

Core PHP translation API	
<code>t('Home')</code>	Translates 'Home' to the requested language.
<code>format_plural(\$count, '1 comment', '@count comments')</code>	Returns the proper translation for the text based on plural rules in the requested language.
Optional variable replacements array and language code can be provided in this order.	

JavaScript translation API	
<code>Drupal.t('Home')</code>	Translates 'Home' to the language used for page display.
<code>Drupal.formatPlural(count, '1 comment', '@count comments')</code>	Returns the proper translation for the text based on plural rules in the page's language.
Optional variable replacements array can be provided, <b>language is always the page language!</b>	

Other PHP functions using translation features	
<code>format_date(\$timestamp, 'medium', ...)</code>	Translates parts of the date/time formatting.
<code>format_interval(\$timestamp, ...)</code>	Displays the time interval in human terms. Translates names of the segments.
<code>format_size(\$size, ...)</code>	Formats a file size provided in bytes. Translates the MB, KB and byte labels.
All functions accept an optional language code at the end.	

Replacement formats (based on prefix characters)	
<code>@count</code>	Replaced with the value escaped with <code>check_plain()</code> for output.
<code>%size</code>	Replaced with the value escaped and formatted as a placeholder (theme ('placeholder')).
<code>!html</code>	Replaced with the value as is, without any additional formatting. <b>Insecure! Only use if you already sanitized your text.</b>
Example: <code>t('%name moved to @directory', array('%name' =&gt; \$filename, '@directory' =&gt; \$dirname));</code>	

Text extracted for translation and used later	
<code>.info</code> files	Module/theme name; description; package name.
<code>hook_perm()</code> implementations	Literal permission names, when provided with standard array syntax.
<code>hook_menu()</code> implementations	Menu item 'title' and 'description' keys, when provided as literal strings.
<code>watchdog('menu', 'Items saved')</code> calls	Log type and message, when provided as literal strings. Third parameter specifies the replacements array. <b>No plural support.</b>

Translating text in the installer	
<code>st('Home')</code>	Works the same way as <code>t()</code> , but uses a file+memory backend. Use only in the installer and in install profiles. <b>No plural support.</b>
<code>\$t = get_t(); \$t('Home');</code>	Use <code>get_t()</code> in <code>.install</code> files and any other code which could run in the installer too. It returns 't' or 'st' depending on the function to be used at the moment.
Remember: if you have an <code>.install</code> file and need to use translation, your function calls could easily end up running in the installer if your module becomes a core module in an install profile. Use <code>get_t()</code> !	

The most common mistakes		
Using dynamic values, such as <code>t(\$type)</code> . Whenever you use dynamic strings to call the above functions, make sure that all possible values are already in the code elsewhere in literal form using the API. <b>Do not translate user provided content.</b>	Reuse of the same message eg. for <code>watchdog()</code> and <code>drupal_set_message()</code> calls. The former disallows <code>t()</code> , the later requires <code>t()</code> , so this is not possible. <b>Always call the above API functions directly, without indirection.</b>	Usage of <code>t()</code> in a global context, such as <code>define('CONST', t('...'))</code> . The locale system is not yet initialized at that time, so this usage is incorrect and also leads to nasty performance problems. Never do it.

Tools to verify you are using the API properly	
<a href="http://drupal.org/project/potx">http://drupal.org/project/potx</a>	Drupal 6.x-2 or later versions have coder module plugin support, so improper use of the API is reported in reviews. Also feel free to export a template for your module/theme and check whether all strings appear.
<a href="http://drupal.org/project/l10n_client">http://drupal.org/project/l10n_client</a>	The potx module cannot find mistakes when you forgot to use <code>t()</code> or some other function to translate your output. The localization client has a convenient interface to translate text displayed on the page. If certain strings are not showing up for translation, you forgot to use the API.