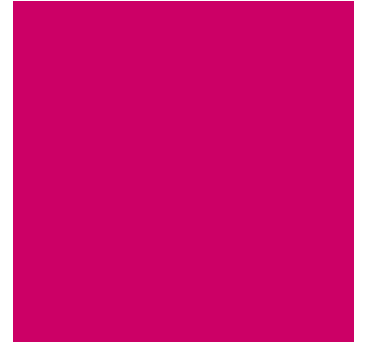




HTW Chur

Hochschule für Technik und Wirtschaft
University of Applied Sciences



Interaktive Medien

Richtlinien für das Codieren Version vom 18. Juni 2014

Martin Vollenweider

Dateinamen im Internet

- Da wir im Internet in gemischten Hard- und Softwareumgebungen (z.B. Windows, Unix, Macintosh, Linux etc.) arbeiten, dürfen nicht beliebige Dateinamen vergeben werden. Es gelten die folgenden Regeln:

Erlaubte Zeichen

- Buchstaben a bis z oder A bis Z
- Zahlen von 0 bis 9
- Unterstrich _

Verbotene Zeichen

- Europäische Sonderzeichen (z.B. ä, ö, ü, é, à, è, ç etc.)
- Leerzeichen
- Sonderzeichen (z.B. Punkt, ?, \$ etc.)

Gross- und Kleinschreibung

- Windows und Macintosh unterscheiden nicht zwischen Gross- und Kleinschreibung in Dateien. Das heisst, die Dateien «produkte.htm» und «Produkte.htm» sind identisch für Windows und Mac.
 - Linux und Unix unterscheiden zwischen Gross- und Kleinschreibung. Das heisst, die Dateien «produkte.htm» und «Produkte.htm» sind für Linux und Unix **nicht** identisch.
 - Dies ist in der Praxis wichtig, weil sehr viele Provider mit Linux- und Unix-Webservern arbeiten.
 - Vorschlag: Wir vergeben wir für Ordner- und Dateinamen und CSS-Definitionen nach Möglichkeit nur Kleinbuchstaben.
- Beispiel eines verbotenen Dateinamens: Über uns.html
Beispiel eines erlaubten Dateinamens: ueber_uns.html

Kamelhöcker-Namen verwenden

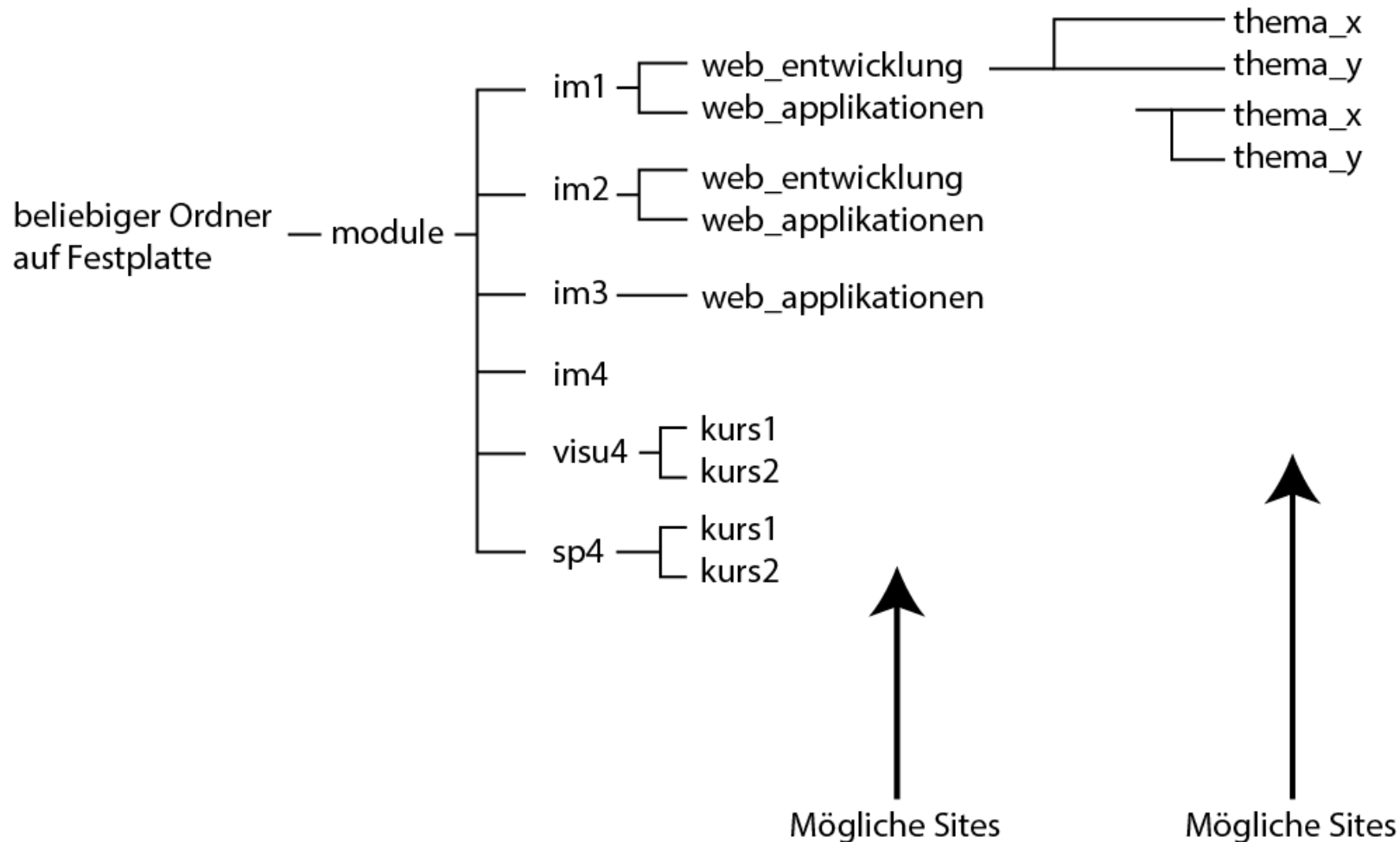
- Um die Lesbarkeit zu verbessern können Kamelhöcker-Namen verwendet werden. Beispiel:
statt `listenausgabe.html` besser `listenAusgabe.html` oder `listen_ausgabe.html`

