



Semantic MediaWiki quick reference

Covers MediaWiki extensions *Semantic MediaWiki 1.4*, *Semantic Result Formats 1.4*, *Semantic Google Maps 0.4*, *Semantic Forms 1.3* and *Semantic Drilldown 0.5*

Properties

Special pages: Types, CreateProperty, Properties, UnusedProperties, Browse, SearchByProperty, Ask

Defining a triple: `[[property-name::value]]`
`{{#set:property-name=value}}` (silent definition)

Defining a property:
`[[Has type::type-name]]`
`[[Allows value::enum-value]]` (for enumerations)

Property types: String, Page, Text, Code, Number, URL, Email, Date, Boolean, Geographical coordinate, Temperature

Inline queries

Syntax: `{{#ask:argument1|argument2|...}}`

Parameters:
query-conditions link={none|subject|all}
? *property-name*=*label* default=
limit= intro=
offset= searchlabel=
sort= format=
order={asc|desc} sep=
headers={show|hide} template=
mainlabel=

Query conditions: `[[Category:category-name]]...`
`[[Concept:concept-name]]...` `[[property-name::value]]...`

Subquery: `[[prop1.prop2.prop3::value]]`

Operators: `::` (equals), `::<` (less than), `::>` (greater than), `::~` (like), `::!` (not)

Format types: *standard:* list, ol, ul, table, broadtable, rss, vcard, icalendar, csv, embedded, template, count, debug; *from Semantic Result Formats:* timeline, eventline, calendar, graph, googlebar, googlepie, sum, average, min, max; *from Semantic Google Maps:* googlemap

Sample query: to create a table of all cities in Europe and their populations, sorted by population:

```
{{#ask:mainlabel=City|[[Category:Cities]]
|[[Has country.Has continent::Europe]]
? Has population=Pop.|sort=Has population}}
```

Defining a concept: Within the 'Concept' namespace:
`{{#concept:query-conditions|description}}`

Semantic templates

Special pages: CreateTemplate, Templates

Simple semantic template definition:

```
Field-label: [[property-name::{{{field-name}}}}]
[[Category:category-name]]
```

Setting a field to hold multiple values:

```
{{#arraymap:|value|delimiter|var|
formula|new_delimiter}}
{{#arraymaptemplate:|value|
template-name|delimiter|
new_delimiter}}
```

Forms

Special pages: CreateForm, Forms, CreateCategory

Simple form definition:

```
{{{for template|template-name}}}
field-label: {{{field|field-name}}}
{{{end template}}}
{{{standard input|free text}}}

{{{standard input|save}}}
```

Form tags:
info field
for template choosers
end template standard input

'info' parameters:
add title= partial form
edit title= page name=

'for template' parameters:
label= chooser=
multiple chooser caption=
strict

'field' parameters:
input type= hidden
size= mandatory
rows= restricted
cols= uploadable
class= values=
maxlength= default=
list preload=
delimiter= autocomplete
no autocomplete
autocomplete on property=
autocomplete on category=
autocomplete on concept=
autocomplete on namespace=
remote autocompletion

'standard input' types:
free text save
minor edit preview
watch changes
summary cancel

'standard input' parameters:
label=

Input types

Field type	Default input type	Other allowed input types
Page, String, Number, etc.	text	textarea
Text, Code	textarea	text
Date	date	
Enumeration	dropdown	radiobutton
Boolean	checkbox	
List of Page, String, etc.	text	textarea
List of Enumerations	checkboxes	listbox

Form input:

```
{{#forminput:form-name|size|value|button_text|
query_string}}
```

Form input query string options:
template_name[*field_name*]=
namespace= super_page=

Attaching red links to forms:

(in a property page)
`[[Has default form::form-name]]`
`[[Has alternate form::form-name]]`
(in a namespace page (*project-namespace*:
namespace-name))
`[[Has default form::form-name]]`

Setting the 'edit with form' tab:
(in a category or namespace page)
`[[Has default form::form-name]]`

Forms that set an automatic page name

In form definition:
`{{{info|page name=...<template-name|field-name|> ... <unique number>}}}`

Linking to the form:
`{{#formlink:form-name|link_text|link_type|
query_string}}`

Drilldown

Special pages: CreateFilter, Filters, BrowseData

Defining a filter:

```
[[Covers property::property-name]]
[[Has value::value]]
[[Uses time period::{Year|Month}]]
[[Gets values from category::category-name]]
[[Has input type::{text|date range}]]
[[Requires filter::Filter:filter-name]]
[[Has label::label]]
```

Adding a filter to a category:
`[[Has filter::Filter:filter-name]]`