



Formulare mit Drupal

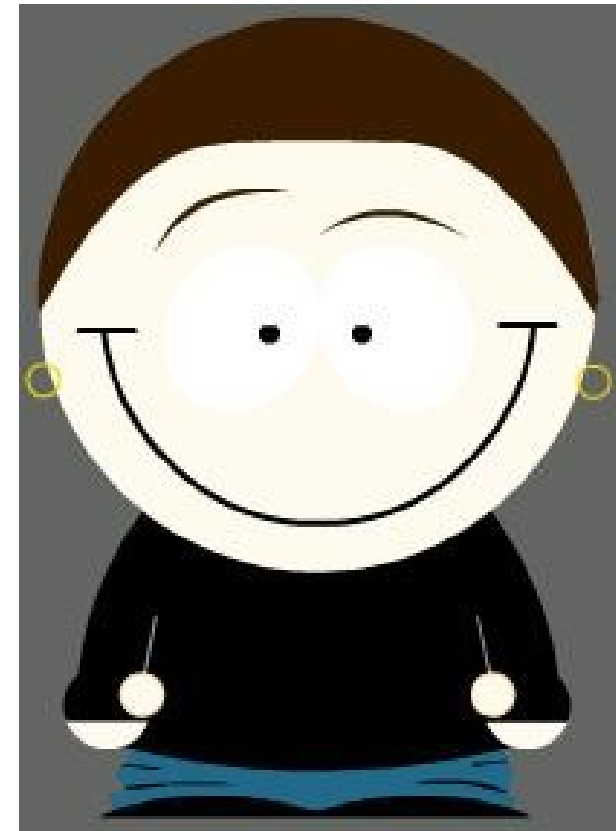
Drupalcamp, Köln 2009

Jutta Horstmann

data in transit
I T - B E R A T U N G

Jutta Horstmann (Dipl. Inform., Dipl. Pol.)
IT-Beraterin (“data in transit”)

- Schwerpunkt: Freie Software
- Content Management Systeme
- Datenbankmanagement-Systeme
- Analyse, Beratung, Konzeption
- Implementation, Web-Programmierung



