Tutorium Skriptsprachen

2009 - Max Maurer

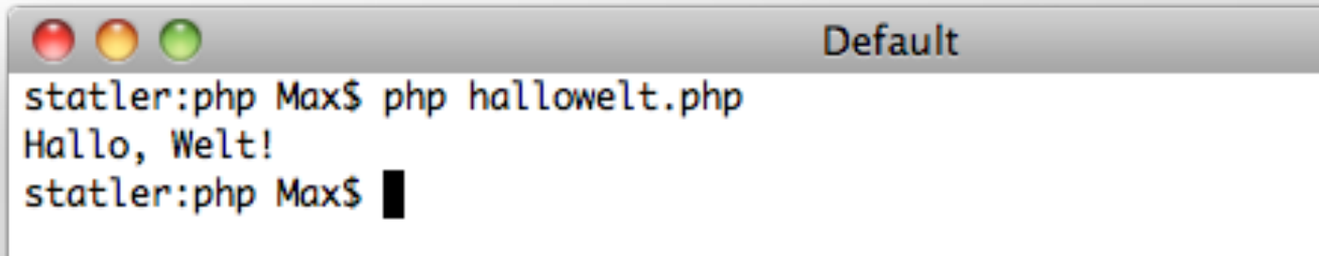
Rekusrives Akronym

PHP: Hypertext Preprocessor
PHP: Hypertext Preprocessor
PHP: Hypertext Preprocessor
PHP: Hypertext Preprocessor
PHP: Hypertext Preprocessor
PHP: Hypertext Preprocessor
PHP: Hypertext Preprocessor

Hello World

Hello World!

```
<?php
echo "Hallo, Welt!\n";
?>
```



Allgemeines

Fakten

Entstanden:	1995
Erfinder:	Rasmus Lerdorf
Firma:	The PHP Group
Lizenz:	PHP License
Stärken:	Sehr einfach HTML-Mix möglich
Anwendungsgebiete:	fast ausschließlich Internetseiten

Rasmus Lerdorf


Rasmus Lerdorf
(c) Wikimedia Commons

- Ursprünglich in Perl geschriebene „Personal Home Page Tools“
- Erfassen der Zugriffe auf Lerdorfs Lebenslauf
- PHP3 komplett neu von Suraski und Gutmans
- Momentan aktuelle Version: 5.3.1

Einsatzgebiete

- Riesige Verbreitung bei Webseiten
- Eher semi-professionell
- Viele bekannte PHP-basierte Webframeworks
  - TYPO3, Joomla!, Drupal (CMS)
  - WordPress (Blog-Software)
  - Mediawiki
  - phpMyAdmin (MySQL-Datenbank Administration)

Characteristika

- Interpreterbasierte Sprache
- Mit einer Erweiterung „bcompiler“ kann intermediate Bytecode erzeugt werden
- Programmierparadigmen: imperativ, objektorientiert
- schwach und dynamisch typisiert
- Garbage Collection und copy-on-write
- <?php ?>-Struktur ermöglicht Co-Existenz mit HTML-Code

Code-Beispiele

Der richtige Mix

- PHP-Code kann nur einen Teil der PHP-Datei ausmachen
- Andere Zeichen werden 1 zu 1 übernommen

```
<html>
<body>
<?php
echo "Hallo, Welt!\n";
?>
</body>
</html>
```

```
Default
statler:php Max$ php codemix.php
<html>
<body>
Hallo, Welt!
</body>
</html>
statler:php Max$
```

Variablen und Sichtbarkeit

- Variablendefinition durch $
- Normalerweise nur lokal sichtbar globale Variablen müssen übernommen werden
- Variablen können mit dem Inhalt einer anderen Variable aufgerufen werden

```
<?php
function func() {
     echo $local;
}
function func2() {
     global $local;
     echo "$local\n";
}
$local = "a";
$a = "Test";
echo $local."\n";
echo $$local."\n";
func();
func2();
?>
```

```
Default
statler:php Max$ php vars.php
a
Test
a
statler:php Max$
```

Konstanten

√

- mit ‚define' können Konstanten definiert werden
- ‚constant' ruft sie ab

```
<?php
define("CONST",
"Constants are accessible everywhere\n");

function func() {
    echo constant("CONST");
}

func();
?>
```

```
Default
statler:php Max$ php const.php
Constants are accessible everywhere
statler:php Max$
```

Funktionen und Parameter

- Funktionsdefinitionen mit Schlüsselwort function
- Parameter erhalten Namen (mit $) und optional einen Standardwert

```
<?php
function sum($a=1, $b=3) {
    return $a+$b;
}
echo sum(2)."\n";
echo sum(3,4)."\n";
echo sum()."\n";
?>
```

```
Default
statler:php Max$ php sum.php
5
7
4
statler:php Max$
```

Klassen

- Können einfach wie Funktionen definiert werden

```php
<?php

$obj = new cc;
echo $obj->var."\n";
$obj->increase();
echo $obj->var."\n";

class cc {
    public $var = 1;
    function __construct() {
            echo "object is beeing created!\n";
    }
    function increase() {
        $this->var++;
    }
}
?>
```

```
Default
statler:php Max$ php klassen.php
object is beeing created!
1
2
statler:php Max$
```