Sinnvolle, «sprechende» Ordner- und Dateinamen vergeben

- `hausAmSee.jpg` ist besser als `has.jpg`

Ordnerstrukturen

- Die oben beschriebenen Regeln gelten innerhalb der vorgeschlagenen Ordnerstrukturen bei `imX`:



Site in Dreamweaver

- Es ist zwingend notwendig, für HTML- oder PHP-Projekte in Dreamweaver eine (oder mehrere) Site zu erstellen!

Programmierung

Aus folgenden Gründen sind Codier Richtlinien (engl. Code Conventions, Coding Guidelines) wichtig:

- Code Conventions erhöhen die Lesbarkeit der Software und ermöglichen es, neuen Code schneller und gründlicher zu verstehen
- 80% der Kosten einer Software entfallen auf Instandhaltung und Wartung
- Kaum eine Software wird für die gesamte Dauer ihrer Existenz vom selben Autor gepflegt
- Für den Fall, dass Sie Ihren Quellcode verkaufen möchten, sollten Sie sicherstellen, dass er genauso gut verpackt und sauber daherkommt, wie jedes andere von Ihnen erzeugte Produkt auch.

(Quelle: <http://www.cmklein.de/it/javacc/codeconv.pdf>, 28.02.2008)

Variablennamen bei Programmiersprachen

Grundsätzlich gelten dieselben Regeln wie bei Ordner- und Dateinamen:

- Buchstaben `a` bis `z` oder `A` bis `Z`
- Zahlen von `0` bis `9`
- Unterstrich `_`
- Gross- und Kleinschreibung beachten
- Sinnvolle, «sprechende» Variablennamen vergeben
- Kamelhöcker-Namen verwenden (Beispiel: `ausgabeWert`)
- Keine Variablennamen von Schlüsselwörtern der Programmiersprache (zum Beispiel `if`, `else`, `for` etc). vergeben
- Sich an Codier Normen halten (z.B. `i` für Schleifen)

Gültige und sinnvolle Variablennamen

- `ausgabe`, `ausgabeWert`, `listenFeld` etc.
- Variablen müssen mit einem Buchstaben (oder dem Unterstrich) beginnen
- Gross- und Kleinschreibung beachten

Gültige und sinnvolle ID- und Klassennamen Regeln bei CSS

- Struktur des Dokuments beachten (Beispiele: `#column-left`, `#column-right`)
- ID- und Klassennamen Variablen müssen mit einem Buchstaben beginnen
- Gross- und Kleinschreibung beachten
- Keine Unterstriche verwenden

Datenbank-, Tabellen- und Feldnamen

- Vorschlag: nur Kleinbuchstaben a bis z und Unterstrich
[Details](#)
- Namen, welche MySQL selbst als Befehle verwendet sollten nicht verwenden werden, zum Beispiel: `alter`, `select`, `from`, `order` etc.

Kommentare

Nicht triviale Befehle müssen kommentiert werden!

