (the name displayed in the user interface) and a machine name (the name used internally). **The label can be changed** after you create the field, if needed, but **the machine name cannot be changed** after you have created the field.
* **What type of data the field will store**Each field can store one type of data (text, number, file, etc.). When you define a field, you choose a particular field type, which corresponds to the type of data you want to store. **The field type *cannot* be changed after you have created the field.**
* **How the data will be entered and displayed**  
  Each field type has one or more available "widgets" associated with it; each widget provides a mechanism for data input when you are editing (text box, select list, checkboxes, file upload, etc.). Each field type also has one or more display options, which determine how the field is displayed to site visitors. **The widget and display options can be changed after you have created the field.**
* **How many values the field will store?**   
  You can store one value, a specific maximum number of values, or an unlimited number of values in each field. For example, an employee identification number field might store a single number, whereas a phone number field might store multiple phone numbers. **This setting can be changed after you have created the field, but if you reduce the maximum number of values after inputting/entering data, you may lose information.**

### Reusing fields

Once you have defined a field, you can reuse it. For example: if you define a custom image field for one content type, and you need to have an image field with the same parameters on another content type, you can add the same field to the second content type, in the Add existing field area of the user interface. You could also add this field to a taxonomy vocabulary, comments, user accounts, etc.

**Some settings of a reused field are unique to each use of the field; others are shared across all places you use the field. For example, the label of a text field is unique to each use, while the setting for the number of values is shared.**

There are two main reasons for reusing fields. **First, reusing fields can save you time over defining new fields. Second, reusing fields also allows you to display, filter, group, and sort content together by field across content types.**

There is one main reason to **not reuse a field: different permissions**. For example, you may need different user roles to have different levels of access to a field, depending on the content type to which it has been added. This can be difficult if you reuse a field.

* **Fields on content items**Fields on content items are defined at the content-type level, on the Manage fields tab of the content type edit page (which you can reach from the Content types page).
* **Fields on taxonomy terms**Fields on taxonomy terms are defined at the taxonomy vocabulary level, on the Manage fields tab of the vocabulary edit page (which you can reach from the Taxonomy page).
* **Fields on user accounts**Fields on user accounts are defined on a site-wide basis on the Manage fields tab of the Account settings page.
* **Fields on comments**Fields on comments are defined at the content-type level, on the Comment fields tab of the content type edit page (which you can reach from the Content types page).

### Remove field

If you need to remove a field completely one would delete it from all existing content types where it exists. The Field UI can tell you what content types a field is being used on before you try and delete the field.

However, some modules create fields automatically and must be uninstalled to have their fields removed. After you disable the module, you'll have to go to the uninstall tab to complete the uninstall process.

field\_delete\_field($field\_name); field\_purge\_batch(); // note: only run this if you need the field deleted immediately!! otherwise it will just be deleted the next time the cron runs

If you manually delete fields from the database, many things can go wrong. For example, hook\_field\_delete will not be called and one of your modules might fail and crash the site. For this reason, it is best to only use the Field UI for deleting fields if you aren't comfortable in the code or with SQL/databases.

The **Field module** allows custom data fields to be attached to Drupal entities (content nodes, users, taxonomy vocabularies, etc.) and takes care of storing, loading, editing, and rendering field data. Most users will not interact with the Field module directly, but will instead use the [Field UI module](http://drupal.org/documentation/modules/field-ui) user interface. Module developers can use the [Field API](http://api.drupal.org/api/group/field/7) to make new entities "fieldable" and allow fields to be attached to their entities.

**The Field module provides the infrastructure for fields and field attachment**; the field types and input widgets themselves are provided by additional modules.

## Field Storage API

Implement a storage engine for Field API data.

**The Field Attach API uses the Field Storage API to perform all "database access". Each Field Storage API hook function defines a primitive database operation such as read, write, or delete.** The default field storage module,[field\_sql\_storage.module](https://api.drupal.org/api/drupal/modules%21field%21modules%21field_sql_storage%21field_sql_storage.module/7.x), uses the local SQL database to implement these operations, but alternative field storage backends can choose to represent the data in SQL differently or use a completely different storage mechanism such as a cloud-based database.  
  
