The Minx Book 1.2

Selection Tags

Posts

Quickly View posts.

Usage

[posts {paged|unpaged} {page=*n*} {count=*n*} {skip=*n*}
{show=all|publish|draft|pending|expire|user|friend|foaf}
{sticky|unsticky|nosticky|stickyonly} {type=post|page|podcast|photo|...}
{sort=posted|updated|title|comments} {order=asc|desc}
{status=publish|draft|hide|delete|...} {search=*options*} {filter=*options*} {default=*options*}
{category=*name*} {site=*address*} {folder=*path*} {sites=*list*} {folders=*list*}]

[posts:here {template=*type.name*} {defaulttemplate=*type.name*} {options}]

Type block tag, here tag

Data Fields

(none)

Text Fields

title, subject, text, keywords, more, memo, meta, misc, abstract, excerpt, blurb, notes, custom, tags, reply, mod, warn
{.raw|.esc|.text|.clean}

Info Fields

data, xml
{.raw|.esc|.text|.clean}

Date Fields

date, time, datetime, shortdate, day, month, year

create.date, update.date, start.date, end.date, comment.date, ping.date
[.date|.time|.datetime|.shortdate|.day|.month|.year]

Status Fields

Template Tags

status, type, category, comments, pings, edits, reads, locked, allow.comments, allow.pings, allow.attach, sticky, hidden, defer, expire, flag, ip, ip.hash, filter, number, even, odd, mod.[3|4|5], bytes, kbytes, words, first, last

Display Fields

show.html, show.bbcode, show.smilies, show.macro, show.dict, show.sig

User Fields

author.name, author.nick, author.mail, author.web, author.aim, author.msn, author.yim, author.icq, author.location, author.occupation, author.interests, author.sig, author.hash

ID Fields

id, tid, pid, fid, etid, cfid

Category Fields

category.id, category.name, category.desc, category.path, category.nospace, category.tagline

Folder Fields

folder.id, folder.name, folder.desc, folder.path, folder.nospace, folder.tagline

Site Fields

site.id, site.name, site.desc, site.short, site.host, site.tagline

Sub-Tags

Sub-Posts

[post:subposts]
[post:comments]
[post:pings]
[post:chat]
[post:discuss]
[post:footnotes]
[post:items]
[post:notes]
[post:pages]
[post:remarks]
[post:replies]

[post:responses]
[post:talk]
[post:trackbacks]

Attachments

[post:attachments]
[post:apps]
[post:audio]
[post:files]
[post:flash]
[post:images]
[post:media]
[post:movies]
[post:music]
[post:programs]
[post:songs]
[post:text]
[post:zips]

Other Details

[post:links]
[post:related]
[post:references]
[post:stats]
[post:categories]
[post:folders]
[post:sites]
[post:next]
[post:prev]
[post:first]
[post:last]
[author:friends]

Description

The [posts] tag selects a set of *entries*, which are the mainstay of many types of websites including blogs and journals. The entries selected are of type post. See the chapter *Objects* for an explanation of entries and related structures in Minx.

The tag can be used either in block form:

Template Tags

[posts]
template code
[/posts]

Or in *here* form:

[posts:here]

In this form, the Item.Post template is used to format each entry.

Options

paged, unpaged

Controls whether the listing of entries is broken up into pages. For this and other thread-level tags, the default is paged. For most other tags, the default is unpaged.

page=*n*

Defines a specific page of entries to be shown by the template. Typically, this would be selected by the user as they navigate through your site, but you are free to specify it if you find this useful

count=*n*

Sets the number of entries to display, either in total or in a given page. If this is not specified, then for paged views the default page size will be used, and for unpaged views, all entries will be shown. The latter case may result in an extremely large page; it is typically only used when either it is know in advance that the number of entries is reasonable, or it is specifically desired that every item is shown, for example, when generating a backup file.

skip=*n*

Skip ahead a given number of entries. For example, if you wish to show the most recent post with a custom layout, followed by a standard listing of the next 20 posts, you could set count=20 skip=1.

show=all|publish|draft|pending|expire|user|friend|foaf

Selects which entries to show based on general criteria.

all	shows all available entries regardless of status.
publish	shows only entries marked as Publish.
draft	shows only entries marked as Draft.
pending	shows entries that have a start date that has not yet been reached.
expire	shows entries that have an end date that has passed.
user	shows entries posted by the current user.
friend	shows entries posted by friends of the current user.
foaf	shows entries posted by friends of friends of the current user.

