Drupal 6 API cheatsheet

GrowingVentureSolutions.com

Important hooks

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

Random functions

drupal_goto($path)*	Stop processing and send a user somewhere. $_REQUEST['destination'] may be more appropriate.
watchdog($type, $message)*	Add information to the log.
drupal_mail($module, $key, $to, $language)*	Send e-mail. See also hook_mail().
drupal_add_js($data, $type)*	Add Javascript to a page.
drupal_json()*	Return an array (object?) as JSON encoded(?) data.
menu_get_object($type, $position)*	Returns the loaded object (node, user, etc.) that has been returned by the menu system.
menu_get_active_trail()	Find the menu ancestors for the current path. Returns an array of menu objects..

Database

db_query($query)	Execute a query.
pager_query($query, $limit)	Perform a paged database query.
db_query_range($query, [printf substitutions], $from, $count)	Execute a query that returns $count results.
db_fetch_array($result)	Fetch a record from a db result set *as an array.*
db_fetch_object($result)	*... as an object.*
db_rewrite_sql($query)*	Add limitations to a query to respect the node access system.
db_result($result)	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

Theming

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

Textual

l($text, $path)*, url($path)*	Create a URL or an HTML formatted link.
t($string, $args)*	Use the translation system to translate a [dynamic] message using placeholders to insert dynamic text. Remember: **@** and **%** filter text to prevent XSS. **!** avoids it.
drupal_set_message($message)*	Set a message at the top of the page.
drupal_set_title($title)	Or set the title.
truncate_utf8($string, $len, $dots)*	Cut off a chunk of text to a specific size in a word-boundary & UTF-8 safe way.

Text filtering

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

User-specific

user_load(array('uid' =>1))	Load a fully populated user object. Note: different data/format than {global $user}.
user_save($account, $array)*	Save a user object—can be used to create users.
user_access($string)*	Test whether a user (defaults to global $user) has a specific permission.
hook_user()	(See **Important hooks**.)

Node-specific

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

Drupal 6 API cheatsheet

GrowingVentureSolutions.com

Form

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

Command-line mini cheatsheet

Install module

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

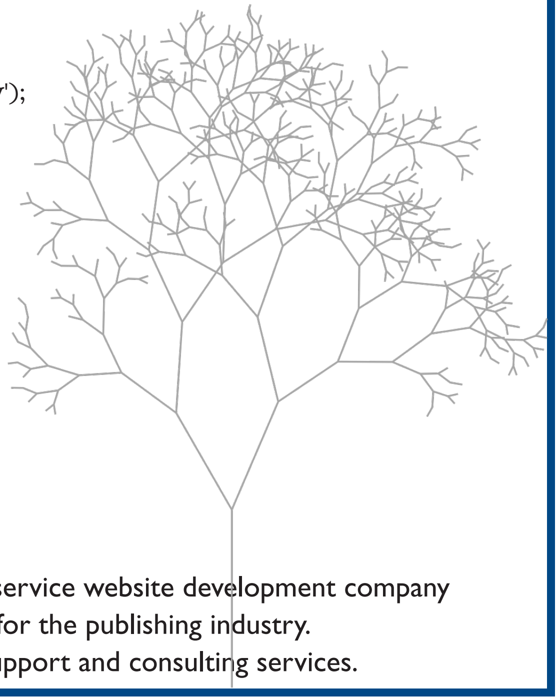
Form API example

```
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.