

Ansichten anpassen mittels Templates

1. Die Template-Sprache

Im Ordner /tpl/MyCMS/default/ liegen die Templatedateien für die Standardansichten. Dies sind namentlich die Gruppen-Liste (groups.tpl), Artikel-Liste (articles.tpl), Detail-Ansicht (detail.tpl) und die Home-Ansicht (home.tpl). Des weiteren befinden sich in diesem Ordner die Datei "navigation.tpl", welche die Blätterfunktion der Listen-Ansichten darstellt und "search-form.tpl", welche das Formular für die Suche darstellt.

In diesen Dateien haben Sie die Möglichkeit, oben genannte Ansichten nach Ihren Wünschen anzupassen. Die Darstellung geschieht grundsätzlich mit HTML. Zusätzlich gibt es eine Templatesprache. Diese ist schlicht gehalten und beinhaltet wenige Elemente. Hier eine kurze Erklärung:

Syntax	Erklärung/Anwendung
{ \$Platzhalter-Name }	<p>Mit diesem Code implementieren Sie einen Platzhalter. "Platzhalter-Name" muss natürlich durch einen bestehenden Namen wie z.B. "TITLE" ersetzt werden. In diesem Fall wird dann an der Stelle, an der { \$TITLE } steht, der Titel eines Artikels ausgegeben. Mit dem Platzhalter { \$CONTENT } wird der Inhalt/Text eines Artikels ausgegeben. { \$CREATEDATE } steht für das Erstelldatum. Eine Liste aller verfügbaren Platzhalter finden Sie im Zusatzdokument. Nun können Sie diese Platzhalter in den HTML-Code einfügen. Hier ein Beispiel:</p> <pre> <table> <tr> <td> { \$TITLE } </td> </tr> <tr> <td> { \$CONTENT } </td> </tr> <tr> <td> { \$CREATEDATE } </td> </tr> </table> </pre> <p>Der HTML-Code generiert eine Tabelle mit drei Zeilen und einer Spalte. In der ersten Zeile steht der Titel, in der zweiten der Inhalt eines Artikels und in der dritten das Erstelldatum des Artikels..</p>
{ \$# ({ \$BEDINGUNG }) # } ... { \$# End # }	<p>Mit dieser Syntax erstellen Sie eine Bedingung. Wenn die Bedingung wahr also TRUE ist wird der Code dazwischen dargestellt. Wenn sie nicht wahr also FALSE ist wird der Code zwischen diesen beiden Zeilen ignoriert. Auch hier gibt es eine</p>

Liste aller Bedingungen für die Standard-Ansichten im Zusatzdokument.

Nehmen wir als Beispiel die Bedingung `$SHOWDATE`. `$SHOWDATE` ist so definiert: Wenn das Datum für ein Artikel angezeigt werden soll, ist diese Bedingung wahr (TRUE). Soll kein Datum unter diesem Artikel angezeigt werden ist sie nicht wahr (FALSE).

Nun ein Beispiel dazu:

```
{ $# ( { $SHOWDATE } ) # }  
  
Erstellt am: { $CREATEDATE }  
  
{ $# End # }
```

Ausschlaggebend ob die Bedingung `{ $# ({ $SHOWDATE }) # }` TRUE oder FALSE ergibt ist die Einstellung des jeweiligen Artikel in DynPG. Ist das Häkchen unter „Ja, Datum in der Detail- und Listenansicht anzeigen“ gesetzt, ist diese Bedingung wahr. Somit wird der Code bis zu `{ $# End # }` interpretiert. Da ja `{ $CREATEDATE }` der Platzhalter für das Erstellungsdatum ist, wird in diesem Falle ausgegeben: "Erstellt am 09.08.2006". Ist das Häkchen nicht gesetzt so ist diese Bedingung FALSE und der Code wird übersprungen.

