



Grundlagen der Web-Entwicklung

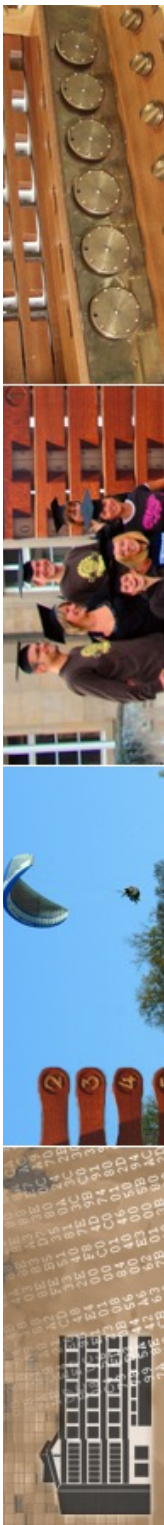
INF3172

Template-Engines am Beispiel Smarty

Thomas Walter

16.12.2021

Version 1.0





Digital Services Act: EU-Abgeordnete für Recht auf wirksame Verschlüsselung

Im EU-Parlament steht ein Kompromiss zum Gesetz für digitale Märkte. Plattformen sollen nicht an Ende-zu-Ende-Verschlüsselung gehindert werden dürfen.

Lesezeit: 7 Min.  In Pocket speichern

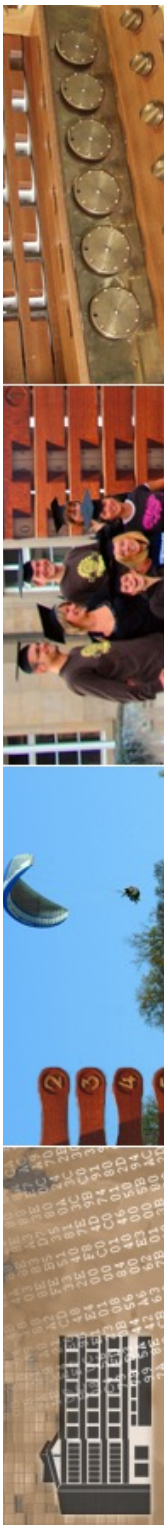


(Bild: Cristian Storto/Shutterstock.com)

07:40 Uhr

Von Stefan Krempf

"Die Mitgliedstaaten dürfen Anbieter von Vermittlungsdiensten nicht daran hindern, Ende-zu-Ende-verschlüsselte Dienste anzubieten." Ein entsprechendes Recht auf durchgehende Verschlüsselung sehen heise online vorliegende Kompromissanträge im EU-Parlament zum Digital Services Act (DSA) vor, über die der federführende Ausschuss für Binnenmarkt und Verbraucherschutz (IMCO) am Montag und Dienstag abstimmen soll. Mit der Annahme eines Großteils der Übereinkunft wird gerechnet, sodass die Verhandlungsposition der Abgeordneten zu dem Gesetz für digitale Dienste damit prinzipiell feststeht. Sie muss dann nur noch in einer Plenarsitzung bestätigt werden.



 Alert!

Firefox 95: Neue Sandbox-Technik verbessert Browser-Sicherheit

Die jetzt veröffentlichte Version 95 von Firefox schließt 13 Sicherheitslücken und liefert eine neuartige Sandbox mit. Firefox ESR erscheint in Version 91.4.

Lesezeit: 4 Min.  In Pocket speichern

   72



(Bild: Sundry Photography/Shutterstock.com)

08.12.2021 14:38 Uhr | Security

Von Dirk Knop

Die Firefox-Entwickler haben in der gerade veröffentlichten Version 95 insbesondere die Sicherheit des Webbrowsers im Blick gehabt. Die neue Fassung schließt insgesamt 13 Sicherheitslücken mit teils als "hoch" eingestuftem Risiko. Zudem ergänzt sie eine neue Sandbox namens "RLBox", deren Schutzfunktion deutlich über die Möglichkeiten der klassischen Sandbox-Technologie zur Prozess-Isolierung hinausgehen soll. Die Version 91.4.0 mit Langzeit-Unterstützung, Firefox ESR, haben die Entwickler ebenfalls abgesichert.





