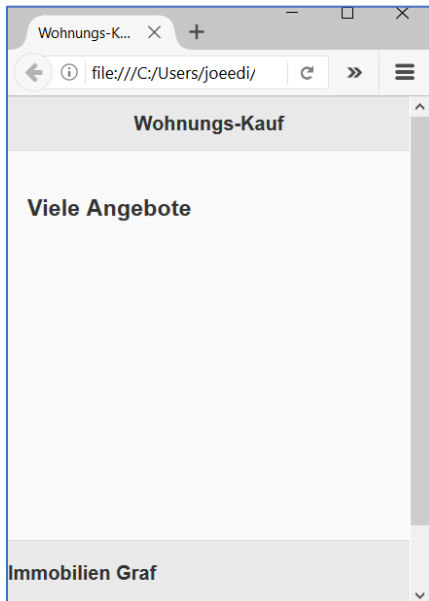


jQuery – 3

Inhalt:

1)Header-Widget, 2)Footer-Widget, 3)Buttons, 4)Übung: Wohnungskauf, 5)Controlgroups, 6)Listview, 7)Panel-Widget

Erstelle folgendes einfache Beispiel mit einem Header, einem Content und einer fixierten Fußzeile:



Code:

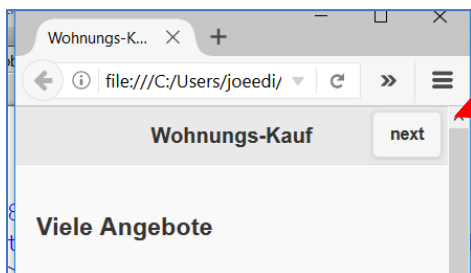
```
3_jquerymobile.html x
jQuery mobile-1.4.5.min.css jquery-2.1.4.min.js jquery.mobile-1.4.5.min.js
Code Teilen Entwurf Live-Ansicht Titel: Wohnungen
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Wohnungen</title>
7 <link rel="stylesheet" href="lib/jquery.mobile-1.4.5.min.css">
8 <script src="lib/jquery-2.1.4.min.js"></script>
9 <script src="lib/jquery.mobile-1.4.5.min.js"></script>
10 </head>
11 <body>
12 <!--Start page 1-->
13 <div data-role="page" id="p1" >
14     <div id="startPageHeader" data-role="header">
15         <h3>Wohnungs-Kauf</h3>
16     </div> <!--Ende Header-->
17     <div id="startPageContent" data-role="content">
18         <h3>Viele Angebote</h3>
19     </div> <!-- Ende content-->
20     <div id="startPageFooter" data-role="footer" data-position="fixed">
21         <p>Immobilien Graf</p>
22     </div> <!-- Ende footer-->
23 </div> <!--Ende page 1-->
24 </body>
25 </html>
```

1)Header mit einem Button:

Wenn nur ein Button im Header Verwendung finden soll, gibt es nur zwei Möglichkeiten: entweder der Button ist links, was ohne Konfiguration bewerkstelligt werden kann, da es der Standard ist, oder er soll rechts sein. In diesem Fall muss man eine CSS-Klasse vergeben:

Der Button soll auf die nächste Seite verlinken, die aber noch nicht vorhanden ist:

```
14 <div id="startPageHeader" data-role="header">
15   <a href="page_zwei.html" class="ui-btn-right">next</a>
16   <h3>Wohnungs-Kauf</h3>
```



Hinweis: für die linke Seite gibt es die Klasse: „ui-btn-left“.

Hyperlinks im Header werden standardmäßig in Buttons umgewandelt.

Zurück-Button im Header:

Über das data-Attribut „data-add-back-btn“ lässt sich im Header der Seite ein Zurück-Button einbinden, der analog zum Browser-Zurück-Button funktioniert. Mit dem Wert „true“ wird der Button angezeigt und mit dem Wert „false“ ausgeblendet. Weiters kann man mit dem data-Element „data-back-btn-text“ einen beliebigen Text für den Zurück-Button vergeben. Vergibt man keinen eigenen Text, so wird der voreingestellte Standardtext „Back“ angezeigt.

```
12 <!--Start page 1-->
13 <div data-role="page" id="p1" >
14   <div id="startPageHeader" data-role="header" data-add-back-btn="true"
15     data-back-btn-text="zurück">
16     <a href="page_zwei.html" class="ui-btn-right">next</a>
17     <h3>Wohnungs-Kauf</h3>
18   </div> <!--Ende Header-->
```

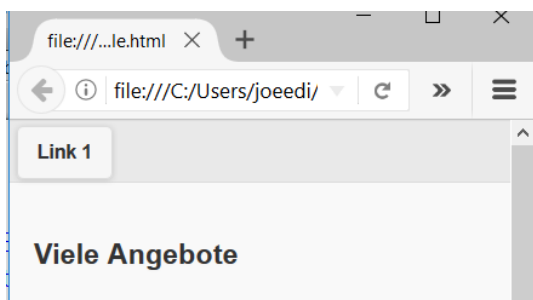
Hier muss man beachten, dass man dem rechten Button die Klasse „ui-btn-right“ mitgibt, ansonsten würde der Button nicht angezeigt. Der Grund ist nämlich, dass der rechte Button durch den im Header definierten Button überlagert werden würde.