Man kann nun die Sache auch umdrehen. Dies geschieht mit einem vorangestellten Ausrufezeichen (!). Dann wird der Code interpretiert, wenn die Bedingung nicht wahr also FALSE ist.

Ein Beispiel dazu:

```
<table>  
  <tr>  
    <td> { $TITLE } </td>  
  </tr>  
  { $# ( { $SHOWDATE } ) # }  
  <tr>  
    <td> Erstellt am: { $CREATEDATE } </td>  
  </tr>  
  { $# End # }  
  
  { $# ( ! { $SHOWDATE } ) # }  
  <tr>  
    <td> Anzeige ohne Erstellungsdatum </td>  
  </tr>  
  { $# End # }  
  <tr>  
    <td> { $CONTENT } </td>  
  </tr>  
</table>
```

Der HTML-Code erstellt eine Tabelle mit einer Spalte. In der ersten Zeile dieser Tabelle wird der Titel angezeigt.

	<p>Nun prüft die erste Bedingung, ob das Datum angezeigt werden soll. Im Falle von TRUE gibt er das Datum in einer neuen Zeile aus. Bei FALSE überspringt er den Code zwischen der Bedingung und dem Ende der Bedingung.</p> <p>Danach prüft er in der zweiten Bedingung, ob das Datum nicht angezeigt werden soll. Dies geschieht durch das Ausrufezeichen. Dadurch ergibt die Bedingung TRUE, wenn das Häkchen in DynPG nicht gesetzt ist, da ja die Bedingung „Datum nicht anzeigen“ wahr ist. In diesem Falle erstellt er eine neue Zeile in der steht: „Anzeige ohne Erstelldatum“.</p> <p>Logischerweise kann und muss in diesem Beispiel nur eine der beiden Bedingungen TRUE sein. Zum Schluss wird eine dritte Zeile erstellt, die den Inhalt des Artikel/Textes ausgibt, unabhängig von den Bedingungen.</p> <p>Natürlich kann auch eine Bedingung innerhalb einer Bedingung stehen.</p> <p>Prinzipiell dürfen Sie innerhalb des Bereiches BEDINGUNG beliebige Operatoren verwenden, solange das Resultat ein boolescher Wert bleibt.</p>
<p>{<code>PHP (PHP-CODE)</code> #}</p>	<p>Mit dieser Syntax können Sie PHP-Code implementieren. Alles was in dieser Klammer steht, wird als PHP-Code interpretiert. Es kommt dem Tag „<?php ... ?>“ einer PHP-Seite gleich.</p> <div data-bbox="560 969 1374 1016" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>{<code>PHP (substr('Teilwort',4);)</code> #}</pre> </div> <p>Dieser Code ermöglicht das Einbinden von PHP-Code. In diesem simplen Beispiel wird das Wort „Teilwort“ nach 4 Zeichen abgetrennt und das Wort „Teil“ bleibt übrig.</p>

2. Liste der Platzhalter in den Templates

2.1 Template groups.tpl

Dieses Template ist zuständig für die Darstellung von Ansichten der Gruppen-Listen.

Platzhalter	Bedeutung/Ausgabe
{ \$ROOT_NAVIGATION }	Erstellt eine Übersichts-Navigation. Gruppe >> Untergruppe
{ \$PAGES_NAVIGATION }	Erstellt eine Navigation zum Blättern bei mehreren Seiten. Also falls mehr Gruppen angezeigt werden sollten, als in der config.php unter config_AllGroups_Limit eingestellt sind.
{ \$LIST_REFERER_TO }	Enthält die URL-Adresse zur nächsttieferen Anzeige-Stufe (im Falle groups.tpl den Link zur Artikelliste oder zur Untergruppe).
{ \$NAME }	Gibt den Namen der Gruppe aus.
{ \$SHORTTEXT }	Gibt den Kurztext der Gruppe aus.
{ \$CREATEDATE }	Gibt das Erstelldatum der Gruppe aus.
{ \$LIST_SHOW_VISTS }	Gibt die Anzahl Klicks auf den Link der entsprechenden Gruppe aus.
{ \$LIST_HAS_IMAGE }	Gibt zurück, ob die Gruppe ein Leadbild hat. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus DynPG.
{ \$LIST_IMAGE_ALIGN }	Gibt die gewünschte Position des Leadbildes zurück. TOP für oben, BOTTOM für unten, LEFT für links und RIGHT für rechts. Entnimmt den Wert aus DynPG.
{ \$IMAGE_URL }	Gibt das Quellverzeichnis des Leadbildes aus.
{ \$IMAGE_NAME }	Gibt den Namen des Leadbildes aus.
{ \$IMAGE_DESCR }	Gibt die Beschreibung des Leadbildes aus.