Template Engine

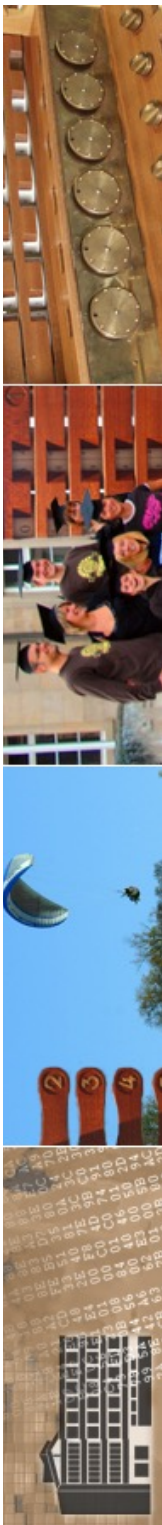
- Software, welche in Vorlagen (den Templates) Platzhalter mit konkretem Inhalt ausfüllt
- in Templates soll *keine* Business-Logik enthalten sein!
- in MVC-Paradigma: Abspaltung der View durch Template-Engine





Bewertung

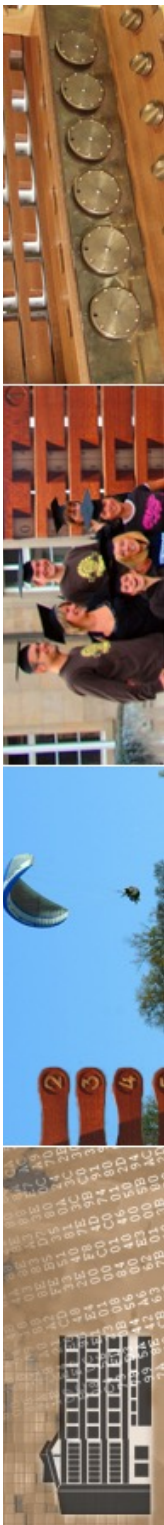
- durch Template-Engine *saubere Trennung* der View vom Programm-Code
- Designer unabhängig vom Programmierer, kann sich besser einbringen
- zusätzlicher Overhead
- zusätzliche Strukturen
- Performance





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, Lammas-(ZEND)-Framework
 - zahlreiche weitere





- Home
- Download
- Documentation
- Forum
- Mailing Lists

Search... in English

Advertisement

Sponsors [info]

- sports betting app
- CryptoCasinos.com
- Bookmakers
- Pro Sites by John
- non UK casinos
- internetcasino.jp
- Instagram Followers - Skweezer
- ufabet
- slotxo
- slotbar888
- sbobet
- BSV DevCon
- Buy RDP
- CasinoSlam
- bankid casino
- casasapuestasdeportivas.es
- rahapelit
- casinoonlineespana.es
- Buy Instagram Followers
- buy_rdp
- CasinoHEX.in
- Nya nätcasino online
- Nettikasinot
- CasinoTop.com
- Energie vergelijken
- Eettafels Leasen
- new uk bingo sites
- Netticasinot
- Intetics - Software Development Company
- VPS Server

I.T Certifications

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

Smarty 4.0.0 Released! Dec 3, 2021

Smarty 4.0.0 is released! This version requires PHP 7.1 or greater. If you need PHP before 7.1 please continue to use Smarty 3.1.4.

- [4.0.0](#)
- [v4 Change Log](#)

Smarty 3.1.4 Released Oct 13, 2021

- [3.1.4](#)
- [v3 Change Log](#)
- [v3 New Features](#)

Smarty discussion forum is moving! Sep 9, 2021

The Smarty forum is becoming a larger burden and security concern and mostly just spam control. Sometime in the coming weeks I will be retiring the forum portion of the website. Fear not, because there will still be a place to discuss! And you can use it today.

We will move discussion to the r/smarty subreddit:

<https://www.reddit.com/r/smarty/>

Feel free to post there now, it's created and ready!