Kein Text im Header:

Für den Fall, dass man keinen Titeltext im Header haben möchte, sondern nur Buttons, braucht man eine CSS-Klasse. Man muss in Header einen leeren „span-Tag“ erstellen und ihm die Klasse „ui-title“ zuweisen.

Erstelle diese Variante zur Übung, aber stelle danach wieder mit STRG+z den Vorzustand wieder her.

```
12 <!--Start page 1-->
13 <div data-role="page" id="p1" >
14   <div id="startPageHeader" data-role="header" >
15     <a href="">Link 1</a>
16     <span class="ui-title">
17   </div> <!--Ende Header-->
```



2)Das Footer Widget

Dieser kann im Gegensatz zum Header Widget nicht nur zwei sondern mehrere Buttons verwenden. Zusätzlich kann man diese in ein Controlgroup Widget packen, welches noch schöner aussieht.

Aber nicht nur Buttons, sondern auch andere Elemente kann man im Footer Widget platzieren.

Soll ein Footer über mehrere Seiten konstant bleiben, verwendet man für diese den gleichen Quelltext mit der gleichen „data-id“.

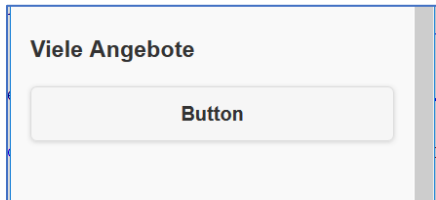
Hyperlinks im Footer werden standardmäßig in Buttons umgewandelt.

3)Buttons

Im Header und Footer werden Hyperlinks automatisch in Buttons umgewandelt.

Syntax:

```
<a href="" data-role="button">Button</a>
```



Icons:

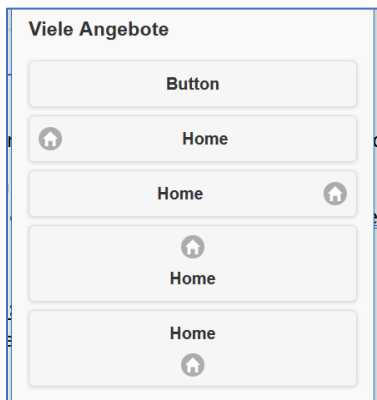
Mit dem data-Attribut „data.icon“ kann man dem Button Icons zuweisen.

Standardmäßig werden die Icons immer links im Button abgebildet. Mit „data-iconpos“ kann man diese Position ändern, mit folgenden Werten:

left, right, top und bottom.

Übung:

Erstelle im Content folgende Buttons:



Code:

```
<h3>Viele Angebote</h3>
<a href="" data-role="button">Button</a>
<a href="" data-role="button" data-icon="home" data-iconpos="left">Home</a>
<a href="" data-role="button" data-icon="home" data-iconpos="right">Home</a>
<a href="" data-role="button" data-icon="home" data-iconpos="top">Home</a>
<a href="" data-role="button" data-icon="home" data-iconpos="bottom">Home</a>
</div> <!-- Ende content-->
```

Buttons platzsparend anordnen:

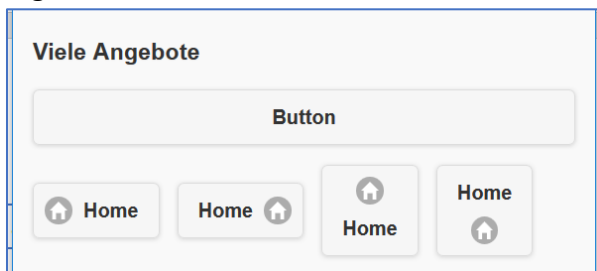
```
data-inline="true"
```

Um platzsparend zu arbeiten, verwendet man das data-Attribut „data-inline=“true“. Er verwendet dann nur so viel Platz, wie er benötigt, um den Text vollständig darzustellen.

Übung:

Versieh alle erstellten vier Buttons mit dem „inline“-Element.

