



Grundlagen der Web-Entwicklung

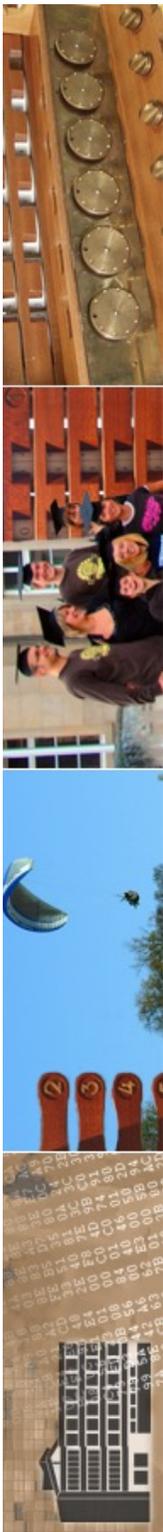
INF3172

Template-Engines am Beispiel Smarty

Thomas Walter

07.12.2023

Version 1.0





ZDV-Rundmail

Liebe Kolleginnen, liebe Kollegen,

wie Sie der Medienberichterstattung entnehmen konnten, hat es in den vergangenen Monaten erneut zahlreiche erfolgreiche Angriffe auf die IT-Infrastruktur öffentlicher Einrichtungen gegeben. In vielen Fällen gab es wochenlange Totalausfälle der IT-Infrastruktur an Krankenhäusern, Schulen und auch Universitäten. Leider muss während der bevorstehenden Weihnachts- und Neujahrstage erneut damit gerechnet werden, dass Hacker-Gruppen diese Zeit des Jahres für besonders gravierende Angriffe nutzen werden.

Das Rektorat der Universität hat uns daher aufgefordert, für diese Zeit besondere Vorkehrungen zu treffen. Um einen wirksamen Schutz zu gewährleisten, haben wir daher beschlossen, in der Zeit von

- * 23.12.2023 bis 26.12.2023
- * 30.12.2023 bis 01.01.2024

die Netzwerkanbindung der Universität von außen nach innen auf das absolut Nötigste zu reduzieren. Ihnen stehen dann außerhalb des universitären Netzes nur die verschlüsselten Web-Dienste über https zur Verfügung.

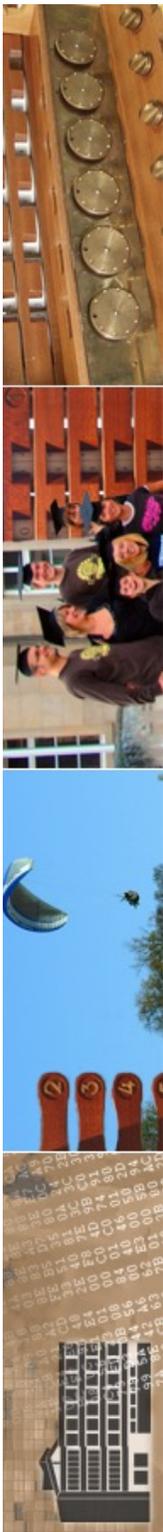
Um diese Maßnahme vorzubereiten, ist ein Probelauf für

- * Freitag, 08.12.2023 16h bis 17h



Frameworks

- Framework: *wiederverwertbares* Softwaresystem mit bereits implementierter, genereller Funktionalität
 - Spezialisierung führt zu konkreter Anwendung
 - Framework *setzt Architektur um*
 - Framework folgt Design Pattern
 - Beispiele
 - Node.js
 - ZF: Laminas (Zend Framework)
 - Ruby on Rails
 - JEE
 - Angular
 - dojo





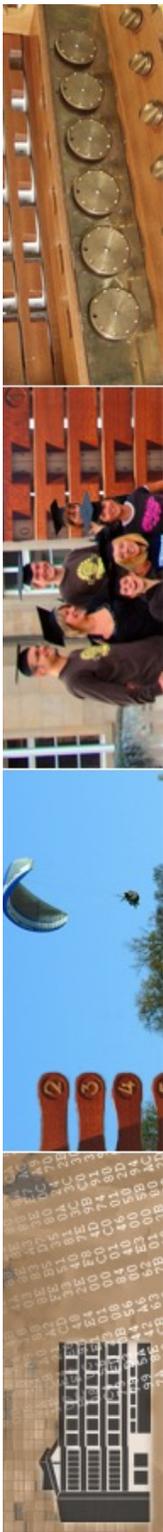
Frameworks (serverseitig) und MVC

	V	C	M
Smarty	X		
Laminas (ZF)	X	X	
ROR	X	X	X
CakePHP	X	X	X



Template Engine

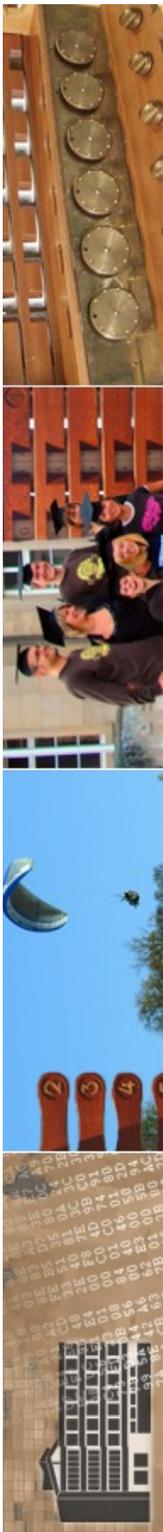
- Software, welche in Vorlagen (den Templates) Platzhalter mit konkretem Inhalt ausfüllt
- in Templates soll *keine* Business-Logik enthalten sein!
- in MVC-Paradigma: Abspaltung der View durch Template-Engine





Bewertung

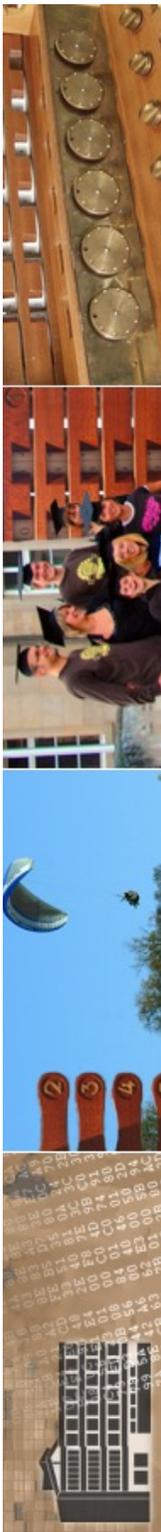
- durch Template-Engine *saubere Trennung* der View vom Programm-Code
- Designer unabhängig vom Programmierer, kann sich besser einbringen
- zusätzlicher Overhead
- zusätzliche Strukturen
- Performance