Error Handling

- Klassisch: try, catch, throw

```php
<?php
function inverse($x) {
    if (!$x) {
        throw new Exception('Division by zero.'
    }
    else return 1/$x;
}

try {
    echo inverse(5) . "\n";
    echo inverse(0) . "\n";
    echo inverse(4) . "\n";
} catch (Exception $e) {
    echo 'Caught exception: ',
                  $e->getMessage(), "\n";
}

// Continue execution
echo "Hello World\n";
```

```
Default
statler:php Max$ php exceotions.php
0.2
Caught exception: Division by zero.
Hello World
statler:php Max$
```

Module

- PHP ist (mehr oder weniger) all-inclusive
- Module bereits beim Kompilieren festgelegt
- phpinfo() hat nützliche Informationen

```
<?php
phpinfo();
?>
```

Default

```
Configuration

bcmath

BCMath support => enabled

Directive => Local Value => Master Value
bcmath.scale => 0 => 0

bz2

BZip2 Support => Enabled
Stream Wrapper support => compress.bz2://
Stream Filter support => bzip2.decompress, bzip2.compress
BZip2 Version => 1.0.5, 10-Dec-2007

calendar

Calendar support => enabled

Core

PHP Version => 5.3.0
```

Include und require

- Eigene Dateien können als Module eingelesen werden
- Falsch: include lädt dynamisch, require lädt immer
- Richtig: require bricht bei Fehlern ab, include warnt nur

Datei: includeMe.php

```
<?php
echo "Included!\n";
?>
```

```
<?php
    echo "one\n";
    include "includeMe.php";
    echo "two\n";
?>
```

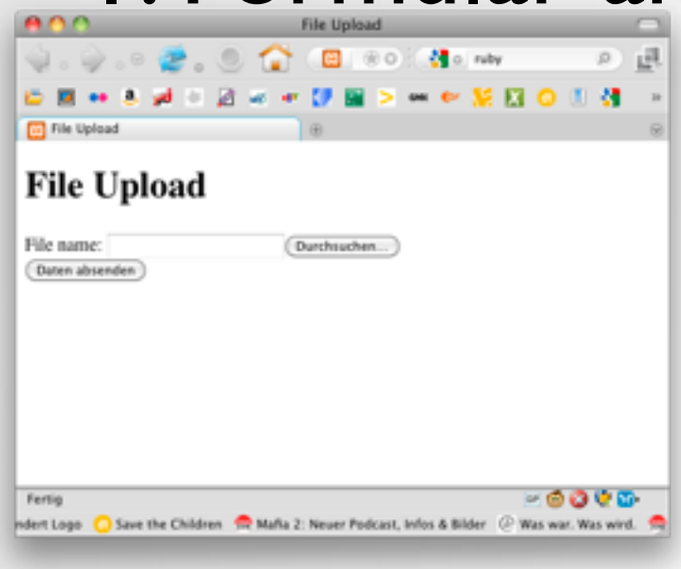
Default

```
statler:php Max$ php include.php
one
Included!
two
statler:php Max$
```

