



# CONTENT MANAGEMENT SYSTEMS

14.03.2014 @ SAE INSTITUTE  
BJOERN ZAPADLO

# ABOUT ME

Bjoern Zapadlo  
Konstanz  
34 Jahre

Team Manager / Lead Developer HolidayCheck AG

Informatik Studium 1999 - 2002  
3 Agenturen in Stuttgart

HolidayCheck International Websites / new Framework  
Neckermann / Thomas Cook

Dozent an der SAE, Dualen Hochschule Stuttgart,  
Hochschule Furtwangen

PHP, Java, Scala, Javascript, CSS, Html, MySQL, MongoDB,  
Elasticsearch, ...

# CONTACT ME

[bjoern.zapadlo@gmail.com](mailto:bjoern.zapadlo@gmail.com)

<http://www.zapadlo.de>

@BjoeZap

[https://www.xing.com/profile/Bjoern\\_Zapadlo](https://www.xing.com/profile/Bjoern_Zapadlo)

<http://de.linkedin.com/pub/bjoern-zapadlo/36/889/1a5>

Facebook

Google+

# HOLIDAYCHECK AG

Größtes deutsches Meinungsportal für Reise und Urlaub

Vermittlung von Reisen

Sitz in der Schweiz, direkt am Bodensee

Börsennotiert über Tomorrow Focus AG

Existiert seit 1999

Ausgründungen in mehreren europäischen Ländern

Über 300 Mitarbeiter

[HTTP://WWW.HOLIDAYCHECK.DE  
/JOBS](http://www.holidaycheck.de/jobs)

JETZT ABER SCHLUSS MIT DER  
WERBUNG... ;)

# AGENDA

1. Template-Engines
2. Basics
3. Typen von CMS
4. Bestandteile eines CMS
5. Marktübersicht
6. Wordpress
7. Contao
8. Joomla
9. Drupal
10. Typo3
11. Übung - Meine eigene CMS Seite
12. Q&A

# 01. TEMPLATE ENGINES

Was sind Template Engines und wozu werden sie eingesetzt, Voraussetzungen, Handbuch und Links, jeweils  
=> Twig, Smarty, Fluid, Jeweils ein Beispiel

smarty-Syntax, Sprachkonstrukte (Variablen, Schleifen, Kontrollstrukturen), `$smarty`, `includes`, Variablen-Modifizier, Funktionen (`html_*`, `include`, `literal`, etc.)



# BEFORE WE START

TEMPLATE ENGINES SIND DIE  
BASIS VON MODERNEN  
WEBAPPLIKATIONEN

- Frameworks: Zend, Symfony, ...
- CMS

# TEMPLATES?!?

Templates (der englische Begriff für Schablonen), sind Vorlagen, die mit Inhalt gefüllt werden können

Eine Vorlage (englisch template) dient in der Datenverarbeitung zur Erstellung von Dokumenten oder Dokumentteilen. Sie stellt eine Art „Gerüst“ dar, die einen Teil des Inhaltes oder der Gestaltung des Dokumentes vorgibt.

Durch Einsetzen der fehlenden Bestandteile wird die Vorlage zu einem vollständigen Dokument ergänzt. Sie werden hauptsächlich für DTP- oder Textverarbeitungsprogramme und Webseiten genutzt.

# TEMPLATE ENGINES

Eine Template-Engine (von engl. Vorlage und Maschine) ist eine Software, die eine Datei (das Template) verarbeitet, und bestimmte Platzhalter darin mit jeweils aktuellen Inhalten füllt. Die Bezeichnungen Templateklasse und Templatesystem werden oft als Synonym für eine Template-Engine verwendet.

# GRUNDLEGENDES

Klassisch in einer PHP-Applikation repräsentiert eine PHP-Datei eine Html-Seite

Mit Template-Engines erfolgt eine Trennung in eine Datei, die die PHP Logik enthält und ein Template, welches das Html + Template-Code enthält.

# WARUM TEMPLATES?

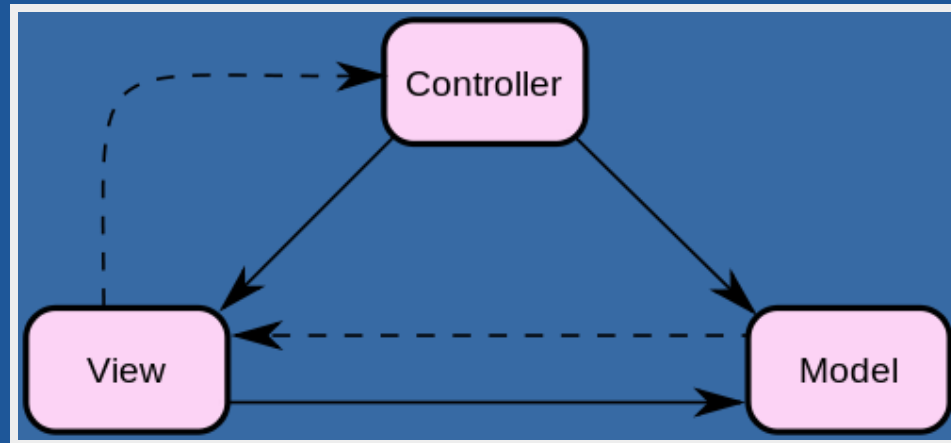
- MVC-Architektur
- Aufteilung von Aufgaben möglich
- Sicherheit (XSS)
- Komfort über Helper-Funktionen