Beispiele

- Beispiele für Template Engines
 - Smarty
 - www.smarty.net
 - Fluid/FLOW3
 - alle weiteren serverseitigen Frameworks in dieser Veranstaltung haben eine Art von Template Engine: CakePHP, Ruby on Rails, Laminas-(ZEND)-Framework
 - zahlreiche weitere







Smarty
TEMPLATE ENGINE

Home | Download | Documentation | Forum | Mailing Lists

in English

Advertisement

Sponsors [info]

Play online casino
[casasapuestasdeportivas.es](#)
[rahapelit](#)
[casinonlineespaña.es](#)
[máquinas tragaperras](#)
[comparador de casinos](#)
[casinosonlineespaña.es](#)
[Casinon Utan Licens](#)
[allcasinos.jp](#)
[Casinowired.com](#)
[CryptoCasinos.com](#)
[Bookmakers](#)
[Pro Sites by John](#)
[non UK casinos](#)
[BSV DevCon](#)
[Buy Instagram](#)
[Followers](#)
[Nya nätcasino online](#)
[Eettafels Leasen](#)
[new uk bingo sites](#)
[ilmaiskierroksia ilman](#)
[talletusta](#)
[Casinoburst.com](#)
[Netticasinot](#)
[Intetics - Software](#)
[Development Company](#)
[VPS Server](#)

Sponsors

Want your site showcased? [email us a press release.](#)

Smarty 4.2.1 Released! *Sep 14, 2022*

Smarty 4.2.1 is released!

[4.2.1](#)
[v4 Change Log](#)

Smarty 4.2.0 Released! *Aug 1, 2022*

Smarty 4.2.0 is released!

[4.2.0](#)
[v4 Change Log](#)

Smarty 4.1.0 Released! *Feb 7, 2022*

Smarty 4.1.0 is released! This version adds PHP 8.1 compatibility.

[4.1.0](#)
[v4 Change Log](#)

PARTNERS

Check out the latest Finnish online casinos from [Netticasinot24.com](#)

Visit [Kasinolinna.com](#) and find all the best Finnish casinos.

Looking to play great [online casino](#) games? Betiton is your place to be!

Get Smarty

Download

About Smarty

All About Smarty

Why use it?

Use Cases and Work Flow

Syntax Comparison

Template Inheritance

Best Practices

Crash Course

Version 3 Overview

Testimonials

Resources

Smarty 2->3 upgrade notes

README (from distro)

Quick Install

Documentation

Discussion Forums

Example App

IRC (chat)

Mailing Lists

Links

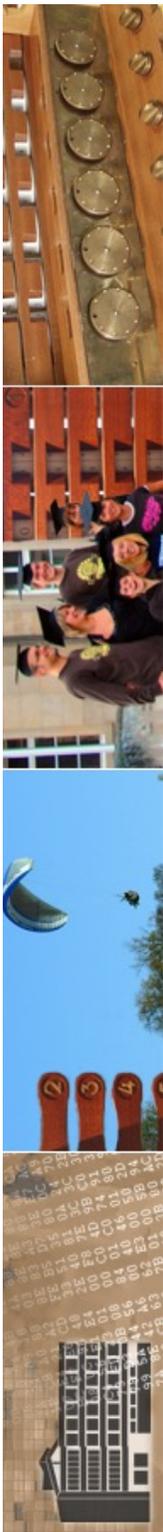
[www.php.net](#)





Einleitung Smarty

- Smarty ist eine „kompilierende Template-Engine“ für PHP
- Ziel: Trennung der in PHP geschriebenen Anwendungslogik von der Sicht/Formatierung
 - keine Applikationslogik im Template, keine Präsentationslogik in der PHP-Anwendung





Separation of presentation from application code

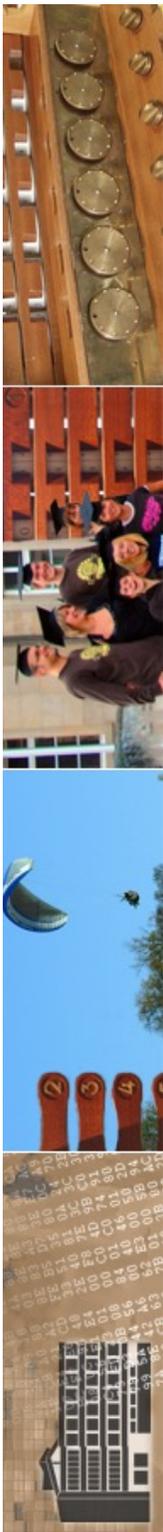
- This means templates can certainly contain logic under the condition that it is for presentation only. Things such as **including** other templates, **alternating** table row colors, **upper-casing** a variable, **looping** over an array of data and rendering it are examples of presentation logic.
- This does not mean however that Smarty forces a separation of business and presentation logic. Smarty has no knowledge of which is which, so placing business logic in the template is your own doing.
- Also, if you desire *no* logic in your templates you certainly can do so by boiling the content down to text and variables only.





Vorteile von Smarty

- Smarty ist
 - sehr performant
 - nur einmaliges Compilieren der Templates (wenn unverändert)
 - if/elsif/else/endif -Konstrukte im Template
 - unterstützt Caching
 - erweiterbar durch PlugIn-Architektur



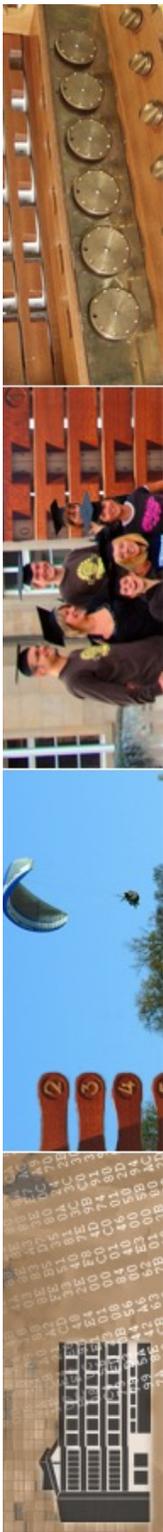


Ressourcen

- zentral:

<http://www.smarty.net/>

- darüber hinaus zahlreiche Ressourcen im Web
 - auch gute Dokumentation
(Monte Ohrt, Andrei Zmievski)





smarty

Smarty is a template engine for PHP, facilitating the separation of presentation (HTML/CSS) from application logic.

[View the Project on GitHub](#)
smarty-php/smarty

Smarty 4 Documentation

Smarty is a template engine for PHP, facilitating the separation of presentation (HTML/CSS) from application logic.