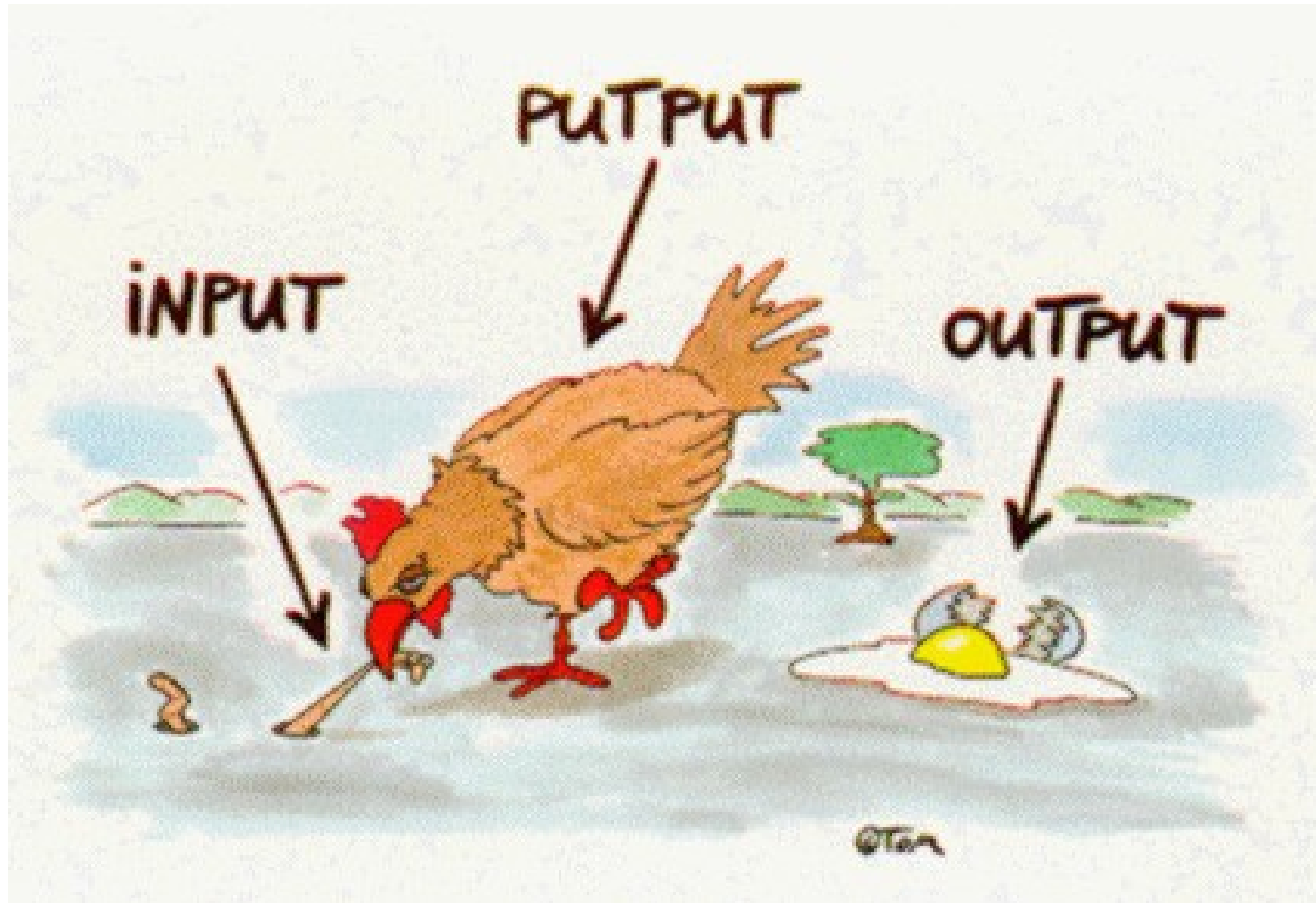
data in transit

- IT-Beratung
- gegründet 2005 in Berlin
- 2006 Umzug nach Bonn
- www.dataintransit.com
- Letzte Drupal-Projekte:
 - select.postbank.de (Postbank AG)
 - www.lisa-freundeskreis.de (Burda)

data in transit
I T - B E R A T U N G

**Wozu Formulare?
Ein Beispielprojekt
Formulare in Drupal
CCK vs. Webforms vs.
Programmierung
Hot New Stuff**


Wozu Formulare?



©TOM (Jochen Enterprises)

Ein Beispielprojekt

- Der Kunde ist ein bundesweit operierendes Unternehmen.
- Auf Stellenangebote des Unternehmens soll man sich online bewerben können.
- Es gibt 3 Arten von Bewerbungsbögen: Trainee, Berufseinsteiger, Führungsposition. Diese werden pro Ausschreibung vom HR-Team inhaltlich erweitert und angepasst. Es gibt veränderbare und nicht veränderbare Felder.
- Abgesendete Daten werden per E-Mail verschickt, in Reports zusammengefasst und exportiert.



Anforderungen: PUTPUT
Anforderungen: INPUT
Anforderungen: OUTPUT



Anforderungen PUTPUT: Das Formular und seine Erstellung

PUTPUT: Das Formular und seine Erstellung

- Redakteure erstellen und bearbeiten das Formular
- Benutzerfreundlicher Formulareditor
- Anfang und Ende der Veröffentlichung des Formulars terminieren

Stellenangebote bearbeiten

- Neue Bewerbungsbogen-Vorlage erstellen
- Neue Ausschreibung für Führungsposition
- Neue Ausschreibung für Berufseinsteiger
- Neue Ausschreibung für Trainee

Neue Ausschreibung erstellen auf der Basis von Bewerberbogen Trainee

Anzeigen Bearbeiten Ergebnisse Versionen Beiträge **Duplizieren**

Dieses Duplikat wird erst in der Datenbank gespeichert, wenn der Beitrag gespeichert wird.