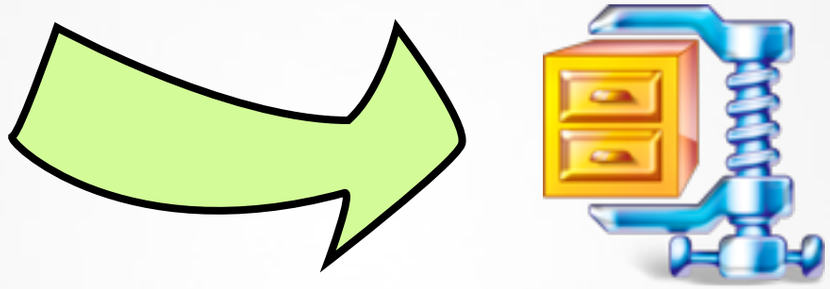
Die Aufgabe

Galerie Baukasten

1. Formular anzeigen

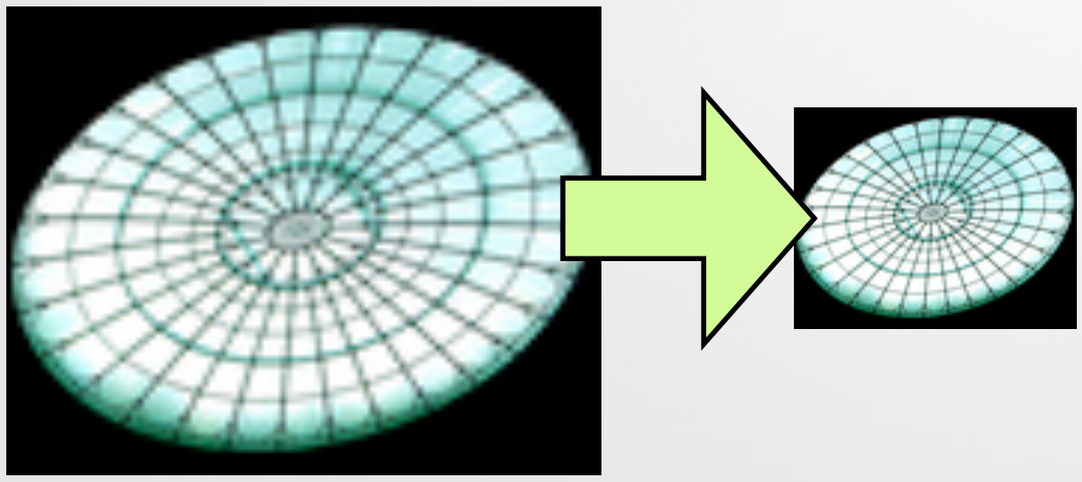


2. Zip-Datei hochladen



3. Entpacken

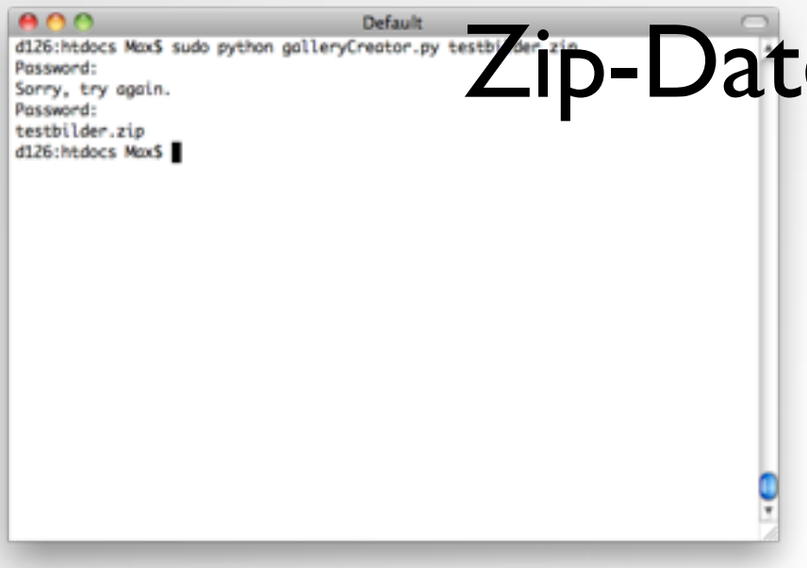
4. Thumbnails rendern



5. Gallerie Seite anzeigen

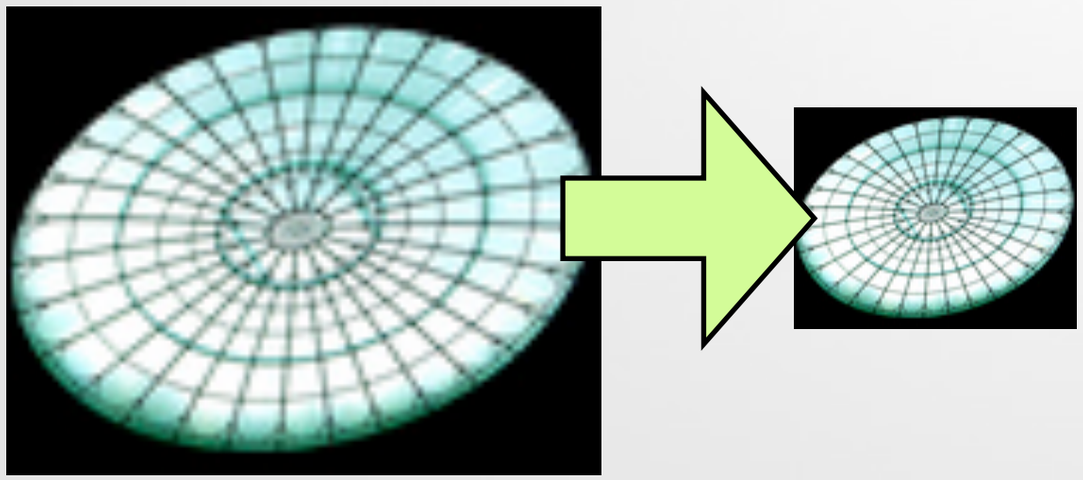
Kommandozeilenversion

1. Kommandozeilenaufruf mit Zip-Datei



2. Entpacken

3. Thumbnails rendern



4. Gallerie Seite erzeugen

Grundskript und Kommandozeile

Grundskript

```
<?php
function error($text) {
    echo "<html><body><h1>".$text."</h1></body></html>";
    exit();
}


if ($argc>1) {
    # wir haben ein command line argument!
    echo "Command line";
    $filepath = $argv[1];
} else {
    echo $HTML_TEMPLATE;
}

?>
```