Each field defines which storage backend it uses. The Drupal system variable 'field\_storage\_default' identifies the storage backend used by default.

# 

# 

# Create a custom Content type

1. Navigate to the Content types page (*Administer > Structure > Content types*).
2. Click **Add Content Type**.
3. In the **Name** field, add a name for the content type. The name is required.
4. In the **Description** field, you can write a description of the content type that will show up on the Add new content page (*Administer > Content > Add new content*). The description is not required.
5. Set your preferred default values for this content type. Any content that you create of this content type will have these values by default. The values can be overridden for each content created. You can specify default values for **Submission form settings**, **Publishing options**, **Display settings**, **Comment settings** and **Menu settings**.
6. Once your preferred default values are saved, you can either click **Save Content Type** or **Save and Add Fields**. Save Content Type leaves the content type with the Title and Body fields, while Save and Add Fields allows you to add more fields to the content type. You can always add more fields later.
7. When adding fields, you have the choice to either add a new field or add an existing field.
8. Once your content type has the desired fields, you can create an instance of it as a node by navigating to the content page (*Administer > Content*). Then select the name of your content type.

**Submission form settings**

**In the Title field label, you may add a title for your content type.** For example, if your content type is "Furniture," you can choose a title of "Furniture Type." Note: **The default for this field is "Title."**

**Publishing options**

The default publishing options are **Published** and **Promoted to front page**. Additional options are available as **Sticky at top of lists** and **create new revision**.

**Display settings**

The Display Settings has one option which allows the **Author username and publish date** to be **displayed with the content**.

**Comment settings**

You may leave comments **open, closed, or hidden**. The **default** values allow for **50 comments** to be published per page in a **threaded list**. There are also default options as **Allow comment title** and **Show reply form on the same page as comments**

**Menu settings**

In the Menu settings you can choose which default parent item to use. This comes in handy when using taxonomy in your site.

# **Add a field to a content type**

1. Navigate to the Content types page (*Administer > Structure > Content types*).
2. In the table, locate the row that contains your content type and click the **MANAGE FIELDS** link. If you can't find the **MANAGE FIELDS** link, be sure to enable the Field UI module.
3. In the Add new field section, enter a label for the field.
4. In the Field name field, enter the machine name for the field. You can only use numbers, lower-case letters and the underscore character (\_). Node that the machine name is automatically generated and it is not necessary that it be changed.
5. In the field type list, select one of the following types.

* **Boolean**has one of two values (for example: "yes/no" or "true/false" or "small/large")
* **File**reference to a file (such as a PDF) on the Drupal file system
* **Image**reference to an image file such as GIF, JPG, PNG on the Drupal file system
* **Integer**a whole number, such as a year (e.g. 2012) or value (e.g. 1, 2, 5, 305); it does not allow decimals.
* **Float**a number that can use decimals, such as 0.0123456789
* **Decimal**a number that allows **exact** decimal values; often used for price or cost (such as $199.99)
* **List (float)**select from a drop-down list of floats
* **List (integer)**select from a drop-down list of integers
* **List (text)**select from a list of text options (which can be formatted as either a drop-down list or checkboxes)
* **Text**short text such as a name (limited to 255 characters)
* **Long text**long, multi-line alphanumeric text such as a biography
* **Long text and summary**same as long text, but with an additional summary text
* **Term reference**reference id to an existing Taxonomy Term (or tag)

1. In the Widget field, select a widget. The options vary depending on the field type.
2. Click **Save**.

# **Rearrange the order of fields**

Rearranging fields changes their order as displayed when a user creates or edits content with these fields.

1. Navigate to the Content types page (*Administer > Structure > Content types*).
2. In the table, locate the row that contains your content type and click the **manage fields** link.
3. Locate the field you want to re-order, and drag the four-headed arrow icon up or down.
4. Click **Save**.