It allows you to write **templates**, using **variables**, **modifiers**, **functions** and **comments**, like this:

```
<h1>{$title|escape}</h1>

<p>
    The number of pixels is: {math equation="x * y" x=$hei
</p>
```

When this template is rendered, with the value "Hello world" for the variable \$title, 640 for \$width, and 480 for \$height, the result is:

```
<h1>Hello world</h1>

<p>
    The number of pixels is: 307200.
</p>
```

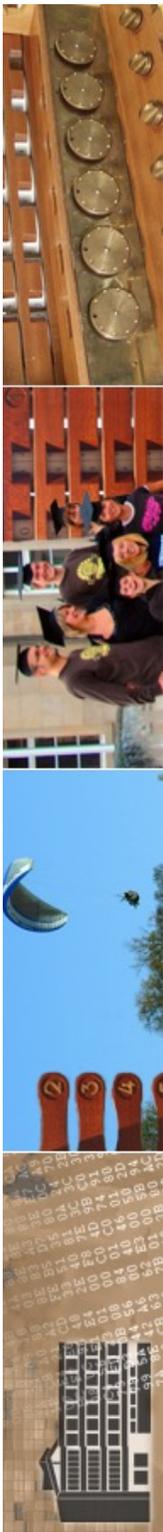
Introduction

- [Philosophy](#) - or "Why do I need a template engine?"
- [Features](#) - or "Why do I want Smarty?"
- [Getting Started](#)



Installation

- die Grundinstallation ist einfach:
 - Herunterladen von Smarty
(aktuell sind 2.6.31, 3.1.4 und 4.2.1)
 - Entpacken
 - Kopieren des Smarty-Verzeichnisses an geeignete Stelle, etwa `PHPDIR/smarty`
 - Anpassen der Direktive `include_path` in `php.ini`
 - aktuelles PHP vorausgesetzt





Installation

- oder:
Composer (wieder)



A Dependency Manager for PHP

Installation

Smarty versions 3.1.11 or later can be installed with [Composer](#).

To get the latest stable version of Smarty use:

```
composer require smarty/smarty
```

To get the latest, unreleased version, use:

```
composer require smarty/smarty:dev-master
```

To get the previous stable version of Smarty, Smarty 3, use:

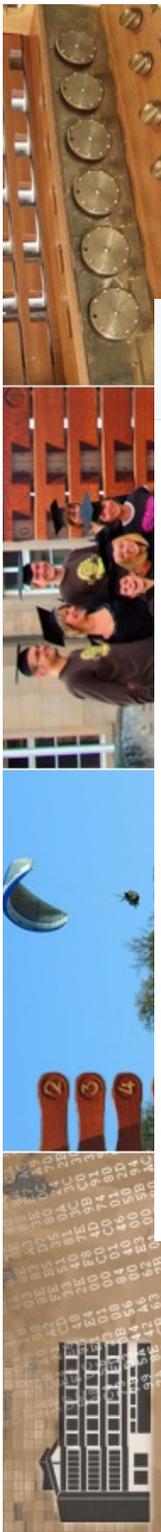
```
composer require smarty/smarty:^3
```

Here's how you create an instance of Smarty in your PHP scripts:

```
<?php

require 'vendor/autoload.php';
$smarty = new Smarty();
```

Now that the library files are in place, it's time to setup the Smarty directories for your application.



smarty-php / smarty Public

Notifications Fork

<> Code Issues 33 Pull requests 7 Discussions Actions Projects Wiki Security 5 Insights

master 24 branches 58 tags

Go to file Code

About

Smarty is a template engine for PHP, facilitating the separation of presentation (HTML/CSS) from application logic.

php smarty

Readme View license Security policy 2.1k stars 179 watching 670 forks

Releases 38

v4.3.0 Latest last month + 37 releases

Packages

No packages published

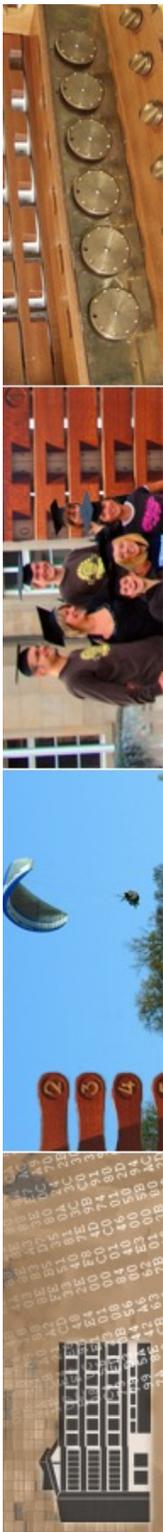
wisskid Allow dereferencing of non-objects across all supported PHP vers... ✓ a34ee98 last month 4,892 commits		
.github/workflows	PHP8.2 compatibility (#775)	last month
demo	fix PHP 8.1 deprecation notices in demo/plugins/cacheresource.pdo...	last year
docs	PHP8.2 compatibility (#775)	last month
lexer	Dropped remaining references to removed PHP-support in Smarty 4 ...	3 months ago
libs	Allow dereferencing of non-objects across all supported PHP versio...	last month
tests	Allow dereferencing of non-objects across all supported PHP versio...	last month
utilities	PHP8.2 compatibility (#775)	last month
.gitattributes	Include docs en demo in the releases.	3 months ago
.gitignore	add local testrunners for all supported PHP versions using docker. (#...	5 months ago
CHANGELOG.md	Allow dereferencing of non-objects across all supported PHP versio...	last month
LICENSE	Clarify correct LGPL version.	2 years ago
README.md	PHP8.2 compatibility (#775)	last month
SECURITY.md	Update SECURITY.md	4 months ago
composer.json	changed homepage links in composer.json	last year
docker-compose.yml	Simplify test running to support all phpunit cmdline options	last month





Verzeichnisstruktur

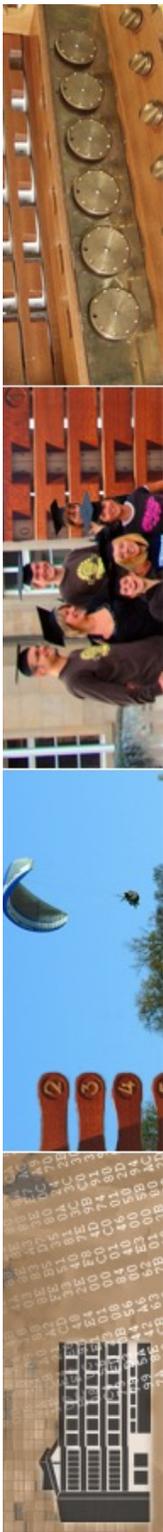
- Smarty sollte unbedingt mit einer vernünftigen Verteilung auf Verzeichnisse betrieben werden
 - spezielle Verzeichnisse für compilierte Templates, Templates, Konfigurationen, ...
 - nur wenig unterhalb von „htdocs“





erste Schritte

- Beispielanwendung mit Smarty:
 - Smarty-PHP setzt eine Variable „message“
 - Template fügt den Wert der Variablen ein und verwendet unser CSS