# EXKURS: MVC

## MODEL VIEW CONTROLLER

Der englischsprachige Begriff model view controller (MVC, englisch für Modell-Präsentation-Steuerung) ist ein Muster zur Strukturierung von Software-Entwicklung in die drei Einheiten Datenmodell (engl. model), Präsentation (engl. view) und Programmsteuerung (engl. controller). Manche Autoren stufen es als Architekturmuster ein, andere als Entwurfsmuster. Ziel des Musters ist ein flexibler Programmentwurf, der eine spätere Änderung oder Erweiterung erleichtert und eine Wiederverwendbarkeit der einzelnen Komponenten ermöglicht. Es ist dann zum Beispiel möglich, eine Anwendung zu schreiben, die das gleiche Modell benutzt, aber einerseits eine Windows- oder Linux-Oberfläche realisiert, andererseits aber auch eine Weboberfläche beinhaltet. Beides basiert auf dem gleichen

# MVC - KOMPONENTEN



# MODELL (MODEL)

Das Modell enthält die darzustellenden Daten und gegebenenfalls (abhängig von der Implementierung des MVC-Patterns) auch die Geschäftslogik. Es ist von Präsentation und Steuerung unabhängig. Die Bekanntgabe von Änderungen an relevanten Daten im Modell geschieht nach dem Entwurfsmuster „Beobachter“. Das Modell ist das zu beobachtende Subjekt, auch Publisher, also „Veröffentlicher“, genannt



# STEUERUNG (CONTROLLER)

Die Steuerung verwaltet eine oder mehrere Präsentationen, nimmt von ihnen Benutzeraktionen entgegen, wertet diese aus und agiert entsprechend. Zu jeder Präsentation existiert eine eigene Steuerung. Die Steuerung sorgt dafür, dass Benutzeraktionen wirksam werden, z.B. durch Änderung der Präsentation (z.B. Verschieben des Fensters) oder durch Weiterleiten an das Modell (z.B. Übernahme von Eingabedaten oder Auslösen von Verarbeitungen). Als es noch keine Objektorientierung gab, bestand ein Modell nur aus Daten, und die Steuerung hat die Daten oft direkt aktualisiert. In einer objektorientierten Umgebung ist es dagegen besser, wenn das Modell die Geschäftsobjekte enthält und die Steuerung sich darauf beschränkt, Benutzereingaben (Daten und Methodenaufrufe)

# PRÄSENTATION (VIEW)

Die Präsentationsschicht ist für die Darstellung der benötigten Daten aus dem Modell und die Entgegennahme von Benutzerinteraktionen zuständig. Sie kennt sowohl ihre Steuerung als auch das Modell, dessen Daten sie präsentiert, ist aber nicht für die Weiterverarbeitung der vom Benutzer übergebenen Daten zuständig. Im Regelfall wird die Präsentation über Änderungen von Daten im Modell mithilfe des Entwurfsmusters „Beobachter“ unterrichtet und kann daraufhin die aktualisierten Daten abrufen. Die Präsentation verwendet oft das Entwurfsmuster „Kompositum“.

# EXKURS: XSS

## CROSS SITE SCRIPTING

Cross-Site-Scripting (XSS; deutsch Websiteübergreifendes Scripting) bezeichnet das Ausnutzen einer Computersicherheitslücke in Webanwendungen, indem Informationen aus einem Kontext, in dem sie nicht vertrauenswürdig sind, in einen anderen Kontext eingefügt werden, in dem sie als vertrauenswürdig eingestuft werden. Aus diesem vertrauenswürdigen Kontext kann dann ein Angriff gestartet werden.

Ziel ist es meist, an sensible Daten des Benutzers zu gelangen, um beispielsweise seine Benutzerkonten zu übernehmen (Identitätsdiebstahl).

Cross-Site-Scripting ist eine Art der HTML Injection. Cross-Site-Scripting tritt dann auf, wenn eine Webanwendung

# VORTEILE

- Programm-Code (PHP, Python, Perl) wird vom Markup (HTML) getrennt.
- Designer und Programmierer können gleichzeitig an einem Projekt arbeiten.
- WYSIWYG-Editoren können verwendet werden.

# NACHTEILE

- Template-Engines erzeugen immer zusätzlichen Overhead.
- Template-Engines müssen im Gebrauch erlernt werden. Neben der neuen Syntax sind auch oft grundlegende Kenntnisse der objektorientierten Programmierung Voraussetzung.
- Template-Engines sind auf eine textuelle Ausgabe beschränkt.

# MARKTÜBERSICHT

Übersicht:

[http://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_template\\_engin](http://en.wikipedia.org/wiki/Comparison_of_web_template_engin)

# ÜBUNG 01

## LEBENS LAUF ALS PHP

# SMARTY

Smarty ist eine freie (unter der LGPL veröffentlichte) Template Engine, die als PHP-Bibliothek vorliegt. Sie wurde mit dem Ziel entworfen, bei der Entwicklung von Webapplikationen die Trennung von Code und Ausgabe zu ermöglichen. Die Ausgabe erfolgt meist in HTML, möglich ist jedes textbasierte Dateiformat, zum Beispiel auch XML.



# HANDBUCH

<http://www.smarty.net/docsv2/de/>

# VORAUSSETZUNGEN

Smarty benötigt einen Webserver mit PHP  $\geq 4.0.6$  bzw. 5.2.0  
(Smarty 3.x)

# INSTALLATION

Download und entzippen nach libs

```
<?php
require('libs/Smarty.class.php');
$smarty = new Smarty();
$smarty->display('cv.tpl.html');
?>
```

# GRUNDLEGENDE SYNTAX

# KOMMENTARE

Kommentare werden von Asterisks umschlossen, und mit Trennzeichen umgeben. Beispiel: `{* das ist ein Kommentar *}`

Smarty-Kommentare werden in der Ausgabe nicht dargestellt und vor allem dazu verwendet, die Templates verständlicher aufzubauen. Smarty Kommentare werden sind in der endgültigen Ausgabe NICHT dargestellt. (im Gegensatz zu `<!-- HTML Kommentaren -->`). Sie sind nützlich um in den Templates interne Anmerkungen zu hinterlassen.

```
<body>
{* Dies ist ein einzeiliger Kommentar *}

{* dies ist ein mehrzeiliger
Kommentar, der nicht zum
Browser gesandt wird.
*}
</body>
```

