

jQuery Mobile -das jQuery-Framework für WebApps



jQuery Mobile (<http://jquerymobile.com>) ist ein Open-Source-Projekt, das auf den weit verbreiteten JavaScript-Bibliotheken jQuery und jQuery UI aufbaut. Es steht unter den Lizenzen MIT und GPL zur Verfügung.

Es ist ein sogenanntes Web-Framework, das ein Grundgerüst für die Entwicklung mobiler Web-Applikationen zur Verfügung stellt. Es basiert auf den Technologien HTML5, CSS3 und JavaScript bzw. Ajax. Allerdings ist ein Mobilgerät kein Desktop-Computer und verfügt meist weder über Tastatur noch Maus als Eingabegeräte. Vielmehr haben sich zunehmend Touchdisplays als Eingabemedium durchgesetzt. Dies hat entsprechend Folgen für das Event-Handling, die überdacht werden müssen:

- Touch-Events sind keine Mouse-Events
- Wischen und Blättern
- Es treten neue Events auf wie z.B. Mehrfingergeräten für Zoom

jQuery-Mobile-Websites laufen u.a. unter folgenden mobilen Systemen Apple iOS, Android. Der Webbrowser ist meist „Webkit“ also Safari, bzw. Google Chrome oder Gecko, sprich Firefox oder Opera mini.

Demoseiten findet man unter <http://jquerymobile.com/demos>

Weitere Sites: m.ikea.com, m.stanford.edu, m.disneyworld.disney.go.com.

Meist werden die mobilen Websites über eine Subdomain „m“ aufgerufen, statt über „www“.

Alternativen zu jQuery Mobile:

Sencha Touch (<http://www.sencha.com/products/touch>) bietet noch mehr Möglichkeiten als jQuery Mobile. Damit erstellte Websites sehen meist zu 100% wie native Apps aus.

jQuery-Mobile baut auf sauberen, semantisch korrekt geschriebenen HTML5 auf, was die Kompatibilität mit den meisten anderen webfähigen Geräten sichert. Es wird einerseits die Barrierefreiheit gewährleistet. Da jQuery-Mobile die Barrierefreiheit nach WAI-ARIA implementiert, unterstützt das Framework auch Endgeräte wie Bildschirmleser oder andere Hilfstechnologien.

Die wichtigsten jQuery-Mobile-Links

jQuery Mobile	http://jquerymobile.com
jQuery-Mobile-Blog	http://jquerymobile.com/blog/
jQuery-Mobile-Forum	http://forum.jquery.com/jquery-mobile/
jQuery-Mobile-Dokumentation	http://api.jquerymobile.com
jQuery-Dokumentation	http://docs.jquery.com

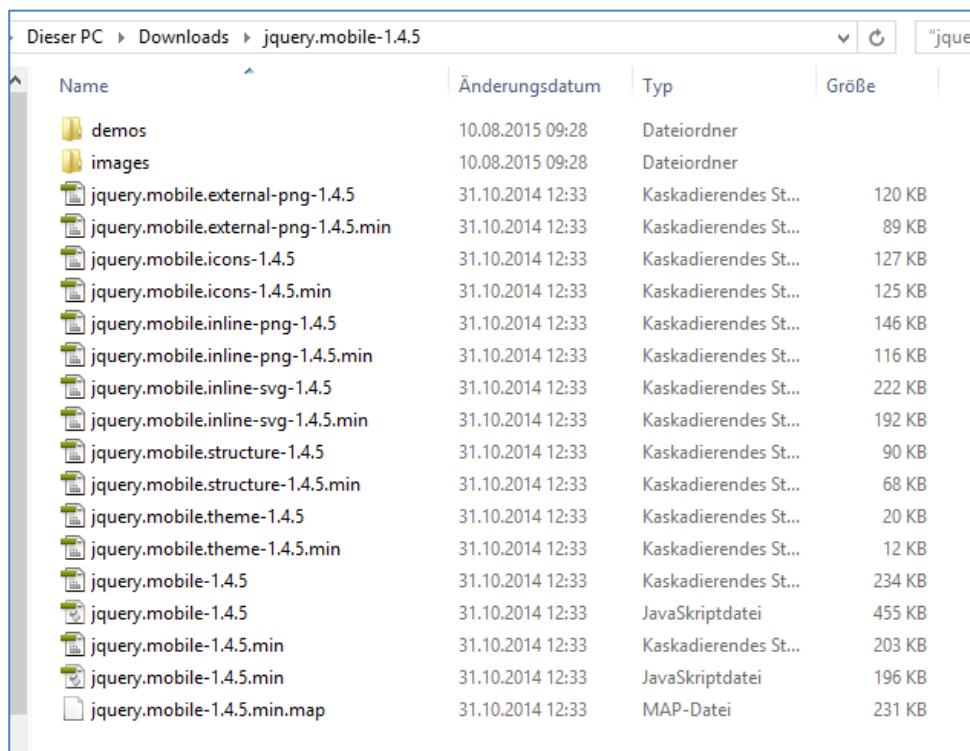
<http://www.w3schools.com/jquerymobile/default.asp>