PHP - hellosmarty.php - Eclipse SDK

File Edit Navigate Search Project PHP/Apache Run Latex Window Help

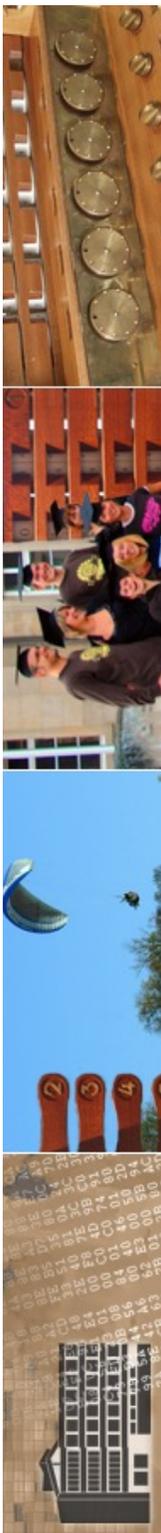
EuroCalc2.java Euro4.java hellosmarty.php hellosmarty.tpl 58

```

1  <?php
2  /*
3   * Created on 15.12.2005
4   *
5   * Fortgeschrittene Programmierung fuer das Internet
6   *
7   * Beispiel fuer Smarty
8   */
9  ?>
10
11
12 <?php
13 require 'Smarty.class.php';
14 $smarty = new Smarty;
15
16 $smarty->template_dir="D:\\www\\webst2\\smarty\\templates";
17
18 $smarty->assign("message","Hello Webst2<BR>here's Smarty");
19
20 $smarty->display('hellosmarty.tpl');
21 ?>

```

Writable Smart Insert 1 : 1





PHP - hellosmarty.tpl - Eclipse SDK

File Edit Navigate Search Project PHP/Apache Run Latex Window Help

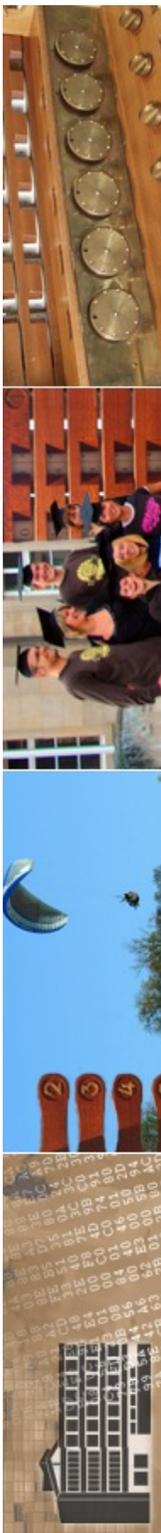
EuroCalc2.java hellosmarty.php hellosmarty.tpl smarty.lib.php »59

```

1 {* webst2: Smarty-Template *}
2
3 <HTML>
4   <HEAD>
5     <TITLE>webst2: HelloWorld mit Smarty</TITLE>
6     <link rel="stylesheet" type="text/css" href="/css/webst.css">
7   </HEAD>
8   <BODY>
9     <HR>
10    <CENTER><H2>
11      { $message }
12    </H2></CENTER>
13    <HR>
14  </BODY>
15 </HTML>

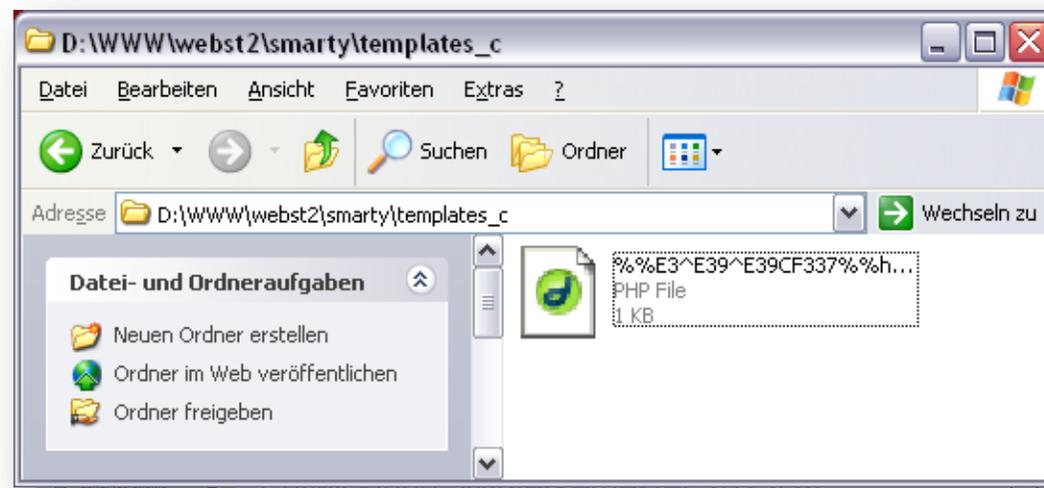
```

Writable Insert 1 : 1



beim Ausführen...

- ...wird das Template „compiliert“ zu reinem PHP
- großer Performance-Vorteil, da keine Neucompilation, solange sich das Template nicht geändert hat