- ▷ Menüeinstellungen
- ▷ Formular-Einstellungen
- ▷ Zugriffsberechtigungen

Name	Typ	Wert	Pflichtfeld	E-Mail	Operationen
+ Angaben zu Ihrer Person	fieldset	-			Bearbeiten
+ Nachname	textfield	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Bearbeiten
+ Vorname	textfield	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Bearbeiten
+ Alter	textfield		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Add"/>

Start und Ende der Ausschreibung einrichten

Start der Ausschreibung:

Format: 15.01.2009 19:42:10. Leer lassen, wenn Ausschreibung sofort starten soll.

Ende der Ausschreibung:

Format: 15.01.2009 19:42:10. Leer lassen, wenn Ausschreibung nicht automatisch beendet werden soll.



Anforderungen INPUT: Strukturierte Dateneingabe

INPUT: Strukturierte Dateneingabe: Gruppierung von Feldern

- 2 Felder im Formular,
1 Feld in der DB
z.B. Vorwahl + Anschluß
- Feldgruppe als
“multiple” Feld

Beschäftigt bei/als:

+

Dauer der Beschäftigung (MM.JJJJ - MM.JJJJ):

Arbeitgeber/Dienststelle:

Tätigkeit:

Dauer der Beschäftigung (MM.JJJJ - MM.JJJJ):

+

Arbeitgeber/Dienststelle:

Tätigkeit:

Weiteren Eintrag hinzufügen

INPUT: Strukturierte Dateneingabe: Abhängigkeiten zwischen Feldern

- Wenn Option A gewählt, dann aktiviere Feld X
- Wenn Feld A ausgefüllt, dann sind Feld B und Feld C Pflichtfelder

Bitte rufen Sie mich zurück	<input checked="" type="radio"/> schnellstmöglich*
	<input type="radio"/> an dem folgenden Termin
Datum (TT.MM.JJJJ)	<input type="text"/>
Tageszeit	vormittags <input type="text"/>

- Bitte geben Sie auch die Note des Abschlusses der Berufsausbildung an.
- Bitte geben Sie auch den Zeitpunkt des Abschlusses der Berufsausbildung an.

► Schulausbildung

Abgeschlossene Berufsausbildung

Berufsbezeichnung:

(max. 40 Zeichen; Beispiele: Fachinformatiker/in, Verwaltungsfachangestellte/r)

Note:

Zeitpunkt des Abschlusses der Berufsausbildung (MM.JJJJ):

INPUT: Strukturierte Dateneingabe: Mehrseitige Formulare

- Ein Formular wird auf mehrere Seiten verteilt
- Navigation mit “Weiter”, “Zurück”
- Fehlermeldungen auf zugehöriger Seite
- Unabhängig von Javascript



The screenshot shows a web form with four tabs: 'Persönliche Angaben', 'Ausbildung / Studium', 'Berufserfahrung', and 'Sonstiges'. The 'Berufserfahrung' tab is currently selected and highlighted in blue. Below the tabs, there is a label 'Aktuelle Berufsbezeichnung:' followed by a dropdown menu showing '- Keine -'. At the bottom of the form, there are two buttons: '< Zurück' on the left and 'Weiter >' on the right.

INPUT: Strukturierte Dateneingabe: Anzeige nach Absenden

Wenn Formular vollständig ausgefüllt wurde:

- Bestätigungsnachricht
oder
- Weiterleitung auf Bestätigungsseite

Bestätigung

Vielen Dank. Ihre Nachricht wurde mit den folgenden Inhalten erfolgreich an uns übermittelt:

Nachricht:	Nur ein Test.
Name:	Prof. Jutta Horstmann
Telefonnummer:	123 / 456
Termin:	10.04.1999, nachmittags

Nach oben ↗

INPUT: Strukturierte Dateneingabe: Weitere Anforderungen

- Zwischenspeichern von Eingaben
- Übernahme von Daten aus anderen bereits gesendeten Formularen als Default-Wert



Anforderungen OUTPUT: Die Weiterverarbeitung der Daten

OUTPUT: Die Weiterverarbeitung der Daten

Anforderungen:

- Versand per E-Mail
- Speicherung in der Datenbank
- Datenexport
- Übersicht über alle abgeschickten
Formulareingaben (zu einem Formular / zu allen
vorhandenen Formularen)
- Reports, Statistiken



Drupal!