You can also submit bugs/issues to the official smarty github repo:

<https://github.com/smarty-php/smarty/issues>

I'm not sure when I'll turn down the forums or exactly how I will do it, but eventually

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
- www.phpinsider.com

Donate





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





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



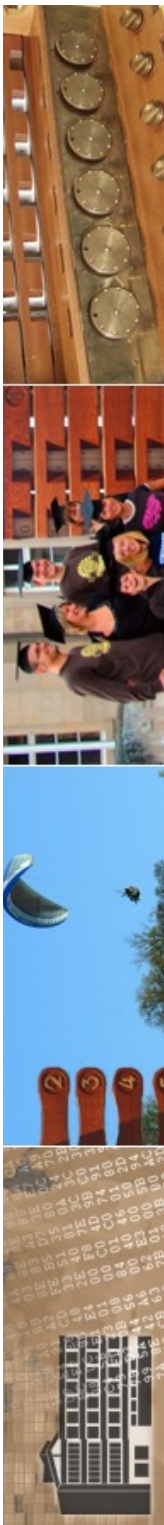


Ressourcen

- zentral:

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

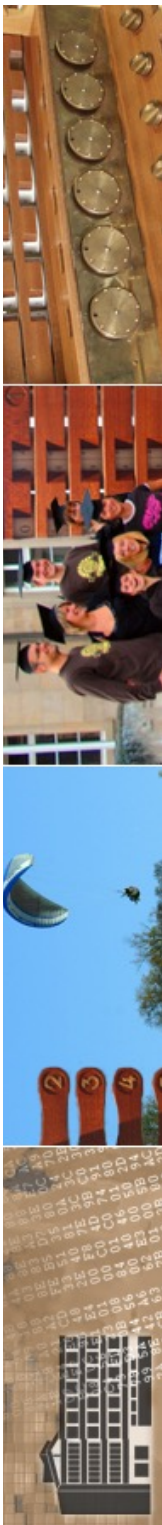
- darüber hinaus zahlreiche Ressourcen im Web
 - auch gute Dokumentation
(Monte Ohrt, Andrei Zmievski)





Installation

- die Grundinstallation ist einfach:
 - Herunterladen von Smarty
(aktuell sind 2.6.31, 3.1.4 und 4.0.0)
 - Entpacken
 - Kopieren des Smarty-Verzeichnisses an geeignete Stelle, etwa `PHPDIR/smarty`
 - Anpassen der Direktive `include_path` in `php.ini`
 - aktuelles PHP vorausgesetzt





smarty-php / smarty Public

Notifications Fork 628

Code Issues 39 Pull requests 22 Discussions Actions Projects Wiki Security 2 Insights

Releases / v4.0.0

v4.0.0

Latest

Compare

wisskid released this 20 days ago · 6 commits to master since this release v4.0.0 baebd59

Jay! Smarty4, with support for PHP8.0 is alive!

Assets 2

Source code (zip)

Source code (tar.gz)

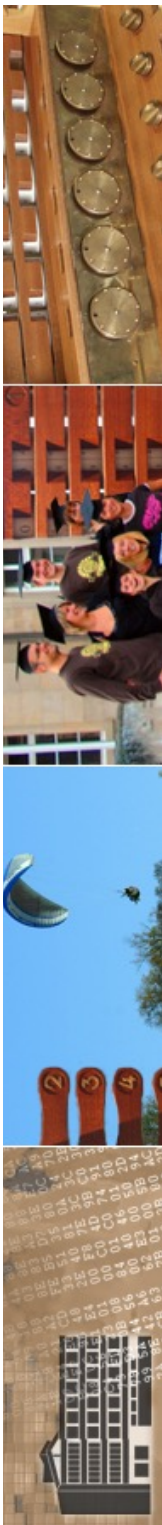
18 1 10 4 29 people reacted





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

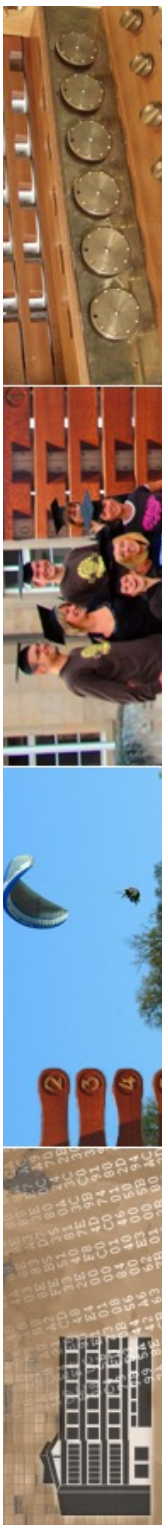
EuroCalc2.java Euro4.java hellosmarty.php hellosmarty.tpl 58

```

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

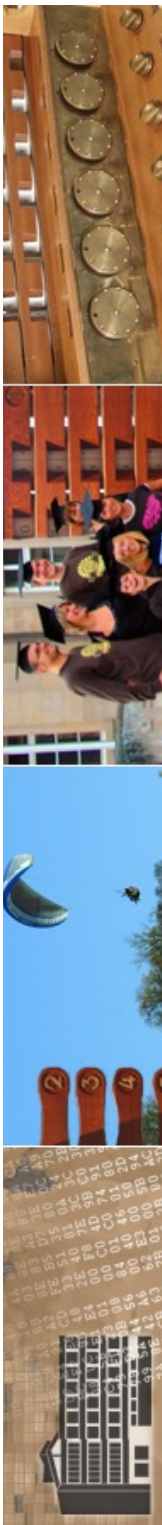
EuroCalc2.java hellosmarty.php hellosmarty.tpl smarty.lib.php »59

