

Calls

```
<!--[ function ]-->
<!--[ function attribute="parameter" attribute=$variable ]-->
<!--[ $variable ]-->
<!--[ $variable|modifier ]-->
<!--[ $variable|modifier:parameter ]-->
<!--[ $variable|phpFunction:parameter ]-->
```

Assign

```
<!--[ assign var='name' value='Bob' ]-->
```

Backticks

```
<!--[ assign var=var3 value=`$var1+$var2` ]-->
```

Loops

```
<!--[ section name=id loop=$variable ]-->
    <!--[ $variable[id] ]-->
<!--[ sectionelse ]-->
    $variable empty!
<!--[ /section ]-->

<!--[ foreach item=currentItem from=$items ]-->
    <!--[ $currentItem ]-->
<!--[ foreachelse ]-->
    $item empty!
<!--[ /foreach ]-->
```

math

```
<!--[math equation="x+y" x=$height y=$width]-->
or
<!--[math equation="x+y" x=$height y=$width assign="var"]-->
<!--[ $var ]-->
or with parenthesis
<!--[ math equation="(( x + y ) / z)" x=2 y=10 z=2 ]-->
```

pnRender Modifier

```
<!--[ $var|activatelinks ]-->
Plugin to turn URLs within a string into links
<!--[$var|n2paragraphs]-->
each newline (\n) is converted to a paragraph pair </p><p>
<!--[$var|pnm]-->
converts lanugage define into language string
<!--[$var|pnmodcallhooks modname="name"]-->
run the transform hooks
<!--[$var|pnvarcensor]-->
remove censored words

<!--[ $var|pndate_format:"dateSpecifiers" ]-->
date_format
<!--[ $var|pndate_format:"":"dateString" ]-->
Defines: datebrief/datelong/datestring ...

<!--[$uname|userprofilelink]-->
create a link to a users profile
<!--[$uname|userprofilelink:classname]-->
Simple version, shows $username, using
class="classname"
<!--[$uname|userprofilelink:"img/profile.gif"]-->
Using profile.gif instead of username, no class

<!--[$var|pnvarprepfordisplay]-->
prepare a variable for display
<!--[$var|pnvarprepfstore]-->
prepare variable for database storage
<!--[$var|pnvarprephtmldisplay]-->
prepare for display, preserving some HTML tags
<!--[$var|htmlspecialchars]-->
converting special characters to HTML entities

<!--[$var|yesno|pnvarprepfordisplay]-->
returns Yes if $myvar = 1 and No if $myvar = 0
<!--[$var|onlineoffline]-->
returns Online if $var = 1 and Offline if $var = 0
<!--[$var|activeinactive]-->
returns Active if $var = 1 and Inactive if $var = 0

<!--[$var|xslt:"your_xsl_file.xml"]-->
apply a xml stylesheet to a variable
```

Standard Modifiers

```
capitalize
cat
count_characters
count_paragraphs
count_sentences
count_words
date_format
default
escape
indent
lower
nl2br
regex_replace
replace
specify
string_format
strip
strip_tags
truncate
upper
wordwrap

All php-functions can be used
as modifiers implicitly
```

Locations

```
/themes/YourTheme/templates/
modules/moduleName/
modified module templates
/themes/YourTheme/templates/
modules/moduleName/style/
modified/extra stylesheets
/themes/YourTheme/templates/
modules/moduleName/plugins/
modified/extra plugins
/themes/YourTheme/templates/
modules/moduleName/images/
modified/extra images
```

Date Specifiers

Standard

```
%a abbr. weekday name
%A full weekday name
%b abbr. month name
%B full month name
%d day of month (range 01 to 31)
%D same as %m/%d/%y
%e day of month (range 1 to 31)
%H hour (range 00 to 23)
%l hour (range 01 to 12)
%j day of year (range 001 to 366)
%k Hour (range 0 to 23)
%l hour (range 1 to 12)
%m month (range 01 to 12)
%M minute as a decimal number
%n newline character
%p either "am" or "pm"
%r time in am. and pm notation
%R time in 24 hour notation
%S second as a decimal number
%t tab character
%T equal to %H:%M:%S
%u weekday as a decimal number
%V ISO 8601:1988 week number
%X preferred time for current locale
%y year (range 00 to 99)
%Y year including century
%Z time zone or name or abbr.
% % a literal "%" character
```