2.2 Template articles.tpl

Dieses Template ist zuständig für die Darstellung von Ansichten der Artikel-Listen.

Platzhalter	Bedeutung/Ausgabe
{ \$ROOT_NAVIGATION }	Erstellt eine Übersichts-Navigation. Gruppe >> Artikel
{ \$PAGES_NAVIGATION }	Erstellt eine Navigation zum Blättern bei mehreren Seiten. Also falls Sie mehr Gruppen angezeigt werden sollten, als in der config.php unter 'config_AllArticles_Limit' eingestellt sind.
{ \$LIST_USER_DEFINED_CSS }	Gibt den Wert zurück, den Sie beim Implementieren unter <code>SetParam_AA_SingleCss(...)</code> ; angegeben wurde.
{ \$LIST_INDEX }	Gibt die aktuelle Anzahl (den Index) der angezeigten Artikel zurück (also beim 3. Artikel den Wert 3, beim 6. Artikel den Wert 6).
{ \$LIST_REFERER_TO }	Enthält die URL-Adresse zur nächsttieferen Anzeige-Stufe (im Falle articles.tpl den Link zur Detail-Ansicht).
{ \$CONTENT }	Gibt die ID des zum Artikel gehörigen Textes aus.
{ \$TITLE }	Gibt den Namen/Titel des Artikels aus.
{ \$LEAD }	Gibt den Lead-Text des Artikels aus.
{ \$SHOWDATE }	Gibt zurück, ob das Erstellungsdatum angezeigt werden soll. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus DynPG.
{ \$CREATEDATE }	Gibt das Erstellungsdatum des Artikels aus
{ \$LIST_SHOW_VISTS }	Gibt die Anzahl Klicks auf den Link des entsprechenden Artikels aus.
{ \$LIST_HAS_IMAGE }	Gibt zurück, ob der Artikel ein Leadbild hat. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus DynPG.
{ \$LIST_IMAGE_ALIGN }	Gibt die gewünschte Position des Leadbildes zurück. TOP für oben, BOTTOM für unten, LEFT für links und RIGHT für rechts. Entnimmt den Wert aus DynPG.
{ \$IMAGE_URL }	Gibt das Quellverzeichnis des Leadbildes aus.
{ \$IMAGE_NAME }	Gibt den Namen des Leadbildes aus.
{ \$IMAGE_DESCR }	Gibt die Beschreibung des Leadbildes aus.

2.3 Template detail.tpl

Dieses Template ist zuständig für die Darstellung der Detailansicht eines Artikels.

