



Web-basierte Anwendungen

Praktikumsaufgabe 5: JavaScript für HTML-Seiten - die Taschenrechner-Aufgabe

Studiengänge AI (4140) & WI (4120)

Taschenrechner: Hintergrund

- Hintergrund:
 - Wir verbessern und erweitern eine bestehende kleine JavaScript-Anwendung schrittweise
 - Dabei legen wir Wert auf eine klare Trennung der Bestandteile HTML, CSS und JS
 - Bei dieser Gelegenheit entfernen wir Code-Redundanzen
 - Wir nutzen die Gelegenheit, erste Versuche mit der JS-Bibliothek JQuery zu unternehmen
 - Nur optional:
Zur Vereinfachung des Codierungsaufwands entwickeln wir nicht direkt JavaScript-Code, sondern CoffeeScript- oder TypeScript-Code

Taschenrechner: Vorbereitungen

- Erweitern Sie die Ergebnisse aus Aufgabe 1-3
 - Wechseln Sie in den Basisorder **wba1** Ihres Rails-Projekts
`$ cd mein/pfad/zu/wba1`
 - Erzeugen Sie nun leere HAML-Seiten für einen Controller **pr05**:
`$ rails generate controller pr05 tr`
 - Editieren Sie `./app/controllers/pr05_controller.rb`. Ergänzen Sie den Verweis auf ein neues Layout **pr05** als Zeile 2
`layout "pr05"`
 - Ergänzen Sie in `./config/initializers/assets.rb` die Zeile
`Rails.application.config.assets.precompile += %w(pr05.css)`
 - Wechseln Sie in `./app/views/layouts`. Legen Sie ein neues, einfaches Layout **pr05.html.haml** an (Kopie von `application.html.haml`). Ersetzen Sie darin die Zeile
`= stylesheet_link_tag "application", ...`
durch
`= stylesheet_link_tag params[:controller]`
- Wählen Sie ferner als Inhalt von head/title:
WBA-Übung 05: Ein Taschenrechner

Taschenrechner: Ausgangslage schaffen

- Reproduzieren Sie das Taschenrechner-Beispiel
 - Wechseln Sie in den Order `./public` Ihres Rails-Projekts
 - Kopieren Sie den HTML-Quellcode des Taschenrechner-Beispiels in die neue Datei `tr.html`
 - Quelle: `~werntges/lv/wba/tr.html` auf dem Fileserver der Informatik (ext. Zugriff z.B. über `sftp login1.cs.hs-rm.de`)
 - Original-Quelle: selfhtml.org (Stand 2015/2016)
 - Starten Sie den Web-Server von Rails:
`$ rails s`
 - Testen Sie die korrekte Funktion der so erzeugten statischen HTML-Seite „Taschenrechner“ unter dem URL `http://localhost:3000/tr.`

Taschenrechner: Komponenten einordnen

- CSS:
 - Verlagern Sie den CSS-Code aus dem HTML-Header in die Datei **app/assets/stylesheets/pr05.scss**
- Hinweis:
Dieser CSS-Code beeinflusst
auch frühere wba1-Aufgaben!
- JavaScript-Code:
 - Verlagern Sie den JavaScript-Code aus dem HTML-Header in die Datei **app/assets/javascripts/packs/tr.js**
 - Ändern Sie in **app/assets/javascripts/packs/tr.js** alle Funktionsdefinitionen wie in folgendem Beispiel:
 - Vorher: `function Check (Eingabe) { ...`
 - Nachher: `window.Check = function(Eingabe) { ...`
 - HTML Body:
 - Verlagern Sie nun den Inhalt des gesamten Body-Elements (**<body>...</body>**) in die Datei **app/views/pr05/tr.html.haml**
 - Bearbeiten Sie die Datei **tr.html.haml**. Wandeln Sie den HTML-Code in HAML-Code um, wie bereits in früheren Aufgaben
 - Verwenden Sie dabei die HTML-Variante der Attribut-Codierung, insbesondere bei den **input**-Elementen. Beispiel:
`%input(type="button" value=" 7 " onclick="Hinzufuegen('7') ")`

Taschenrechner: Erster Test mit JavaScript

- Test 1:
 - Unter **http://localhost:3000/pr05/tr** sollte der Taschenrechner bereits funktionieren.
 - Verwendung findet momentan noch der (nur leicht veränderte) JS-Code von SelfHTML
- Debugging mit Firefox:
 - „Firebug“ einschalten (F12), Reiter „Konsole“ wählen, Seite neu laden
 - Laufzeitfehler von JS-Code werden in der „Konsole“ angezeigt

Taschenrechner: Installation von jQuery

- jQuery-Paket installieren:

- In `./wba1` ausführen:

```
$ yarn add jquery
```

- jQuery-Paket konfigurieren:

- In `./wba1/config/webpack/` eintragen (blaue Texte):

```
const { environment } = require('@rails/webpacker')

const webpack = require('webpack')
environment.plugins.prepend('Provide',
  new webpack.ProvidePlugin({
    $: 'jquery/src/jquery',
    jQuery: 'jquery/src/jquery'
  })
)

module.exports = environment
```

Taschenrechner: jQuery testen

- Test 2:

- In **app/javascript/packs/tr.js** am Ende eintragen:

```
jQuery(document).ready(function() {  
    alert("jQuery läuft");  
});
```

- Laden Sie die Taschenrechner-Seite im Browser neu:
http://localhost:3000/pr05/tr
 - Wenn ok, erscheint die Alert-Box mit Text „jQuery läuft“
 - Bem.: alert-Code wird später durch anderen Code zur Initialisierungszeit ersetzt
 - Wenn nicht, sehen Sie in der Firefox-„Konsole“ (F12) Fehlermeldungen aus der JavaScript-Ebene mit Hinweisen zu möglichen Ursachen

Taschenrechner: Refactoring

- Code-Redundanzen entfernen:
 - Datei **pr05.js**: Vereinfachen Sie den Code!
 - Nutzen Sie Variablen (insb. für `window.document.Rechner.Display`), um die wiederholte Verwendung von Ausdrücken wie `window.document.Rechner.Display.value` zu vermeiden
 - Nutzen Sie Möglichkeiten von ECMAScript wie `switch..case..default`
 - Verbessern Sie insbesondere „Sonderfunktion“ so, dass pro Funktion nur eine neue Zeile erforderlich ist
 - Tipp: Code zum Initialisieren gehört in den Codeblock des `alert`-Aufrufs aus Test 2.
 - Datei **tr.html.html**:
 - Entfernen Sie Leerzeichen aus den `value`-Attributen von **input**
 - Entfernen Sie alle Attribute von **form** außer `name` (ersatzlos)
 - Hier STOP für Woche 1 – Fortsetzung folgt.
Abgabe von P05 erst in zwei Wochen!
 - Ziel für Woche 1: Umstellungen vorführen können.