```

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>

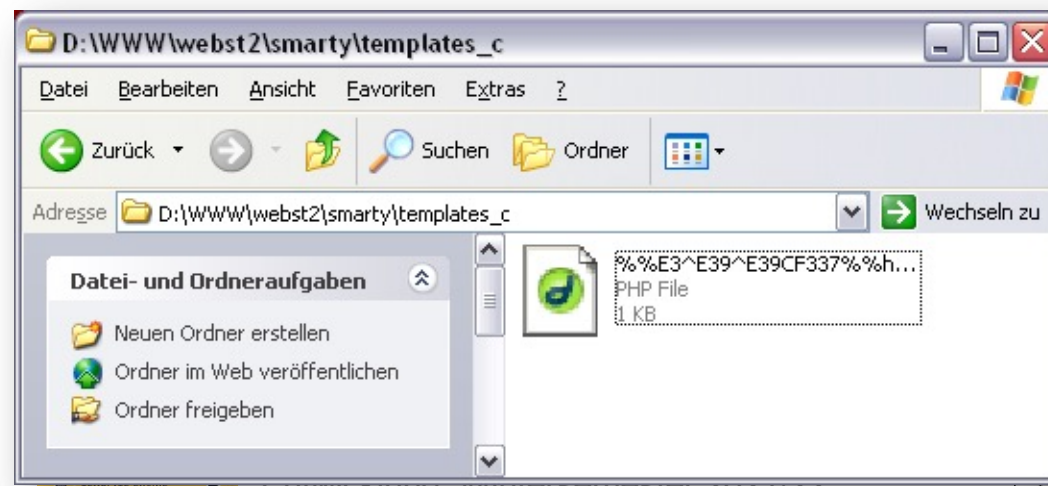
```

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, Direk, 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>

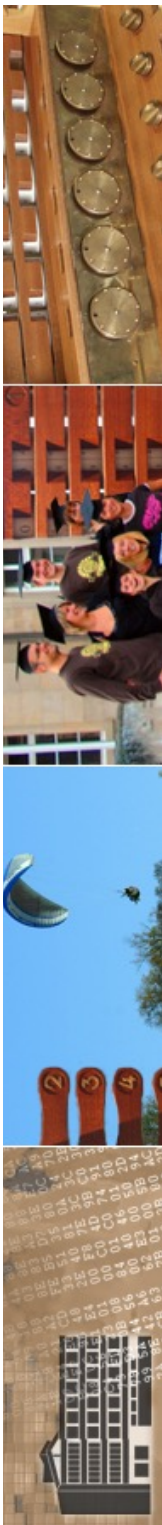
```

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



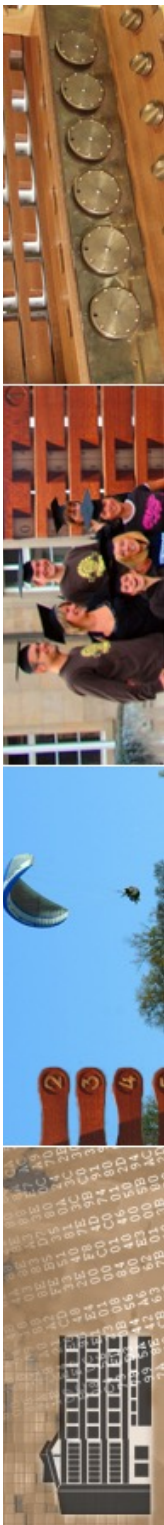


Smarty programmieren

- Grundlegendes:

Smarty-Code innerhalb von Templates
beginnt standardmäßig mit den Tags

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

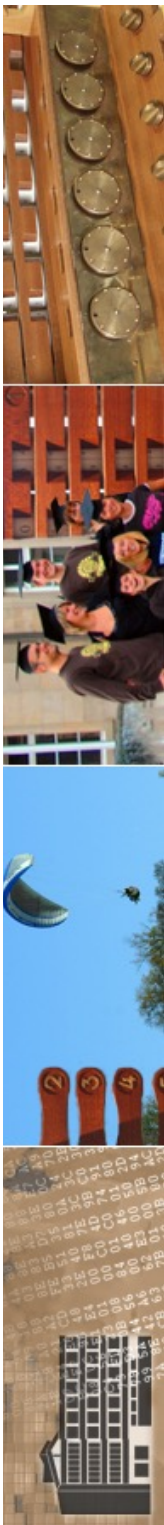




Kommentare

- 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)

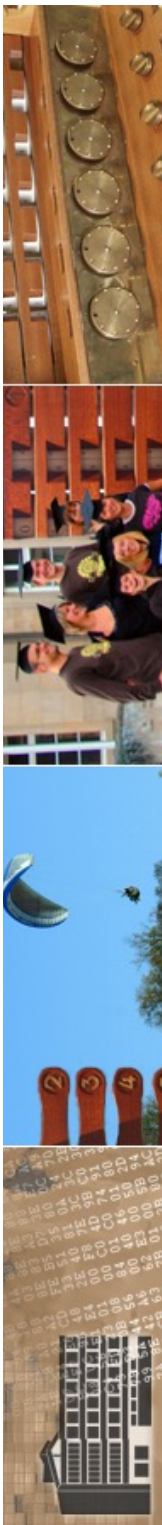
- mittels
 - `assign(variablename, wert);`
- wird in PHP eine Variable für das Template bereitgestellt
- diese Variable hat im Template den Bezeichner `$variablename` und den Wert **wert**





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 | variablen.tpl

```

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

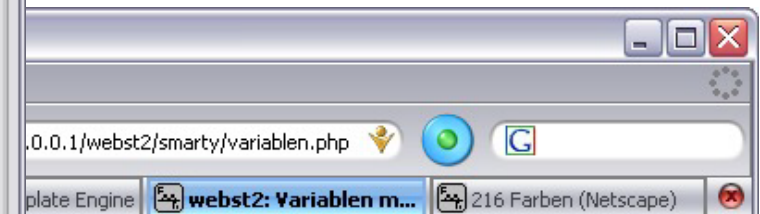
hellosmarty.php | hellosmarty.tpl | variablen.php | variablen.tpl x »61

