



Web-basierte Anwendungen

Praktikumsaufgabe 2: (S)CSS und Layouts

Studiengänge AI (4140) & WI (4120)



Die Layout-Aufgabe

- Erweitern Sie die Ergebnisse aus Aufgabe 1
 - Beachten Sie dabei die Unterschiede!
 - Wechseln Sie in den Basisorder Ihres Rails-Projekts aus Aufgabe 1
`$ cd mein/pfad/zu/wba1`

- Erzeugen Sie nun leere HAML-Seiten für einen Controller „pr02“ (wie in pr01, aber ohne ‚index‘):

g: Kurzform für
„generate“

```
$ rails g controller pr02 einleitung kap1 kap2 tab
```

- Grund: Wir verzichten auf eine eigene „Index-Seite“ und bauen stattdessen ein mehrteiliges Seiten-Layout, das die Index-Information als Navigationsbalken enthalten soll.



Die Layout-Aufgabe

- Vorbereitungen (Forts.):
 - Wechseln Sie in den Ordner **app/views/pr02**
 - Überschreiben Sie die dortigen Datei-Prototypen mit den Inhalten der entsprechenden haml-Dateien aus Aufgabe 01:

```
$ cp ../pr01/*.haml .  
$ rm index.html.haml
```
 - Entfernen Sie nun in allen *.haml-Dateien alle Links auf die Index-Datei
 - Ausnahme: In „kap1.html.haml“ lenken Sie das Link von „redback.gif“ um auf „einleitung“ statt „index“



Die Layout-Aufgabe

- Das Layout: Vorbereitungen
 - Wechseln Sie in den Ordner **app/views/layouts**
 - Erzeugen Sie dort die Datei **pr02.html.haml** (durch Kopieren):
`$ cp application.html.haml pr02.html.haml`
 - Ergänzen Sie in **app/controllers/pr02_controller.rb** als Zeile 2:
`layout "pr02"`
- Das Layout: Grundsätzliches
 - Die HTML-Ausgabe nimmt hier ihren Anfang. Einzelne abgerufene Seiten („Views“) werden von Rails dort eingesetzt, wo Sie das Schlüsselwort „yield“ platzieren!
 - Sie können sich also in „Views“ auf das Wesentliche konzentrieren und allen Ihren Webseiten ein einheitliches Aussehen geben.
 - Bem.: Bei Bedarf sind auch mehrere Layouts möglich.
- Wir wollen nun ein Layout mit Navigationsbalken schaffen!



Die Layout-Aufgabe

- Das Layout – die Aufgabe
 - Gliedern Sie den „body“ Ihrer Dokumente in folgende Teile: „logo“, „headline“, „sidebar“, „main“ und „footer“.
 - Verwenden Sie **div**-Elemente mit **id**-Attributen namens „logo“, „headline“ usw.
 - Erinnerung: HAML bietet dazu sehr praktische Kurzschreibweisen
 - „logo“ (links) und „headline“ bilden einen gleich hohen Block („header“)
 - Bilden Sie kein Wrapper-Element „header“, platzieren Sie „logo“ und „headline“ nur nebeneinander
- Inhalte:
 - Verlagern Sie die „yield“-Anweisung in den Teil „main“
 - Im Teil „sidebar“ erzeugen Sie eine Art Überschrift „Index“ und eine Liste von Links zu den in „index.txt“ benannten vier Seiten. Behalten Sie die Link-Texte bei.
 - In „headline“ fügen Sie die feste Überschrift „WBA-Praktikum 02“ ein
 - In „footer“ soll eine Copyright-Zeile stehen, siehe Beispiel-Abbildung
 - In „logo“ soll sich ein Link zur Homepage der HSRM befinden. Der Inhalt des Links soll das Logo der HSRM sein.



Die Layout-Aufgabe

- Das Layout – die Aufgabe: Erste Tests
 - Starten Sie den Web-Server: **rails s**
 - Testen Sie im Browser das Aussehen der nun erscheinenden Seiten
 - Z.B. beginnend auf <http://localhost:3000/pr02/einleitung>
 - Das Inhaltsverzeichnis sollte auf jeder Seite vor dem jeweiligen Inhalt stehen, alle Links sollten bereits korrekt funktionieren