Platzhalter	Bedeutung/Ausgabe
{ \$COND_SEGM_CATNAV }	Gibt zurück, ob eine Übersichts-Navigation angezeigt werden soll. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus der Implementation: <code>SetParam_A_CategoryNavigation(TRUE/FALSE);</code>
{ \$ROOT_NAVIGATION }	Erstellt eine Übersichts-Navigation . Gruppe >> Artikel >> Titel
{ \$LIST_USER_DEFINED_CSS }	Gibt den Wert zurück, den Sie beim Implementieren unter <code>SetParam_A_ArticleCss(...)</code> ; angegeben wurde.
{ \$COND_SEGM_HEAD }	Gibt zurück, ob der Head-Bereich (im Normalfall Headtext und Beschreibung) angezeigt werden soll. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus der Implementation: <code>SetParam_A_ArticleHeader(TRUE/FALSE);</code>
{ \$TITLE }	Gibt den Namen/Titel des Artikels aus.
{ \$LEAD }	Gibt den Lead-Text des Artikels aus.
{ \$SHOWDATE }	Gibt zurück, ob das Erstelldatum angezeigt werden soll. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus DynPG.
{ \$CREATEDATE }	Gibt das Erstelldatum des Artikels aus
{ \$COND_SEGM_TEXT }	Gibt zurück, ob der Artikelinhalt/-text angezeigt werden soll. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus der Implementation: <code>SetParam_A_ShowText(TRUE/FALSE);</code>
{ \$CONTENT }	Gibt den zugehörigen Text des Artikels aus
{ \$COND_SEGM_PRINTNAV }	Gibt zurück, ob ein Link um den Artikel zu drucken angezeigt werden soll. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus der Implementation: <code>SetParam_A_PrintNavigation(TRUE/FALSE);</code>
{ \$PRINT_HREF }	Gibt den Link zur Druckansicht des Artikels aus.
{ \$COND_SEGM_SEPARATOR }	Gibt zurück, ob der Seperator im Text ignoriert werden soll. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus der Implementation: <code>SetParam_A_IgnoreSeperator(TRUE/FALSE);</code>
{ \$MORE_HREF }	Gibt die URL aus, die vom durch den Seperator getrennten Text zum gesamten Text führt.

2.4 Template home.tpl

Dieses Template ist zuständig für die Darstellung der Home-Ansicht.

Platzhalter	Bedeutung/Ausgabe
{ \$LIST_INDEX }	Gibt die aktuelle Anzahl (den Index) der angezeigten Artikel zurück (also beim 3. Artikel den Wert 3, beim 6. Artikel den Wert 6).
{ \$GRP_NAME }	Gibt den Gruppennamen aus.
{ \$LIST_HEADER_EACH }	
{ \$TITLE }	Gibt den Namen/Titel des Artikels aus.
{ \$LEAD }	Gibt den Lead-Text des Artikels aus.
{ \$LIST_SHOWTEXT_EACH }	
{ \$CONTENT }	Gibt den zugehörigen Text des Artikels aus.
{ \$SHOWDATE }	Gibt zurück, ob das Erstelldatum angezeigt werden soll. Falls ja wird TRUE, bei nein FALSE zurückgegeben. Entnimmt den Wert aus DynPG.
{ \$CREATEDATE }	Gibt das Erstelldatum des Artikels aus.
{ \$LIST_FWD_TEXT }	
{ \$LIST_REFERER_TO }	Enthält die URL-Adresse zur nächsttieferen Anzeige-Stufe (im falle home.tpl den Link zum gesamten Artikel).

2.5 Template navigation-page.tpl

Dieses Template ist zuständig für die Darstellung von Ansichten der Artikel-Listen.

Platzhalter	Bedeutung/Ausgabe
{ \$PG_NAV_LASTPAGE }	Gibt die Seitenzahl der Seite davor aus.
{ \$PG_NAV_NEXTPAGE }	Gibt die Seitenzahl der nächsten Seite aus.
{ \$PG_NAV_RELPAGE }	Gibt die URL der Seite aus.
{ \$PG_NAV_SEPARATION }	
{ \$PG_NAV_ISCURRENT }	

2.6 Template search-form.tpl

Dieses Template ist zuständig für die Darstellung von Ansichten der Artikel-Listen.

Platzhalter	Bedeutung/Ausgabe
{ \$SEARCH_RESULTPAGE }	Gibt die URL der Seite aus, auf der die Suchresultate erscheinen.