

Vokabeltrainerprojekt

Vorstellung der fertigen Anwendung

Uta Lemke, Sahana CG

January 16, 2019

Projektbeschreibung

- Tool zum Üben von Vokabeln
- Funktionen:
 - Abfrage der Vokabeln in zufälliger Reihenfolge
 - Hinzufügen neuer Vokabeln
 - Export- & Import-Funktion
- verwendete Technologien:
 - HTML
 - CSS
 - JavaScript
 - localStorage-API
 - DOM-API

Hier eine kleine Demonstration: <http://localhost/home.html>

HTML



import vocabulary



back to home



export vocabulary

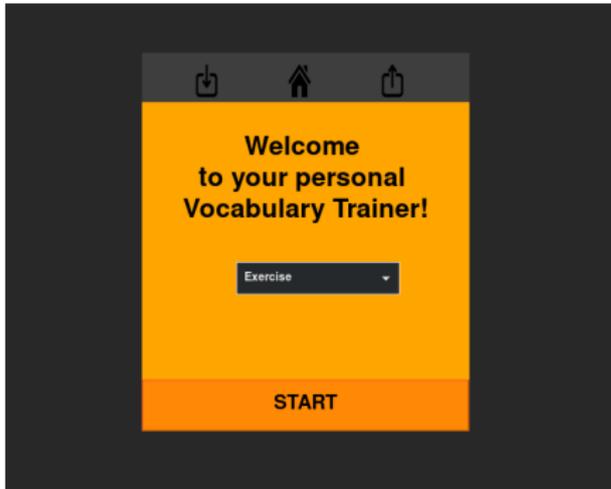
**Welcome
to your personal
Vocabulary Trainer!**

Exercise ▾

START

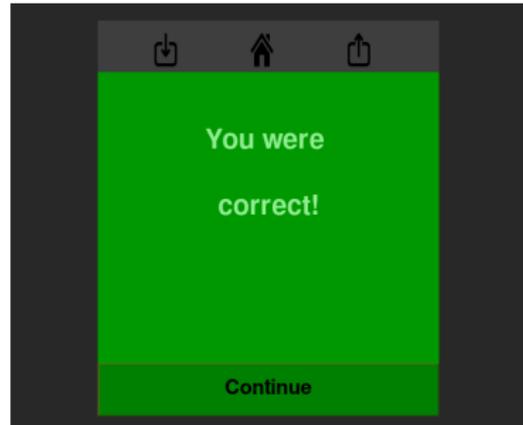
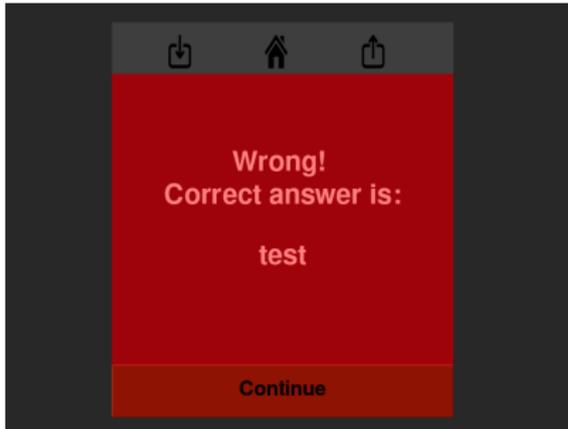
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="width=device-width, initial-scale=1"/>
5 </head>
6 <body>
7 <div class="outer">
8   <div class="topnav">
9     <div id="import">
10      
11      <span class="tooltiptext">import vocabulary</span>
12    </div>
13    <div id="home">
14      
15      <span class="tooltiptext">back to home</span>
16    </div>
17    <div id="export">
18      
19      <span class="tooltiptext">export vocabulary</span>
20    </div>
21  </div>
22  <div class="home_card">
23    <h3>Welcome <br/> to your personal <br/> Vocabulary Trainer!</h3>
24    <div class="select">
25
26      <select id="mode_chooser">
27        <option id="exercise_mode_option" value="exercise">Exercise</option>
28        <option id="add_mode_option" value="add">Add vocabulary</option>
29      </select>
30      <div>
31      </div>
32    </div>
33    <button type="button" id="start_btn">START</button>
34  </div>
35 </div>
36 </body>
37 </html>
```

CSS

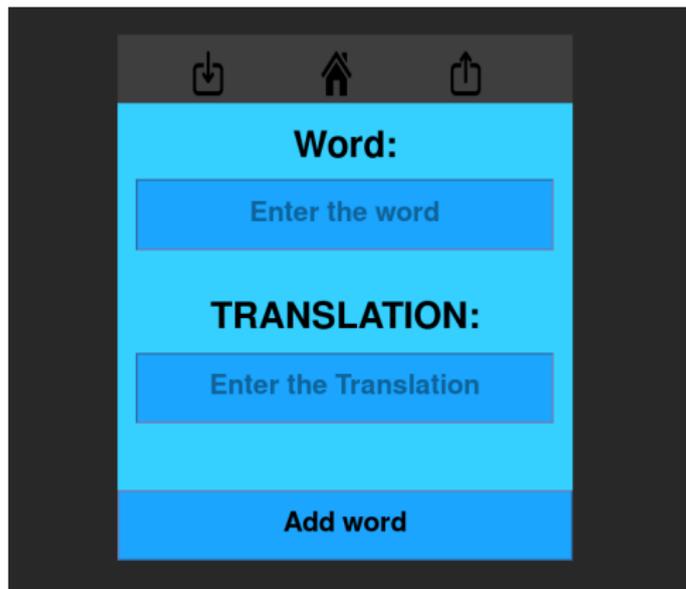