HTML-Formulare

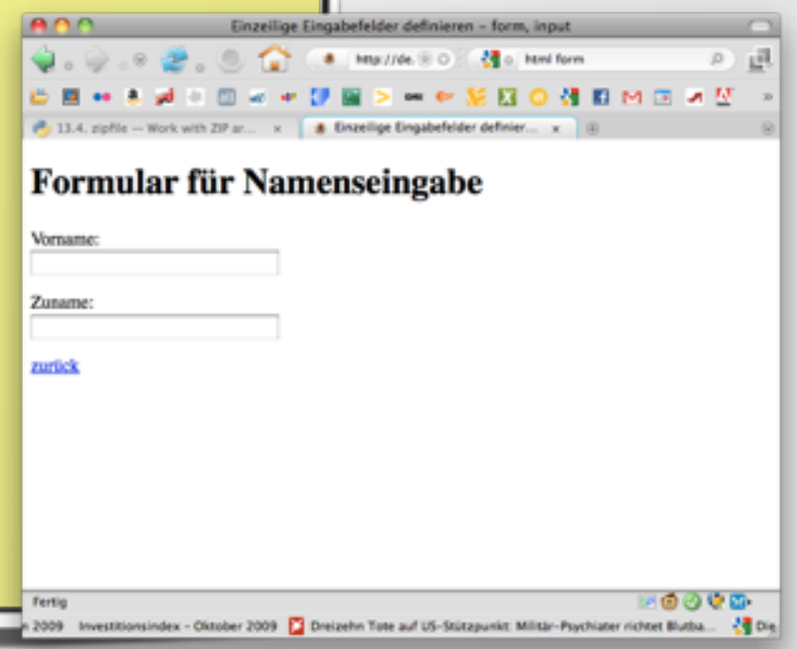
HTML-Formulare

- Darstellung verschiedener Eingabemöglichkeiten
  - Eingabefelder, Dropdown, Checkbox, Radio Button, Datei Upload, TextArea, Button

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
       "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Einzeilige Eingabefelder definieren</title>
</head>
<body>

<h1>Formular f&uuml;r Namenseingabe</h1>

<form action="input_text.htm" method="post">
  <p>Vorname:<br><input name="vorname" type="text" size="30"
maxlength="30"></p>
  <p>Zuname:<br><input name="zuname" type="text" size="30"
maxlength="40"></p>
</form>
```



Formulardaten allgemein

- Zwei Methoden: GET oder POST
- GET
  - Übergabe über die URL: „test.py?action=hallo&var1=wert1“
  - Direkt sichtbar und manipulierbar. Variablen bleiben bei Copy&Paste in E-Mails z.B. erhalten (z.B. Google Maps)
- POST
  - Nicht im Browser sichtbar auch nicht im Browser Cache gespeichert, werden im Anfrage-Header von HTTP übergeben

Formulardaten in PHP

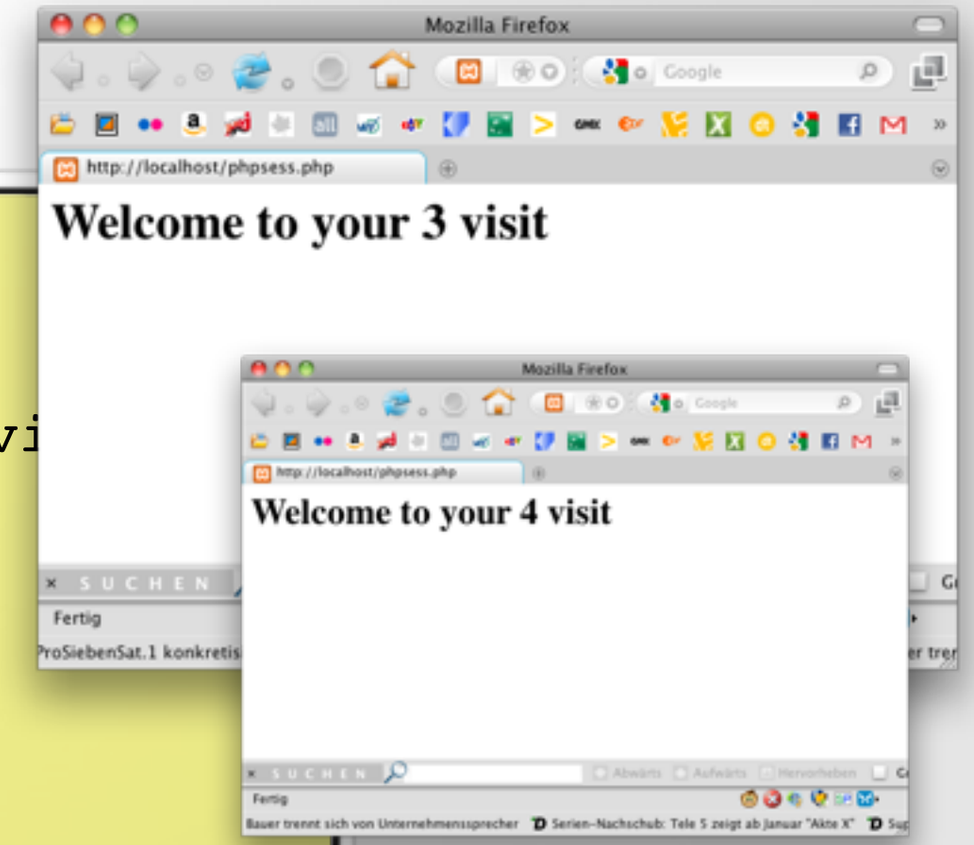
- Viele Sondervariablen ($_SERVER, $_GET, $_POST, $_FILES, $_SESSION, $_COOKIE)
- Immer bereits gefüllt

```
<?php
echot $_GET['action'];
?>
```

Session-Variablen

- Sessions werden benötigt um Benutzer über mehrere Seitenaufrufe zu identifizieren
- Normalerweise kompliziert per Cookies
- Einfach in PHP!