# **Specify how fields are displayed**

Content can be displayed differently depending on whether it's in a list with similar content, in search results, or being viewed in full. For example, content displayed in a list is typically trimmed so it doesn't take up too much space on the page and may only display the most important fields.

## **What is a view mode?**

Drupal refers to the different ways a piece of content can be displayed as *View modes*. All entities, including user profiles, taxonomy terms, and blocks (D8 only) have at least one view mode, plus a ***Default*** view mode which is used when a view mode has not been explicitly set.

## **Hiding, showing and ordering fields and their labels**

For each view mode, Drupal allows you to:

* show and hide fields,
* change the order that fields are rendered in,
* hide or show the field's label, and,
* change how certain types of fields are displayed (**Default, Plain text, Trimmed, Summary or trimmed and hidden**).

### 

## **Changing how nodes are displayed**

Drupal lets you set different display settings for each content type. To change how a content type is displayed:

1. Navigate to the *Content types* page (*Administer > Structure > Content types*).
2. In the table, locate the row that contains the content type you wish to change, and click the **manage display** link. This takes you to the *Default*view mode settings page.
3. At the bottom of the page, in the *Custom display settings* section, check the checkboxes next to the view modes you want to customize and click **Save**.
4. Once the page re-loads, you will see links under the tabs at the top of the page. There will be a link for each view mode you selected in the previous step. Clicking one of these links will take you to a screen where you can hide, show, reorder, and customize fields.

**Out-of-the-box, Drupal defines five view modes for nodes**:

### Full content

The most detailed view of a node. You see this when you visit the node's URL. **This is normally used only when visiting the URL for a particular node.**

### Teaser

A short summary of the node, intended to catch visitors' interest so they will view the Full content. Displayed on many of Drupal's default listing pages, such as the front page and taxonomy term listing pages. **This is used on two list types – the front page and taxonomy term lists.**

### RSS

How the node is displayed in RSS feeds. Visitors or webservices subscribed to the RSS feed will see content that looks like this. **These settings are used when including the node in RSS feeds.**

### Search index

The content that Drupal's internal search sees when it goes to index the page (search engines such as Google typically see the *Full content* view mode). **This is used when Drupal core's Search module indexes content on the site. Fields that are hidden in this view mode will not be searchable.**

Typically, you only customize this view mode if you want to *hide* data from Drupal's internal search.

### Search result

A short summary of the node when Drupal's internal search displays it as a search result (search engines such as Google generate their own search result view, typically from the *Full content* view mode). **This is used when presenting search results on the website with Drupal core's Search module.**

## Taxonomy terms

1. Navigate to the *Taxonomy* page (*Administer > Structure > Taxonomy*).
2. In the table, locate the row that contains the vocabulary you wish to change, and click the **edit vocabulary** link.
3. Click the **Manage display** tab. This takes you to the *Default* view mode settings page. Since Drupal only defines one view mode for taxonomy terms, you may hide, show, reorder, and customize fields here.

**Out-of-the-box, Drupal defines one view mode for taxonomy terms:  
Taxonomy term page**The most detailed view of a taxonomy term. You see this when you click on a taxonomy term link from a node page, or you visit the URL of the taxonomy term directly.

You can only customize the upper portion of the taxonomy term page (the part that contains data about the term itself). Drupal displays a listing of all nodes tagged with the term (in their *Teaser* view mode) in the lower portion.

## Users

1. Navigate to the *Account settings* page (*Administer > Configuration > People > Account settings*).
2. Click the **Manage display** tab. This takes you to the *Default* view mode settings page. Since Drupal only defines one view mode for user accounts, you may hide, show, reorder, and customize fields here.

**Out-of-the-box, Drupal defines one view mode for user accounts:  
User account**The most detailed view of a user account. You see this when you log in, or when you visit the URL of the user directly.

## **Field label display options**

Field labels can be displayed in the following ways:

**Above**

The field label is displayed on it's own line above the field value(s).

**Inline**

The field label is displayed before the field value(s), on the same line.

**Hidden**

Completely removes the field label for all users.