Ergebnis:



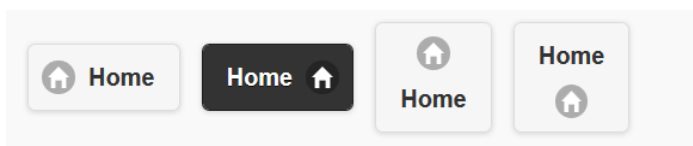
Optische Anpassung des Buttons:

In der Standardversion von jQuery Mobile werden zwei Themes mitgeliefert. Alle Elemente innerhalb einer Page werden standardmäßig im Theme „c“ dargestellt. Möchte man einen Button besonders hervorheben, kann man ihn über ein anderes Theme einfärben:

```
data-theme="b"
```

Übung:

Färbe den zweiten Button um.



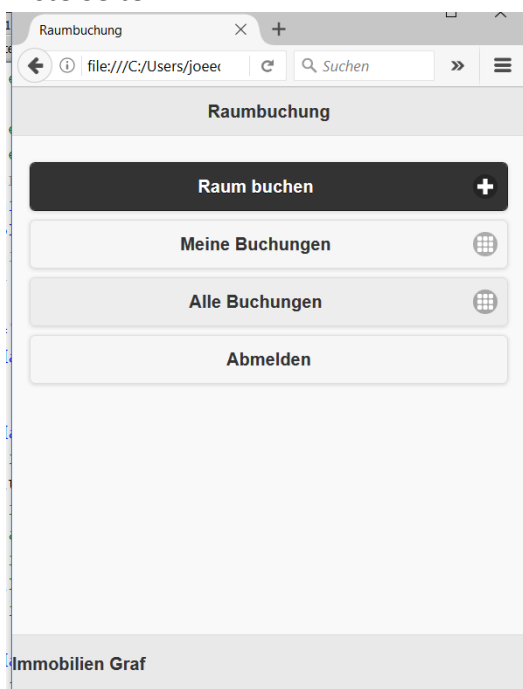
4)Übung Wohnungskauf:

Arbeite mit dem bereits erstellten Beispiel weiter. Erstelle zwei weitere Seiten für die Verwaltung von Raumreservierungen.

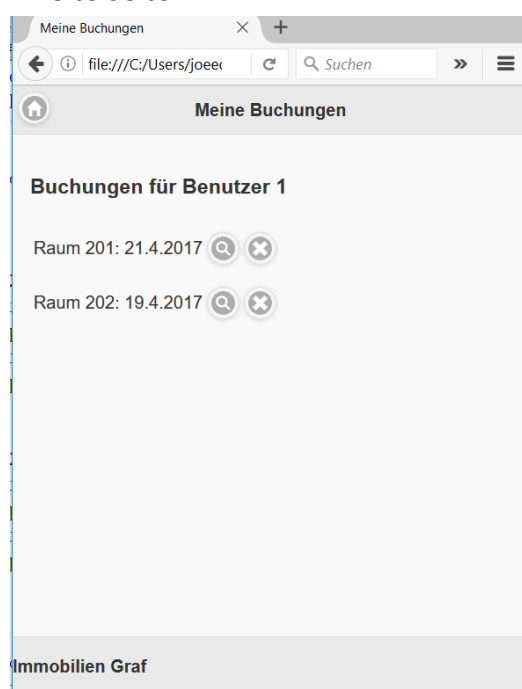
- Die erste Seite soll nach dem Klick auf „Button“, der noch in „Räume“ umbenannt werden soll, sich öffnen. Die Page soll heißen. „raumBuchung“. Die Menübuttons sollen alle ein Icon auf der rechten Seite haben. Einmal „plus“, zweimal „grid“, bei „Abmelden“ kein Icon. Am unteren Ende befindet sich noch ein „Abmelden“-Button. Der erste Button oben „Raum buchen“ soll farblich hervorgehoben sein. Erstelle den gleichen Footer wie schon oben.
- Die zweite Seite soll eine Liste gebuchter Räume anzeigen. Sie soll heißen „raumGebucht“. Sie soll sich öffnen, wenn man auf den Button „Meine Buchungen“ klickt. Im Header soll sie ein Icon für die erste Seite-Verlinkung haben. Sie beinhaltet unterhalb der Überschrift „Buchungen für Benutzer 1“ zwei Zeilen, die die jeweils einen Raum mit Nummer , einem Datum und zwei Buttons aufweisen. Die Buttons haben „notext“ (data-iconpos=“notext“) und sind einmal „search“ und das andere mal „delete“. Verwende für die beiden Einträge „Raum 201 und 202“ eine Tabelle <table> für die Anzeige.