Defines

```
datebrief
%b %d, %Y
datelong
%A, %B %d, %Y
datestring
%A, %B %d @ %H:%M:%S
datetimebrief
%b %d, %Y - %l:%M %p
datetimelong
%A, %B %d, %Y - %l:%M %p
dateinput
%Y-%m-%d
datetimeinput
%Y-%m-%d %H:%M
timebrief
%l:%M %p
```

Modifier Examples

```
<!--[ $title|default:'no title' ]-->
set default value incase variable is empty
<!--[ $title|upper ]-->
apply modifier to a variable
<!--[ $title|truncate:40:'...' ]-->
modifier with parameters
<!--[ html_table loop=$myvar|upper ]-->
apply modifier to a function parameter
<!--[ html_table loop=$myvar|truncate:40:'...' ]-->
with parameters
<!--[ 'foobar'|upper ]-->
apply modifier to literal string
<!--[ $smarty.now|date_format:"%Y/%m/%d" ]-->
using date_format to format the current date
<!--[ mailto|upper address='smarty@example.com' ]-->
apply modifier to a custom function
<!--[ '='|str_repeat:80 ]-->
using php's str_repeat
<!--[ $myArray|@count ]-->
php's count
<!--[ $smarty.server.SERVER_ADDR|shuffle ]-->
php's shuffle on servers's ip
<select name="name_id">
<!--[ html_options output=$myArray|upper|truncate:20 ]-->
</select>
this will uppercase and truncate the whole array
```

pnRender Functions

```
<!--[ pndebug ]-->
Popup with all available variables
<!--[ pnvarcleanfrominput name="param" assign="var" ]-->
fetch the value of an URL parameter
<!--[ pnusergetvar name="user_icq" uid=1 ]-->
fetch user data (no uid = present user)
<!--[ pnusergetidfromname name="name" assign="uid" ]-->
return the user ID for a given username
<!--[ pnimg src="image.png" ]-->
returns an img-tag
src: file name of the image
modname: module name (default: the current module)
width, height: If none is set, they are obtained from the image
alt: If not set, an empty string is being assigned
altml: If true, alt string is assumed to be a ML constant
title: If not set, an empty string is being assigned
titleml: If true, title string is assumed to be a ML constant
optional: If set, the plugin will return np error if image isn't found
default: If set, a default image is used should the requested image
not be found (Note: full path required)
set: If modname is 'core' then the set parameter is set to define the
directory in /images/
nostoponerror: If set and error occurs, do not trigger_error, but
return false and fill pnimg_error instead
assign: If set, the results are assigned to a variable
All remaining parameters are passed to the module function
<!--[ pnml name="_EXAMPLESTRING"]-->
read a language constant
Available parameters:
name: Name of the language constant to return
html: Treat the language define as HTML
assign: If set, the results are assigned to a variable
noprocess: If set, no processing is applied to the constant value
All remaining parameters are passed to the module function
<!--[ pnmodurl modname="Name" type="user" func="display" ]-->
create a URL for a specific module function
Available parameters:
modname: modulename (required)
type: currently one of 'user' or 'admin' (default is 'user')
func: module function (default is 'main')
fragment: The fragment to target within the URL
ssl: set to constant null, true, false
append: a string to be appended to the URL
assign: If set, the results are assigned to a variable
All remaining parameters are passed to the module function
<!--[ pnmodfunc modname="Name" type="user" func="main"]-->
execute a module function (pnuser.php, pnadmin.php ...)
Available parameters:
modname: modulename (required)
type: currently one of 'user' or 'admin' (default is 'user')
func: module function (default is 'main')
assign: If set, the results are assigned to a variable
All remaining parameters are passed to the module function
<!--[ pnmodapifunc modname="Name" type="user" func="main"]-->
execute a module API function (pnuserapi.php, pnadminapi.php ...)
Available parameters:
modname: modulename (required)
type: currently one of 'user' or 'admin' (default is 'user')
func: module function (default is 'main')
assign: If set, the results are assigned to a variable
All remaining parameters are passed to the module function
```