



Important note: When you edit this page, you agree to release your contribution into the **public domain**. If you don't want this or can't do this because of license restrictions, please don't edit.

This page is one of the *Public Domain Help Pages*, which can be freely copied into fresh wiki installations and/or distributed with Foresight Wiki software; see [Help:Contents](#) for an overview of all pages.



If you have standard texts you want to include on several pages, the MediaWiki template feature comes into play.

Contents

- [1 Creation](#)
- [2 Usage](#)
 - ♦ [2.1 Parameters](#)
 - ♦ [2.2 Control template inclusion](#)
 - ♦ [2.3 Organizing templates](#)
- [3 See also](#)

Creation

Templates are standard wiki pages whose content is designed to be transcluded (embedded) inside other pages. Templates follow a convention that the name is prefixed with "Template:", assigning it to that namespace. Therefore you can [create them like any other wiki page](#).

Usage

Templates are wiki pages which can be used in other pages in three ways:

- `{{Name}}` 'transcludes' (i.e. includes a copy of) the content of the template (stored in the page `[[Template:Name]]`) whenever the page containing the template transclusion is fetched and displayed; i.e. if the template is later changed, the displayed transcluding page will automatically change too
- `{{subst:Name}}` replaces that string with the contents of the template, in the source of the transcluding page, when you save that page; the copy of the template contents can then be edited normally (and separately from the original in the template page). **Note:** don't use this if you are looking to continually propagate changes from the source template to the page(s) that references it.
- `{{msgnw:Name}}` includes the template in a form that displays it as raw wiki syntax (the way `<nowiki>` does) when the page containing it is fetched

In fact, an ordinary wiki page can also be used as a template, simply by specifying the namespace it resides in, so:

- `{{Template:Pagename}}` includes `[[Template:Pagename]]`
- `{{Foo:Pagename}}` includes `[[Foo:Pagename]]`
- `{{:Pagename}}` includes `[[Pagename]]`

If no such namespace exists, the full title is assumed to be a template:

- `{{Foo:Bar}}` includes `[[Template:Foo:Bar]]`

Parameters

Template with numbered parameters	
<pre>'''A little thank you...'''
 <small>for {{{1}}}.
 hugs, {{{2}}}</small></pre>	
You type	You get
<code>{{Thankyou all your hard work Joe}}</code>	 Thank you! We invite everyone to contribute constructively to our foresight website. Your apology or cleanup of <u>all your hard work</u> was greatly appreciated. I look forward to your further positive contributions. Joe
with named parameters	
<pre>'''A little thank you...'''
 <small>for {{{reason}}}.
 hugs, {{{signature}}}</small></pre>	
You type	You get
<code>{{Thankyou reason=all your hard work signature=Joe}}</code>	 Thank you! We invite everyone to contribute constructively to our foresight website. Your apology or cleanup of <u>all your hard work</u> was greatly appreciated. I look forward to your further positive contributions. Joe

You can define parameters in templates either numbered as `{{{1}}}` or named `{{{param}}}`.

Example: You want a little thank you note you can put on the talk page of other users. It will contain a reason and your signature. You could create [Template:Thankyou](#) to enter your text, as in the example in the table.

When using the template on a page, you fill in the parameter values, separated by a pipe char (|): `{{Thankyou|all your hard work|Joe}}`. For named parameters use "name=value" pairs separated by a pipe char: `{{Thankyou|reason=all your hard work|signature=Joe}}`. The advantage of using named parameters in your template is that they are flexible in order. It also makes the template easier to understand if you have many parameters. If you want to change the order of numbered parameters, you have to mention them explicitly: `{{Thankyou|2=Joe|1=all your hard work}}`.

You can also provide default values for parameters, i.e. values that are going to be used if no value is provided for a parameter. For example, `{{{reason|all your hard work}}}` would result in *"all your hard work"* if no value was provided for the parameter `reason`.

Control template inclusion

You can control template inclusion by the use of `<noinclude>` and `<includeonly>` tags.

Anything between `<noinclude>` and `</noinclude>` will be processed and displayed only when the template's page is being viewed directly.

Possible applications are:

- Categorising templates
- Interlanguage links to similar templates in other languages
- Explanatory text about how to use the template

The converse is `<includeonly>`. Text between `<includeonly>` and `</includeonly>` will be processed and displayed only when the page is being included. The obvious application is to add all pages containing a given template to a category, without putting the template itself into that category.

Note: when you change the categories applied by a template, the categorization of the pages that use that template may not be updated until some time later: this is handled by the job queue.

Organizing templates

For templates to be effective users need to find them and be able to use them. A simple technique is to include an example on the template page. For example:

```
<noinclude>
== Usage ==
Allows to establish a link to a subject:
{{NameOfTemplate|Term1+Term2+Term3}}
</noinclude>
```

Then, an editor can simply copy and paste the example to create a similar page.

See also

- [Help:External searches](#) ? a template special use case example
- [Help:Magic words](#) ? fancy stuff you may find in some templates
- [Help:Parser functions in templates](#)