**Tipp: Zum Weiterleiten gib den ID-Namen in der Weiterleitung an: **

Erste Seite:



Zweite Seite:



Tipp: Als Bezeichnung für Header, Content und Footer wähle auch immer den Page-Namen, z.B. <div id="raumGebuchtHeader" data-role="header">, <div id="raumGebuchtContent" data-role="content">

Code:

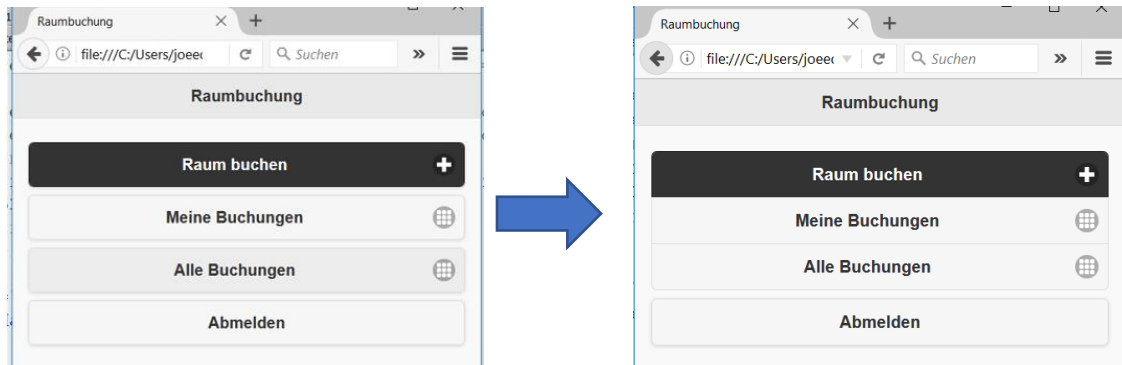
```
31 <!--Start page 2-->
32 <div data-role="page" id="raumBuchung" >
33   <div id="raumBuchungHeader" data-role="header">
34     <h3>Raumbuchung</h3>
35   </div>
36   <div id="raumBuchungContent" data-role="content">
37     <a href="#" data-role="button" data-icon="plus"
38       data-iconpos="right" data-theme="b">Raum buchen</a>
39     <a href="#raumGebucht" data-role="button" data-icon="grid"
40       data-iconpos="right">Meine Buchungen</a>
41     <a href="#" data-role="button" data-icon="grid"
42       data-iconpos="right">Alle Buchungen</a>
43     <a href="#" data-role="button">Abmelden</a>
44   </div>
45   <div id="raumBuchungFooter" data-role="footer" data-position="fixed">
46     <p>Immobilien Graf</p>
47   </div> <!-- Ende footer-->
48 </div> <!--Ende page 2-->
```

```
49 <!--Start page 3-->
50 <div data-role="page" id="raumGebucht" >
51   <div id="raumGebuchtHeader" data-role="header">
52     <h3>Meine Buchungen</h3>
53     <a href="#raumBuchung" data-role="button" data-icon="home"
54       data-iconpos="notext"></a>
55   </div>
56   <div id="raumGebuchtContent" data-role="content">
57     <h3>Buchungen für Benutzer 1</h3>
58     <table>
59       <tr>
60         <td>Raum 201: 21.4.2017</td>
61         <td><a href="#" data-role="button" data-icon="search"
62           data-iconpos="notext"></a> </td>
63         <td><a href="#" data-role="button" data-icon="delete"
64           data-iconpos="notext"></a> </td>
65       </tr>
66       <tr>
67         <td>Raum 202: 19.4.2017</td>
68         <td><a href="#" data-role="button" data-icon="search"
69           data-iconpos="notext"></a> </td>
70         <td><a href="#" data-role="button" data-icon="delete"
71           data-iconpos="notext"></a> </td>
72       </tr>
73     </table>
74   </div>
75   <div id="raumGebuchtFooter" data-role="footer" data-position="fixed">
76     <p>Immobilien Graf</p>
77   </div> <!-- Ende footer-->
78 </div> <!--Ende page 3-->
79 </body>
```

5)Controlgroup:

Controlgroups werden zur Gruppierung von Buttons verwendet, um sie optisch zusammengehörig darzustellen.

Im obigen Beispiel kann man z.B. die Menübuttons mit einer Controlgroup zu einem zusammengehörigen Block bündeln.



Man benötigt einen zusätzlichen div-Tag und vergibt dem data-role-Attribut den Wert „controlgroup“.

Übung: erstelle dies bei den „Raumbuchungen“ und zwar für die ersten drei Buttons.