# VARIABLEN

Templatevariablennamen beginnen mit einem \$dollar-Zeichen. Sie können Ziffer, Buchstaben und Unterstriche ('\_') enthalten, sehr ähnlich den Variablen in PHP. Numerische Arrayindizes können referenziert werden und auch Nichtnumerische. Zugriff auf Objekteigenschaften und -methoden ist auch möglich.

```
{ $foo }          <-- Zeigt einfache Variable an (kein Array oder Objekt)
{ $foo[4] }       <-- Zeigt 5. Element von einem Array an, deren Schlüssel bei 0
{ $foo.bar }      <-- Zeigt das Element zum Schlüssel "bar" des Arrays an (wie PH
{ $foo.$bar }     <-- Zeigt das Element eines variablen Schlüssels an (wie PHPs $
{ $foo->bar }      <-- Zeigt eine Eigenschaft "bar" des Objekts $foo an
{ $foo->bar() }    <-- Zeigt den Rückgabewert der Objectmethode "bar" an
```

# VARIABLENZUWEISUNG

```
<?php
require('libs/Smarty.class.php');
$smarty = new Smarty();

$smarty->assign('name', 'george smith');
$smarty->assign('address', '45th & Harris');

$smarty->display('cv.tpl.html');
?>
```

# FUNKTIONEN

Jedes Smarty-Tag gibt entweder eine Variable aus oder ruft eine Funktion auf. Funktionen werden aufgerufen indem der Funktionsname und die Parameter mit Trennzeichen umschlossen werden. Beispiel: {funcname attr1="val" attr2="val"}.

```
{mySimpleFunction}  
{html_select_date display_days=true param2="hallo welt"}
```



# MODIFIKATOREN

Variablen-Modifikatoren können auf alle Variablen angewendet werden, um deren Inhalt zu verändern. Dazu hängen sie einfach ein | (Pipe-Zeichen) und den Modifikatornamen an die entsprechende Variable an. Ein Modifikator über Parameter in seiner Arbeitsweise beeinflusst werden. Diese Parameter werden dem Modifikatorname angehängt und mit : getrennt.

```
{* Modifikator auf eine Variable anwenden *}
{$titel|upper}
{* Modifikator mit Parametern *}
{$title|truncate:40:"..." }

{* Modifikator auf Funktionsparameter anwenden *}
{html_table loop=$myvar|upper}
{* mit Parametern *}
{html_table loop=$myvar|truncate:40:"..." }

{* formatierung einer Zeichenkette *}
{"foobar"|upper}

{* mit date_format das aktuelle Datum formatieren *}
{"now"|date_format:"%Y/%m/%d" }
```

# IF / ELSE

{if}-Statements in Smarty erlauben die selbe Flexibilität wie in PHP, bis auf ein paar Erweiterungen für die Template-Engine. Jedes {if} muss mit einem {/if} kombiniert sein. {else} und {elseif} sind ebenfalls erlaubt. Alle PHP Vergleichsoperatoren und Funktionen, wie ||, or, &&, and, is\_array(), etc. sind erlaubt.

```
{if $name == "Fred"}      {* ein Beispiel mit == und 'eq' (gleich) *}
    Willkommen der Herr.
{elseif $name eq "Wilma"}
    Willkommen die Dame.
{else}
    Willkommen, was auch immer Du sein magst.
{/if}

{if ( $anzahl < 0 or $anzahl > 1000 ) and $menge >= #minMengeAmt#} {* Klammer
    ...
{/if}

{if count($var) gt 0} {* einbetten von php Funktionsaufrufen ('gt' steht für
    ...
{/if}
```

# ITERATION - SECTION

Template-{sections} werden verwendet, um durch Arrays zu iterieren (ähnlich wie {foreach}). Jedes section-Tag muss mit einem /section-Tag kombiniert werden. name und loop sind erforderliche Parameter. Der Name der 'section' kann frei gewählt werden, muss jedoch aus Buchstaben, Zahlen oder Unterstrichen bestehen. {sections} können verschachtelt werden

```
<?php
$data = array(1000,1001,1002);
$smarty->assign('custid',$data);
?>
```

```
{* dieses Beispiel gibt alle Werte des $KundenId Arrays aus *}
{section name=kunde loop=$custid}
    id: {$KundenId[kunde]}<br>
{/section}
{* alle Werte in umgekehrter Reihenfolge ausgeben: *}
{section name=kunde loop=$custid step=-1}
    id: {$KundenId[kunde]}<br>
```

# ITERATION - FOREACH

Die foreach Schleife ist eine Alternative zu section. foreach wird verwendet, um ein assoziatives Array zu durchlaufen. Die Syntax von foreach-Schleifen ist viel einfacher als die von section. {foreach} Tags müssen mit {/foreach} tags kombiniert werden. Erforderliche Parameter sind: from und item.

```
<?php
$data = array(1000,1001,1002);
$smarty->assign('custid',$data);
?>
```

```
{* dieses Beispiel gibt alle Werte aus dem $KundenId Array aus *}
{foreach from=$custid item=aktuelle_id}
    id: {$aktuelle_id}<br>
{/foreach}
```

# ITERATION - FOREACH

Foreach-Loops haben auch eigene Variablen welche die Foreach Eigenschaften enthalten. Diese werden wie folgt ausgewiesen: {\$smarty.foreach.foreachname.varname}.  
foreachname ist der Name der als name Attribut von Foreach übergeben wurden.