```
<?php
session_start();
$_SESSION['times'] = $_SESSION['times']+1;
echo "Welcome to your ".$_SESSION['times']." vi
?>
```

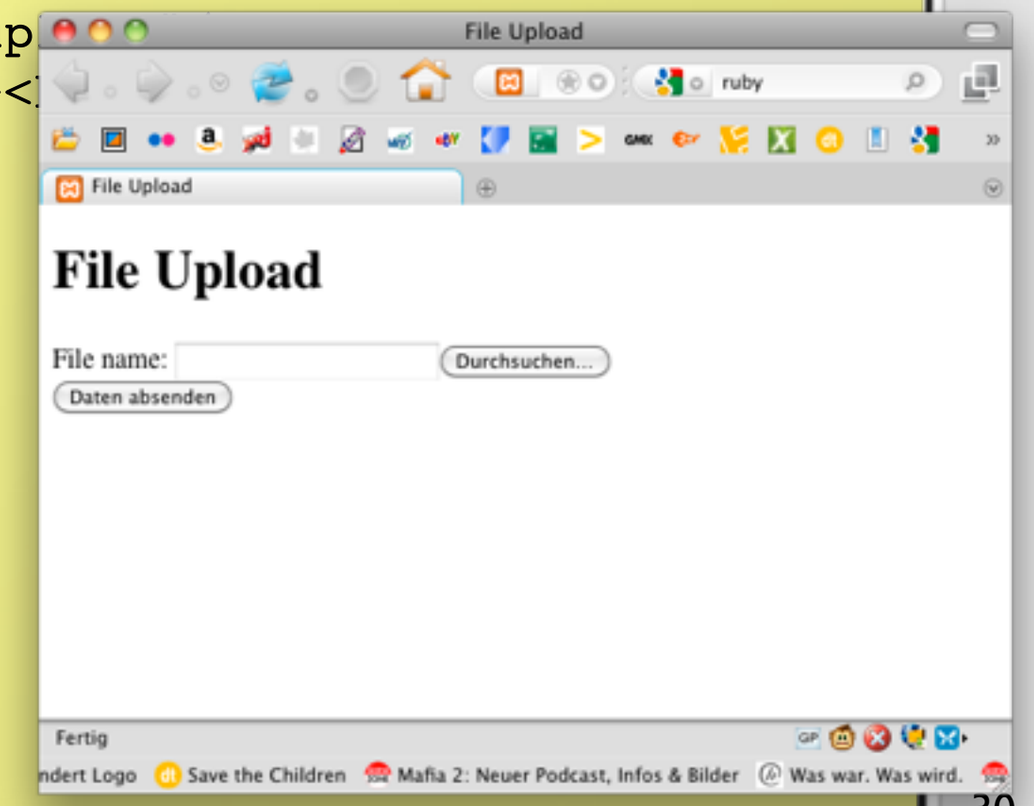


Datei-Upload

Formular ausgeben

```
<?php
$HTML_TEMPLATE = <<<eot
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
       "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<html><head><title>File Upload (PHP)</title>
</head><body><h1>File Upload (PHP)</h1>
<form action="?action=upload" method="post" enctype="multipart/form-
data">
<input type="hidden" name="action" value="up
File name: <input name="file" type="file"/><
<input name="submit" type="submit"/>
</form>
</body>
</html>
eot;

   echo $HTML_TEMPLATE;
?>
```



Upload auslesen

- Gelesene Datei temporär hinterlegt
- Kann einfach verschoben werden

```
<?php

$UPLOAD_DIR = "./tmp";
function processFile($key) {
   global $UPLOAD_DIR;
   $filepath = $UPLOAD_DIR."/" . basename( $_FILES[$key]['name']);
   move_uploaded_file($_FILES[$key]['tmp_name'], $filepath);
   return $filepath;
}
$filepath = processFile('file');
?>
```

ZIP-Datei entpacken

Zugriff auf ZIP-Dateien

```php
<?php
$zip = new ZipArchive;
$zip->open('test.zip')
$zip->extractTo('destination/dir/');
$zip->close();
?>
```

Das ‚ZipArchive'-Modul

- Umgang mit ZIP-Dateien
- Wichtige Parameter und Funktionen
  - $zip = new ZipArchive erzeugt ein neues Objekt für diese Zip-Datei
  - $zip->numItems gibt die Anzahl der Element in der Datei zurück
  - $zip->statIndex($i) gibt einen Informationsarray mit detaillierten Informationen zu einem Objekt zurück.
  - $zip->extractTo($dest, $item) entpackt ein oder mehrere Items. Problem: Die Ordnerstruktur der Datei bleibt immer erhalten und wird unterhalb von $dest angelegt

Zugriff auf ZIP-Dateien

- Zip-Datei ‚flat' entpacken

