Custom Modifiers

Custom Modifiers are subclasses of patTemplate_Modifier and have to be placed inside patTemplate/Modifier. The filename has to match the last part of the classname. That means if you would like to create a filter that truncates sentences you will have to create a new class 'patTemplate_Modifier_Truncate' and place it into patTemplate/Modifier/Truncate.php.

The class has to implement one method modify(string value, array params) that has to return the modified value. You may do whatever you like inside this method to modify the value.

patTemplate will pass two parameters to your method:

Â Â Â * string $value, the value of the variable
Â Â Â * array $params, an associative array containing all attributes of the var tag except the ones, that are used internally (name, default, copyFrom, modifier)

The truncate modifier could look like this:

```
01Â Â Â Â Â Â Â <?PHP
02Â Â Â Â Â Â Â /**
03Â Â Â Â Â Â Â * patTemplate modfifier Truncate
04Â Â Â Â Â Â Â *
05Â Â Â Â Â Â Â * $Id: modifiers.xml,v 1.1 2004/05/11 19:46:09 schst Exp $
06Â Â Â Â Â Â Â *
07Â Â Â Â Â Â Â * @packageÂ Â Â Â Â Â Â patTemplate
08Â Â Â Â Â Â Â * @subpackageÂ Â Â Â Modifiers
09Â Â Â Â Â Â Â * @authorÂ Â Â Â Â Â Â Stephan Schmidt <schst@php.net>
10Â Â Â Â Â Â Â */
11Â Â Â Â Â Â Â class patTemplate_Modifier_Truncate extends patTemplate_Modifier
12Â Â Â Â Â Â Â {
13Â Â Â Â Â Â Â Â Â Â Â /**
14Â Â Â Â Â Â Â Â Â Â Â * truncate the string
15Â Â Â Â Â Â Â Â Â Â Â *
16Â Â Â Â Â Â Â Â Â Â Â * @accessÂ Â Â public
17Â Â Â Â Â Â Â Â Â Â Â * @paramÂ Â Â stringÂ Â Â Â Â Â Â value
18Â Â Â Â Â Â Â Â Â Â Â * @returnÂ Â Â stringÂ Â Â Â Â Â modified value
19Â Â Â Â Â Â Â Â Â Â Â */
20Â Â Â Â Â Â Â Â Â Â Â function modify( $value, $params = array() )
21Â Â Â Â Â Â Â Â Â Â Â {
22Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â /**
23Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â * no length specified
24Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â */
25Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â if( !isset( $params['length'] ) )
26Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â return $value;
27Â Â Â Â Â Â Â Â Â Â Â
28Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â /**
29Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â * is shorter than the length
30Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â */
31Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â if( $params['length'] >strlen( $value ) )
32Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â return $value;
33Â Â Â Â Â Â Â Â Â Â Â
34Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â return substr( $value, 0, $params['length'] ) . '...';
35Â Â Â Â Â Â Â Â Â Â Â }
36Â Â Â Â Â Â Â }
37Â Â Â Â Â Â Â ?>
```

If you copy this file to the Modifier folder, you may instantly use it in your template files, by specifying it in the modifier attribute of any <patTemplate:var/> tag.

```
01Â Â Â <patTemplate:tmpl name="page">
02Â Â Â Â Â Â Â Â <div>
```

```
03Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â  <patTemplate:var name="myVar" modifier="Truncate" length="50"/>
04Â Â Â Â Â Â Â Â Â Â  </div>
05Â Â Â  </patTemplate:tmpl>
```

If you now pass a value that is longer than 50 chars, the modifier will automatically truncate it, before inserting it into the template.