**iteration**: gibt die aktuelle iteration aus, beginnend bei 1

**first** ist TRUE wenn die aktuelle Iteration die erste ist

**last** ist TRUE wenn die aktuelle Iteration die letzte ist

**total** gibt die Anzahl Iterationen des Foreach Loops aus und kann in- oder nach- Foreach Blöcken verwendet werden.

# INCLUDE

{include}-Tags werden verwendet, um andere Templates in das aktuelle Template einzubinden. Alle Variablen des aktuellen Templates sind auch im eingebundenen Template verfügbar. Das {include}-Tag muss ein 'file' Attribut mit dem Pfad zum einzubindenden Template enthalten.

```
<html>
<head>
  <title>{$title}</title>
</head>
<body>
  {include file='page_header.tpl'}

  {* hier kommt der body des Templates *}
  {include file="$tpl_name.tpl"} <-- $tpl_name wird durch eine Wert ersetzt

  {include file='page_footer.tpl'}
</body>
</html>
```

# BEISPIEL

```
<?php
include('Smarty.class.php');

$smarty = new Smarty;
$smarty->assign('name', 'george smith');
$smarty->assign('address', '45th & Harris');

$smarty->display('index.tpl');
?>
```

```
<html>
<head>
  <title>Info</title>
</head>
<body>

  <pre>
  User Information:

  Name: {$name}
  Address: {$address}
  </pre>

</body>
</html>
```

```
<html>
<head>
  <title>Info</title>
</head>
<body>

  <pre>
  User Information:

  Name: george smith
  Address: 45th & Harris
  </pre>

</body>
</html>
```



# ÜBUNG 02

## LEBENS LAUF ALS SMARTY TEMPLATE

# TWIG

Twig ist eine moderne Template Engine von den Machern des Symfony Frameworks und daher auch Teil dieses

# HANDBUCH

<http://twig.sensiolabs.org/documentation>

# VORAUSSETZUNGEN

Smarty benötigt einen Webserver mit PHP 5.2.4

# INSTALLATION

Download und entzippen nach lib/Twig

```
<?php
require_once 'lib/Twig/Autoloader.php';
Twig_Autoloader::register();
$loader = new Twig_Loader_Filesystem('.');
$twig = new Twig_Environment($loader, array());

echo $twig->render('index.tpl.html', array('bla' => 'blubb'));
?>
```

# KOMMENTARE

```
{# note: disabled template because we no longer use this
  {% for user in users %}
    ...
  {% endfor %}
#}
```

# VARIABLEN

```
{{ foo.bar }}  
{{ foo['bar'] }}
```

# VARIABLEN ZUWEISEN

```
{% set foo = 'foo' %}  
{% set foo = [1, 2] %}  
{% set foo = {'foo': 'bar'} %}
```

```
echo $twig->render('index.tpl.html', array('bla' => 'blubb', 'blubb' => 'bla'
```



# FILTER

```
{{ name|striptags|upper }}  
{{ list|join(', ') }}
```

# FUNKTIONEN

```
{{ date() }}  
{% for i in range(0, 3) %}  
    {{ i }},  
{% endfor %}
```

# IF / ELSE

```
{% if kenny.sick %}  
    Kenny is sick.  
{% elseif kenny.dead %}  
    You killed Kenny! You bastard!!!  
{% else %}  
    Kenny looks okay --- so far  
{% endif %}
```

# ITERATION

```
<h1>Members</h1>
<ul>
{% for user in users %}
    <li>{{ user.username|e }}</li>
{% endfor %}
</ul>
```

# INCLUDE

```
{% include 'sidebar.html' %}  
  
{% for box in boxes %}  
    {% include "render_box.html" %}  
{% endfor %}
```

# VERERBUNG

```
<!DOCTYPE html>
<html>
<head>
  {% block head %}
  <link rel="stylesheet" href="style.css" />
  <title>{% block title %}{% endblock %} - My Webpage</title>
  {% endblock %}
</head>
<body>
  <div id="content">{% block content %}{% endblock %}</div>
  <div id="footer">
    {% block footer %}© Copyright 2011 by <a href="http://domain.invalid/">yo
  </div>
</body>
</html>
```

# VERERBUNG

```
{% extends "base.html" %}

{% block title %}Index{% endblock %}
{% block head %}
    {{ parent() }}
    <style type="text/css">
        .important { color: #336699; }
    </style>
{% endblock %}
{% block content %}
    <h1>Index</h1>
    <p class="important">
        Welcome to my awesome homepage.
    </p>
{% endblock %}
```

# ÜBUNG 03

## LEBENS LAUF ALS TWIG TEMPLATE



# TYPOSCRIPT

Die Ausgabe von Inhalten im Frontend wird in erster Linie über die Metasprache TypoScript gesteuert. Genauer genommen ist TypoScript eine Konfigurationssprache: Was in TypoScript definiert ist, wird in ein systemweites PHP-Array geparkt. Dieses wiederum steuert, welche PHP-Funktionen beim Aufruf der Seite ausgeführt werden. Damit können Eigenschaften und Erweiterungen mit wenigen Handgriffen global für die gesamte Website verwaltet werden. TypoScript ist (bezogen auf die Syntax) objektorientiert.

```
page = PAGE
page.10 = TEXT
page.10.value = Hello, world!
page.10.wrap = <h2>|</h2>
```

# FLUID

## TypoScript + Extension in Typo3

```
<div class="box box-{boxColor}">  
  <h3>{boxHeader}</h3>  
  <div class="contains">  
    {boxContent}  
  </div>  
</div>
```

# 02. BASICS

# WAS IST EIN CMS

Ein Informationssystem, dass Inhalte organisiert und verwaltet

So gesehen sind auch Office Produkte, Email-Clients und z.B. Adserver (OpenX) Systeme, um Inhalte zu Pflegen

# INHALTE???

- Text
- Bilder
- Videos
- Audiodateien
- PDF-Dokumente
- ...