Typically, you only choose this if you only have two fields which are supposed to appear as if they are a single field.

## **Manage Display Page Details**

Manage Display page contains a table with 3 columns with following titles:

**Field**

The field column contains all the visible and hidden field names.

**Label**

The label column contains options to show label *Above, Inline or Hidden* respectively. These values are described above.

**Format**

The format column contains the options to show the field value in different formats. *Hidden* is the global value which is present in all the format option list, if you choose hidden then it will hide the whole field from showing on the content page.

# Demonstrate ability to create and use Taxonomy vocabularies and terms for classification and organization of content

## About taxonomies

Taxonomy is the practice of classifying things. In Drupal, the Taxonomy module allows you to classify your website content, and it can be an important part of your information architecture.

### Planning taxonomies

The first step in establishing a taxonomy is **creating a new vocabulary**. Next you provide **the terms that fall within that vocabulary.** **The arrangement can be "flat," as in a tagging system, or hierarchical, with parents and children.**

An alternative model for organizing information is the use of "tags." User-defined tags can be added to Drupal content on the fly. In **Drupal 7**, a ready-to-use "Tags" vocabulary is included by default, so users can immediately begin adding tags to their content.

### Taxonomy principles

* Each vocabulary consists of a set of terms.
* A site can have an unlimited number of vocabularies.
* Each vocabulary has an associated ID number. Each taxonomy term has an associated ID number.
* Each vocabulary can contain an unlimited number of terms.
* Within a vocabulary, terms can be ordered into hierarchies. All vocabularies can be arranged into a hierarchy without taking additional steps, if you find this useful.
* In Drupal 7, you don't need to predefine the type of vocabulary in order to create a "tagging" vocabulary.

### Adding a vocabulary

* Choose a name for your vocabulary.
* Give your vocabulary a description. Modules may use this description in different ways. (For example, a module may show the description when users hover over a link.)
* In Drupal 7, vocabularies are only assigned to content types through the addition of a "term reference" field for that content type. One step in setting the "term reference" field will be to choose the vocabulary from which the term will be chosen. Once you've assigned a vocabulary to a content type, content creators will be able to choose to assign a term from that list to their content.
* You can also give your vocabulary "help text" to help your users choose. Note that when creating a new content type, existing vocabularies do not have the new content type selected; you need to edit them to add the type.
* Vocabularies can have hierarchies of terms. In Drupal 7 and 6, you simply arrange items to create a hierarchy.
* Finally, you can delete the vocabulary altogether, thereby also deleting all its terms, but not the nodes to which they were assigned.

### Working with Terms

To view or manage the terms of each vocabulary, click on its *list terms* link. On the *list terms* page you can edit each term by clicking the *edit* link. Now, on the *edit term* page you have several kinds of choices.

* You can put the term in its place in the hierarchy by choosing the term's "parent" .
* You must assign your term a name as a "nameless term" does not exist.
* You can decide the order in which your term will appear in lists by assigning it a "weight".
* You can delete a term altogether.

To add new terms to your vocabulary, click its *add terms* link. The *list terms* page also has an *add terms* link that does the same thing. When you add new terms you have the same options as when you edit them.

### Using Categories in Menus

The menus on your site can call for items that match specific taxonomy terms. To create a menu using Taxonomy, follow these steps:

* Find the number given to your taxonomy term (to find the term number, go to the *categories* page, *"list terms"* for the category to which your term belongs, and hover over the term. You'll see the number.)
* Go to the *menus* page (*Administer > Site building > Menus*)
* Select *Add item,*
* When you fill in the *Path* field you add your term as follows:
* taxonomy/term/1
* If the term "Sonatas" is term 1, this would call for all the nodes of that category.

