

PHP Fortgeschrittenen Kurs Sommercampus 2006

09.08.2006

Christoph Hermann, Dominik Benz
{hermann,dbenz}@informatik.uni-freiburg.de

php.ini

Konfigurations-Datei für PHP: php.ini

- ▶ befindet sich oft in `/etc/php5/apache2/php.ini`; Pfad wird auch in `phpinfo()` angezeigt
- ▶ Einstellungen können in dieser Datei vorgenommen werden oder dynamisch zur Laufzeit per Funktion `ini_set` (nicht alle, siehe <http://www.php.net/manual/en/ini.phpini.list>)
- ▶ Veränderungen an php.ini werden erst nach Neustart des Webservers wirksam

`register_globals <boolean>` schaltet an / aus, ob per POST/GET übergebene Variablen automatisch global verfügbar sind. Sollte ausgeschaltet sein!

`post_max_size <integer>` definiert die maximale Datenmenge in Bytes, die per POST hochgeladen werden kann (auch Dateien). Muss für grosse Dateien grösser sein als `upload_max_filesize <integer>`

php.ini - 2

`register_globals` <boolean> schaltet an / aus, ob per POST/GET übergebene Variablen automatisch global verfügbar sind. Sollte ausgeschaltet sein!

`memory_limit` <integer> legt den maximalen Speicherverbrauch eines Skripts fest. Der Standard von 8 MB muss für einige Anwendungen, besonders CMS, hochgesetzt werden

`include_path` <string> gibt eine Liste von Pfaden an, in denen nach einzubindenden Dateien gesucht wird. Praktisch für Libraries wie PEAR

`file_uploads` <boolean> erlaubt / verbietet das Hochladen von Dateien

`upload_tmp_dir` <string> gibt den Pfad an, in dem hochgeladene Dateien zwischengespeichert werden

php.ini - 3

`session.save_path <string>` legt den Pfad fest, in dem Session-Daten gespeichert werden

`session.use_trans_sid <boolean>` erlaubt / verbietet die Übergabe von Session IDs per URL (Sicherheitsrisiko!)

- ▶ weitere Direktiven: in den jeweiligen Kapiteln im Handbuch

Pear Quickform

- ▶ Einfache Formulargenerierung
- ▶ Vorbelegung der Felder möglich
- ▶ Validierung der Formularfelder (Server- und Clientseitig)
- ▶ Komplexe Regelerstellung möglich
- ▶ Hierarchical Selects möglich

Pear Quickform Beispiel



```
<?php
require_once "HTML/QuickForm.php";
$user = array("lastname"=>"Mouse", "sal_id"=>4, "user_id"=>7);
$salutations = array("0"=>"Mr", "1"=>"Miss", ..., "4"=>"Sir");
$form = new HTML_QuickForm('frmTest', 'get');
$form->addElement('header', 'MyHeader', 'Edit_username');
$form->addElement('hidden', 'user_id');
$form->addElement('select', 'sal_id',
    'Address_me_as:', $salutations);
$form->addElement('text', 'lastname', 'Last_name:');
$form->addRule('lastname',
    'Your_lastname_is_required', 'required');
$form->addElement('reset', 'btnClear', 'Clear');
$form->addElement('submit', 'btnSubmit', 'Submit');
if ($form->validate()) {
    // Formular ist validiert, daten verarbeiten
    $form->freeze(); $form->process('process_data', false);
}
$form->setDefaults($user);
$form->display();
```

?>

Pear Quickform - Datenauswertung

```
<?php
function process_data ($values) {
    echo "<pre>";
    foreach ($values as $key=>$value) {
        echo $key."=".$value."<br>";
    }
    echo "</pre>";
}
?>
```

Smarty

- ▶ Smarty ist eine Template-Engine für PHP.
- ▶ Erlaubt es die einfache Trennung von Applikations-Logik und Design/Ausgabe.
- ▶ Sehr schnell im Gegensatz zu anderen Template-Engines
- ▶ Unbegrenzte Verschachtelung von 'section', 'if' und anderen Blöcken.
- ▶ Konfigurierbare Syntax für Template-Tags: `{}`, `{{}`}, `<`, `,`, `>`, `!` ...