# CRUD

- Create
- Update
- Delete

MANIPULATION VON INHALTEN

# WIKIPEDIA SAGT

Ein Content-Management-System (kurz: CMS, deutsch „Inhaltsverwaltungssystem“) ist eine Software zur gemeinschaftlichen Erstellung, Bearbeitung und Organisation von Inhalten (Content) zumeist in Webseiten, aber auch in anderen Medienformen. Diese können aus Text- und Multimedia-Dokumenten bestehen. Ein Autor mit Zugriffsrechten kann ein solches System in vielen Fällen mit wenig Programmier- oder HTML-Kenntnissen bedienen, da die Mehrzahl der Systeme über eine grafische Benutzeroberfläche verfügen.

# AUFGABE EINES CMS

WIR DEFINIEREN  
PFLEGE VON WEBSEITEN



# MERKMALE EINES CMS

Was sind die Grundsätze eines CMS?

# 01 - STRIKTE TRENNUNG VON INHALT UND LAYOUT

Im Gegensatz zu statischen Webseiten werden die Inhalte (Texte, Bilder, Videoclips etc.) sowie die Formatvorlagen (Templates) in einem CMS separat gespeichert. Wenn eine entsprechende Webseite aufgerufen wird, wird diese dynamisch generiert, indem in ein entsprechendes Template die verschiedenen Inhalte geladen und angeordnet werden.

<http://www.csszengarden.com/>

Medienneutralität ist hier ebenfalls ein Grundsatz, so dass Ausgabe auch als PDF, RSS, ... möglich sein soll.

# 02 – KOMPONENTEN-MANAGEMENT

In Content Management Systemen werden die einzelnen von den Autoren gelieferten Inhalte mit Metadaten versehen und in einer Komponenten-Datenbank (content component database) abgelegt. Redakteure können nun aus diesen einzelnen Komponenten (Texte, Bilder, ...) Artikel zusammensetzen, die dann publiziert werden können. Oft wird die auch Asset-Management genannt.

## 02 - WORKFLOW-MANAGEMENT

Ein CMS bietet Mechanismen, die eine Definition und Kontrolle des Workflows (Ablauf der Arbeitsschritte) ermöglichen. So werden die von den Redakteuren zusammengesetzten Artikel vom Chefredakteur überprüft, bei Bedarf redigiert und von diesem für die Online-Publikation freigegeben. Die auf der Webseite publizierten Artikel bleiben für eine bestimmte Zeit online und werden nach Ablauf dieser Zeit im Archiv abgelegt.

# BRAUCH ICH EIN CMS?

- Deine Seite ist groß
- Wird oft geändert
- Hat mehrere Autoren

DANN JA!

# ABER

## EIN CMS BRAUCHT LIEBE

- Updates
- Sicherheitslöcher
- Geld für Webspace
- ...

# VORTEILE

- Direktes editieren von Inhalt, teilweise auch über das Frontend möglich
- Globales Update z.B. von Links
- Zeitgesteuerte Inhalte möglich
- Einfaches editieren
- Versionierung der Inhalte
- Rollback auf frühere Versionen
- Suchen
- Wiederverwendbarkeit von Inhalten

EFFIZIENTER, SCHNELLER,  
WENIGER ARBEIT UND  
DUPLIZIERUNG

# NACHTEILE

- Ausgaben für Webspace
- Aufwand / Kosten für die Implementierung
- Kenntnisse der eigenen Prozesse notwendig
- Planung im Vorfeld notwendig => Anforderungen



# 03. TYPEN

Kommerziell, open source, spezielle CMS (eingeschränkt auf bestimmte Typen)

# VOLLDYNAMISCHE SYSTEME

## AUCH ONLINE ODER WEB-CMS GENANNT

Volldynamische Systeme erzeugen angeforderte Dokumente bei jedem Aufruf dynamisch neu, das heißt, Vorlagen und Inhalte werden erst bei Abruf interpretiert bzw. zusammengeführt und ausgegeben. Vorteile: Die Seite ist immer „aktuell“; eine Personalisierung für den Surfer ist in der Regel sehr einfach oder sogar bereits vorhanden. Nachteile: Die Neuberechnung bei jeder Seitenauslieferung kann unter hoher Last (zum Beispiel hoher Besucherandrang) zu einer verzögerten Auslieferung der Seiten oder bei mangelhafter Ausstattung/Konfiguration an Rechenkapazität im Verhältnis zur Anzahl gleichzeitig

# STATISCHE SYSTEME

## OFFLINE CMS

Statische Systeme erzeugen die einzelnen Webseiten aus den Vorlagen und Inhalten als statisch abgelegte Datei im Dateisystem oder ggf. in einer Datenbank. Als Endprodukt erhält man somit Dokumente, die keinerlei Interpretation seitens einer Servertechnologie wie z. B. ASP, JSP oder PHP mehr benötigen und daher direkt durch den Webserver ausgegeben werden können, was sich in der Ausgabegeschwindigkeit zeigt. Dies hat den Vorteil, dass selbst einfachere Webhosting-Produkte als Basis ausreichend sein können. Nachteil kann sein, dass möglicherweise gewachsene Anforderungen durch größere Komplexität in Verbindung mit dem Wunsch nach sehr kurzen Aktualisierungszyklen ein solches System als ungeeignet entpuppen. Rein statische Systeme bilden den

# HYBRIDE SYSTEME

Hybride Systeme kombinieren die Vorteile der statischen und der volldynamischen Seitenerzeugung. Lediglich die Inhalte, die dynamisch aus einer Datenbank generiert werden müssen (z. B. News, Suchabfragen, personalisierte Inhalte oder Shopdaten), werden zur Laufzeit aus der Datenbank ausgelesen. Alle anderen Inhalte, die nicht laufend Änderungen unterzogen werden (wie etwa das Seitengerüst, die Navigation, aber auch bestimmte Texte und Bilder), liegen statisch vor.