Die CSS-Aufgabe

- Vorab: CSS oder SCSS?
 - Wechseln Sie in den Ordner `app/assets/stylesheets`. Sie finden dort die Dateien: „`application.css`“, „`pr01.scss`“ und „`pr02.scss`“
 - „`application.css`“ ist eine reine CSS-Datei mit „globalen“ Anweisungen für alle Teile einer Rails-Anwendung.
 - Ihre Einbindung im Layout bewirkt das Einbinden aller Stylesheets
Es ist ein verbreiteter Irrtum anzunehmen, dass Views zum Controller `pr02` nur die CSS-Anweisungen aus `pr02.scss` erhalten. Da alle Stylesheets geladen werden, wirken tatsächlich hier auch die CSS-Anweisungen aus `pr01.scss` weiter – und umgekehrt!
 - „`pr02.scss`“ erhält nun Anweisungen, die für die Seiten des Controllers „`pr02`“ gedacht sind. Ihre Inhalte bestehen aus SCSS, einer Obermenge von CSS, die das Arbeiten mit CSS erleichtert. Ihr Inhalt wird von Rails vor der Auslieferung automatisch in CSS umgewandelt.
 - Tipp (hier nicht sinnvoll): Wenn Sie *nur* die (S)CSS-Anweisungen in `pr02.scss` laden lassen wollen, so ersetzen Sie im betroffenen Layout die Zeile
`stylesheet_link_tag "application"`
durch
`stylesheet_link_tag params[:controller]`
 - Da SCSS zu CSS abwärtskompatibel ist, haben Sie die Wahl, die folgenden Aufgaben mit CSS oder SCSS zu lösen
 - SCSS wird sehr empfohlen, weil es hilft, Redundanzen bei CSS-Code zu vermeiden



Die CSS-Aufgabe

- Zur Arbeitsweise
 - Kern Ihrer Aufgabe ist die eigenständige Recherche in den angegebenen Quellen zur Umsetzung der Vorgaben.
 - Bei CSS gibt es viel zu entdecken. Machen Sie sich vertraut mit dem Material. Dann werden Sie die benötigten Eigenschaften effizient finden und hier anwenden können.
 - Es ist ok, wenn Sie nicht alle Anforderungen zu 100% umsetzen können. Versuchen Sie aber, so weit wie möglich zu kommen.
 - Lernziel ist der selbstständige Umgang mit dem Quellen-Material.
 - Die eigentliche Schreibarbeit (SCSS-Quellcode) ist vergleichsweise kurz.



Die CSS-Aufgabe

- CSS-Teil 1: Umsetzung der Anforderungen aus Aufgabe 1
 - Realisieren Sie zunächst die Gestaltungswünsche aus Aufgabe 1, indem Sie die erforderlichen Einträge in Datei „pr01.scss“ schreiben / erweitern.
 - Wer dies bereits in Aufgabe 01 erledigt hat, ist hier schon fertig – weiter mit Teil 2.
 - Die **View-Dateien** selbst (einschließlich der Layouts) sollten möglichst **frei von CSS-Code** sein
 - Über Ausnahmen können wir diskutieren... wenn die Regel „sitzt“



Die CSS-Aufgabe

- CSS-Teil 2: Layout-Gestaltung mit „grid“
 - Ergänzen Sie CSS-Anweisungen in pr02.scss, die die Teile so auf dem Bildschirm positionieren, wie ihre Namen es suggerieren
 - Verwenden Sie als Ausgangspunkt das Grid-Beispiel aus der Vorlesung, variieren und ergänzen Sie die CSS-Anweisungen nach Bedarf
 - Klassische Alternative: CSS 2.1, Kapitel 9 „Visual formatting model“, insb. 9.6
 - Der „header“-Bereich soll 80px hoch sein, „footer“ 20px (feste Werte!)
 - Der Navigationsbalken „sidebar“ soll 10em breit sein, der Text im Hauptteil „main“ einen linken Zusatzabstand von 1em haben
 - Der Inhalt von „headline“ soll am linken Rand denselben Abstand haben wie der von „main“
 - Die Höhen von „main“ und „sidebar“ sollen mit dem Browserfenster variieren, also keine feste Höhe haben, wobei der Inhalt von „main“ bei Überlauf scrollbar sein soll
 - Der Inhalt von „logo“ (die Logo-Bilddatei der HS RM) soll mit CSS-Mitteln so skaliert werden, dass er in die linke Spalte passt
 - Der Text (die Link-Einträge) in „sidebar“ soll einen größeren Zeilenabstand als normal erhalten