```

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`

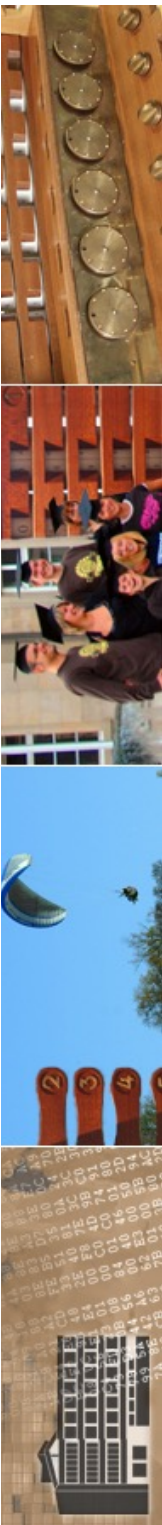
- `$smarty.env.PATH`

- `$smarty.session.id`

- `$smarty.cookies.benutzer`

- `$smarty.version`

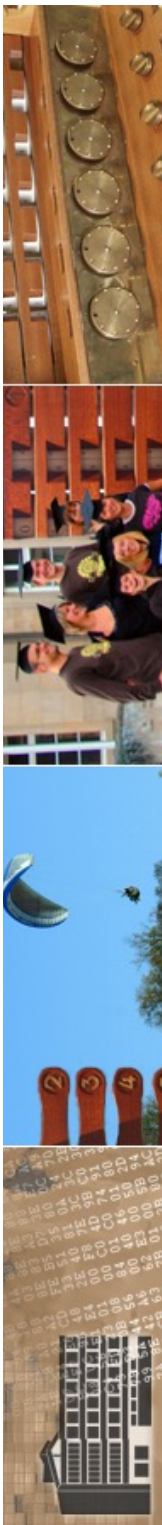
- `$smarty.now`





Smarty Funktionen

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





Laden einer Konfiguration

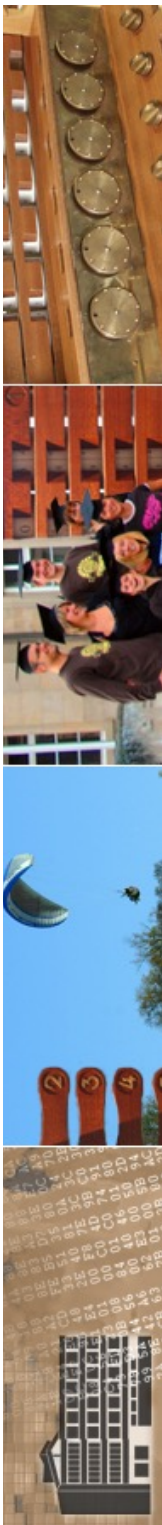
- die Direktive `config_load` lädt eine Konfiguration in Smarty
- zwingender Parameter:
 - `file`





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

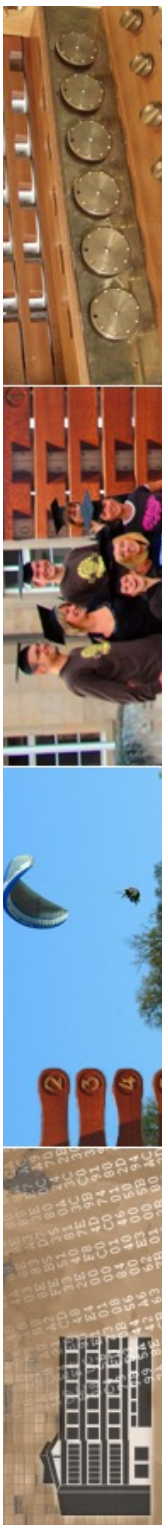
variablen.tpl | variablen.php | debugging.php | schleifen.php | schleifen.tpl

```

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

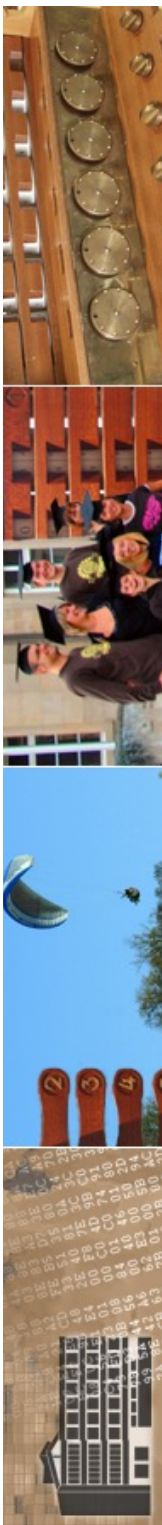
variablen.tpl | variablen.php | debugging.php | schleifen.php | schleifen.tpl

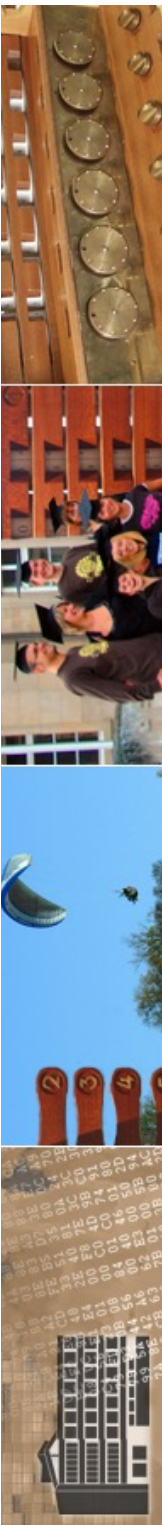
```