Einfaches Smarty Beispiel

```
{* Smarty *}
<table>
<tr>
<td class="text">
{section name=count loop=$news}
    {if $smarty.section.count.index % 5 == 0}
        </td><td class="text">
    {/if}
    <a href="news.php?id={$news[count].id}">
    <b>{$news[count].thema}</b>
    </a><br />
{/section}
</td>
</tr>
</table>
```

Smarty Beispiel

▶ Eingebauter Caching-Support!

```
<?php
require('Smarty.class.php');
$smarty = new Smarty(); $smarty->caching = true;
if(!$smarty->is_cached('index.tpl')) {
    // kein Cache gefunden, also Variablen zuweisen
    $contents = get_database_contents();
    $smarty->assign($contents);
}
$smarty->display('index.tpl');
?>
```

▶ Per-User Caching z.B. durch cacheids möglich:

```
<?php
require('Smarty.class.php');
$smarty = new Smarty(); $smarty->caching = true;
// ... cacheid erzeugen
$smarty->display('index.tpl', $cacheid);
?>
```

Pear Quickform Controller (Multipage Forms)

- ▶ Vorteile von Quickform
- ▶ Mehrseitige Formulare
- ▶ Vor- / Zurückblättern
- ▶ Integration mit Smarty auch möglich: `smarty-dynamic.tpl`
Siehe QuickForm docs: `docs/renderers/templates`
- ▶ Code rel. komplex (lang), daher hier kein Beispiel



phpDocumentor

- ▶ phpDoc(umentor) ist ein Dokumentationsstandard der JavaDoc sehr ähnelt.
- ▶ Installation 1/2
 - Download&unzip: <http://pear.php.net/package/PhpDocumentor/download>
 - PHP konfigurieren: \$ setup lang/php5
 - \$ which php
/usr/local/lang/php/5.1.4/bin/php

default

[class tree: default] [index: default] [all elements]