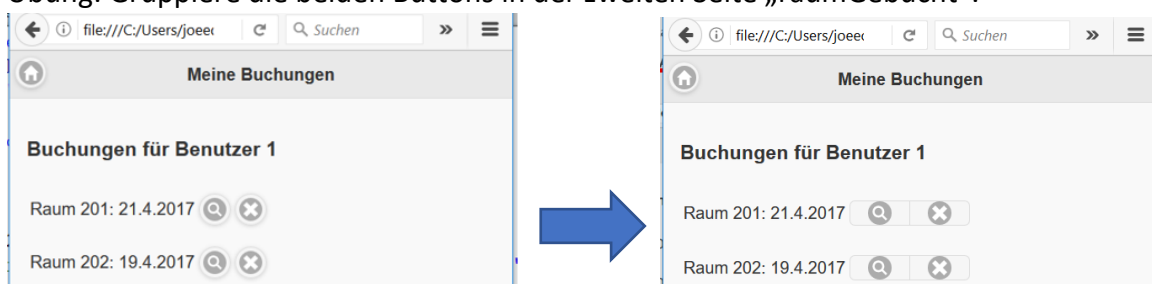
Beginne nach dem <div> und vor dem <a href> und beende das neue <div> vor „Abmelden“:

```
36 <div id="raumBuchungContent" data-role="content">
37 <div data-role="controlgroup">
38 <a href="#" data-role="button" data-icon="plus"
39 data-iconpos="right" data-theme="b">Raum buchen</a>
40 <a href="#raumGebucht" data-role="button" data-icon="grid"
41 data-iconpos="right">Meine Buchungen</a>
42 <a href="#" data-role="button" data-icon="grid"
43 data-iconpos="right">Alle Buchungen</a>
44 </div>
45 <a href="#" data-role="button">Abmelden</a>
46 </div>
```

Horizontale Controlgroup:

Man kann statt dem vertikalen auch eine horizontale Anwendung erstellen. Hierfür benötigt man das Attribut „data-type=“horizontal“.

Übung: Gruppier die beiden Buttons in der zweiten Seite „raumGebucht“.



Füge dazu ein `<div>` und das „`data-role="controlgroup"`“ ein und „`data-role="horizontal"`“ vor dem ersten ``:

```
<tr>
  <td>Raum 201: 21.4.2017</td>
  <td><div data-role="controlgroup" data-type="horizontal">
    <a href="#" data-role="button" data-icon="search"
    data-iconpos="notext"></a>
    <a href="#" data-role="button" data-icon="delete"
    data-iconpos="notext"></a>
  </div><td>
</tr>
```

Beachte, dass die Struktur der Tabelle nicht verloren geht. Dafür muss mit dem `<td>` und `</td>` gut umgegangen werden.

6)Listview

Damit kann man sehr gut Daten auflisten, bzw. auch eine Navigation bauen (Siehe 6.jquerymobile2.docx).

Die Basis der Listview ist ein HTML-Listen-Tag mit `` oder ``.

Die einzelnen Datenknoten werden in ``-Tags eingebettet. Das Einzige neue ist das „data-role-Attribut“.

Übung:

Erstelle eine neue Page 4, indem du die Elemente außer dem `<table>` von „page 3“ kopierst und die „id“ ersetzt durch „listview“:

```
84 </div> <!--Ende page 3-->
85 <!--Start page 4-->
86 <div data-role="page" id="listview" >
87   <div id="listviewHeader" data-role="header">
88     <h3>Meine Buchungen</h3>
89     <a href="#raumBuchung" data-role="button" data-icon="home"
90       data-iconpos="notext"></a>
91   </div>
92   <div id="listviewContent" data-role="content">
93     <h3>Listview</h3>
94   </div>
95   <div id="listviewFooter" data-role="footer" data-position="fixed">
96     <p>Immobilien Graf</p>
97   </div> <!-- Ende footer-->
98 </div> <!--Ende page 4-->
99 </body>
100 </html>
```

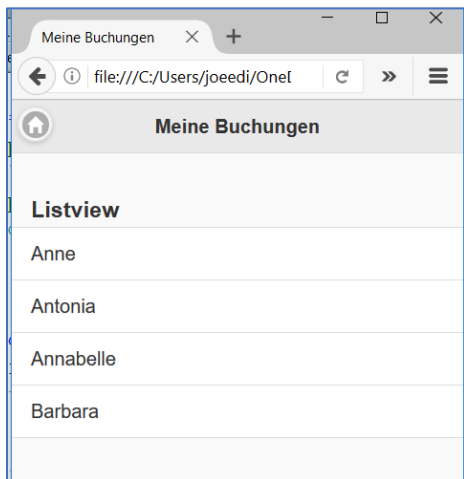
Damit die Seite auch aufgerufen werden kann, muss man oben auch die „id“ eintragen: in Zeile 42 für den Button „Alle Buchungen“

```
41     <a href="#" data-role="button" data-icon="grid"
42       data-iconpos="right">Meine Buchungen</a>
43     <a href="#listview" data-role="button" data-icon="grid"
44       data-iconpos="right">Alle Buchungen</a>
45   </div>
46   <a href="#" data-role="button">Abmelden</a>
```