<http://demos.jquerymobile.com/1.4.5/>

1. Download: latest stable



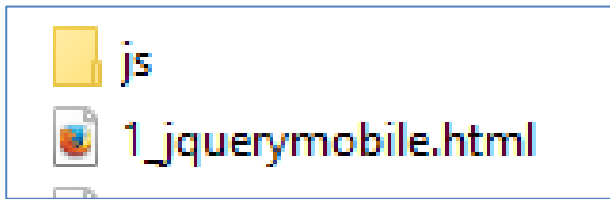
Nach dem Download der aktuellsten Version entzippe den Ordner:



Wir benötigen die folgenden beiden Dateien und den Ordner „images“:

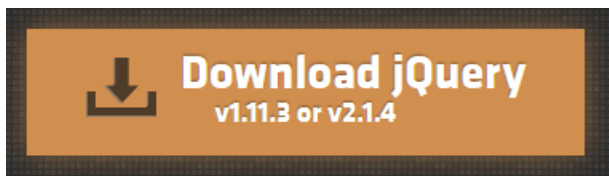
- jquery.mobile-1.4.5.min.js
- jquery.mobile-1.4.5.min.css

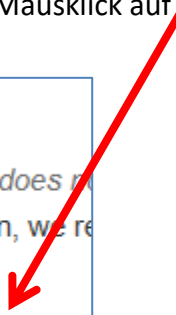
Diese kopiere in einen neuen Ordner „js“ , der sich im gleichen Verzeichnis wie die Haupt-HTML befindet. Es kann auch der Ordner „lib“ für „library“ sein.

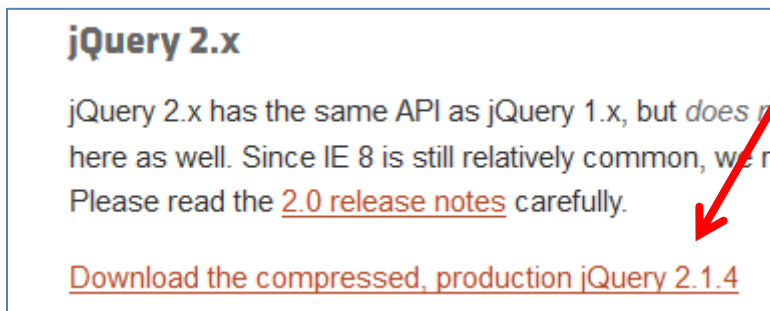


Im „images“ befinden sich die verschiedenen Pfeile und Glyphicons.

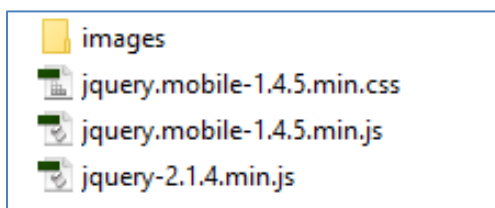
2. Download jQuery von www.jquery.com



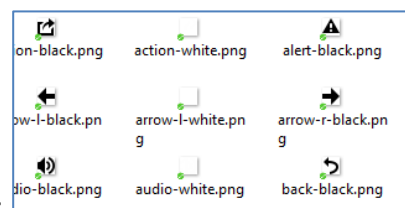
Nutze die jQuery 2.1.4 compressed version: rechter Mausklick auf  und dann „Ziel speichern unter“ den Ordner „lib“ bzw. „js“ wählen:



Ergebnis:



im images-Ordner:



Sinnvoll: Wenn man will, kann man die Dateien umbenennen in

- jquery.js
- jquery.mobile.js
- jquery.mobile.css.