The screenshot shows the XEmacs editor interface. The title bar reads 'XEmacs %%E3^E39^E39CF337%%hellosmarty.tpl.php - XEmacs'. The menu bar includes 'File', 'Edit', 'View', 'Cmds', 'Tools', 'Options', 'Buffers', 'Top', 'Bot', and 'Help'. The left sidebar contains icons for Open, Dired, Save, Print, Cut, Copy, Paste, Undo, and Spell. The main text area contains the following code:

```

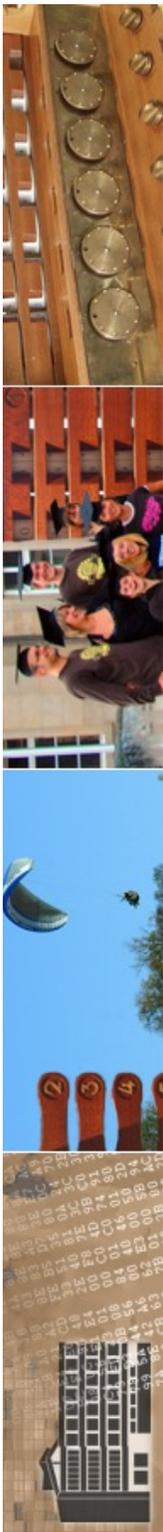
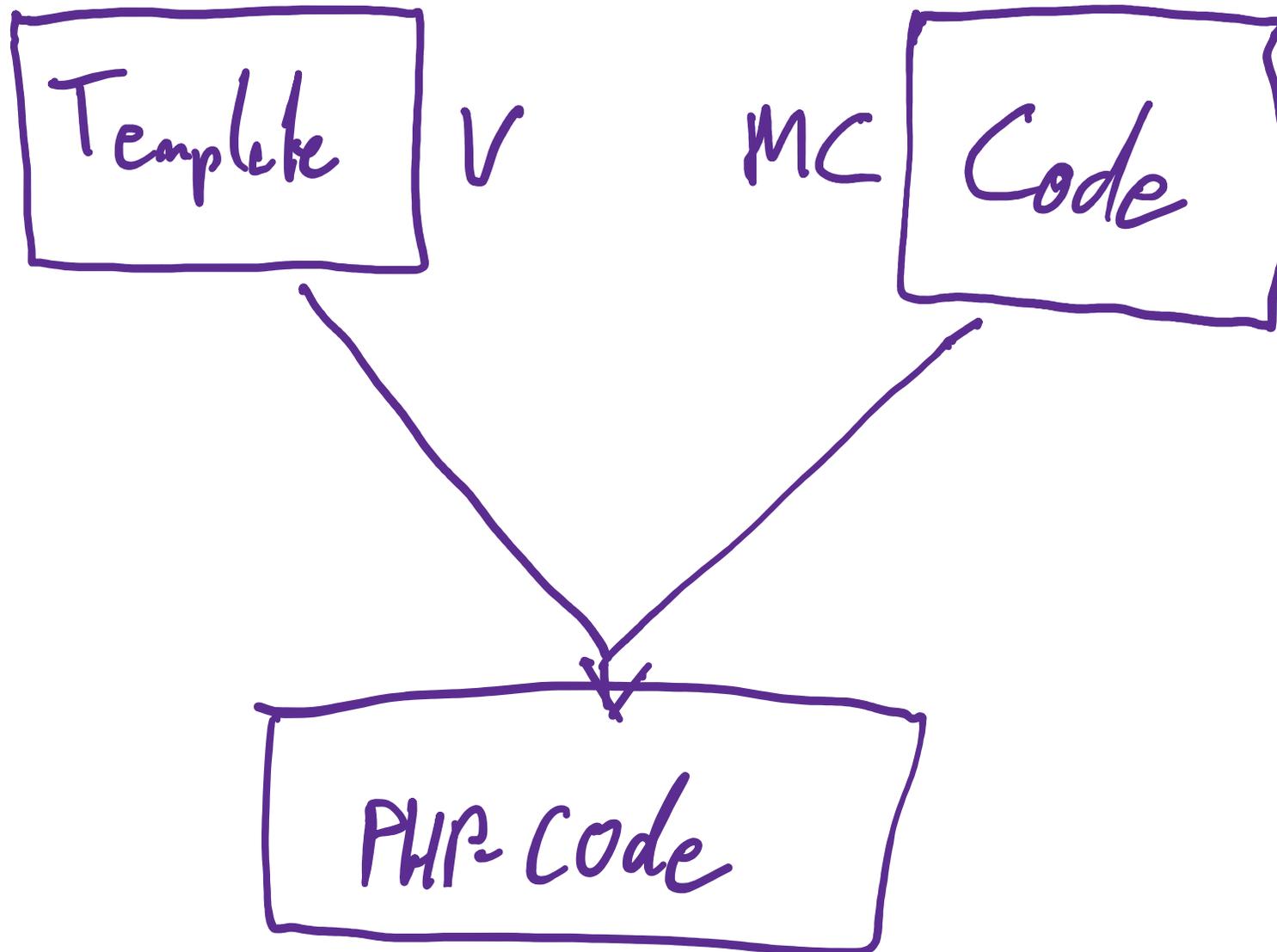
<?php /* Smarty version 2.6.11, created on 2005-12-16 08:43:44
      compiled from hellosmarty.tpl */ ?>

<HTML>
  <HEAD>
    <TITLE>webst2: HelloWorld mit Smarty</TITLE>
  </HEAD>
  <BODY bgcolor="ORANGE">
    <HR>
    <CENTER><H2>
      <?php echo $this->_tpl_vars['message']; ?>
    </H2></CENTER>
    <HR>
  </BODY>
</HTML>

```

The status bar at the bottom shows 'Raw-----XEmacs: %%E3^E39^E39CF337%%hellosmarty.tpl.php (C Font Abbrev)' and a message 'Scanning buffer... (100%) done'.

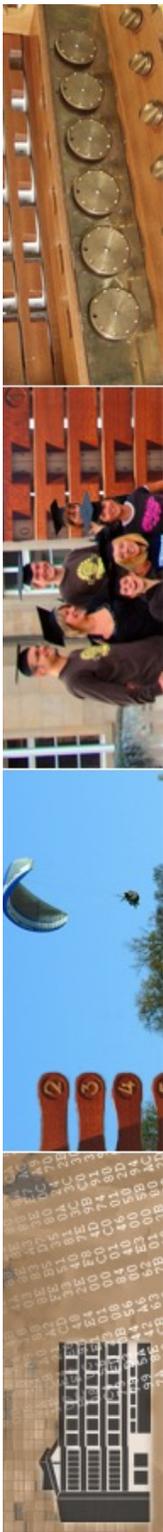






Konfiguration über Datei

- zahlreiche Smarty-Parameter können sinnvollerweise über eine zentrale include-Datei verwaltet werden



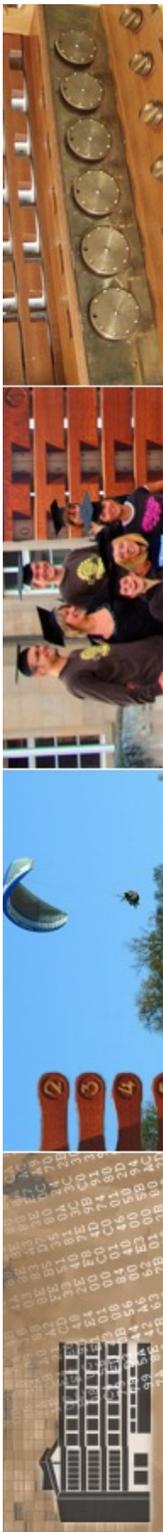


Smarty programmieren

- Grundlegendes:

Smarty-Code innerhalb von Templates beginnt standardmäßig mit den Tags

```
{ <smarty-code> }
```