```
function unzip($filepath, $dir) {
    $zip = new ZipArchive;
    $zip->open($filepath);
    for ($i=0; $i<$zip->numFiles;$i++) {
        $item = $zip->statIndex($i);
        preg_match('/([^\/]*)$/',$item['name'],$regs);
        $filename = $regs[1];
        if ($filename=="") continue;
        if (preg_match('/^\./', $filename)) continue;
        $fp = $zip->getStream($item['name']);
        $contents = '';
        while (!feof($fp)) {
        $contents .= fread($fp, 2);
    }
    fclose($fp);
        file_put_contents($dir."/". $filename,$contents);
    }
    $zip->close();
}
```

Lesen und Schreiben von Dateien

- Übliche vorgehensweise: Öffen, Lesen/ Schreiben, Schließen
  - $fp = fopen($file, $mode)
  - Modi: r, r+, w
  - $content = fread($fp, $bytes) oder fwrite($fp, $bytes)
  - fclose($fp) schließt die Datei
- Einfachere Möglichkeit
  - file_put_contents($file, $contents)
  - erzeugt Datei mit Inhalten und schließt diese sofort

Zugriff auf ZIP-Dateien

- Zip-Datei ‚flat' entpacken

```
function unzip($filepath, $dir) {
    $zip = new ZipArchive;
    $zip->open($filepath);
    for ($i=0; $i<$zip->numFiles;$i++) {
        $item = $zip->statIndex($i);
        preg_match('/([^\/]*)$/',$item['name'],$regs);
        $filename = $regs[1];
        if ($filename=="") continue;
        if (preg_match('/^\./', $filename)) continue;
        $fp = $zip->getStream($item['name']);
        $contents = '';
        while (!feof($fp)) {
        $contents .= fread($fp, 2);
    }
    fclose($fp);
        file_put_contents($dir."/". $filename,$contents);
    }
    $zip->close();
}
```

Thumbnails schreiben

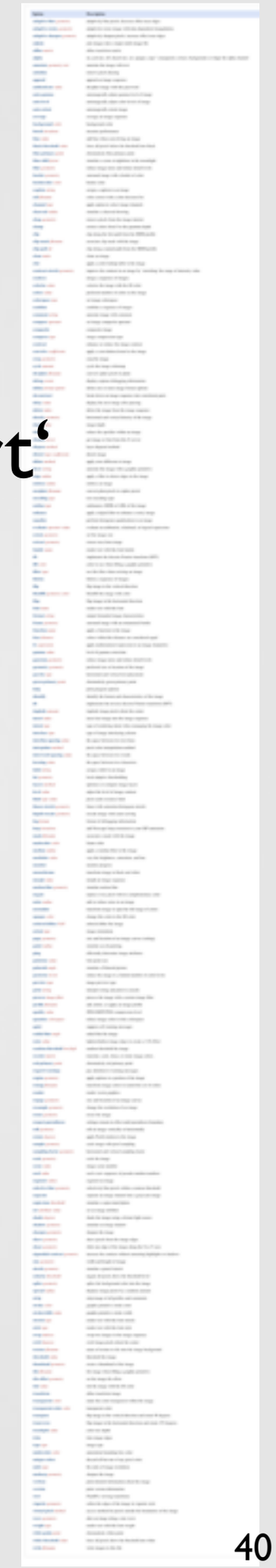
Thumbnails schreiben

- Kein Modul verfügbar
- Extern per Kommandozeilenaufruf (‚convert‘)