**(In Drupal 7, you can no longer specify paths such as "taxonomy/term/1,2", "taxonomy/term/1+2","taxonomy/term/2/2" or "taxonomy/term/2/all". Function taxonomy\_term\_page (Drupal 7) does not parse the string as it did in Drupal 6.)**

# 

## Add a field to a taxonomy term (Drupal 7 and later)

Taxonomy now depends on the [Field](https://www.drupal.org/handbook/modules/field) module. This allows you to add new fields to your Vocabulary Terms as you would to a Content Type. Please note that in order to add new fields you will need to have the [Field UI](http://drupal.org/handbook/modules/field-ui) module enabled. Adding a field to a Vocabulary will add that field to all Taxonomy Terms of the Vocabulary - not the Vocabulary object itself.

Because Drupal 7 entities (content nodes, users, taxonomy vocabularies, etc.) uses the [Field UI](http://drupal.org/handbook/modules/field-ui) module to manage fields you can just follow the directions over at <http://drupal.org/handbook/modules/field-ui> in order to add/remove a field to/from vocabularies.

## Add a vocabulary to a content type

To use taxonomy terms to organize your content, the vocabulary must be associated with the content type. To add vocabulary, go to structure->taxonomy->add vocabulary. Here you can add a vocabulary list. Once created, you will see it appear in the list and can click to "view terms" and then will be able to add terms to the vocabulary list.

## Add a term to a vocabulary

Once you have finished defining the vocabulary, you may populate it with terms using Add. Add terms to a vocabulary by navigating to admin/structure/taxonomy/[your-vocabulary-name]/add. From there, Drupal will prompt for:

* **Term name** (Required) -- The name for this term. Example: Technology.
* **Description** (Optional) -- Description of the term (this item may be used by some modules and feeds).

Advanced options:

* **Parents** (Optional): Select the term(s) under which this term is a subset.
* **Weight** (Optional): The weight is used to sort the terms of this vocabulary; by default they will be sorted alphabetically.
* **URL Alias** (Optional, D7): Default is "/taxonomy/term/" but here you can specify an alias like "term1" and the final url will be "/term1".

## 

## 

## Create a vocabulary

In D7, Vocabularies are used to group, organize and, in many cases, categorize a set of taxonomy terms. Vocabularies are fieldable entities and are given a name and vocabulary id (vid) making it referenceable by other Drupal components. So, vocabularies can be thought of as parent or root containers for taxonomy terms. Creating a vocabulary can be as simple as assigning the container a name. You can, optionally, add fields (e.g., an image field, etc.) to your vocabulary at admin/structure/taxonomy/[your-vocabulary-name]/fields.

To create a vocabulary in D7, go to admin/structure/taxonomy, then click Add Vocabulary. Drupal will prompt for:

* **Vocabulary name** (Required) -- A name for this vocabulary; for example, Topics.
* **Description** (Optional) -- A description of the vocabulary (this item may be used by some modules and feeds).

That's it, your vocabulary is created and waiting for you to add some terms to it. Other contributed modules like the [Taxonomy Menu](https://www.drupal.org/project/taxonomy_menu) module may add additional configurable settings to the admin/structure/taxonomy/add page.

**Note that D7 comes with a empty 'Tags' vocabulary already setup for you. The 'Tags' vocabulary is a 'free-tagging' vocabulary that will hold user created terms that are added to the Tags vocabulary when content is created or edited. This is generally accomplished by using a term reference field with an autocomplete widget on the content creation page.**

## 

## Guidelines for taxonomy design

## **General Rules**

**1.** **Unless a vocabulary is well-known to all anticipated users -- for example, an alphabetical list of world countries -- keep it below 30-40 terms.**

**2.** **Use parent-child relationships with caution. If your taxonomy relies on these structures, think about dividing up the vocabulary.** An unwieldy parent-child scheme is often a sign of poor design.

**Example**

An online art shop has taxonomize its products as follows:

* Europe
  + Lapp
  + Sami
  + Celtic
* Australia
  + Aboriginal

Even though this accurately shows how the shop classifies its items, its insistence on many small categories prevents a user from saying, "show me some European art."

Multiple-select is one way around this, but not the best way. It's better to use *two vocabularies*: one for region, one for culture. Make them both multiple-select, and our art collector can now ask: "What have you got which is Lapp, or Celtic or Chinese?"