Ändere in Zeile 88 die Überschrift auf „Listview“, ebenfalls im Content in Zeile 93. Nun erstelle die Listview im „content“ unter der Überschrift `<h3>Listview</h3>`:

```
92   <div id="listviewContent" data-role="content">
93     <h3>Listview</h3>
94     <ul data-role="listview">
95       <li>Anne</li>
96       <li>Antonia</li>
97       <li>Annabelle</li>
98       <li>Barbara</li>
99     </ul>
100   </div>
```

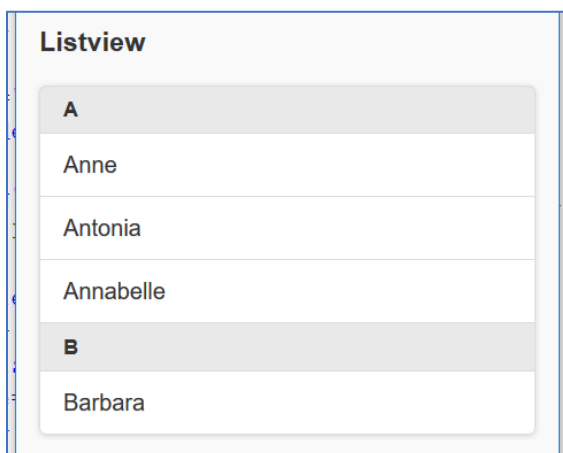
Ergebnis:



Konfiguration: Mit dem **„data-inset=“true“-Attribut“** wird ein Abstand zum Rand erzeugt und die Ecken abgerundet dargestellt.

Zur besseren Übersicht kann man an passenden Stellen Trennelemente einfügen. Dies geschieht über die Verwendung eines „data-role-Attributs“ mit **„list-divider“**. Standardmäßig wird für das Trennelement das b-Theme verwendet. Man kann aber auch andere Themes auswählen, indem man das data-Attribut „data-divider-theme“ verwendet.

```
93     <h3>Listview</h3>
94     <ul data-role="listview" data-inset="true">
95         <li data-role="list-divider">A</li>
96         <li>Anne</li>
97         <li>Antonia</li>
98         <li>Annabelle</li>
99         <li data-role="list-divider">B</li>
100        <li>Barbara</li>
101    </ul>
102 </div>
```



7)Panel Widget

Dieses stellt eine weitere Content-Ebene dar, die zusätzlich zum Seiteninhalt eingeblendet werden kann. Man kann es gut für umfangreiche Menüs, Zusatzinformationen oder kleinere Formulare (z.B. Login) nutzen.

Panels müssen in einer Page vor dem Header oder nach dem Footer definiert, dazwischen ist es nicht erlaubt.

Die Kennzeichnung als Panel erfolgt über das data-Attribut **data-role="panel"**.

Geöffnet können Panels werden, mit einem Link auf die ID des Panels, z.B. #panelPage01“.

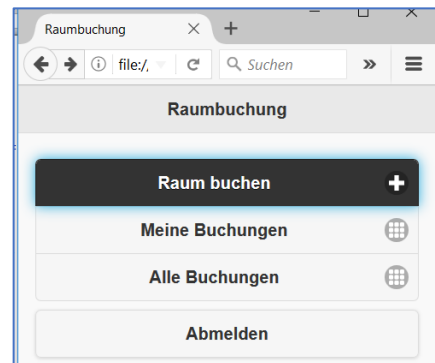
Beispiel:

Der Button „Raum buchen“ soll eine Seite öffnen, die ein Panel aufweist.

Kopiere daher „page 4“ und mache „page 5“ daraus. Lösche alles bezüglich „listview“ und ersetze bei header, content und footer die entsprechenden Elemente.

In Zeile 108 soll die ID „panelPage01“ heißen und einen einfachen Text „Text im Panel geschrieben...“ beinhalten.