Dies vereinfacht die Einbindung. Bei einem späteren Wechsel der Version muss man zudem nicht die Links in allen Dateien ändern.

Oder direkt hier mit der Verlinkung direkt zu den aktuellen Dateien:

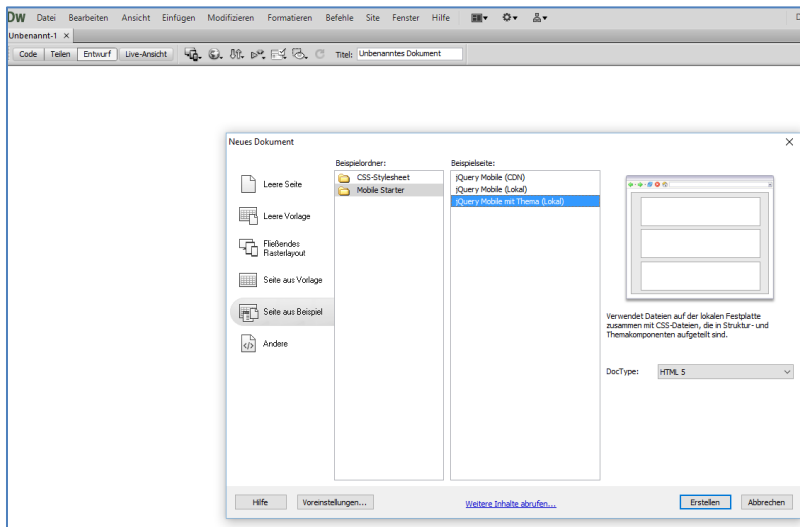
Die lokale Kopie von jQuery Mobile kann unter <http://jquerymobile.com/download/> heruntergeladen werden. In jQM ist jQuery noch nicht enthalten, deshalb muss man es separat unter <http://jquery.com/download/> besorgen. Auf der Download-Seite von jQM steht jeweils, welche jQuery-Version welche jQM-Version unterstützt. Die Dateien kopieren wir in das lib-Verzeichnis und passen die Pfade im <head>-Element der index.html-Datei an.

Info:

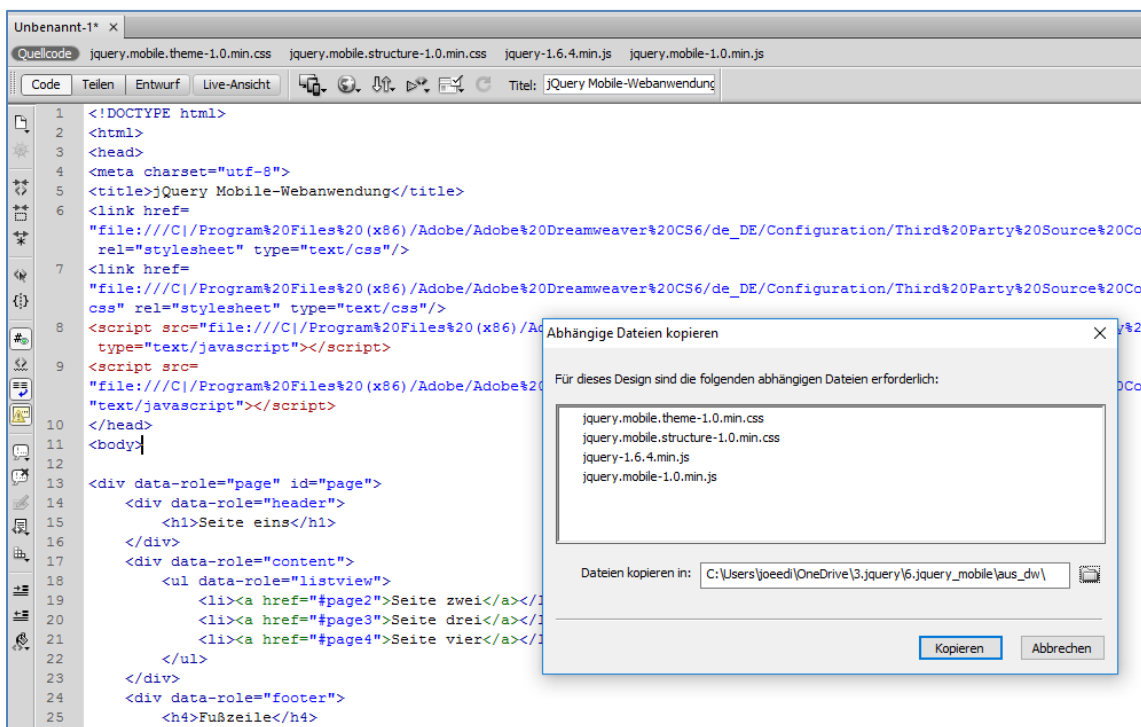
jQuery Mobile im Dreamweaver

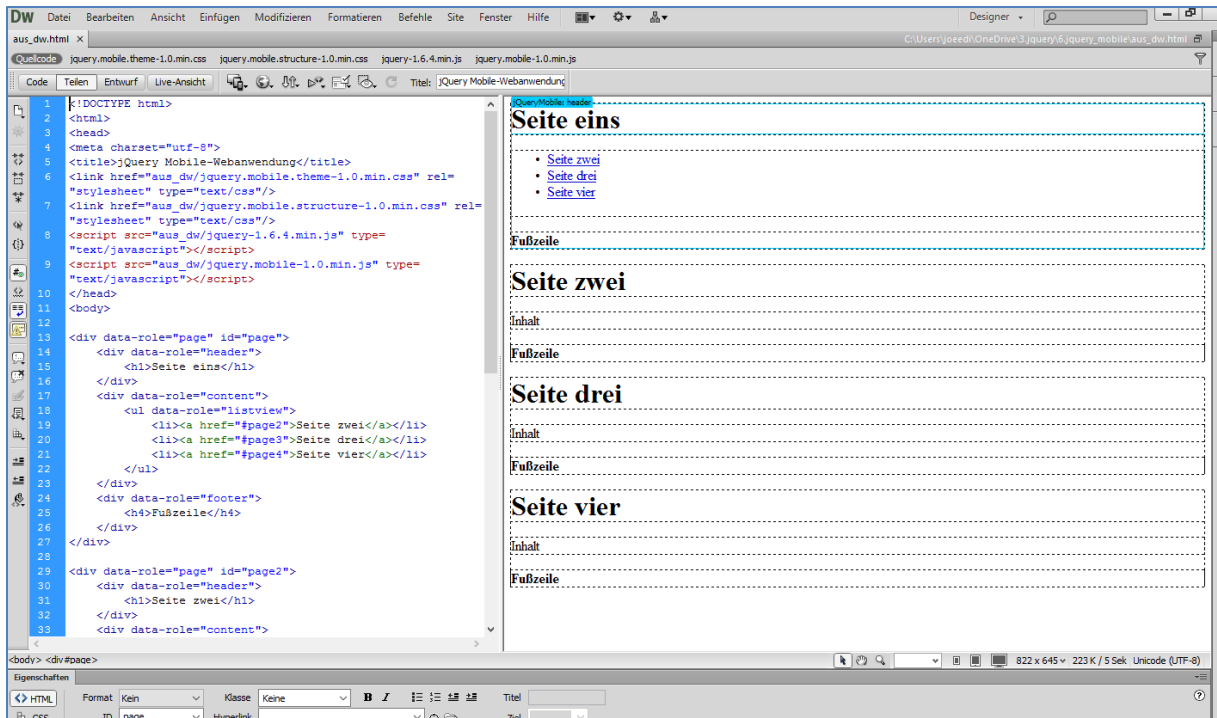
Mit **Dreamweaver** ab der Version CS5.5 lassen sich mit einer integrierten Funktion jQuery Mobile Web-Apps erstellen.