The default setting is publish.

sticky, unsticky, nosticky, stickyonly

Controls the behaviour of posts marked as announce, sticky, or unsticky.

The default behaviour is sticky, in which case posts marked as announce appear first, followed by sticky posts, then normal posts, and finally unsticky posts.

If unsticky is specified, then the sticky settings on the posts are ignored, and the posts are shown in whatever order they would appear naturally.

If nosticky is set, only posts marked as normal as shown, i.e., no announce, sticky, or unsticky posts.

If stickyonly is set, only posts marked as announce, sticky, or unsticky are shown.

You cannot specify both sticky and unsticky, or both nosticky and stickyonly. Other combinations are allowed, though they may not make sense.

type=*list*

Selects entries of a specific type. By default, the posts tag will select entries of type post (naturally enough), but you can override this to select other types, or to select multiple types of item in a single stream.

To select multiple types, simply list the types separated by commas , or pipes |

[posts type=post|page] ... [/posts] will select both posts and pages and combine them together in whatever order you choose.

[posts type=post,page] ... [/posts] is exactly equivalent. You can't mix commas and pipes in the one list, though – use one or the other.

[posts type=all] will select all types of entries.

See also: *Thread Types, Entry Types, Post Types*

sort=*option*

Sort the list of entries in one of the following orders

date	The date of the entry. (Default descending.)
posted	Same as date.
blog	Same as date.
updated	The date the entry was lasted updated or commented. (Default descending.)
forum	Same as updated.
title	Alphabetically by title. (Default ascending.)
comments	By number of comments. (Default descending.)

order=*option*

Modifies the sort option, as follows:

asc	Sort in ascending order.
desc	Sort in descending order.

status=*list*

Selects entries marked with a specific status. The default is publish. As with the type option, you can select multiple statuses by separating them with commas or pipes.

[posts status=publish,draft] ... [/posts] will list both published entries and entries still in draft.

See also: *Entry Statuses*

template=*type.name*
defaulttemplate=*type.name*

In *here* form (i.e., when using [posts:here]), overrides the default template for the individual entries, normally Item.Post.

The defaulttemplate is used if the initial query returns no results and the default search options are used instead.

search=*options*
filter=*options*
default=*options*

The search, filter, and default options can be used the specify a search string to be applied when selecting entries. Search terms are separated by slashes / with the exact same syntax as when specified in a URL.

You can use either search or filter. If you use both, the search options may be combined in unpredictable ways.

The default search string will be applied if the initial query returns no entries.

The following search options are available:

search	Performs the (site-defined) default search.
filter	Performs the (site-defined) default filter.
archive	Searches for entries within a specified year, month, or day.
date	Searches for entries posted on a specific date.
author	Searches for entries by a specific author.
tag	Searches for entries labeled with a specific tag.
keyword	Searches for entries containing posts labeled with a specific keyword.
title	Searches for entries where the title contains the search term.
page	Specifies the page of results to return.

search and filter options may be further specified by the following attributes:

.a	Search by author name.
.b	Search text of both posts and comments (and any other sub-posts).
.c	Search text of comments only.

Template Tags

.i	Title search.
.j	Subject search.
.k	Keyword search.
.p	Search text of posts only.
.q	Quick search.
.s	Smart search – searches multiple fields.
.t	Tag search.
.w	Wiki search (exact title match).
.x	Extended search.
.z	Extra-extended search.

(See *Searching and Filtering* for more information on these options.)

category=*name*

Select entries in a specific category.

site=*address*
sites=*list*

Selects entries from another site (if that site has granted permission for this) by specifying the address of that site. You can optionally specify the path of a folder within that site as well, thus:

site=example.mee.nu
site=example.mee.nu/news

In the sites form of this option, you can specify a comma- or pipe-separated list of sites (and optionally folders):

sites=example.mee.nu/news,demo.mee.nu/features

folder=*path*
folders=*list*

Selects entries from another folder by specifying the path of that folder. If you have specified an alternate site, this will select or override the folder within that site; otherwise it selects a folder within your own site.

folder=news/local

In the folders form of this option, you can specify a comma- or pipe-separated list of folders (again, by the path of the folder). You can use this in conjunction with the site option, but not with the sites option.

folders=news,sport,weather
site=example.mee.nu folders=news/national|features|opinion/politics

Fields

When specifying a field, you generally need to use the form [item.field], thus, for example:

[post.title]
[post.create]

Some sub-fields are also made available directly, so you can use either of:

[post.author.name]
[author.name]

String and date sub-element options follow the field name:

[post.title.raw]
[post.create.shortdate]

See *Tag Formatting* for more information.

