

Drupal 6 API cheatsheet

Important hooks

<code>hook_menu()</code>	Tell Drupal what path your module will handle. "node/1" is defined by the <code>node_menu</code> .
<code>hook_nodeapi(&\$node, \$op)*</code>	Take action in response to node events.
<code>hook_user(\$op, &\$edit, &\$account)*</code>	Take action in response to user events.
<code>hook_comment(&\$a1, \$op)</code>	...comment events. Relatively inconsistent/weak :(Comments as nodes! http://groups.drupal.org/node/18705
<code>hook_form_alter(&\$form, \$form_state, \$form_id)</code>	Modify any form. <i>Modify. Any. Form.</i>
<code>hook_boot(), hook_init(), hook_exit()*</code>	Take actions as the page request starts ends.

User-specific

<code>user_load(\$uid)</code>	Load a fully populated user object. Note: different data/format than <code>{global \$user}</code> .
<code>user_save(\$account, \$array)*</code>	Save a user object—can be used to create users.
<code>user_access(\$string)*</code>	Test whether a user (defaults to global <code>\$user</code>) has a specific permission.

Node-specific

<code>node_load(\$nid)</code>	Loads a fully populated node object. Infamous for being slow (if you need one piece of data, query directly instead of node_loading it).
<code>node_save(\$node)</code>	Save a node. Consider <code>drupal_execute()</code> ing it instead.
<code>node_access(\$op, \$node)*</code>	Test whether a user has access to a specific node.

Database

<code>db_query(\$query)</code>	Execute a query.
<code>pager_query(\$query, \$limit)</code>	Perform a paged database query.
<code>db_query_range(\$query, [printf substitutions], \$from, \$count)</code>	Execute a query that returns <code>\$count</code> results.
<code>db_fetch_array(\$result)</code>	Fetch a record from a db result set <i>as an array</i> .
<code>db_fetch_object(\$result)</code>	<i>... as an object</i> .
<code>db_rewrite_sql(\$query)*</code>	Modify a query to respect node, comment, and taxonomy access restrictions.
<code>db_result(\$result)</code>	Fetch a single result from a db result set.

Interactive version at growingventuresolutions.com/drupal6-api-cheatsheet

Function signatures show the most common usage which omits defaults and uncommon optional arguments. Hook functions explicitly show pass-by-reference modifier (&) where relevant.

* Asterisk indicates additional optional parameters not listed.

GrowingVentureSolutions.com

Theming

<code>theme()</code>	Applies appropriate theme function to unformatted content. Allow overriding of functions.
<code>*_preprocess_hookname (&\$variables)</code>	Add variables for templates (<i>tpl.php</i>) files, tell the theme layer which code (<code>theme_override</code> or <i>tpl.php</i>) to use to render a specific page.
<code>*_preprocess(&\$variables, \$hook)</code>	
<code>theme('item_list', \$items)*</code>	Turn an array of items into an HTML list.

Textual

<code>l(\$text, \$path)*, url(\$path)*</code>	Create a URL or an HTML formatted link.
<code>t(\$string, \$args)*</code>	Translate a message using placeholders to insert dynamic text. <i>Remember: @ and %</i> filter text to prevent XSS. ! avoids it.
<code>drupal_set_message(\$message)*</code>	Display a message to the user.
<code>drupal_set_title(\$title)</code>	Set the page title.
<code>truncate_utf8(\$string, \$len, \$dots)*</code>	Cut off a chunk of text to a specific size in a word-boundary & UTF-8 safe way.

Text filtering

<code>check_plain(\$text)</code>	Represent text as "plain" text using html entities.
<code>check_markup(\$text, \$format)*</code>	Filter text through the appropriate input format.
<code>drupal_html_to_text(\$string)*</code>	Convert HTML to plain text by stripping html tags—useful for mail.
<code>filter_xss_admin(\$string)</code>	Filter text liberally. Appropriate for text entered by admins like the site mission.

Random functions

<code>drupal_goto(\$path)*</code>	Stop processing and send a user somewhere. <code>\$_REQUEST['destination']</code> may be better.
<code>watchdog(\$type, \$message)*</code>	Add information to the log.
<code>dpm(\$var)</code>	Output variable for debugging. Requires Devel module.
<code>drupal_mail(\$module, \$key, \$to, \$language)*</code>	Send e-mail. See also <code>hook_mail()</code> .
<code>drupal_add_js(\$data, \$type)*</code>	Add Javascript to a page.
<code>drupal_json()*</code>	Convert a variable to JSON, set the HTTP MIME type header, and exit().
<code>menu_get_object(\$type, \$position)*</code>	Returns the loaded object (node, user, etc.) that has been returned by the menu system.
<code>menu_get_active_trail()</code>	Find the menu ancestors for the current path. Returns an array of menu objects..
<code>drupal_http_request(\$url)</code>	An HTTP client implementation capable of GET, POST, and other requests.