# HALBSTATISCHE SYSTEME

Halbstatische Systeme generieren den Inhalt so, dass dieser statisch ist, aber gleichzeitig auch dynamisch, d. h., es werden alle Daten direkt in statisch generierten Dateien gespeichert, die dann bei Abruf sofort ausgegeben werden. Die dynamischen Inhalte werden dann generiert, wenn ein Code in der Programmiersprache in die Datei eingebunden wird oder einzelne Datensätze geändert beziehungsweise neu angelegt werden.

# ENTERPRISE CONTENT MANAGEMENT

Enterprise-Content-Management (ECM) umfasst die Methoden, Techniken und Werkzeuge zur Erfassung, Verwaltung, Speicherung, Bewahrung und Bereitstellung von Content und Dokumenten zur Unterstützung organisatorischer Prozesse im Unternehmen. ECM führt strukturierte, schwach strukturierte und unstrukturierte Informationen zusammen.

Die Bezeichnung Enterprise-Content-Management ist ein modernes Kunstwort, das Produkte, Lösungen, einen Markt und eine Branche beschreiben soll.

# ENTERPRISE CONTENT MANAGEMENT

Enterprise-Content-Management geht vom Ansatz aus, alle Informationen eines Unternehmens auf einer einheitlichen Plattform zur Nutzung intern, im Partnerverbund und extern bereitzustellen („Unified-Federated-Repository“, Data-/ Document-/ Content-Warehouse). ECM umfasst herkömmliche Informationstechniken wie Dokumentenmanagement, Knowledge Management (Wissensmanagement), Workflow-Management, Archivierung, etc. und integriert die Host- und Client/Server-Welt mit Portal- und anderen Internet-Techniken. Ziel von ECM ist, Daten- und Dokumentenredundanz zu vermeiden (jede Information existiert nur einmal), den Zugriff einheitlich zu regeln, unabhängig von Quelle und Nutzung beliebige Informationen bereitzustellen und als Dienst allen

# OPEN SOURCE CMS

## KOSTENLOS

Open Source bzw. quelloffen sind Werke, die ihren Quelltext offenlegen und einige weitere Bedingungen erfüllen. Im engeren Sinne handelt es sich dabei um Software, die unter einer von der Open Source Initiative (OSI) anerkannten Lizenz steht.

Open-Source-Software (OSS) ist nahezu deckungsgleich mit Freier Software. Der Unterschied liegt in den vertretenen Werten: Für Freie Software ist die Nutzerkontrolle über die Software sowie die Kooperation mit Anderen ein wichtiges soziales, politisches und ethisches Anliegen. Bei der OSI ist der vertretene Wert primär der praktische Nutzen und die Entwicklungsmethode.



# KOMMERZIELLE CMS

## KOSTEN GELD

- Einmalig
- Jahreslizenz
- SaaS

# SPEZIELLE CMS

## EHER EIN SPEZIELLER ANWENDUNGSBEREICH

- Mediawiki
- PHPBB
- Magento
- ...

# 04. BESTANDTEIL E

# WAS DENKT IHR?

Was gehört alles zu einem CMS?

# SYSTEMARCHITEKTUR

# DATEISYSTEM

- Beinhaltet die Dateien des CMS
- Meistens Speicherung von Medieninhalten wie zum Beispiel Bilder und Videos

# DATENBANK

- Speichert Daten wie zum Beispiel Artikel, Seiten und Einstellungen
- Speicherung von Metadaten zu Medieninhalten
- Sehr oft MySQL
- Oft auch andere Datenbanken (SQL & NoSQL)

# CMS - CORE

- Enthält die Kernfunktionalität des CMS
- Kombinieren von Daten und Layout
- Geschrieben in einer Programmiersprache (z.B. PHP)



# SEITEN

- Unterteilung in Titel, Text, Teaser und evtl. weitere Unterelemente
- Basis Inhalt eines CMS
- Mix aus Text, Links, Bildern, ...
- Wiederverwendbarkeit
- Speicherung vs. Publikation
- Haben Metadaten

# ARTIKEL / NEWS

- Kommt aus der Welt der Blogs
- Eher ein kurzer Text mit Überschrift
- Mix aus Text, Links, Bildern, ...
- Darstellung meist als Liste
- Haben Metadaten

# MENÜS

- Navigation innerhalb der Webseite
- Kann mehrere geben
- Meistens beliebig verlinkbar
- Personalisierbar

# ASSETS

- Dateien innerhalb eines CMS
- Oft via DMS gepflegt
- Haben einen Typ
- Haben Metadaten
- Lassen sich innerhalb von Artikeln benutzen

# USER, RECHTE UND ROLLEN

- Benutzer melden sich mit Benutzername und Passwort an
- Benutzer haben eine Rolle, z.B. Administrator, die zeigt welche Funktionen sie ausführen dürfen (Rechte)
- Benutzer haben Zugriff auf ihre erstellen Artefakte
- Anmeldung kann extern erfolgen, OAuth (z.B. Google)

# ADMIN-SCHNITTSTELLE

- Die Pflege erfolgt meist hier
- Konfiguration des Systems
- Getrennt von der eigentlichen Webseite
- Geschützte mit Passwort

# FRONTEND

- Ausgabe der Inhalte
- Herzstück des CMS
- Im angemeldeten Zustand manchmal andere Ansicht verfügbar
- Frontend-Editing

# STAGING-UMGEBUNG / PREVIEW

- Ansicht nicht finalisierter Inhalte
- Staging bezeichnet meist ein eigenes System z.B. bei Offline-CMS
- Preview meist wie "fertiger" Artikel, also im Frontend



# WORKFLOWS

- Definition von Prozessen, wer was wann im System tun darf
- Anlegen, Ändern, Speichern, Freigeben, Publizieren, ...
- Meistens mit dem Rollensystem verknüpft