Die CSS-Aufgabe

- CSS-Teil 2: Layout-Gestaltung (Forts.)
 - Hintergrundfarben:
 - Gestalten Sie die Teilbereiche mit geeigneten Hintergrundfarben. „Geeignet“ bedeutet, dass die Lesbarkeit und Übersichtlichkeit der Seiten gefördert werden soll
 - Nehmen Sie Einfluss auf die Farbgebung der Links nur in „sidebar“:
 - Unterscheiden Sie zwischen (1) unberührten und (2) bereits besuchten Links sowie Links, über die Sie (3) mit der Maus schweben, sowie (4) angeklickten (aktiven) Links
 - Wählen Sie andere als die Standard-Einstellungen. Ihre Wahl sollte ferner zur Wahl Ihrer Hintergrundfarben passen. Die Sidebar-Links sollen ferner nicht unterstrichen erscheinen.
 - Tipp: CSS 2.1 Kap. 5.11 „Pseudo classes“, W3Schools:
 - SASS/SCSS
 - Sehr nützlich: Variablen, Berechnungen, Kaskadierbarkeit – insb. beim Layout
 - Versuchen Sie, die vorgegebenen Maße nur an jeweils einer Stelle im Code anzugeben, und abgeleitete Werte berechnen zu lassen



WBA-Praktikum 02

Index

Einleitung

Kapitel 1

Kapitel 2

Anhang (Tabelle)

Einleitung

In der Einleitung soll der Umgang mit Texten geübt werden. Hierzu gehören Überschriften, Absätze, Zeilenumbrüche sowie verschiedene Formatierungsmöglichkeiten. Setzen Sie außerdem in allen Dokumenten die Umlaute korrekt um.

Erzeugen Sie am Ende der Zeile einen Zeilenumbruch.

Erzeugen Sie am Ende der Zeile einen Zeilenumbruch.

Erzeugen Sie am Ende der Zeile einen Zeilenumbruch.

Hier bitte keinen Zeilenumbruch. Hier bitte keinen Zeilenumbruch. Hier bitte keinen Zeilenumbruch. Hier bitte keinen Zeilenumbruch. Sehen Sie einen Unterschied?

Ich bin ein fast endlos scheinender Satz, der einfach kein Ende nehmen will und eigentlich für die Anzeige im Browser viel zu la

Dieser Abschnitt soll zentriert werden.

Dieser Abschnitt soll linksbündig formatiert werden.

Dieser Abschnitt soll rechtsbündig formatiert werden.

Im ersten Teilsatz soll ein Wort *kursiv* dargestellt sein, im zweiten soll eines in **Fettschrift** erscheinen, und am Ende soll ein Wort durchgestrichen sein.

Hoffentlich haben Sie nicht vergessen, die Umlaute korrekt umzusetzen. Das Umsetzen der Umlaute gilt auch für die anderen Seiten.



Bedingungen

- Abgabe
 - Vor Beginn Ihrer nächsten Praktikumsgruppe, also in einer Woche
 - Zusammen mit Aufgabe 01
- Art des Leistungsnachweises
 - Zu vergeben: **1 Punkt für Aufgabe 02 (+2 Punkte für Aufgabe 01)**
 - **Einzel-Arbeit – keine Teams!**
 - Abgabe der Dateien:
 - Im Rails-Verzeichnis „wba1“ ausführen:

```
$ rake log:clear  
$ rake tmp:clear  
$ cd ..
```
 - Sie sind nun im Elternverzeichnis von „wba1“. Jetzt noch ausführen:

```
# Ordner ./wba1 verpacken, ohne unnötige Unterordner:  
$ tar czf 02-wba1-<matnr>.tar.gz --exclude wba1/tmp \  
    --exclude wba1/node_modules --exclude wba1/log ./wba1  
Datei „02-wba1-<matnr>.tar.gz“ abgeben
```
- Kurzes Abnahmegespräch / Online-Demo der korrekten Funktion
 - Online, per Breakout-Session.

Fortsetzungszeile

- **Zu HTML, aber auch zu XML und CSS:**
 - **<https://wiki.selfhtml.org>**
 - Früher der Standard: de.selfhtml.org
 - Ergänzungen zum Stand von 2007 inzwischen vorhanden
 - **<https://www.w3schools.com>**
 - umfassend, gut, aktuell; auf Englisch
 - Tipp: Gute Doku zu Grid-Layouts unter https://www.w3schools.com/css/css_grid.asp
 - **<https://www.w3.org/TR/CSS2/>**
 - Die CSS-Spezifikation in Version 2.1 + errata
 - **<https://css-tricks.com/snippets/css/complete-guide-grid/>**
 - Ein Tutorial zu Grid-Layouts mit CSS3
- **Zu HAML**
 - **haml.info**
 - „Tutorial“ und „Documentation“ genügen völlig
- **Zu SASS/SCSS**
 - **sass-lang.com**
 - „Learn Sass“ und „Documentation“ genügen völlig
 - Beachte: Wir nutzen die SCSS-Variante und kein Kommandozeilen-Tool „sass“.
- **Zu Rails**
 - **rubyonrails.org**
 - Für den Anfang genügt: http://guides.rubyonrails.org/getting_started.html