Template Tags

Text Fields

title	A title for the entry as a whole.
subject	A title for the individual post.
text	The main or introductory text of the post.
more	Extended or additional text.
memo	Additional text.
meta	Additional text.
misc	Additional text.
abstract	An overview of the content of the post.
excerpt	A short excerpt from the post.
blurb	A short, catchy introduction to the post.
notes	Notes for the author/editor.
custom	A custom text field for any purpose.
tags	A comma-separated list of tags.
keywords	A comma or space-separated list of keywords.
reply	A reply to the post. Typically used for author responses to comments.
warn	A response to the post. Typically used for official moderator action on comments.
mod	A response to the post. Typically used for moderator response to comments that does not rise to the level of an official warning.

Info Fields

data	Additional data.
xml	Additional data in XML format.

Text Sub-Elements

By default, text fields automatically have all applicable processing done for you, ready for use. This includes HTML sanitising, BBCode parsing, macro and dictionary processing, smilie (emoticon) handling, and optional text filters.

You can also easily access the text in alternate formats, either via the format option, or via a set of sub-element specifiers:

.raw	The raw, unprocessed content of the field.
.esc	The raw field, with XML escaping applied.
.text	The text only, stripped of all HTML and BBCode markup.
.clean	The field with any HTML sanitised.

Date Fields

date	The date of the entry.
time	The time of the entry.
datetime	Both the date and the time of the entry.
shortdate	The short-form date of the entry.
day	The day of the entry.
month	The month of the entry.
year	The year of the entry.
create.*date*	The date the entry was created, as recorded by the system.
update.*date*	The date the entry was last updated, either by editing, or by a comment, ping, or other related item
start.*date*	The start date of the entry.
end.*date*	The end date of the entry.
comment.*date*	The date of the most recent comment.
ping.*date*	The date of the most recent ping.

The .date sub-element in a date-related tag can be replaced with any of the following:

.date	Show only the date.
.time	Show only the time.
.datetime	Show the date and the time.
.shortdate	Show the short form of the date.
.day	Show the day only.
.month	Show the month only.
.year	Show the year only.

Status Fields

status	The status of the entry, typically either *Publish* or *Draft*. See *entry statuses*.
type	The type of the entry. See *entry types*.
category	The name of the primary category.
comments	The number of comments.
pings	The number of pings.
edits	The number of times the entry has been edited.
reads	The number of times the entry has been read.
locked	Indicates that the entry has been locked, preventing all changes.
allow.comments	Indicates whether comments are allowed.
allow.pings	Indicates whether pings are allowed.

Template Tags

allow.attach	Indicates whether attachments are allowed.
sticky	The (numeric) stickiness of entry, in a range of -10 to +10. Standard values are 2 for announcements, 1 for sticky posts, 0 for normal posts, and -1 for unsticky posts.
hidden	Indicates whether the entry has been hidden.
defer	Indicates whether publication has been deferred.
expire	Indicates whether the entry has expired.
flag	Indicates that the entry has been flagged
ip	The IP address of the author at the time he or she created the entry. The last part of the IP address is obscured for privacy reasons.
ip.hash	A 6-character base-64 hash of the IP address of the author. Useful in screening out "sock puppets" and fake commenters.
filter	An optional text filter to be applied automatically to the entry.
number	The sequence of the entry within the overall selection range.
even	The number is even.
odd	The number is odd.
mod.[3|4|5]	The number is evenly divisible by 3, 4, or 5 respectively.
bytes	Length of the entry in bytes.
kbytes	Length of the entry in kilobytes.
words	Length of the entry in words.
first	Indicates that this is the first entry in the current selection (page).
last	Indicates that this is the last entry in the current selection (page).
isauthor	Indicates that the current user is the author of this entry.
edit	A link to the edit screen for this entry, if the user is allowed to edit it.
authorlink	The author's name as a link back to his or her web site.
name	The name specified for this post.
mail	The email address specified for this post.
web	The web address specified for this post.

Display Fields

With the exception of show.sig, these fields indicate whether certain types of markup are to be applied to a post. This is handled automatically by the system, but the information is also made available for your own use.

show.html	Is HTML markup allowed?
show.bbcode	Is BBCode markup to be processed into HTML?
show.smilies	Are smilies to be converted into embedded images?
show.macro	Are macro tags to be processed?
show.dict	Are dictionary entries to be processed?
show.sig	Is the user's signature to be displayed.