<p>Packages:</p> <ul style="list-style-type: none"> default HTML_QuickForm QuickForm <p>Files:</p> <ul style="list-style-type: none"> advcheckbox.php Array.php ArraySmarty.php Autocomplete.php button.php Callback.php checkbox.php checkboxset.php date.php Default.php element.php elements.php Email.php file.php filters.php FileDynamic_example.php FileStatic_example.php formrule.php group.php groups.php header.php hidden.php hiddenselect.php hierselect.php html.php image.php input.php ITDynamic.php ITDynamic_example.php ITDynamic_example2.php ITStatic.php ITStatic_example.php link.php multiple-labels.php Object.php ObjectKey.php password.php QuickForm.php 	<p>Class: HTML_QuickForm_Renderer_ArraySmarty</p> <p>Source Location: /HTML_QuickForm-3.2.6/QuickForm/Renderer/ArraySmarty.php</p> <p>Class Overview</p> <pre>HTML_QuickForm_Renderer --HTML_QuickForm_Renderer_Array --HTML_QuickForm_Renderer_ArraySmarty</pre> <p>A static renderer for HTML_QuickForm, makes an array of form content useful for a Smarty template</p> <p>Author(s):</p> <ul style="list-style-type: none"> • Aleksey Borozov <boroz_off@cs.msu.su> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Inherited Variables</th> <th style="width: 50%;">Inherited Methods</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> <p>Class: HTML_QuickForm_Renderer_Array</p> <ul style="list-style-type: none"> HTML_QuickForm_Renderer_Array::\$staticLabels HTML_QuickForm_Renderer_Array::\$ary HTML_QuickForm_Renderer_Array::\$collectHidden HTML_QuickForm_Renderer_Array::\$currentGroup HTML_QuickForm_Renderer_Array::\$currentSection HTML_QuickForm_Renderer_Array::\$elementStyle HTML_QuickForm_Renderer_Array::\$elementStyle HTML_QuickForm_Renderer_Array::\$sectionCount </td> <td style="vertical-align: top;"> <p>Class: HTML_QuickForm_Renderer_Array</p> <ul style="list-style-type: none"> HTML_QuickForm_Renderer_Array:HTML_QuickForm_Renderer_Array() HTML_QuickForm_Renderer_Array:Constructor HTML_QuickForm_Renderer_Array:flushGroup() HTML_QuickForm_Renderer_Array:renderElement() HTML_QuickForm_Renderer_Array:renderHeader() HTML_QuickForm_Renderer_Array:renderHidden() HTML_QuickForm_Renderer_Array:renderSectionStyle() HTML_QuickForm_Renderer_Array:starForm() HTML_QuickForm_Renderer_Array:starFormGroup() HTML_QuickForm_Renderer_Array:toArray() HTML_QuickForm_Renderer_Array:toArray() HTML_QuickForm_Renderer_Array:toArray() </td> </tr> </tbody> </table>	Inherited Variables	Inherited Methods	<p>Class: HTML_QuickForm_Renderer_Array</p> <ul style="list-style-type: none"> HTML_QuickForm_Renderer_Array::\$staticLabels HTML_QuickForm_Renderer_Array::\$ary HTML_QuickForm_Renderer_Array::\$collectHidden HTML_QuickForm_Renderer_Array::\$currentGroup HTML_QuickForm_Renderer_Array::\$currentSection HTML_QuickForm_Renderer_Array::\$elementStyle HTML_QuickForm_Renderer_Array::\$elementStyle HTML_QuickForm_Renderer_Array::\$sectionCount 	<p>Class: HTML_QuickForm_Renderer_Array</p> <ul style="list-style-type: none"> HTML_QuickForm_Renderer_Array:HTML_QuickForm_Renderer_Array() HTML_QuickForm_Renderer_Array:Constructor HTML_QuickForm_Renderer_Array:flushGroup() HTML_QuickForm_Renderer_Array:renderElement() HTML_QuickForm_Renderer_Array:renderHeader() HTML_QuickForm_Renderer_Array:renderHidden() HTML_QuickForm_Renderer_Array:renderSectionStyle() HTML_QuickForm_Renderer_Array:starForm() HTML_QuickForm_Renderer_Array:starFormGroup() HTML_QuickForm_Renderer_Array:toArray() HTML_QuickForm_Renderer_Array:toArray() HTML_QuickForm_Renderer_Array:toArray()
Inherited Variables	Inherited Methods				
<p>Class: HTML_QuickForm_Renderer_Array</p> <ul style="list-style-type: none"> HTML_QuickForm_Renderer_Array::\$staticLabels HTML_QuickForm_Renderer_Array::\$ary HTML_QuickForm_Renderer_Array::\$collectHidden HTML_QuickForm_Renderer_Array::\$currentGroup HTML_QuickForm_Renderer_Array::\$currentSection HTML_QuickForm_Renderer_Array::\$elementStyle HTML_QuickForm_Renderer_Array::\$elementStyle HTML_QuickForm_Renderer_Array::\$sectionCount 	<p>Class: HTML_QuickForm_Renderer_Array</p> <ul style="list-style-type: none"> HTML_QuickForm_Renderer_Array:HTML_QuickForm_Renderer_Array() HTML_QuickForm_Renderer_Array:Constructor HTML_QuickForm_Renderer_Array:flushGroup() HTML_QuickForm_Renderer_Array:renderElement() HTML_QuickForm_Renderer_Array:renderHeader() HTML_QuickForm_Renderer_Array:renderHidden() HTML_QuickForm_Renderer_Array:renderSectionStyle() HTML_QuickForm_Renderer_Array:starForm() HTML_QuickForm_Renderer_Array:starFormGroup() HTML_QuickForm_Renderer_Array:toArray() HTML_QuickForm_Renderer_Array:toArray() HTML_QuickForm_Renderer_Array:toArray() 				

```

239.
240.
241.
242.
243.
244.
245.
246.
247.
248.
249.
250.
251.
252.
253.
254.
255.
256.
257.
258.
259.
260.
261.
262.
263.
264.
265.
266.
267.
268.
269.
270.
271.
272.
273.
274.
275.
276.
277.
278.
279.
280.
281.
282.
283.
284.
285.
286.
287.
/**
 * Creates an array representing an element
 *
 * @access private
 * @param object An HTML_QuickForm_element object
 * @param bool Whether an element is required
 * @param string Error associated with the element
 * @return array
 */
function toArray($element, $required, $error)
{
    $ret = array(
        'name' => $element->getName(),
        'value' => $element->getValue(),
        'type' => $element->getType(),
        'frozen' => $element->isFrozen(),
        'required' => $required,
        'error' => $error
    );
    // render label(s)
    $labels = $element->getLabel();
    if (is_array($labels) && $this->staticLabels) {
        foreach($labels as $key => $label) {
            $key = is_int($key)? $key + 1: $key;
            if (1 == $key) {
                $ret['label'] = $label;
            } else {
                $ret['label' . $key] = $label;
            }
        }
    } else {
        $ret['label'] = $labels;
    }
}
// set the style for the element
if (isset($this->elementStyles[$ret['name']])) {
    $ret['style'] = $this->elementStyles[$ret['name']];
}
if ($group == $ret['type']) {
    $ret['_separator'] = $element->_separator;
    $ret['elements'] = array();
} else {
    $ret['html'] = $element->toHtml();
}
return $ret;
}
    
```