Kommentare

- Kommentare in Smarty-Templates:

```
{
    * Kommentar im Smarty-Template *
}
```





Variablen

- Variablenbezeichner beginnen mit `$`
- Arrays werden mit `[i]` indiziert
- assoziative Arrays werden `$assocArray.key` indiziert
- Objektattribut: `$referenz->attribut`
- Objektmethode: `$referenz->methode()`
- Systemvariable: `#PATH#`





Übergabe von Variablenwerten aus PHP-Script (I)

- mittels
 - `assign(variablename, wert);`
- wird in PHP eine Variable für das Template bereitgestellt
- diese Variable hat im Template den Bezeichner `$variablename` und den Wert **wert**





Übergabe von Variablenwerten aus PHP-Script (II)

- analog für PHP-Arrays:
 - `assign(arrayname, array(0,1,2, "drei")) ;`

- und für assoziative Arrays in PHP:
 - `assign(hashname, array(key1 => value1, key2 => value2)) ;`





PHP - variablen.php - Eclipse SDK

File Edit Navigate Search Project PHP/Apache Run Latex Window Help

EuroCalc2.java hellosmarty.php hellosmarty.tpl smarty.lib.php variablen.php x variablen.tpl 59

```

1  <?php
2  /*
3   * Created on 16.12.2005
4   *
5   * Fortgeschrittene Programmierung fuer das Internet
6   *
7   * Variablen in Smarty
8   */
9
10 require 'Smarty.class.php';
11
12 $smarty = new Smarty;
13 $smarty->template_dir="D:\\www\\webst2\\smarty\\templates";
14
15 $smarty->assign("lokaleVariable","drei");
16 $smarty->assign("anArray",array(0,1,2,"drei"));
17 $smarty->assign("aHash",array("eins" => 1, "zwei" => "zwei", "drei" => "zweiundvierzig"));
18
19 $smarty->display('variablen.tpl');
20
21 ?>
22

```

Writable Smart Insert 1:1



PHP - variablen.tpl - Eclipse SDK

File Edit Navigate Search Project PHP/Apache Run Latex Window Help

hellosmarty.php | hellosmarty.tpl | variablen.php | variablen.tpl x »61

```

1 {* webst2: Smarty-Template zu Script "variablen.php" *}
2
3 <HTML>
4   <HEAD>
5     <TITLE>webst2: Variablen mit Smarty</TITLE>
6     <link rel="stylesheet" type="text/css" href="/css/webst.css">
7   </HEAD>
8   <BODY>
9     <HR>
10    <CENTER><H2>
11      Variablen mit Smarty
12    </H2></CENTER>
13    <H4><CODE>
14      lokaleVariable = { $lokaleVariable } <BR>
15      anArray[0] = { $anArray[0] } <BR>
16      anArray[3] = { $anArray[3] } <BR>
17      aHash[eins] = { $aHash.eins } <BR>
18      aHash[zwei] = { $aHash.zwei } <BR>
19      aHash[$lokaleVariable] = { $aHash.$lokaleVariable } <BR>
20    </CODE></H4>
21
22
23    <HR>
24  </BODY>
25 </HTML>

```

Writable Insert 1 : 1



Variablen mit Smarty

```

lokaleVariable = drei
anArray[0] = 0
anArray[3] = drei
aHash[eins] = 1
aHash[zwei] = zwei
aHash[$lokaleVariable] = zweiundvierzig

```





die Referenz `$smarty`

- die Variable `$smarty` referenziert auf ein Objekt, welches wesentliche Informationen zur Webapplikation speichert, etwa

- `$smarty.server.SERVER_NAME`

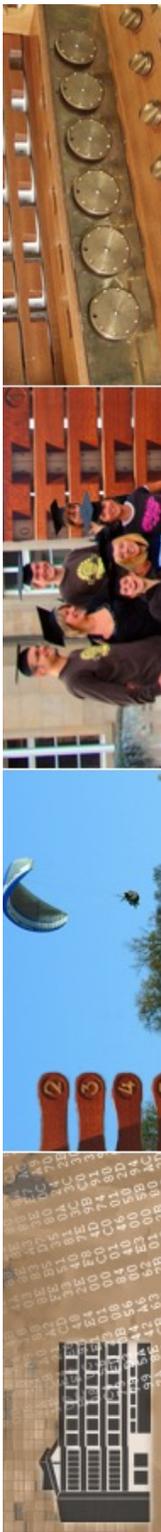
- `$smarty.env.PATH`

- `$smarty.session.id`

- `$smarty.cookies.benutzer`

- `$smarty.version`

- `$smarty.now`





Smarty Funktionen

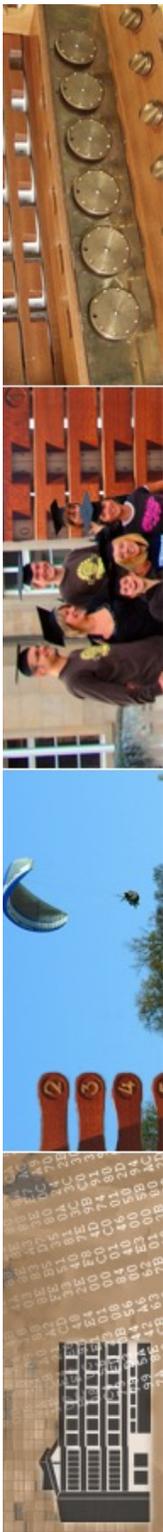
- Smarty verfügt über einige nützliche Funktionen, etwa
 - `lower`
 - `upper`
 - `n12br`
 - `replace`
 - `regex_replace`
 - ...





Laden einer Konfiguration

- die Direktive `config_load` lädt eine Konfiguration in Smarty
- zwingender Parameter:
 - `file`





Kontrollstrukturen

- Smarty stellt in den Templates auch Kontrollstrukturen bereit
 - Verzweigungen
 - Schleifen





Verzweigung

- Smarty stellt die übliche wenn-dann-sonst-Struktur mit elseif zur Verfügung
- Syntax:

```

- {if $a == 1}
    a hat Wert 1
{elseif $a == 2}
    a hat Wert 2
{else}
    a hat anderen Wert
{/if}
    
```



Schleifen

- foreach-Schleife durchläuft Array

- Syntax:

– ...

```
{foreach from=$anArray item=i}
    Wert = <B>{$i}</B>
{/foreach}
```



PHP - schleifen.php - Eclipse SDK

File Edit Navigate Search Project PHP/Apache Run Latex Window Help

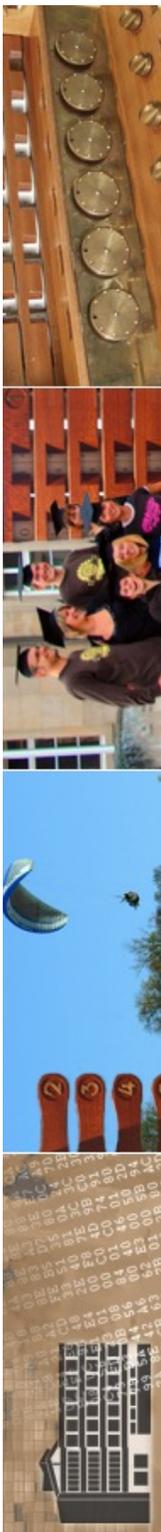
variablen.tpl | variablen.php | debugging.php | schleifen.php | schleifen.tpl

```

1  <?php
2  /*
3   * Created on 16.12.2005
4   *
5   * Fortgeschrittene Programmierung fuer das Internet
6   *
7   * Beispiel fuer Kontrollstrukturen in Smarty: Schleifen
8   */
9
10 require 'Smarty.class.php';
11
12 $smarty = new Smarty;
13 $smarty->template_dir="D:\\www\\webst2\\smarty\\templates";
14
15 $myArray = array(0,1,2,"drei");
16
17 $smarty->assign("anArray", $myArray);
18
19 $smarty->display('schleifen.tpl');
20
21 ?>

```

Writable Smart Insert 1 : 1





PHP - schleifen.tpl - Eclipse SDK

File Edit Navigate Search Project PHP/Apache Run Latex Window Help

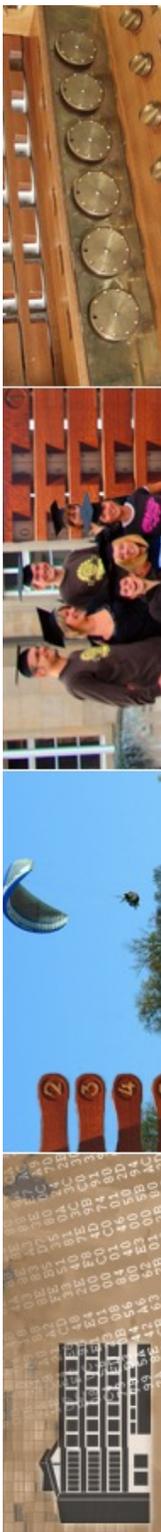
variablen.tpl | variablen.php | debugging.php | schleifen.php | schleifen.tpl

```

1 { * webst2: Beispiel fuer Kontrollstrukturen in Smarty: Schleifen * }
2
3 <HTML>
4   <HEAD>
5     <TITLE>webst2: Variablen mit Smarty</TITLE>
6     <link rel="stylesheet" type="text/css" href="/css/webst.css">
7   </HEAD>
8   <BODY>
9     <HR>
10    <CENTER><H2>
11      Schleifen mit Smarty
12    </H2>
13    <H4>
14      { foreach from=$anArray item=i}
15        Wert = <B>{ $i }</B><BR>
16      { /foreach}
17
18    </H4>
19    </CENTER>
20    <HR>
21  </BODY>
22 </HTML>

```

Writable Insert 1 : 1





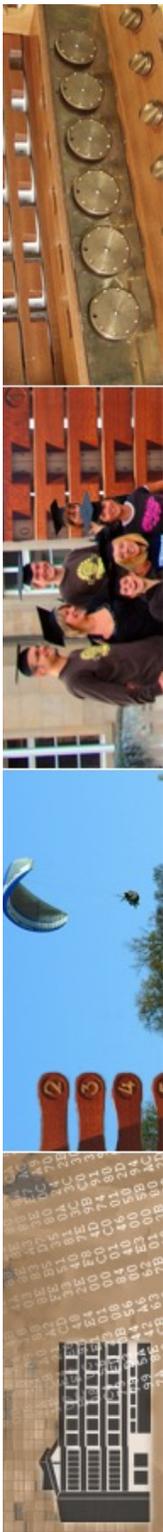


Importieren

- nützlich ist der Import von Dateien in die Template-Datei:

```
{ include file="footer.tpl" }
```

- neben `include` gibt es noch `insert`, nur findet hier kein Caching statt

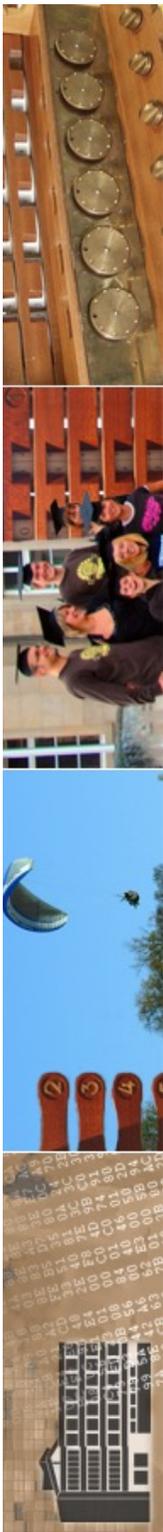




Wertzuweisung innerhalb eines Templates

- auch innerhalb eines Templates kann einer Smarty-Variablen ein Wert zugewiesen werden:

```
{
  assign var="name" value="value"
}
```





die Anweisung `fetch`

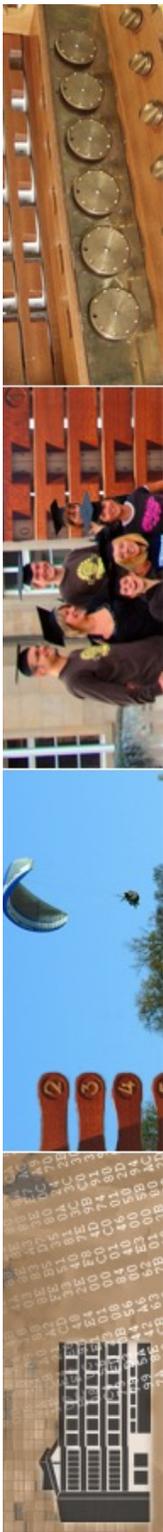
- mittels `{ fetch file="datei" }` kann mit den Protokollen
 - http
 - ftp
 - (Zugriff auf eine Datei auf dem lokalen Filesystem)
- ein Dokument angefordert werden





HTML-Output

- ähnlich wie die HTML-Ausgabefunktionen des PERL-Moduls CGI.pm bzw. Python-Modul CGI kann mit Smarty-Funktionen HTML-Code einfach erzeugt werden
- es steht eine Vielzahl derartiger Funktionen für die jeweiligen HTML-Elemente zur Verfügung
 - Funktionen `html_xxx`