User Fields

These are a subset of the user object, representing useful public information about the author of the entry.

author.name	The author's name.
author.nick	The author's nickname.
author.mail	The author's public email address.
author.web	The author's web page.
author.aim	The author's AIM name.
author.msn	The author's MSN name.
author.yim	The author's YIM name.
author.icq	The author's ICQ name.
author.location	The author's (geographic) location.
author.occupation	The author's occupation.
author.interests	The author's interests.
author.sig	The author's signature.
author.hash	A (mostly unique) 6-digit base-64 has of the author's id.

ID Fields

You will rarely need to access ID fields directly; they are mostly used in edit and moderation screens. However, they are available should you wish to (for example) construct your own edit or moderation screens.

Template Tags

id	The id of this entry, which is the same as the thread id.
tid	The thread id of this entry.
pid	The post id of this entry.
fid	The id of the folder containing this entry.
etid	The effective thread id of this entry.
cfid	The id of the category containing this entry.

Category Fields

A subset of the fields for the Folder object, these represent useful public information about the category folder containing this entry.

category.id	The ID of the category.
category.name	The name of the category.
category.desc	The description of the category.
category.path	The path of the category.
category.nospace	The name of the category, with any spaces stripped out.
category.tagline	A tag line for the category, an alternate description or title.

Folder Fields

A subset of the fields for the Folder object, these represent useful public information about the folder containing this entry.

folder.id	The ID of the folder.
folder.name	The name of the folder.
folder.desc	The description of the folder.
folder.path	The path of the folder.
folder.nospace	The name of the folder, with any spaces stripped out.
folder.tagline	A tag line for the folder, an alternate description or title.

Site Fields

A subset of the fields for the Site object, these represent useful public information about the site containing this entry.

site.id	The ID of the site.
site.name	The name of the site.
site.desc	A description of the site.
site.short	A short name for the site.
site.host	The hostname (web address) of the site.
site.tagline	A tag line for the site.

Sub-Tags

Sub-Posts

These tags are used to lists the posts within an entry – comments, pings, and other similar items.

Options

[post:*items* {paged|unpaged} {page=*n*} {count=*n*} {skip=*n*} {show=all|junk} {type=comment|ping|page|...} {order=asc|desc} {search=*options*} {filter=*options*}]

[posts:*items*:here {template=*type.name*} {options}]

post:posts	Shows posts within the entry. (Typically there is just one post.)
post:subposts	Shows all subposts within the entry.
post:comments	Shows comments.
post:pings	Shows pings.
post:chat	Shows chat.
post:discuss	Shows discussions.
post:footnotes	Shows footnotes.
post:notes	Shows notes. (This is different to the BBCode notes tag.)
post:items	Shows items.
post:pages	Shows pages. Pages can be used both as an entry type, or to break up a length post into smaller sections.
post:remarks	Shows remarks.
post:replies	Shows replies.
post:responses	Shows responses.

Template Tags

post:talk	Shows talk.
post:trackbacks	Shows trackbacks.

Attachments

These tags list attachments – files uploaed the the server and inked to the entry. You can have any number of attachments of any type linked to a given entry.

Options

[post:*attachments* {paged|unpaged} {page=*n*} {count=*n*} {skip=*n*} {type=image|audio|movie|...} {sort=date|file|name|size|type} {order=asc|desc} {search=*options*} {filter=*options*}]

[posts:*attachments*:here {template=*type.name*} {options}]

post:attachments	List all attachments.
post:apps	List applications.
post:audio	List audio files.
post:files	List other files.
post:flash	List Flash files.
post:images	List images.
post:media	List media files (audio and movies).
post:movies	List movies.
post:music	List music (audio).
post:programs	List programs.
post:songs	List songs (audio).
post:text	List text files.
post:zips	List zip files.

Other Details

post:links	Lists links attach to the entry.
post:related	Lists other related entries.
post:references	Lists references.
post:stats	Displays statistics for the entry.
post:categories	Lists categories the entry appears in.
post:folders	Lists folders the entry appears in.

post:sites	Lists sites the entry appears on. (Note: This only works if the entry is cross-posted. It doesn't reflect ad-hoc requests using site tag options or smart folders.)
post:next	The next entry in the sequence.
post:prev	The previous entry in the sequence.
post:first	The first entry in the sequence.
post:last	The last entry in the sequence.

Template Tags

Statistics:Summary

Quickly Site statistics summaries.