```
home.css
24 height:40px;
25 font-family: Arial, Helvetica, sans-serif;
26 text-align: center;
27 padding: 0;
28 margin-left:115px;
29 margin-top: 50px;
30 font-weight: bold;
31 }
32
33 .home_card select:hover {
34 background-color: black;
35 color: white;
36 }
37
38 .home_card select-box {
39 cursor: pointer;
40 background-color: bisque;
41 }
42
43 .home_card button {
44 padding: 15px 32px;
45 font-family: Arial, Helvetica, sans-serif;
46 text-align: center;
47 background-color:rgb(255, 137, 5);
48 display: inline-block;
49 font-size: 24px;
50 text-decoration: none;
51 color: black;
52 border-block-end-width: 100%;
53 border-color: rgba(165, 42, 42, 0.38);
54 width:100%;
55 font-weight:bold;
56
57 position: absolute;
58 bottom: 0;
59 margin-bottom: 0;
60
61 /*margin-top: 105px;*/
62 }
63
64 .home_card button:hover {
65 background-color: rgba(255, 255, 255, 0.01);
```

Exercise



Add Vocabulary



The screenshot shows a mobile application interface for adding vocabulary. At the top, there is a dark grey navigation bar with three icons: a download arrow, a house icon, and an upload arrow. Below this is a light blue main area. The first section is labeled "Word:" and contains a blue button with the text "Enter the word". The second section is labeled "TRANSLATION:" and contains a blue button with the text "Enter the Translation". At the bottom of the light blue area is a blue button with the text "Add word".

JavaScript