Drupal 6 API cheatsheet

Basic functions

<code>hook_distract_chx()</code>	Must always return FALSE. See also <code>chxcannotbedistracted.com</code> .
<code>hook_throwyourhandsintheair(\$like_you_care = FALSE)</code>	Use with the block regions "left" and "right" so you can wave them side to side.
<code>hook_mood_alter(&\$user)</code>	Useful when your friends get upset with you for working with Drupal too much.
<code>theme_tinylittleduckies()</code>	Reserved for when Ze Frank upgrades his site to Drupal (and for fan sites in the mean time).

Command-line mini cheatsheet

Install module

```
wget http://ftp.drupal.org/files/projects/views-6.x-2.3.tar.gz
tar xzf views-6.x-2.3.tar.gz
```

or

```
drush pm install views
drush mm enable views
```

Checkout code, make patch, update to next release

```
cd sites/all/modules/
export CVSROOT=:pserver:anonymous:anonymous@cvs.drupal.org:/cvs/
drupal-contrib
cvs co -d views -r DRUPAL-6--2-3 contributions/modules/views
cvs diff -up > path_to_share.patch
patch -p0 < path_to_apply.patch
cd sites/all/modules/views
cvs up -r DRUPAL-6--2-4
```

MySQL database backup

```
mysqldump -u mysqluser p mysqlpass > output_file.sql
mysql -u mysqluser -p mysqlpass < input_file.sql
```

Get the db authentication information:

```
grep db_url sites/default/settings.php
```

Resources worth reading that are longer than a card

http://api.drupal.org/api/file/developer/topics/forms_api_reference.html/6

http://api.drupal.org/api/file/developer/topics/forms_api.html

<http://drupal.org/theme-guide>

Copyright © 2009 Growing Venture Solutions, LLC. ♻️ Printed on recycled paper.

Some rights reserved. Licensed under Creative Commons Attribution-Share Alike 3.0:



<http://creativecommons.org/licenses/by-sa/3.0/>

Attribute by linking to growingventuresolutions.com/drupal6-api-cheatsheet

GrowingVentureSolutions.com

Form

<code>drupal_get_form(\$form_id)</code>	Render a form object into an HTML form.
<code>system_settings_form(\$form)</code>	Render a form object with "save/reset to defaults" buttons. Automatically saves elements as variables in the variable table.
<code>form_set_error(\$name, \$message)*</code>	Useful during form validation. Adds a CSS class and shows an error message to the user.
<code>drupal_execute(\$form_id, \$form_state)</code>	Programmatically submit a form through the validation/submit processing functions.

Condensed Form API example

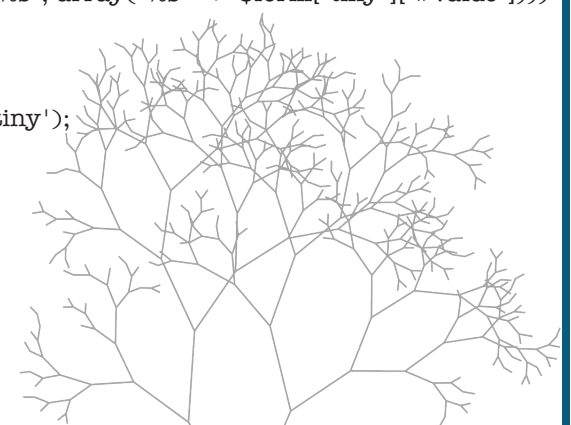
For coding standards see <http://drupal.org/node/318>

```
function little_menu() {
  $items['form'] = array('access arguments' => array('access content'),
    'page callback' => 'little_form', 'type' => MENU_CALLBACK);
  return $items;
}

function form_tiny() {
  $form['tiny'] = array('#type' => 'textfield', '#title' => t('Tiny is'));
  $form['submit'] = array('#type' => 'submit', '#value' => t('Save'));
  return $form;
}

function form_tiny_submit($form, &$form_state) {
  drupal_set_message(t('Tiny is: %s', array('%s' => $form['tiny']['#value'])));
}

function little_form() {
  return drupal_get_form('form_tiny');
}
```



gvs

Growing Venture Solutions is a full service website development company focused on Open Source solutions for the publishing industry.

We provide site building, training, support and consulting services.