Usage [statistics:summary {count=n} {mode=sum|avg|min|max} {order=asc|desc}]
[statistics:summary:here {template=type.name}]

Type block tag, here tag

Data Fields site, year, month, pages, bytes, bytes.text, pages.bytes, pages.bytes.text, files.bytes, files.bytes.text, feeds, templates, queries, rows, ctime, ctime.text, etime, etime.text, qtime, qtime.text

Description

The [statistics:summary] tag retrieves a set of monthly summary statistics for your site. The values available are:

site	The id of your site.
year	The year for this summary record.
month	The month for this summary record.
pages	The number of pages (html, text, xml) retrieved.
files	The number of files retrieved.
feeds	The number of RSS/Atom feed requests.
templates	The number of templates processed.
queries	The number of SQL queries executed.
rows	The number of database rows retrieved.
ctime	The number of CPU seconds used.
etime	The number of seconds elapsed during processing.
qtime	The number of seconds elapsed during queries.
bytes	The total number of bytes transferred.
pages.bytes	The number of bytes transferred for pages.
files.bytes	The number of bytes transferred for files.

Pre-formatted versions are available for each of the time and bytes fields by appending .text. The pre-formatting takes account of minutes and seconds for times, and of kilobytes, megabytes, and gigabytes for transfers.

Options

count=*n*

Maximum number of records return.

mode=sum|avg|min|max

Return an aggregate figure, rather than individual records. The values are the total (sum), average (avg), minimum (min) or maximum (max) of the respective fields. In aggregate mode, only one record is returned.

order=asc|desc

Sort records in ascending or descending order. The default is descending.

template

Template to be used in here mode. The default is item.stats.summary

Template Tags

Tag Formatting

You can add formatting to any data tag using the format= option. The main formatting commands are described below.

Date and Time

Dates and times use a format string as described below. You can use any combination of formatting options and text in the format string. If the format string contains spaces, you will need to enclose it in either single or double quotes.

Date Formatting Options

Code	Meaning	Examples
%a	Abbreviated weekday name.	Tue, Thu
%A	Full weekday name.	Tuesday, Thursday
%b	Abbreviated month name	Jan, Aug
%B	Full month name.	January, August
%c	Date and time.	Tue Apr 3 14:53:38 2007
%d	Day of month.	3
%j	Day of year.	093
%m	Month.	1, 8
%U	Week of year (weeks starting Sunday).	13
%W	Week of year (weeks starting Monday).	13
%x	Date (mm/dd/yy).	04/03/07
%y	Year.	07
%Y	Year with century.	2007

Time Formatting Options

Code	Meaning	Examples
%H	Hour (24-hour clock).	11, 23
%I	Hour (12-hour clock).	11, 11
%M	Minute.	03, 29
%p	AM or PM.	AM, PM
%S	Second.	15, 59
%X	Time.	14:53:38

Examples

Code	Result
%A, %B %d	Monday, April 09
%I:%M %p	08:57 PM
%A, %B %d %Y %I:%M %p	Monday, April 09 2007 08:57 PM
%Y-%m-%d	2007-04-09

Text

Multiple text formatting options can be applied by separating the commands with commas. The formatting commands are applied in the order they are listed. Example: [post.text format="first=100,underscore,lower"].

Option	Description
left=n	Returns the first n characters of the text
right=n	Returns the last n characters of the text
first=n	Returns the first n words of the text
last=n	Returns the last n words of the text
trim	Removes any leading and trailing spaces
trim=s	Removes any leading or trailing instances of string s
lower	Converts text to lower case
upper	Converts text to upper case
words	Returns the number of words in the text
letters	Returns the number of letters in the text
nospace	Removes any spaces from the text
underscore	Convert any spaces in the text to underscores
sanitise	Full HTML sanitiser
strip_html	Strip HTML markup
strip_markup	Strip HTML and BBCode markup
escape_html	Escape HTML markup
escape_xml	Escape XML markup
convert_breaks	Convert raw line breaks to HTML
bbcode	Convert BBCode markup to HTML
smilies	Convert smilies to embedded images
macros	Apply macro processing

Template Tags

Numbers

Number formatting works by placing the formatted value into a string containing both formatting options and text.

Option	Description
%d	Signed integer decimal
%i	Signed integer decimal
%o	Unsigned octal
%u	Unsigned decimal
%x	Unsigned hexadecimal (lower case)
%X	Unsigned hexadecimal (upper case)
%e	Floating point exponential (lower case)
%E	Floating point exponential (lower case)
%f	Floating point decimal format
%F	Floating point decimal format
%g	Variable floating point format
%G	Variable floating point format
%s	Default representation
%%	% symbol

As an example, you could have [post.words format="There are %d words in this post."]