# STATISTIKEN

- Anzeige von Zugriffen auf das CMS
- Diagramme
- Oft externe Dienste (z.B. Jetpack bei Wordpress)

# METAINFORMATIONEN

- Tags / Kategorien / Schlagwörter
- Veröffentlichungsdatum, Author
- Dateityp, Exif-Daten, MP3 / Video Metadaten
- Oft durchsuchbar
- Geodaten

# LINK-MANAGEMENT

- Links auf andere Artikel, Assets, ...
- Find von toten Links bei Entfernung des Ziels

# DESIGN

- Erscheinungsbild des CMS
- Begrifflichkeiten meist verschieden
- Themes / Layout:  
Komplettes Erscheinungsbild
- Page:  
Seite eines Themes / Layouts
- Template:  
Datei mit Platzhaltern für Inhalte
- Widgets:  
Elemente (z.B. Tag-Cloud) zum Einbau z.B. im Menü
- Partials:  
Wiederverwendbare Teile eines Templates

# PLUGINS / EXTENSIONS

- Erweiterung der Funktionalität
- Meist aus der Community
- CMS bieten eine API und Hooks
- Potentielle Sicherheitslücke

# MULTILANGUAGE

- CMS in nur einer Sprache
- CMS, bei welchem Artefakte übersetzt werden können => Gleiche Struktur
- CMS, die für jede Sprache eine andere Struktur bieten, z.B. über Mandanten

# MANDANTEN

- Verwalten mehrere Projekte in einem CMS
- Mehrere Seiten
- Mehrere Sprachen der selben Seite



# SECURITY

- Die Admin-Schnittstelle ist DER Angriffsvektor, da die URL bekannt ist
- Regelmäßige Updates
- Keine Standard-Accounts
- Vorsicht bei Plugins, nur offizielle
- Einmal Passwörter benutzen z.B. Google Authenticator

# SEO

- Grundanforderung an jedes CMS
- Schöne URLs über Rewrites
- Google Sitemap
- Interne Verlinkung muss stimmen
- Rich snippets für verschieden Inhaltstypen

# BARRIEFREIHEIT

- Die meisten CMS sind hier noch nicht so gut
- Semantic Web
- Responsiveness schwingt hier mit

# SUCHE

- Mittlerweile eines der zentralen Elemente jeder Webseite
- Suchen ist kompliziert
- Unscharfe Suchen
- Vorschlagssuche
- Meinten Sie...
- Apache Lucene bzw. abgeleitete Produkte im Einsatz
- Geosuche

# IMPORT

- RSS-Feeds
- Twitter
- Social-Media Streams / Comments
- Exporte (Backups)

# EXPORT

- RSS-Feeds
- Twitter
- Social-Media Streams / Comments
- Backups

# DRUCKVERSION / PDF

- Druckfreundliche Darstellung
- Grundfunktionalität der meisten CMS mittlerweile

04.

# MARKTÜBERSICHT



ES GIBT ÜBER 15.000 CMS  
WIE SOLL ICH DAS RICHTIGE  
FINDEN?

# VERGLEICHSSeiten

- <http://cmsmatrix.org/>

# 05. WORDPRESS

# ALLGEMEINES

WordPress ist eine freie Software zur Verwaltung der Inhalte einer Website (Texte und Bilder). Sie bietet sich besonders zum Aufbau und zur Pflege eines Weblogs an, da sie jeden Beitrag einer oder mehreren frei erstellbaren Kategorien zuzuweisen kann und dazu automatisch die entsprechenden Navigationselemente erzeugt.

Parallel kann WordPress auch hierarchische Seiten verwalten und gestattet den Einsatz als CMS.

# BEISPIELE

- Ich ;)
- Coca Cola Frankreich
- Sony Music
- Dallas Mavericks

# VORAUSSETZUNGEN

- PHP version 5.2.4 oder höher.
- MySQL version 5.0 oder höher.

# INSTALLATION

# AUSPROBIEREN

BITTE AUF FOLGENDES ACHTEN /  
AUSPROBIEREN:



# 06. CONTAO

# ALLGEMEINES

Contao (früher TYPOlight) ist ein freies Content-Management-System (CMS) für mittlere bis große Websites. Es ist ein Open-Source-Projekt und erschien erstmals 2006.

Als Datenbank wird MySQL verwendet. Das System lässt sich aus einem Pool von über 1500 Erweiterungen in seiner Funktionalität erweitern.

# BEISPIELE

- Kicken für den guten Zweck
- Contao Webseite
- BKW Energie AG
- Sparkasse Cham

# VORAUSSETZUNGEN

Contao benötigt einen Webserver wie Apache oder IIS mit PHP- und MySQL-Support. PHP muss mindestens in der Version 5.3.2 vorliegen und MySQL in der Version 5.0. Außerdem benötigen Sie die PHP-Erweiterungen "GDlib" (Bildbearbeitung), "DOM" (XML-Dateien) und "SOAP" (Extension Repository) sowie optional "mbstring" (internationale Zeichen) und "mcrypt" (Verschlüsselung). Contao wurde erfolgreich in allen modernen Browsern wie Firefox (ab Version 2) oder Internet Explorer (ab Version 8) getestet.

Der Conato Check

# INSTALLATION

Installation

Und jetzt ihr ;)

# AUSPROBIEREN

BITTE AUF FOLGENDES ACHTEN /  
AUSPROBIEREN:

- Plugins (Installation & Konfiguration)
- Themes (Installation, Anpassen via Admin, Anpassen via CSS)
- Benutzung von Modulen
- Statistik anzeigen
- Seiten anlegen und editieren
- Preview Funktion
- Mehrsprachigkeit
- Import und Export
- SEO
- Verschiedene Benutzerrollen

# 07. JOOMLA

# ALLGEMEINES

Joomla ([dʒu:mlə], Eigenschreibweise Joomla!) ist ein verbreitetes freies Content-Management-System (CMS) zur Erstellung von Webseiten.