```
function createThumbnails($dir) {
    $handle = opendir($dir);
    while (false !== ($file = readdir($handle))) {
        if ($file == ".") continue;
        if ($file == "..") continue;
        if (!preg_match('/\.jpg$/', $file)) continue;
        if (preg_match('/_T\.jpg$/', $file)) continue;
        if (preg_match('/^\./', $file)) continue;
        $inputfile = $dir."/".$file;
        preg_match('/(.*)\.jpg$/',$file,$regs);
        $outputfile = $dir."/".$regs[1]."_T.jpg";
        $cmd = "/usr/local/bin/convert -resize 100x100 $inputfile $outputfile";
        exec($cmd);
    }
}
```

ImageMagick

- Kommandozeilen Bildverarbeitung mit umfangreichem Bildumwandlungstool ‚convert‘
- Dateiformate (> 100!)
  - Beispiele: AVI, BMP, JPEG, MPEG, PCX, PNG, PSD, SVG, TTF, WMF
- Optionen (s. rechts)
  - nur Einige: fill, rotate, resize, white-point
- Mehr Infos: www.imagemagick.org



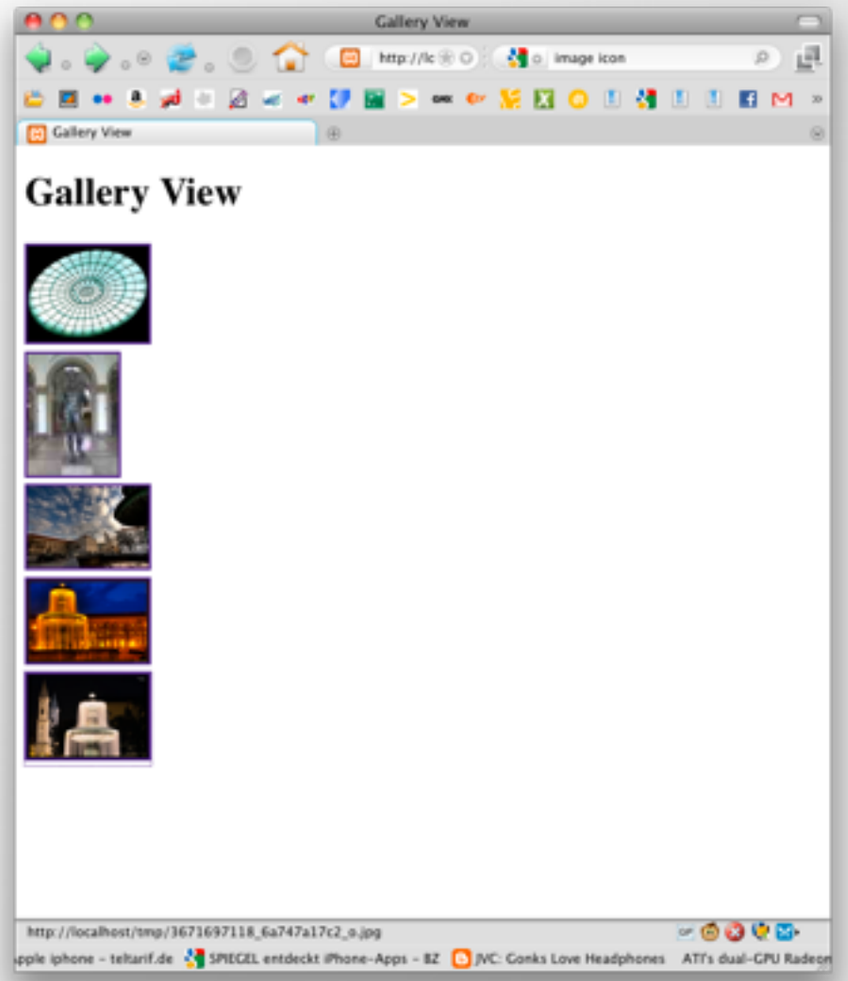
Gallery Seite erzeugen

Thumbnails schreiben

- Seite mit Template bauen und alle gefunden Bilder einfügen

```
$UPLOAD_TEMPLATE = <<<eot
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
       "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<html><head><title>Gallery View (Ruby)</title>
</head><body><h1>Gallery View (Ruby)</h1>
eot;
$UPLOAD_DIR = "./tmp";
function generateWebsite($dir) {
      global $UPLOAD_TEMPLATE,$timeUpload,$timeUnzip,$timeThumbnails;
      $html = $UPLOAD_TEMPLATE;
      $handle = opendir($dir);
      while (false !== ($file = readdir($handle))) {
            if ($file == ".") continue;
            if ($file == "..") continue;
            if (!preg_match('/\.jpg$/', $file)) continue;
            if (preg_match('/_T\.jpg$/', $file)) continue;
            if (preg_match('/^\./', $file)) continue;
            $inputfile = $dir."/".$file;
            preg_match('/(.*)\.jpg$/',$file,$regs);
            $outputfile = $dir."/".$regs[1]."_T.jpg";
            $html .= '<div class="image"><a href="';
            $html .= $inputfile;
            $html .= '"><img src="';
            $html .= $outputfile;
            $html .= '"/></a></div>';
      }
      $html .= "Store file: $timeUpload seconds<br/>";
      $html .= "Unzip file: ".($timeUnzip-$timeUpload)." seconds<br/>";
      $html .= "Create thumbnails: ".($timeThumbnails-$timeUnzip)." seconds<br/>";
      $html .= "Overall: $timeThumbnails seconds<br/>";
      $html .= "</body></html>";
      return $html;
}

echo generateWebsite($UPLOAD_DIR)
```



Zeit messen

Zeitmessung

- Für Sekunden einfach
- Für Millisekunden wird Hilfsfunktion benötigt

```
function microtime_float() {
    list($usec, $sec) = explode(" ", microtime());
    return ((float)$usec + (float)$sec);
}

$start = microtime_float();
$filepath = processFile('file');
$timeUpload = microtime_float()-$start;
```

Kompletter Code

Gallery Uploader Code

```
<?php
$HTML_TEMPLATE = <<<eot
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
       "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<html><head><title>File Upload (PHP)</title>
</head><body><h1>File Upload (PHP)</h1>
<form action="?action=upload" method="post" enctype="multipart/form-data">
<input type="hidden" name="action" value="upload"/>
File name: <input name="file" type="file"/><br/>
<input name="submit" type="submit"/>
</form>
</body>
</html>
eot;

$UPLOAD_TEMPLATE = <<<eot
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
       "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<html><head><title>Gallery View (Ruby)</title>
</head><body><h1>Gallery View (Ruby)</h1>
eot;

$UPLOAD_DIR = "./tmp";
$start = microtime_float();

function error($text) {
     echo "<html><body><h1>".$text."</h1></body></html>";
     exit();
}
```

Gallery Uploader Code