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





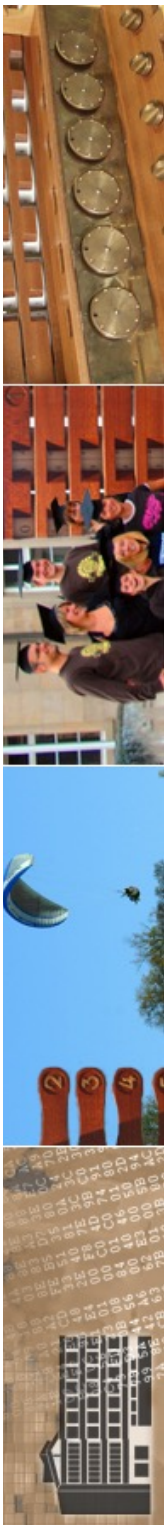


Importieren

- 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

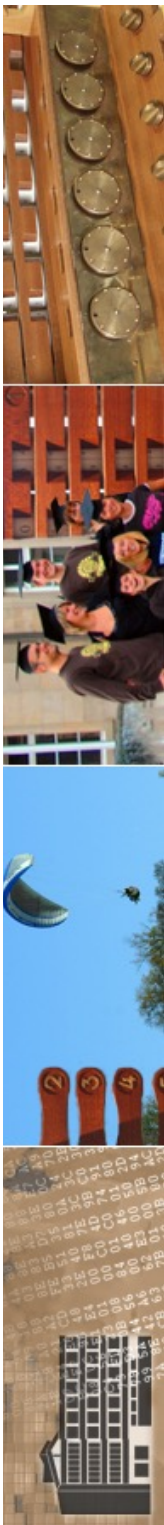




Wertzuweisung innerhalb eines Templates

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

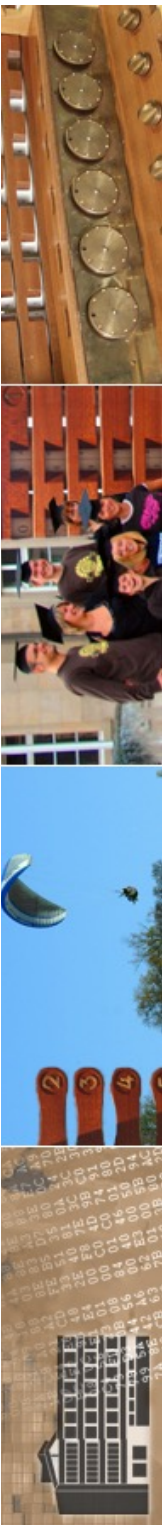
```
{
  assign var="name" value="value"
}
```





die Anweisung `fetch`

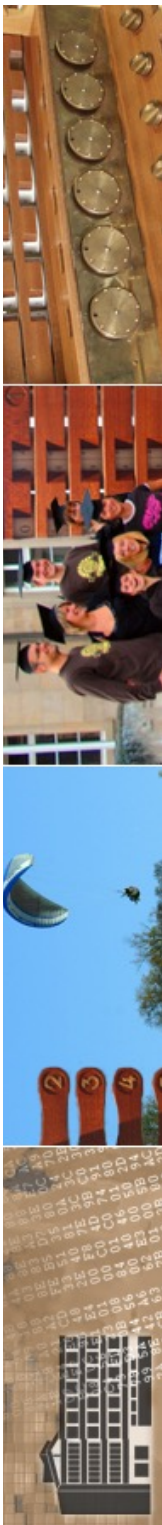
- mittels `{ fetch file="datei" }` kann mit den Protokollen
 - http
 - ftp
 - (Zugriff auf eine Datei auf dem lokalen Filesystem)
- ein Dokument angefordert werden





HTML-Output

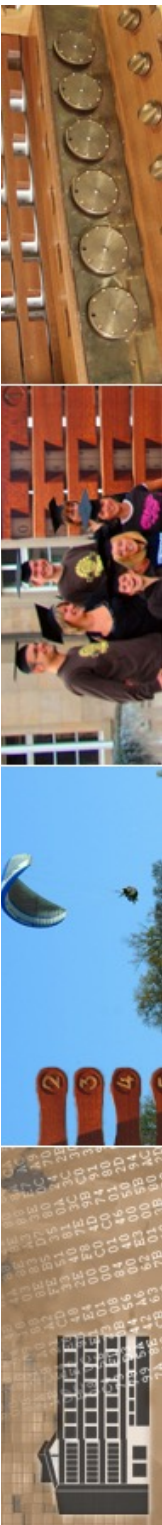
- ähnlich wie die HTML-Ausgabefunktionen des PERL-Moduls CGI.pm bzw. Python-Modul CGI kann mit Smarty-Funktionen HTML-Code einfach erzeugt werden
- es steht eine Vielzahl derartiger Funktionen für die jeweiligen HTML-Elemente zur Verfügung
 - Funktionen `html_xxx`