**3. Too many terms**

"Perfect" taxonomies are always too complex and you need to fight to make them more manageable (especially if you have just cut up a few parent-child vocabularies into several smaller ones).

The advantage of [Views](https://www.drupal.org/project/views) is that the multiple taxonomy terms start to build context, and that can be captured by Views.

There are not many sites that ever need to show more than 3-4 filters to users, even if there are 5-6 more hidden ones.

Plus, building them into structure adds even more granularity.

**Example**

A site for all the small towns of America, where people are interested in their little town's stuff, using [Book](https://www.drupal.org/handbook/modules/book) for basic structure.

Top level - states - 50 items

Off each state - counties

Off each county - settlements

Click the named settlement and get a Views list with filters for news, culture and announcements.

(Hidden filters in the Views filter by State/County/Settlement)

Highly non scary - any users can use that to get what they need.

**4. Clients are not experts on taxonomy, not even their own**

Taxonomy is a communications issue and if there is a budget for the site it's always worth running it past an outsider - but note that they will need to get to understand the purpose of the site and also to an extent the jargon of the subject.

This normally requires at least one decent face to face meeting to force the client to decide what is important, and what can be cut out.

**5. Taxonomy creates legacy issues - so get it right!**

Once you have a load of tagged data, it's hard to make changes to taxonomy structures (apart from adding terms) without rendering existing nodes much harder to find.

**6. Taxonomy is trial and error**

It should be the first thing you do on a site, but by adding test data you'll find flaws, and refine and eventually go live with something that works.

## Organizing content with taxonomies

Taxonomy, a powerful [core](https://www.drupal.org/glossary#core) module, gives your sites use of the organizational keywords known in other systems as categories, tags, or metadata. It allows you to connect, relate and classify your website’s content. In Drupal, these terms are gathered within "vocabularies". The Taxonomy module allows you to create, manage and apply those vocabularies.

Drupal 7 and 8 has the ability to add taxonomy fields to vocabularies and terms.

[Taxonomy](http://en.wikipedia.org/wiki/Taxonomy) is the practice of classifying content. It will come in handy for everything from menu and navigation schemes to view and display options.

Taxonomy can be used in workflow, to customize defined sections of your website with different themes or to display specific content based on taxonomy terms. Although taxonomy can be used in various ways, probably the most important use of taxonomy in Drupal is to relate content.

**Taxonomy should be driven by the business requirements of your website, with an eye towards possible future functional expansion. Here are some questions to help you determine how you may want to use taxonomy:**

1. Are there subsections of your site that you would like to look different from the main theme?
2. Are there content areas of your site that should be edited only by a specific part of your organization?
3. Is there content that can be shared around your site (such as a press release, form or fact sheet)?
4. Is there a business need to support local sites such as service centers or local events?
5. Are there different states you need to set (such as left navigation / breadcrumbs) for site sections?
6. Are there needs around providing default lists of content by taxonomy term or default RSS feeds by term?

Many contributed modules rely on Taxonomy-generating; for example, menus based on existing tags.

## Using taxonomy URLs to display sets of content

When displaying nodes, both in teaser listings on the Drupal home pages and in full, single-node view, many Drupal themes display the categories applied to the node. If the user selects any category term, Drupal will then display a browsable listing for all nodes tagged with that term.

Examine the Taxonomy URL for one such category listing. The end of the URL should look something like this:

taxonomy/term/1

And another Taxonomy URL, for a different term, something like this:

taxonomy/term/2

Note that Taxonomy URLs always contain one or more Term IDs at the end of the URL. These numbers, 1 and 2 above, tell Drupal which categories to display.

In addition to displaying Drupal nodes by category on site, Drupal has category specific RSS feeds for other sites to access your site content. See how the URL format for the RSS feed is very similar to the Taxonomy URL:

taxonomy/term/1+2/0/feed

This feature has been removed in Drupal 7.

Building individual Taxonomy URL's is not the most user friendly way to provide site users access to browsable listings. Nor do administrators necessarily want to build custom blocks for users with links to each category listing.