Joomla steht unter der GNU General Public License. Es ist in PHP 5 geschrieben und verwendet MySQL als Datenbank. Zusammen mit WordPress, TYPO3, Contao und Drupal gehört es zu den bekanntesten und meistverwendeten Open-Source-Content-Management-Systemen. Historisch ist Joomla aus dem Open-Source-Projekt Mambo hervorgegangen.



# BEISPIELE

- Timesquare
- Peugeot
- Harvard University
- MTV Networks Quizilla

# VORAUSSETZUNGEN

- PHP (Magic Quotes GPC off)  $\geq 5.4$
- MySQL (InnoDB support required)  $\geq 5.1$
- Alternativ: MSSQL  $\geq 10.50.1600.1$  oder PostgreSQL  $\geq 8.3.18$
- Apache(with mod\_mysql, mod\_xml, and mod\_zlib)  $\geq 2.0$
- Alternativ: Nginx  $\geq 1.1$  oder Microsoft IIS  $\geq 7$

# INSTALLATION

Installation  
Und jetzt ihr ;)

# AUSPROBIEREN

BITTE AUF FOLGENDES ACHTEN /  
AUSPROBIEREN:

- Plugins (Installation & Konfiguration)
- Themes (Installation, Anpassen via Admin, Anpassen via CSS)
- Statistik anzeigen
- Seiten anlegen und editieren
- Preview Funktion
- Mehrsprachigkeit
- Import und Export
- SEO
- Verschiedene Benutzerrollen

# 08. DRUPAL

# ALLGEMEINES

Drupal ist ein Content-Management-System (CMS) und -Framework. Seine Hauptanwendung findet Drupal bei der Organisation von Websites, zurzeit (März 2014) bei 1,9 % aller Websites mit einem Marktanteil von 5,4 % bei CMS laut W3Techs. Ursprünglich wurde es vom belgischen Informatiker Dries Buytaert konzipiert. Drupal ist freie Software und steht unter der GNU General Public License. Es ist in PHP geschrieben und verwendet MySQL/MariaDB (empfohlen), PostgreSQL (unterstützt), SQLite (ab 7.x), Oracle (in Entwicklung) oder MSSQL (in Entwicklung) als Datenbanksystem.

# BEISPIELE

- [whitehouse.gov](http://whitehouse.gov), die Website des Weißen Hauses
- [economist.com](http://economist.com), die Website des Economist
- [amnesty.org](http://amnesty.org), die Website von Amnesty International
- [WFP.org](http://WFP.org), die Website des Welternährungsprogramms der Vereinten Nationen

# VORAUSSETZUNGEN

- PHP  $\geq 5.4$
- MySQL  $\geq 5.0.15$  mit PDO
- Alternativ: PostgreSQL  $\geq 8.3$  oder SQLite  $\geq 3.3.7$



# INSTALLATION

Installation

Und jetzt ihr ;)

# AUSPROBIEREN

BITTE AUF FOLGENDES ACHTEN /  
AUSPROBIEREN:

- Plugins (Installation & Konfiguration)
- Themes (Installation, Anpassen via Admin, Anpassen via CSS)
- CMS vs. Blog Funktionalität
- Statistik anzeigen
- Seiten anlegen und editieren
- Preview Funktion
- Mehrsprachigkeit
- Import und Export
- SEO
- Verschiedene Benutzerrollen

# 09. TYPO3

# ALLGEMEINES

TYPO3 ist ein freies Content-Management-Framework für Websites/ Internetseiten, welches seit Oktober 2012 offiziell unter dem Namen TYPO3 CMS angeboten wird. Ursprünglich wurde TYPO3 von Kasper Skårhøj entwickelt. TYPO3 basiert auf der Skriptsprache PHP. Als Datenbank kann MySQL oder MariaDB, aber auch PostgreSQL oder Oracle eingesetzt werden.

# BEISPIELE

- Website des Landes Sachsen-Anhalt
- Offizielle Website von Bündnis 90/Die Grünen
- Homepage der Technischen Universität Berlin
- Offizielle Webseite des DFB

# VORAUSSETZUNGEN

- PHP 5.3.7-5.5.x
- MySQL 5.1.x-5.5.x

# INSTALLATION

1. Unzip / untar your downloaded introduction package in the root folder of your web server
2. Make sure that the web server user has write permissions to the folders fileadmin/,typo3conf/,typo3temp/ and uploads/
3. Go to <http://your-site.example.org/> with a browser, you should get redirected to the TYPO3 installer
4. If this does not happen, create a file called ENABLE\_INSTALL\_TOOL in the folder typo3conf/
5. Follow the installer steps

Und jetzt ihr ;)

# AUSPROBIEREN

BITTE AUF FOLGENDES ACHTEN /  
AUSPROBIEREN:

- Plugins (Installation & Konfiguration)
- Themes (Installation, Anpassen via Admin, Anpassen via CSS)
- Statistik anzeigen
- Seiten anlegen und editieren
- Preview Funktion
- Mehrsprachigkeit
- Zeitliche Steuerung von Artikeln
- Import und Export
- SEO
- Verschiedene Benutzerrollen



# 10. ÜBUNG

Vorschlag eigene Bewerbungsseite mit CV, Links zu Projekten, ...

In einem CMS eurer Wahl (Wenn es geht nicht alle das gleiche)

Am Ende kurz 10 - 15 min Grundfunktionalitäten vorstellen und kritische Würdigung

# 11. Q&A

FRAGEN?

# 12. LINKS



# THE END

## VIELEN DANK!

Made with [reveal.js](#)(with Html and JS)