HTML Kommentare

```
<!-- Kommentar -->
```

CSS Kommentare

```
/* Kommentar */
```

Processing Kommentar

```
// Kommentar oder /* Kommentar */
```

Javascript Kommentar

```
// Kommentar oder /* Kommentar */
```

PHP Kommentar

```
// Kommentar oder /* Kommentar */
```

Javascript Ereignisse

- Ausschliesslich nach dem W3C-Modell, Beispiel:

```
window.addEventListener("load", loadFunction, false);
```

```
function loadFunction() {  
    // Code  
}
```

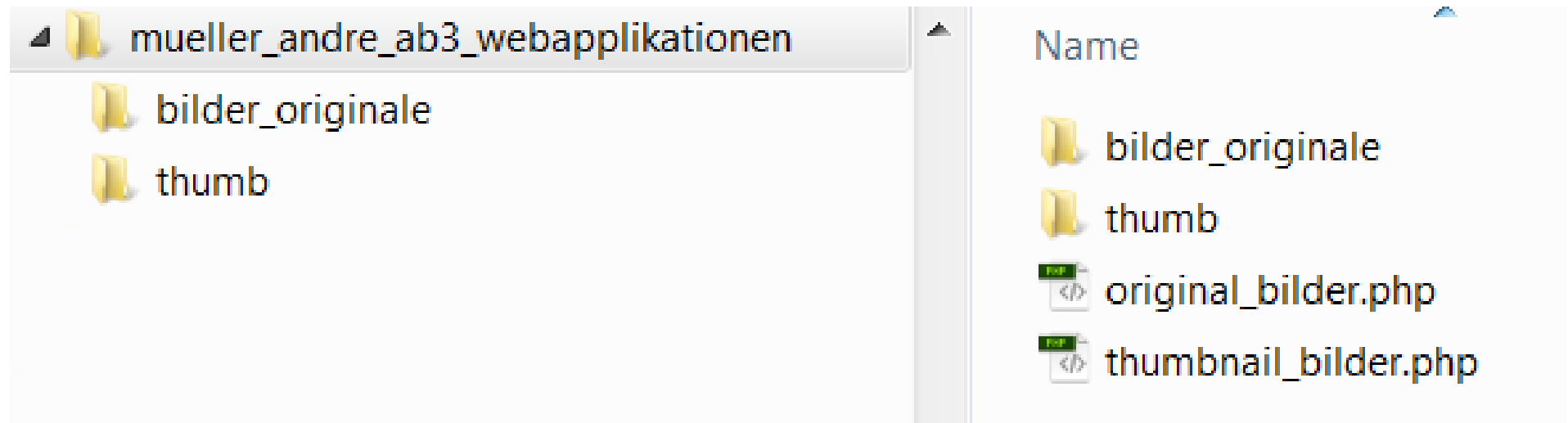
Abgabe von Arbeitsblättern und Projekten

1. Abgabe grundsätzlich nur auf Moodle, niemals via E-Mail
Begründung: Der E-Mail Virenfilter löscht Javascript- und CSS-Dateien.
2. Arbeitsblätter und Projekte müssen als **eine** (1) gezippte Datei abgegeben werden.
3. Die Dateien müssen sich auf der lokalen Festplatte in einem Projektordner befinden. Dieser **muss** genauen Namensbezeichnungen genügen:

name_vorname_abgabeart
4. **Beispiel:** Der Studierende André Müller gibt das Arbeitsblatt 3 des Kurses Web-Applikationen ab. Er speichert alle Dateien in folgenden Ordner:

mueller_andre_ab3_webapplikationen

5. Dieser Ordner mit allen sich darin befindlichen Dateien muss gezippt und auf Moodle gestellt werden:



6. Beispiele von falschen Abgaben:

- Die Dateien befinden sich in einem Ordner ab1

Wir können nicht herausfinden, von wem die Abgabe stammt

- Die Dateien werden ohne den Unterordner gezippt

Wir können nicht herausfinden, von wem die Abgabe stammt

Regeln

1. Abgaben von Dateien per Mail sind **ungültig**
2. Abgaben von nicht gezippten Dateien sind **ungültig**
3. Abgaben von Dateien im nicht-konformen Unterordner *name_vorname_abgabeart* sind **ungültig**
4. Nur Abgaben von Dateien im konformen Unterordner *name_vorname_abgabeart* sind **gültig**. Ordner und darin enthaltene Dateien müssen gezippt & auf Moodle gestellt werden.

Danke für das Verständnis

- In den Interaktiven Medien haben wir manchmal bis zu 70 Arbeitsblätter oder Projekte zu korrigieren
- Wir haben keine Zeit herauszufinden, vom wem die Abgabe der Datei *bildli.js* stammt

Weitere Codier-Regeln

- Siehe «[WordPress Coding Standards](#)»