Formulare in Drupal

- Modul: Webform
- Modul: CCK
- Eigenbau mit Forms API

=> Einsatz je nach Anforderungen im Projekt

(Für die Beispiele: Drupal 6.9, CCK 6.x-2.1, Webform 6.x-2.4)

CCK/Webform vs. Programmierung

CCK und Webform bieten Formulareditor

- Keine Programmierkenntnisse notwendig
- Formulare können von Redakteuren gebaut werden
- Formulare können sehr schnell gebaut werden

Name	Typ	Wert	Pflichtfeld	E-Mail	Operationen
+ Angaben zu Ihrer Person	fieldset	-			Bearbeiten
+ Nachname	textfield	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Bearbeiten
+ Vorname	textfield	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Bearbeiten
+ Alter	textfield		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Add"/>

Webform

Feld-Bezeichner	Name	Typ
+ Titel	Node module form.	
+ Menüeinstellungen	Menu module form.	
+ Geburtsdatum (TT.MM.JJJJ)	field_dat_geb	Datetime
+ Berufsausbildung	field_beruf	Text
Add		
+ Neues Feld		
<input type="text"/>	field_ <input type="text"/>	<input type="button" value="- Feldtyp auswählen -"/>
Feld-Bezeichner	Feldname für die Datenbank	Type of data to store.

CCK

CCK/Webform vs. Programmierung

- **Eigenbau mit Forms API**

→ Volle Flexibilität

```
function my_module_menu() {
  $items = array();
  $items['my_module/form'] = array(
    'title' => t('My form'),
    'page callback' => 'my_module_form',
    'access arguments' => array('access content'),
    'description' => t('My form'),
    'type' => MENU_CALLBACK,
  );
  return $items;
}

function my_module_form() {
  return drupal_get_form('my_module_my_form');
}

function my_module_my_form($form_state) {
  $form['name'] = array(
    '#type' => 'textfield',
    '#title' => t('Name'),
  );

  $form['submit'] = array(
    '#type' => 'submit',
    '#value' => 'Submit',
  );
  return $form;
}
```

<http://drupal.org/node/262422>

CCK/Webform vs. Programmierung

Best of both worlds:

Mit CCK/Webform
erstellte Formulare
können per FAPI und /
oder Themeing noch
weiter angepasst werden

```
function helpers_form_alter(&$form, $form_state, $form_id){
  if ($form_id == 'webform_client_form_3') {
    $form['#validate'][]= 'helpers_customforms_validate';
  }
}

function helpers_customforms_validate($form, &$form_state) {
  if ($form_state['values']['details']['page_num'] == 1) {
    $email = $form_state['values']['submitted']
      ['kontaktdaten']['email'];

    if ($email) {
      if (!eregi("^[_a-z0-9-]+(\\.[_a-z0-9-]+)*@[a-z0-9-]+(\\.[a-z0-9-]+)*(\\.[a-z]*)$", $email)){
        form_set_error('submitted[kontaktdaten][email',
          t('%value ist keine valide E-Mail-Adresse.',
            array('%value' => $email)));
      }
    }
  }
}
```



CCK vs. Webform

Webform vs. CCK

Webform

- Das Formular ist ein Node
- Die Eingaben in das Formular sind keine Nodes



Webform vs. CCK

CCK

- Das Formular ist ein Inhaltstyp
- Die Eingaben in das Formular sind Nodes

Inhaltstypen [Alle anzeigen](#) [Inhaltstyp hinzufügen](#) [Fields](#) [Exportieren](#) [Importieren](#)

Unten folgt eine Liste aller Inhaltstypen der Website. Alle auf der Website existierenden Beiträge sind Instanzen eines dieser Inhaltstypen.