Menü Datei / Neu / „Seite aus Beispiel“:

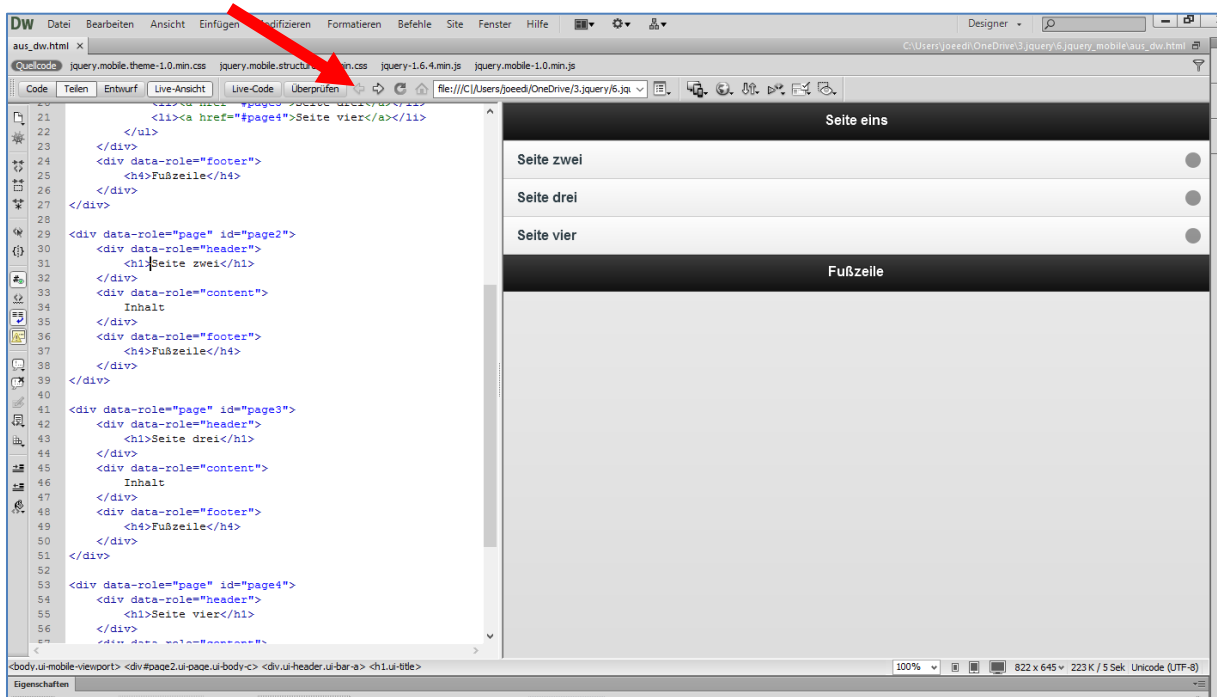


Nach dem Erstellen wird eine fertige Seite bereits angezeigt. Will man diese mit F12 betrachten, muss zuerst die Site gespeichert werden. Dabei erscheint ein Fenster, um die Begleitdateien zu kopieren und speichern. Wähle einen neuen Ordner mit einem entsprechenden Ordernamen z.B. „aus_dw“ aus:

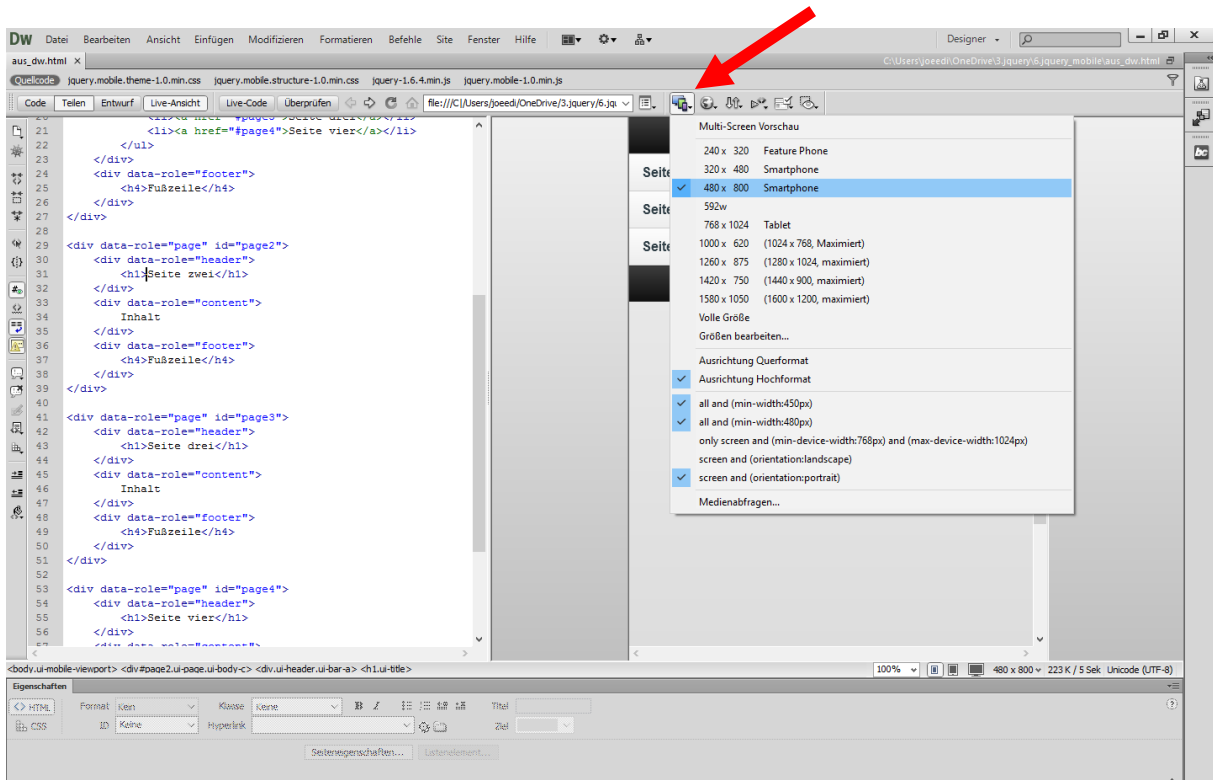




Klicke auf „Live-Ansicht“. Es kann sogar in der Site geklickt werden. Zurück geht auch mit Klick auf den Pfeil

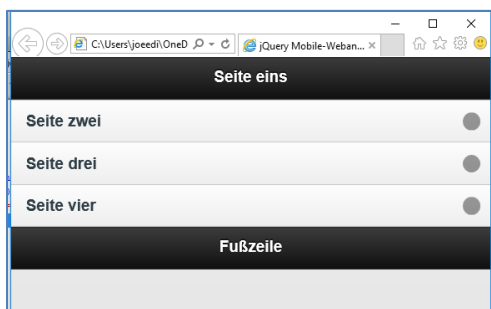


Zusätzlich bietet Dreamweaver einen „Multi-Screen-Modus“, der in einem Popup die aktuell bearbeitete Seite in mehreren Viewport-Simulatoren parallel zeigen kann:

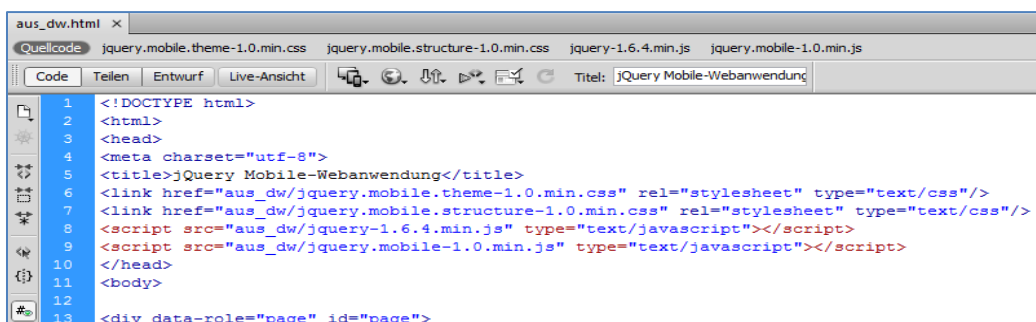


In der Live- wie Multiview kann man zwar ganz gut navigieren. Aber man kann nicht zu viel erwarten. Es ist nämlich kein echter „Emulator“, dennoch ist es ein höchst nützliches Tool