```
function processFile($key) {
     global $UPLOAD_DIR;
     $filepath = $UPLOAD_DIR."/" . basename( $_FILES[$key]['name']);
     move_uploaded_file($_FILES[$key]['tmp_name'], $filepath);
     return $filepath;
}

function unzip($filepath, $dir) {
     $zip = new ZipArchive;
     $zip->open($filepath);
     for ($i=0; $i<$zip->numFiles;$i++) {
          $item = $zip->statIndex($i);
          preg_match('/([^\/]*)$/',$item['name'],$regs);
          $filename = $regs[1];
          if ($filename=="") continue;
          if (preg_match('/^\./', $filename)) continue;
          $fp = $zip->getStream($item['name']);
          $contents = '';
          while (!feof($fp)) {
        $contents .= fread($fp, 2);
    }
    fclose($fp);
          file_put_contents($dir."/". $filename,$contents);
      }
      $zip->close();
}
```

Gallery Uploader Code

```
function createThumbnails($dir) {
     $handle = opendir($dir);
     while (false !== ($file = readdir($handle))) {
          if ($file == ".") continue;
          if ($file == "..") continue;
          if (!preg_match('/\.jpg$/', $file)) continue;
          if (preg_match('/_T\.jpg$/', $file)) continue;
          if (preg_match('/^\./', $file)) continue;
          $inputfile = $dir."/".$file;
          preg_match('/(.*)\.jpg$/',$file,$regs);
          $outputfile = $dir."/".$regs[1]."_T.jpg";
          $cmd = "/usr/local/bin/convert -resize 100x100 $inputfile $outputfile";
          exec($cmd);
     }
}

function generateWebsite($dir) {
     global $UPLOAD_TEMPLATE,$timeUpload,$timeUnzip,$timeThumbnails;
     $html = $UPLOAD_TEMPLATE;
     $handle = opendir($dir);
     while (false !== ($file = readdir($handle))) {
          if ($file == ".") continue;
          if ($file == "..") continue;
          if (!preg_match('/\.jpg$/', $file)) continue;
          if (preg_match('/_T\.jpg$/', $file)) continue;
          if (preg_match('/^\./', $file)) continue;
          $inputfile = $dir."/".$file;
          preg_match('/(.*)\.jpg$/',$file,$regs);
          $outputfile = $dir."/".$regs[1]."_T.jpg";
          $html .= '<div class="image"><a href="';
          $html .= $inputfile;
          $html .= '"><img src="';
          $html .= $outputfile;
          $html .= '"/></a></div>';
     }
     $html .= "Store file: $timeUpload seconds<br/>";
     $html .= "Unzip file: ".($timeUnzip-$timeUpload)." seconds<br/>";
     $html .= "Create thumbnails: ".($timeThumbnails-$timeUnzip)." seconds<br/>";
     $html .= "Overall: $timeThumbnails seconds<br/>";
     $html .= "</body></html>";
     return $html;
}
```

Gallery Uploader Code

```
function microtime_float() {
    list($usec, $sec) = explode(" ", microtime());
    return ((float)$usec + (float)$sec);
}

if ($argc>1) {
     # wir haben ein command line argument!
     echo "Command line";
     $filepath = $argv[1];
     $timeUpload = microtime_float()-$start;
     unzip(filepath, UPLOAD_DIR);
     $timeUnzip = microtime_float()-$start;
     createThumbnails(UPLOAD_DIR);
     $timeThumbnails = microtime_float()-$start;
     $html = generateWebsite(UPLOAD_DIR);
#    File.open("gallery.html", 'w') {|f| f.write(html) }
     exit();
} else {
     if ($_GET['action'] == "upload") {
          $filepath = processFile('file');
          $timeUpload = microtime_float()-$start;
          unzip($filepath, $UPLOAD_DIR);
          $timeUnzip = microtime_float()-$start;
          createThumbnails($UPLOAD_DIR);
          $timeThumbnails = microtime_float()-$start;
          echo generateWebsite($UPLOAD_DIR);
          exit();
     }
     echo $HTML_TEMPLATE;
}
?>
```

Nächste Veranstaltung?

Literatur

- PHP Manual http://www.php.net/manual/en/index.php
- PHP Regular Expressions examples: http://www.roscripts.com/PHP_regular_expressions_examples-136.html
- PHP Tutorials: http://www.tizag.com/phpT/fileupload.php

Bildnachweis

- http://www.creativeuncut.com/gallery-07/art/mlpit-mario-baby-hammer.jpg
- http://www.dotolearn.com/picturecards/images/imageschedule/proud_l.gif
- http://www.hakstpoelten.ac.at/buchoase/Buch.gif
- http://school.discoveryeducation.com/clipart/images/arteasel4c.gif
- http://www.gg-schnitt.at/wp-content/uploads/2009/01/lego_brick.png
- http://www.helliot.com/cms/images/stories/logo/logo_school.gif
- http://www.clker.com/clipart-window-icon.html
- http://globaleuropeans.com/uploads/images/GE_global%20projects%202.jpg
- http://www.sbac.edu/~tpl/clipart/Animals%20and%20Insects/bug%20cartoon%2002.jpg
- http://jasoncirillo.files.wordpress.com/2009/03/pong.jpg
- http://www.poster.net/leavitt-michael/leavitt-michael-mix-9959139.jpg