Name	Typ	Beschreibung	Operationen
Bewerbungsbogen Einsteiger	form_einsteiger		Bearbeiten Felder bearbeiten Löschen
Bewerbungsbogen Trainee	form_trainee		Bearbeiten Felder bearbeiten Löschen

Inhalt [\[ausführlichere Hilfe...\]](#)

Nur Einträge anzeigen mit

Typ gleich **Bewerbungsbogen Trainee**

und wobei Status ist

Sprache

Aktualisierungs-Einstellungen

<input type="checkbox"/>	Titel	Typ	Autor	Status	Sprache	Operationen
<input type="checkbox"/>	Bewerbungsbogen Trainee - Bewerbung von jh@dataintransit.com, 14. Januar 2009 - 16:33	Bewerbungsbogen Trainee	admin	Veröffentlicht	Deutsch	Bearbeiten

Webform vs. CCK - Konsequenzen

Webform:

- Eingaben in Formulare stehen nicht für Views zur Verfügung

CCK:

- Viele Formulare = viele Inhaltstypen



Webform vs. CCK: Formular erstellen & verwalten (PUTPUT)

Formular erstellen/verwalten mit Webform

- Inhalt erstellen → Webform → Titel “Trainee Form”
oder
- Vorhandenes Formular “Trainee Form” auswählen
→ Klonen (*node_clone*)
- Felder anlegen / editieren: Formulareditor
- Veröffentlichung terminieren (*scheduler*)
- **Nicht möglich: Berechtigung pro Feld**

Formular erstellen/verwalten mit CCK

- Verwalten → Inhaltsverwaltung → Inhaltstypen
 - Inhaltstyp hinzufügen: “form_trainee”
oder
 - “Exportieren” → Inhaltstyp “form_trainee” → Code kopieren →
“Importieren” → Code einfügen & editieren → Absenden

Umständlich und fehleranfällig.

- Felder anlegen / editieren: Formulareditor
- **Nicht möglich:**
 - Veröffentlichung terminieren
 - Berechtigung pro Feld




Webform vs. CCK: Strukturierte Dateneingabe (INPUT)

Strukturierte Dateneingabe mit Webform

- Mehrseitige Formulare (per “pagebreak”)
- Bestätigungsseite nach Absenden konfigurierbar
- Nicht (bzw. nur per Code):
 - Gruppierung von Feldern
 - Abhängigkeiten zwischen Feldern
 - Zwischenspeichern von Eingaben
 - Übernahme von gespeicherten Daten aus anderen Formularen als Vorgabe für Felder

Strukturierte Dateneingabe mit CCK

- Zwischenspeichern : “Speichern” ohne “Veröffentlichen”
- Gruppierung von Feldern: *flexifield*
- Bestätigungsseite nach Absenden: *trigger + action*
- Mehrseitige Formulare: **Eingeschränkt:**
 - *pageroute*: **Inhaltstyp pro Seite.**
 - *cck_fieldgroup_tabs*: **Nur mit aktiviertem Javascript.**
 - *multistep*: **Nur mit aktiviertem Javascript. Nicht Drupal 6.**
- Abh. Felder: *conditional_fields* (nicht: abh. Pflichtfelder)
- **Übernahme von Daten als Default: Nur mit Code**



Webform vs. CCK: Weiterverarbeitung der Daten (OUTPUT)

Weiterverarbeitung der Daten mit Webform

- Versand per E-Mail – ist eingebaut
- Datenexport: CSV, XLS – ist eingebaut
- Reports, Statistiken – eingebaut (**eingeschränkt**)
- Speicherung in der DB: `webform`,
`webform_component`, `webform_roles`,
`webform_submissions`, `webform_submitted_data`
- Übersicht über alle abgeschickten
Formulareingaben (nur: zu einem Formular, **nicht:**
zu allen vorhandenen Formularen)

Weiterverarbeitung der Daten mit CCK

- Versand per E-Mail: *trigger + action*
- Speicherung in der DB:
`content_type_form_trainee`
- Export: *views_bonus_pack*: CSV, DOC, TXT
- Übersicht über alle abgeschickten
Formulareingaben: *views*
- Reports, Statistiken: *views*

Fazit: Webform vs. CCK

- Flexibilität in INPUT und OUTPUT:
→ CCK
(Ausnahme: mehrseitiges Formular)
- Benutzerfreundlichkeit im Anlegen und Bearbeiten von Formularen (PUTPUT), schnelle Export-Lösung:
→ Webform