phpDocumentor

- ▶ Installation 2/2
- ▶ Eclipse konfigurieren: `$ setup devel/eclipse`
- ▶ Integration in (PHP)Eclipse:
 - Eclipse → Run → External Tools → External Tools
 - Program → New
 - Name: PHPDocumentor
 - Location: `/usr/local/lang/php/5.1.4/bin/php`
 - Arguments: `path/to/phpDocumentor/phpdoc.inc`
`-t ${project_loc}/doc -o HTML:Smarty:PHP`
`-d ${project_loc} -s on`
 - Tab Refresh: Refresh resources upon completion
 The Project containing the selected resource
 Recursively include subfolders

PHPUnit

PHPUnit installieren

- ▶ Download: <http://pear.php.net/package/PHPUnit2/download>
- ▶ phpunit Skript vorbereiten:
 - pear-phpunit Skript in phpunit umbenennen
 - Den String @php_bin@ durch den Pfad zu PHP ersetzen
 - phpunit skript ausführbar machen
- ▶ @package_version@ in PHPUnit2/Runner/Version.php durch die Versionsnummer der PHPUnit2 Version ersetzen die heruntergeladen wurde
- ▶ `phpunit UnitTest UnitTest.php`
Führt die Tests der Testfall-Klasse `UnitTest` aus.

PHPUnit Beispiel

```
<?php
require_once 'PHPUnit2/Framework/TestCase.php';
class ArrayTest extends PHPUnit2_Framework_TestCase {
    protected $arr;
    protected function setUp() {
        $this->arr = Array();
    }
    public function testNewArrayIsEmpty() {
        // Der erwartete Wert von sizeof($this->arr) ist 0.
        $this->assertEquals(0, sizeof($this->arr));
    }
    public function testArrayContainsAnElement() {
        // Ein Element dem Array hinzufügen.
        $this->arr[] = 'Element';
        // Der erwartete Wert von sizeof($this->arr) ist 1.
        $this->assertEquals(1, sizeof($this->arr));
    }
}
?>
```