Konfigurationsdateien

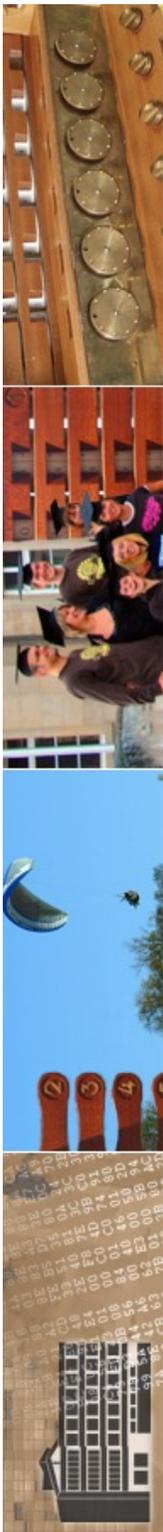
- mittels Konfigurations-Dateien können elegant Template-Variablen übersichtlich gesetzt werden
- wichtige Konfigurationen:
 - SMARTY_DIR
 - \$template_dir
 - sinnvoller weise *nicht* unterhalb von „htdocs“
 - \$cache_dir
 - ...





Caching

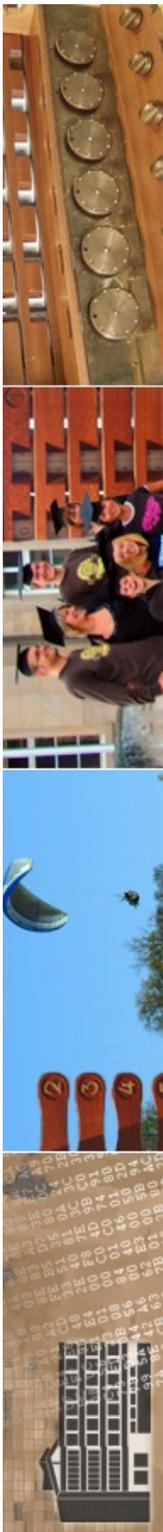
- Smarty hat umfassende Möglichkeiten zur Unterstützung von Caching
- bezieht sich auf die Methoden
 - `fetch()`
 - `display()`
- Steuerung über Variable `$caching`





Plugins

- Smarty kann durch Plugins erweitert werden
- diese sind in PHP geschrieben





Smarty 3

- benötigt PHP5 oder PHP7
- neuer Template-Parser erlaubt u.a. einfache Berechnungen und rekursive Funktionen im Template
- neues Smarty Data Object: **Smarty_Data**
- Vererbung von Templates
- Funktionsdefinition und -aufruf im Template
- weitgehend rückwärtskompatibel zu Smarty 2





Diskussion

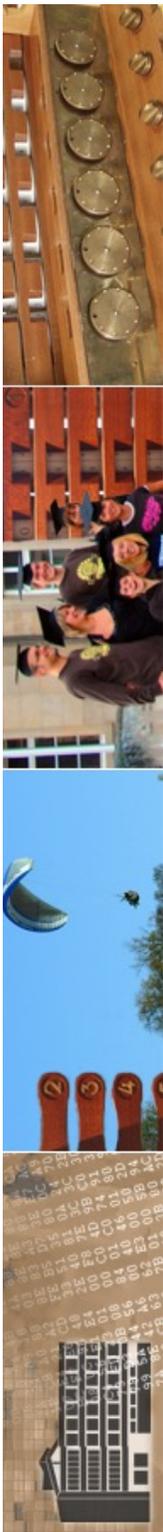
- Vorteile von Smarty
 - modulare Entwicklung durch Konfigurationsfiles, die leicht zu warten sind
 - Sicherheit: Templates enthalten kein PHP, so dass ein reiner Template-Designer keinen „Schaden“ anrichten kann
 - technische Vorteile
 - Caching, Debugging, Compiling, ...





Diskussion: STL

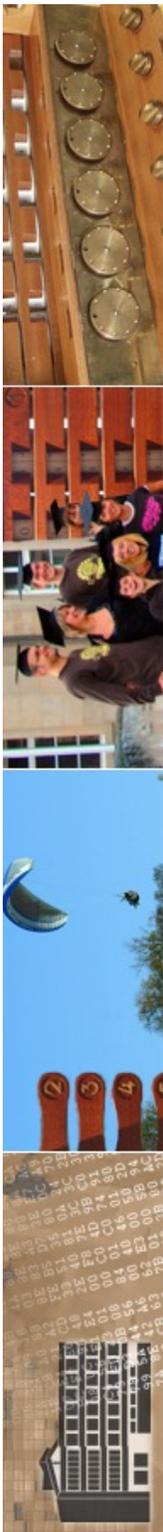
- Smarty Template Language (STL)
 - sehr eingeschränkte Sprache im Template
 - Nachteil: Weitere Sprache notwendig
 - Vorteile
 - leicht zu lernen
 - Erzwingt geradezu die Trennung View-Controller





Ziel

- Trennung von Anwendungscode von Darstellung
 - Designer können den Anwendungscode nicht beschädigen
 - Fehlerbehandlung modular
 - klare Rollentrennung
 - der Designer braucht im wesentlichen nur HTML-Kenntnisse, um die Anwendung auf seine CI anzupassen

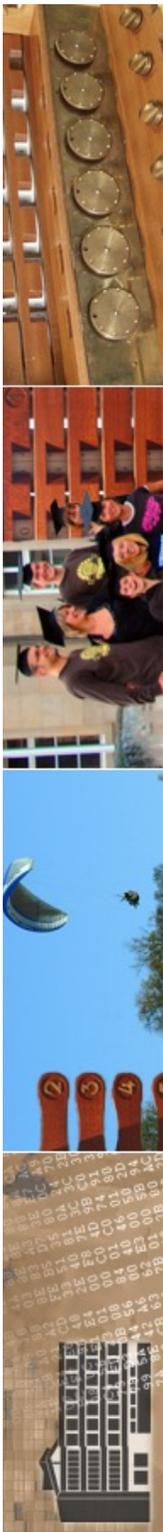




What Smarty is Not

Smarty is not an application development framework. Smarty is not an MVC. Smarty is not an alternative to Laravel, Symfony, CodeIgniter, or any of the other application development frameworks for PHP.

Smarty is a template engine, and works as the (V)iew component of your application. Smarty can easily be coupled to any of the engines listed above as the view component. No different than any other software, Smarty has a learning curve. Smarty does not guarantee good application design or proper separation of presentation, this still needs to be addressed by a competent developer and web designer.



...und nun...

- haben wir ein einfaches Framework für die Komponente View kennen gelernt



- als nächstes: das Laminas-(ZEND-)Framework als zentrales Beispiel für PHP-Frameworks