```
< main.js
1 var mode_chooser, home_card, start_btn, outer, home_btn, import_btn, export_btn;
2
3 document.addEventListener('DOMContentLoaded', function(){
4   initialize();
5 })
6
7 var initialize = function(){
8   outer = document.querySelector('.outer');
9   home_card = document.querySelector('.home_card');
10  mode_chooser = document.querySelector('#mode_chooser');
11  start_btn = document.querySelector('#start_btn');
12  home_btn = document.querySelector('#home_btn');
13  import_btn = document.querySelector('#import_btn');
14  export_btn = document.querySelector('#export_btn');
15  back_to_home();
16
17  start_btn.onclick = start;
18  home_btn.onclick = back_to_home;
19  export_btn.onclick = export_localstorage;
20  import_btn.onclick = import_as_localstorage;
21 }
22
23 var start = function () {
24   if (mode_chooser.options[mode_chooser.selectedIndex].value == 'exercise'){
25     start_exercise();
26   }
27   else{
28     start_add();
29   }
30 }
31
32 var load_html = function (page) {
33   var xmlhttp = new XMLHttpRequest();
34   xmlhttp.open('GET', page, false);
35   xmlhttp.send()
36   return xmlhttp.responseText;
37 }
38 }
```

- Interaktivität → Single Page Application
- Zugriff auf APIs

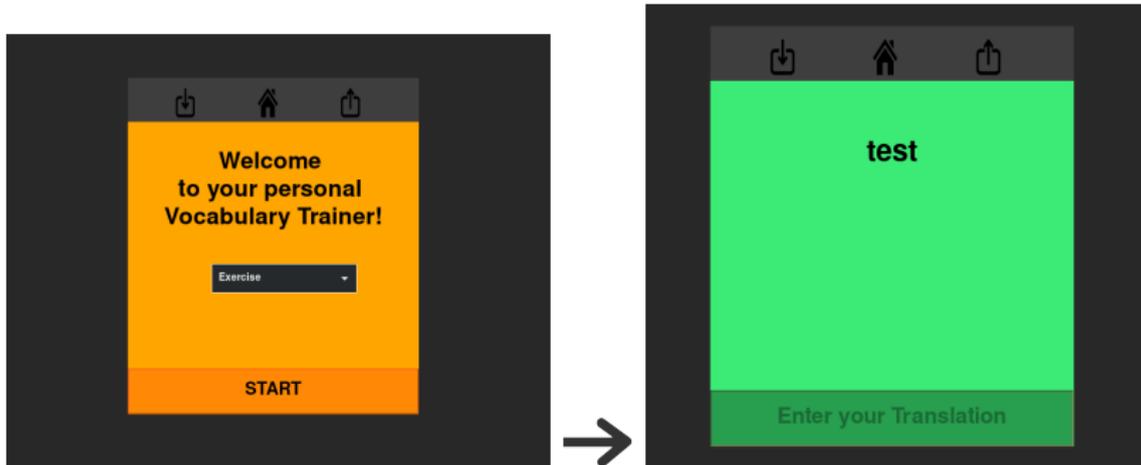
localStorage-API



Key	Value
Baum	tree
Maus	mouse
mouse	Maus
test	test
tree	Baum

- Datenpersistenz → Vokabelkartei
- einfacher Zugriff

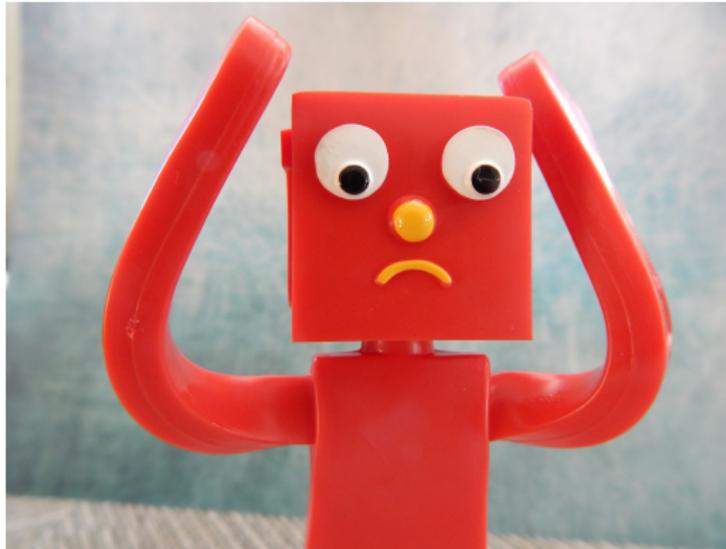
DOM-API



- dynamisches Ersetzen von Inhalten
- Ansprechen der HTML-Document-Tree-Elemente

To be continued...

- mehrere Vokabelkarteien
- Refactoring
- ggf. Bugfixing



Noch Fragen?