Darüber schreibe das <div> mit der data-role="panel" und der entsprechenden ID, die weiter unten im <header> aufgerufen wird – siehe Zeile 114.



```
106 </div> <!--Ende page 4-->
107 <!--Start page 5-->
108 <div data-role="page" id="panelPage01">
109   <div data-role="panel" data-position="left" id="panelPage01Left">
110     <p>Text im Panel geschrieben...</p>
111   </div>
112   <div id="panelPage01Header" data-role="header" >
113     <h3>Panel</h3>
114     <a href="#panelPage01Left">Panel links</a>
115   </div>
116   <div id="panelPage01Content" data-role="content">
117
118   </div>
119   <div id="panelPage01Footer" data-role="footer" data-position="fixed">
120     <p>Immobilien Graf</p>
121   </div> <!-- Ende footer-->
122 </div> <!--Ende page 5-->
123 </body>
124 </html>
```

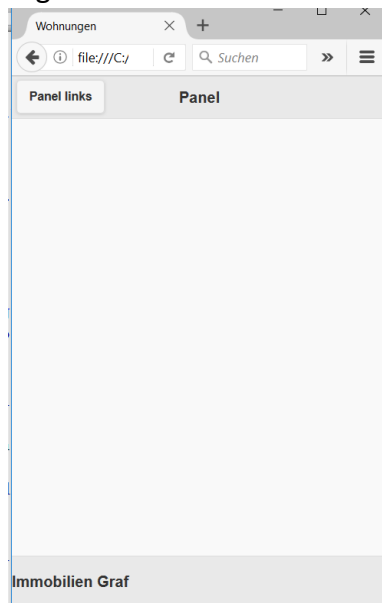
Der Button soll den Inhalt „Panel links“ haben.

Damit die Seite aufgerufen werden kann, muss der noch leere Button auf der „page 2“ in der Zeile 38 noch die ID erhalten, die ihn zur Seite 5 führt:

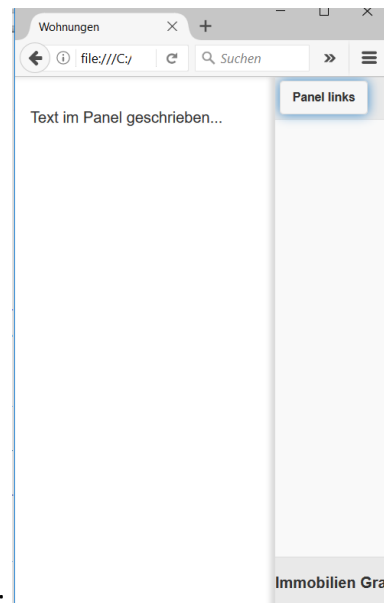
```
<a href="#panelPage01">
```

```
36 <div id="raumBuchungContent" data-role="content">
37 <div data-role="controlgroup">
38 <a href="#panelPage01" data-role="button" data-icon="plus"
39 data-iconpos="right" data-theme="b">Raum buchen</a>
40 <a href="#raumGebucht" data-role="button" data-icon="grid"
41 data-iconpos="right">Meine Buchungen</a>
```

Ergebnis:



Nach dem Klick:



Das Panel kann entweder rechts oder links positioniert werden, was über das data-Attribut „data-position“ mit den Werten **left** oder **right** gesteuert wird. Es kann auch ein eigenes Thema haben, z.B. data-theme="b".

Das Panel kennt drei unterschiedliche Animationseffekte: overlay, reveal und push. Sie können als Wert dem data-attribut „data-display“ mitgegeben werden.

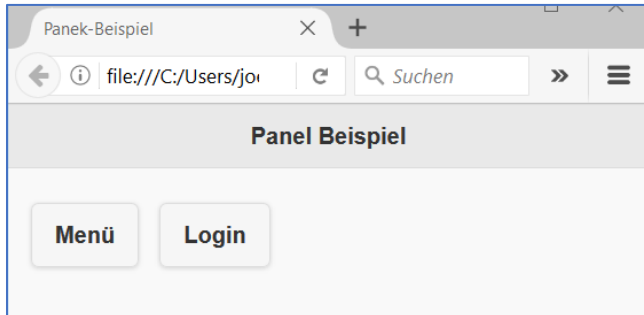
- **data-display="overlay"**: Der „overlay“-Effekt bewirkt, dass sich das Panel optisch über den Inhalt der Seite legt und der Seiteninhalt nicht zur Seite verschoben, sondern teilweise vom Inhalt des Panels verdeckt wird.
- **data-display="reveal"**: Beim Effekt „reveal“ sieht es so aus, als ob der Inhalt der Seite über dem Panel zur Seite rutscht, dabei das Panel statisch und sichtbar wird.
- Der Effekt „push“ ist der Default-Effekt, der verwendet wird, wenn nicht explizit ein anderer Effekt über „data-display“ angegeben ist.