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

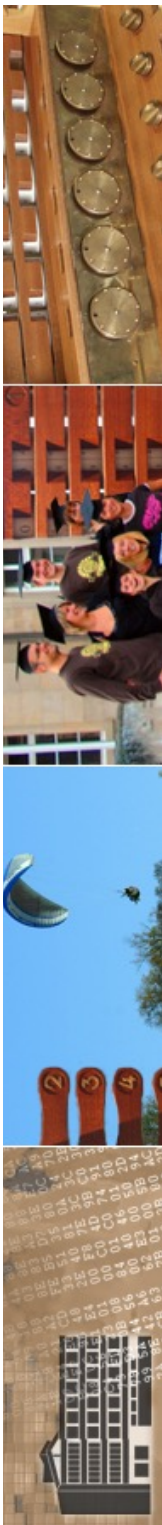
- Smarty hat umfassende Möglichkeiten zur Unterstützung von Caching
- bezieht sich auf die Methoden
 - `fetch()`
 - `display()`
- Steuerung über Variable `$caching`





Plugins

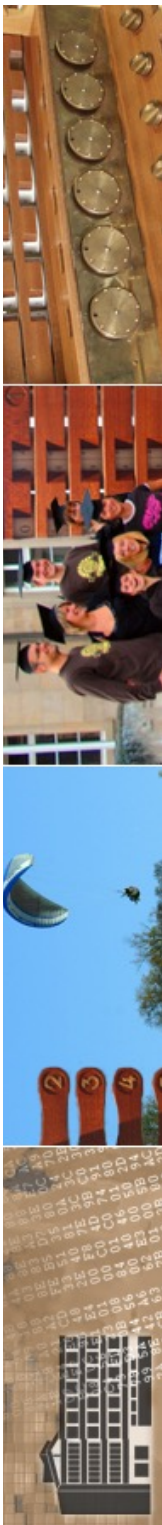
- Smarty kann durch Plugins erweitert werden
- diese sind in PHP geschrieben





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

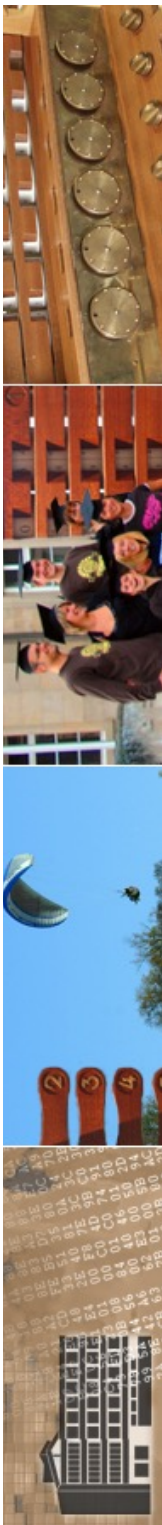




Diskussion

- Vorteile von Smarty

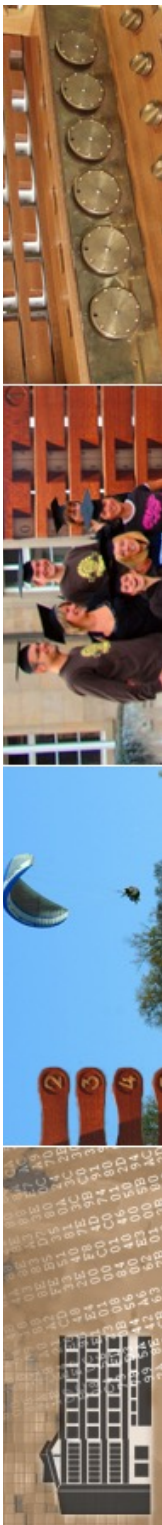
- modulare Entwicklung durch Konfigurationsfiles, die leicht zu warten sind
- Sicherheit: Templates enthalten kein PHP, so dass ein reiner Template-Designer keinen „Schaden“ anrichten kann
- technische Vorteile
 - Caching, Debugging, Compiling, ...





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



...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