Each of the numeric formatting options can take a number of flags:

Flag	Description	Example
0	Zero fill the result.	%0d
-	Left-adjust the result	%-d
	(Space) Leave a space before a positive result.	% d
+	Always show a leading + or -.	%+d

Template Programming

Conditional Tags

if, ifn, iff, and iffn.

[if] tests a single variable. If that variable exists and has a true (non-zero / non-null / non-empty) value, the template code within the bracket of the [if] and the [/if] is evaluated.

If the variable is false (zero / null / empty), the code is not evaluated.

And if the variable does not exist, the [if] tag itself is not evaluated, and is instead included directly in the output. So if you get the name of the variable wrong, and type [if post.coments] instead of [if post.comments], you will be able to see your mistake right there in the page. In this case, the bracketed code is processed as if the [if] tag did not exist.

[ifn] works exactly the same, except that the condition is reversed. Code inside the [ifn] [/ifn] is evaluated if the variable is false.

With [if] and [ifn], if the variable does not exist the behaviour of a following [else] statement is undefined.

[iff] and [iffn] are similar, with one big exception: The bracketed code is evaluated if and only if the variable evaluates to true (or false, for [iffn]). If the variable does not exist, the code is skipped, just as if the variable had been false.

Comparison Tags

if.eq, if.ne, if.gt, if.lt, if.ge, if.le

These extended conditional tags compare two values, to see if, respectively, they are equal or unequal, or that the first value is greater than, less than, greater than or equal, or less than or equal to the second value. They must be closed by a corresponding [/if.XX] tag such as [/if.eq], not by a [/if].

The values can be either variables or constants. The way Minx determines which is the case is somewhat simplistic: If a variable exists which has the name of the value, it assumes that's what you mean; if not, then it assumes the value is a constant. So unlike the simple [if] tag, there's never a case where the variable doesn't exist.

These tags will perform integer comparisons if both values are integers; otherwise they will perform string comparisons. So [if.gt 10 2] will evaluate as true, just as you'd expect, and [if.gt a10 a2] will evaluate as false.

These tags also work if supplied with a single variable. The missing variable is assumed to be zero, if the provided variable is a number; False, if the value is True, blank, if the value is a string, and None otherwise.

Extended Comparison and Conditional Tags

if.z, if.nz, if.odd, if.even

These codes take a single variable.

[if.z] and [if.nz] test whether the value is zero or non-zero. This is effectively identical to using [if.eq] and [if.ne] with a single numeric variable, but may be clearer to the user.

[if.odd] and [if.even] test whether the value is odd or even. If the value is not numeric, the result is undefined.

[if.all], [if.any], [if.one], [if.none]

These codes currently take two variables, but may be extended in the future to take an arbitrary number of values.

[if.all] tests whether both the variables are true (non zero / non-empty) - effectively an and.
[if.any] tests whether either or both of the variables is true - an or.
[if.one] tests if one and only one of the variables is true.
[if.none] tests if neither of the variables is true - a nor.

For all comparison tags and extended conditionals, there is also a matching [ifn.condition], which simply reverses the result.

Else

Code within [else]....[/else] is evaluated if the last tried conditional block within the current template was not evaluated. In other words, it works the way you would expect, with one addition: You can have multiple [else] blocks after one [if] block, and *all* the [else] blocks will be evaluated if the [if] fails.

Variables and Calculations

set

You can set a variable with the [set] tag. Like the extended conditionals, [set] can take either

another variable or a constant, so [set a post.title] or [set a "Wombats in Paradise"] are valid examples. Again like the extended conditionals, [set] assumes a variable if the variable name exists, and a constant otherwise.

The variable can be accessed again with the prefix *var.* So [var.a] will print the value of *a*, and [if var.a] will test it. You can use the var.a terminology inside set as well, and it will be interpreted appropriately.

Variables can be used anywhere a normal data tag could be used.

calc.add, calc.sub, calc.mul, calc.div, calc.mod, calc.min, calc.max

The [calc] tag performs a calculation based on two values, and assigns the result to a variable. The values must be integer variables or constants; anything else will be evaluated as zero.

So [calc.add x var.y var.z] will set the variable x to be the sum of y and z.

inc, dec

Just want to increment or decrement a counter? These are the tags for you! [inc a] or [dec a] and you're done.

If the variable did not exist beforehand, it is created as zero, and then incremented or decremented.

Note that none of these tags can alter the value of an existing data tag, only user variables.