Für den Fall, dass keine der drei Panel-Animationen erwünscht ist, lassen sich diese auch mit dem data-animation-Attribut durch das Setzen des Wertes „false“ abschalten.

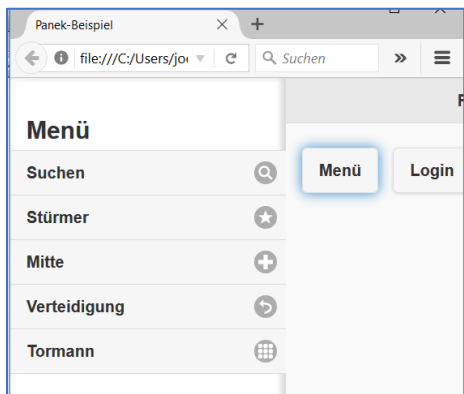
Weiteres Beispiel:

Erstelle eine „3.panel.html“ mit Ziel, dass ein Label links auf den Klick auf den Button „Menü“ reagiert und ein Label rechts aufgehen kann, das ein Login enthalten soll.

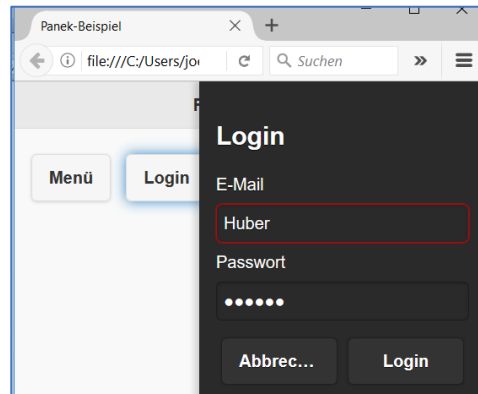
Diese beiden Buttons sollen also nach einem Klick jeweils ein Label einblenden, einmal von links, das andere Mal von rechts.



„reveal“



„overlay“



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Panek-Beispiel</title>
7 <link rel="stylesheet" href="lib/jquery.mobile-1.4.5.min.css">
8 <script src="lib/jquery-2.1.4.min.js"></script>
9 <script src="lib/jquery.mobile-1.4.5.min.js"></script>
10 </head>
11 <body>
12 <div data-role="page">
13   <div data-role="header">
14     <h1>Panel Beispiel</h1>
15   </div>
16   <div data-role="content">
17     <a href="#panel1" data-role="button" data-inline="true">Menü</a>
18     <a href="#panel2" data-role="button" data-inline="true">Login</a>
19   </div>
```

```

20 <!--Panel eins Definition-->
21   <div data-role="panel" id="panel1" data-position="left"
22     data-display="reveal" data-theme="a">
23     <h2>Menü</h2>
24     <ul data-role="listview" data-theme="a">
25       <li data-icon="search"><a href="search.html">Suchen</a></li>
26       <li data-icon="star"><a href="stuermer.html">Stürmer</a></li>
27       <li data-icon="plus"><a href="mitte.html">Mitte</a></li>
28       <li data-icon="back"><a href="verteidigung.html">
29         Verteidigung</a></li>
30       <li data-icon="grid"><a href="tor.html">Tormann</a></li>
31     </ul>
32   </div> <!--panel eins fertig-->

```

```

33 <!--Panel zwei Definition-->
34   <div data-role="panel" id="panel2" data-position="right"
35     data-display="overlay" data-theme="b">
36     <form class="userform">
37       <h2>Login</h2>
38       <label for="email">E-Mail</label>
39       <input type="email" name="email" id="email">
40       <label for="epassword">Passwort</label>
41       <input type="password" name="password" id="pwd">
42
43       <div class="ui-grid-a">
44         <div class="ui-block-a">
45           <a href="#" data-rel="close" data-role="button"
46             data-theme="c">Abbrechen</a>
47         </div>
48         <div class="ui-block-b">
49           <a href="#" data-rel="close" data-role="button"
50             data-theme="b">Login</a>
51         </div>
52       </div>
53     </form>
54   </div> <!--panel zwei fertig-->
55 </div> <!--Page Ende-->
56 </body>
57 </html>

```

Quellen:

Philipp Friberg, in: Web-Apps mit jQuery Mobile, Mobile Multiplattform-Entwicklung mit HTML5 und JavaScript, dpunkt. Verlag, Heidelberg, 2013

Marco Dierenfeldt, in: jQuery Mobile, Einfach mobile Web-Apps entwickeln. Verlag entwickler. press, Kapitel 1-2, Frankfurt, 2014