patTemplate Overview

Last Updated Thursday, 30 December 2004

Tags

There are five xml tags available in patTemplate that form the building blocks for the templating engine:

- The tmpl tag
- The sub tag
- The link tag
- The var tag
- The comment tag

Tag Namespaces

All patTemplate tags must include a namespace which is used to distinguish the patTemplate XML markup tags from regular HTML, xHTML and other XML tags. The default namespace is patTemplate but you my change this as desired by calling the setNamespace method.

```
<?php
$tmpl =& new patTemplate();
// change the namespace to 'cms'
$tmpl->setNamespace( 'cms' );
?>
```

This is a test of my content management system.

Template Types

Templates are parts of html pages, which can be displayed by the template engine. To identify a certain part of the page as a template, patTemplate uses XML tags to assign a template a unique name and a various attributes (see patTemplate Tags and attributes for a list of all tags and attributes). When parsing a template, the parser divides the page in several chunks and treats them as separate templates. By using patTemplate's API you can hide, display or repeat a certain template.

A template can contain as many other templates as you like.

Templates can be of these types (defined by using the type="..." attribute):

- Standard â€“ output of the template will always be the same

- OddEven â€“ you can assign to subtemplates () to this type: Odd and Even, which will be alternated, when the template is repeated

- Modulo â€“ is like the Odd Even template except that you may have more than two subtemplates. The Odd Even template type is simply a special case of this type where the modulo attribute equals 2.

- Condition â€“ you can assign as many subtemplate as you like and you have to assign one variable which will be used to compare it with the specified conditions of the subtemplates. This allows you to define different HTML code for each value your condition variable may have. There are two special conditions you can use: default, which is similar to the default in a switch/case statement and empty which will be displayed, when there is no value assigned to the condition variable.

- SimpleCondition â€“ A situation that often arises is that an error message has to be displayed if an error occured. Normally you would set the error message if it exists or hide the template using setAttribute() if there is none. By using a template of type SimpleCondition you can assign a list of variables (using the requiredvars) that have to be set if the template has to be displayed. If on of these variables is not set, the template will not be visible.

Template Variables

patTemplate supports variables similar to programming languages. There can be local variables (assigned by the addVar method) and global variables (assigned by the addGlobalVar method).

Global variables can only be scalar variables (strings, integers,...) and local variables can also be arrays. If you assign an array to a variable, the template where you assigned the variable will be repeated.

If a condition template uses the useglobals="yes" attribute global variables will be used if no local variables are set.

Variables have to be written in uppercase and may only contain chars, numbers or the underscore (_). Variables are always enclosed in curly brackets, for example, {FOO}.

Variables will be replaced with their values when parsing a template.

There are several predefined system variables that are available when you set the addsystemvars attribute.

- PAT_ROW_VAR â€“ you can use this when you want to enumerate a template that is automatically repeated. Just place the variable PAT_ROW_VAR in your template, where you'd like to enumeration to appear (can also be used for javascript mouseovers).

- PAT_LOOPS â€“

- PAT_IS_FIRST â€“

- PAT_IS_LAST â€“

- PAT_IS_ODD â€“

- PAT_IS_EVEN â€“

- PATTEMPLATE_VERSION â€“

- PAT_MODULO_REP â€“

- PAT_MODULO -

Tag Attributes

Templates can have different attributes, the can be set using setAttribute or by placing the attributes in the Tags.

See patTemplate Tags and attributes for details.
Examples

The downloadable package of patTemplate contains several examples (currently only in German), which explain the basic functions.