

Praktikum Datenbanken und Webprogrammierung

Aufgabenblatt Nr. 3: Dart Teil II

Prof. Dr. Nane Kratzke

1 Hausaufgabe: Dot-o-Mator (50 Punkte)

Sie sollen nun einen Dot-o-mator entwickeln, der Domain Namen Vorschläge macht, die Web 2.0 Charakter haben. Das Prinzip ist dabei ganz einfach. Üblicherweise werden zwei oder drei Begrifflichkeiten genommen, die einen Service oder eine Seite beschreiben. Diese Wörter werden in ihre Silben zerlegt und die Silben miteinander kombiniert. Ein Teil der Silben wird für Endungen genutzt. Für die Begrifflichkeiten *Energy* und *Aware(ness)* könnte diese Zerlegung so aussehen:

Dot-o-Mator

Erste Silben:	<input type="text" value="en er"/>
Zweite Silben:	<input type="text" value="a wa ware"/>
Endungen:	<input type="text" value="gy ness"/>
Top Level Domains:	<input type="text" value="com net de org"/>
<input type="button" value="Senden"/>	

- [enagy.com](#)
- [enagy.net](#)
- [enagy.de](#)
- [enagy.org](#)
- [enaness.com](#)
- [enaness.net](#)
- [enaness.de](#)
- [enaness.org](#)
- [enwagy.com](#)
- [enwagy.net](#)
- [enwagy.de](#)
- [enwagy.org](#)
- [enwaness.com](#)
- [enwaness.net](#)
- [enwaness.de](#)
- [enwaness.org](#)
- [enwaregy.com](#)
- [enwaregy.net](#)

Sie erhalten Punkte wie folgt:

- Sie lösen das Problem mittels einer Command Line App (die Silben werden als Listen kodiert). **[10 Punkte]**
- Sie bieten über einen Server ein Webformular an, die Ausgabe erfolgt mittels einer einfachen HTML Liste von Links **[+ 20 Punkte]**

- Sie sehen eine Behandlung fehlerhafter Eingaben vor (Silben werden durch mehr als ein Whitespace getrennt, es werden nicht alle Eingaben getätigt). In diesen Fällen soll eine Fehlermeldung ausgegeben werden und die Eingabe erneut vorgenommen werden können. [**+ 10 Punkte**]
- Sie beachten zusätzlich Cross Site Scripting und Escapen Nutzereingaben entsprechend. [**+10 Punkte**]

Hinweis: Es ist sinnvoll das Webframework **start** (<https://github.com/lvivski/start>) zu nutzen. Die Formulardatenübergabe mittels GET ist vermutlich die einfachste Form hier.

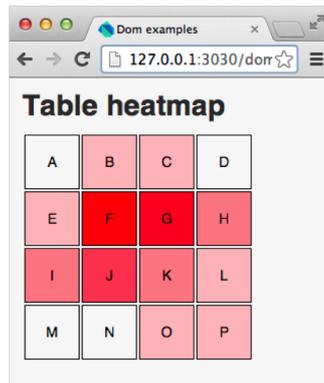
2 Hausaufgabe: Vorbereitung auf Clientseitige Programmierung mit Dart

Bereiten Sie sich bitte auf die Praktikumsaufgabe vor, indem sie noch einmal das Konzept

- des DOM-Trees nacharbeiten,
- das Selektormodell nachvollziehen und wie es in Dart angewendet werden kann,
- wie sie in Dart auf den DOM Tree zugreifen und ihn verändern können (insbesondere mittels `style`, `attribute` und `classes` Properties von `HtmlElementen`),
- wie sie Ereignisse im DOM Tree abfangen können, um auf diese zu reagieren.

3 Praktikumsaufgabe: Implementieren Sie eine interaktive Heatmap (50 Punkte)

Entwickeln sie bitte eine HTML Seite mit folgendem Aussehen (allerdings ohne rotes Highlighting) [10 Punkte].



Verfolgen Sie bitte auf dieser Seite `mouseover` Events innerhalb der Tabelle derart, dass Tabellenelemente mitbekommen, dass ein `mouseover` Event in ihnen stattfindet (indem sie diese Ereignis durch eine Konsolenausgabe kundtun und ihren Inhalt ausgeben) [10 Punkte].

Erweitern sie nun ihre Lösung so, dass sie im DOM Tree speichern wieviele `mouseover` Events über einem Tabellenelement ausgeführt wurden (indem sie diese Anzahl in einem Attribut im DOM Tree speichern und bei neuen Events aktualisieren [10 Punkte], erweitern sie bitte auch ihre Konsolenausgabe um die Anzahl mit auszugeben [5 Punkte]).

Setzen sie diese im DOM Tree gespeicherten Zähler so um, dass Tabellenelemente einen umso roten Hintergrund bekommen, je mehr `mouseover` Events sie erhalten haben (desto häufiger die Maus darüber war). Die Farbdarstellung soll relativ sein. Das Element mit den meisten `mouseover` Events soll die Farbe `rgba(255, 0, 0, 1.0)` erhalten. [15 Punkte]

Hinweis: Die Klassen `MouseEvent` und `HtmlElement` in der Dart API (<http://api.dartlang.org>) sind sicherlich sehr hilfreich.