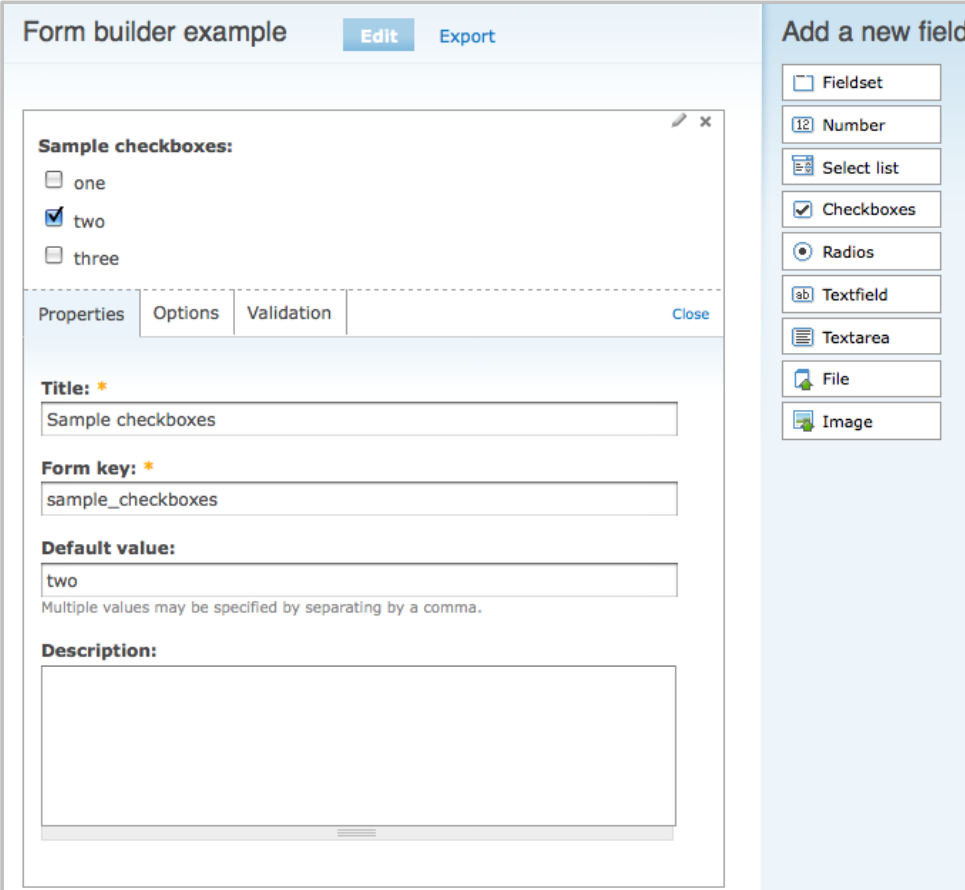
Hot New Stuff & Ausblick

Front End: Form Builder

- Lullabot
- Formulare bauen per Drag&Drop (AJAX)
- Ersatz für alle Drupal Formulareditoren
- Module fügen Formularlogik hinzu

http://drupal.org/project/form_builder

<http://quicksketch.org/demos/form-builder-example>



The screenshot displays a web-based form builder interface. The main window is titled "Form builder example" and contains a preview of a form with three checkboxes labeled "one", "two", and "three". The "two" checkbox is checked. Below the preview are tabs for "Properties", "Options", and "Validation". The "Properties" tab is active, showing fields for "Title" (with a red asterisk), "Form key" (with a red asterisk), "Default value", and "Description". The "Title" field contains "Sample checkboxes", the "Form key" field contains "sample_checkboxes", and the "Default value" field contains "two". A note below the "Default value" field states: "Multiple values may be specified by separating by a comma." To the right of the main window is a sidebar titled "Add a new field" with a list of field types: Fieldset, Number, Select list, Checkboxes, Radios, Textfield, Textarea, File, and Image. The "Checkboxes" option is selected with a checkmark.

Back End: Drupal 7 Fields API

- Felder als 1st level citizens im Core
- CCK und Webforms können auf die selben Felder zugreifen

=> zum Beispiel:

- Webform-Logik (benutzerfreundliches Anlegen von Formularen, einfacher Export)
verheiraten mit
- flexiblen CCK-Feldern

Fragen???



data in transit <http://www.dataintransit.com>

Kontakt jh@dataintransit.com