Fertige Seite:



Tip: Das Aktualisieren der älteren Dateien ist gar nicht erfolgreich. Bleibe daher bei:



Theorie:

Allgemeine Struktur:

jQuery Mobile Dateien im Header einbinden:

Der Aufbau einer solchen HTML-Seite, in der jQuery Mobile verwendet werden soll:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="lib/jquery.mobile-1.4.5.css">
<script src="lib/jquery-2.1.4.min.js"></script>
<script src="lib/jquery.mobile-1.4.5.js"></script>
</head><title>Mobile</title>
</head>
<body>
<!-- ... -->
</body>
```

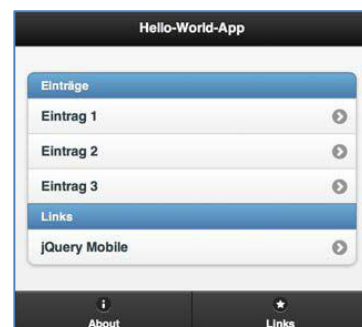
Für jQuery-Mobile-HTML-Seiten muss der HTML5 Doctype **<!DOCTYPE html>** verwendet werden. Dies stellt sicher, dass *data*-Tags korrekt unterstützt werden.

Zunächst konfigurieren wir das Meta-Tag **viewport** so, dass die Skalierung der Seite durch den Benutzer nicht geändert werden kann. Anschließend binden wir das Standard-CSS-Style-Sheet von jQuery Mobile ein und dann die JavaScript-Dateien von jQuery und jQuery Mobile. Wir laden die Dateien aus einem Unterverzeichnis unseres Applikationsverzeichnis.

Andere Möglichkeit: einbinden mit „CDN-Repository“:

Alternativ kann man die Bibliotheken auch direkt aus dem Internet über das Content Delivery Network (CDN) der jQuery-Projekte laden. Dies bietet sich beispielsweise für öffentlich erreichbare Anwendungen an.

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css"/>
<script src="http://code.jquery.com/jquery.min.js">
</script>
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
```



Pages:

Die Oberfläche einer jQuery-Mobile-Applikation ist aufgeteilt in so genannte Pages, die jeweils einen „screenful of content“ präsentieren. Navigiert der Benutzer durch eine App, dann wechselt er zwischen den Pages der App. **Diese Pages dürfen nicht mit einer HTML-Seite verwechselt werden**, deren Quelltext eine oder mehrere jQuery Mobile Pages enthalten kann. Wir sprechen daher im Folgenden von HTML-Dokumenten und jQuery Mobile Pages oder einfach nur Pages.

Im HTML-Markup wird eine Page durch ein **div mit data-role="page"** implementiert und durch die **Vergabe einer id referenzierbar** gemacht. Eine jQuery-Mobile-HTML-Seite kann mehrere solcher Pages enthalten. Man spricht in diesem Fall von einem „Multi-Page-Document“.

```
<body>
<div id="loginPage" data-role="page">
<!-- ... -->
</div>
<div id="welcomePage" data-role="page">
<!-- ... -->
</div>
<div id="profilePage" data-role="page">
<!-- ... -->
</div>
</body>
```

Wird dieses HTML-Dokument vom Browser geladen, dann veranlasst jQuery Mobile die Anzeige der ersten Page, die in dem Dokument gefunden wird. Die nachfolgenden Pages können per Navigation beispielsweise über einen

Link [Go to Page Two](#),

zur Anzeige gebracht werden. Dieses Vorgehen hat den Vorteil, dass alle Pages im Dokument mit einem Request geladen und bei Bedarf ohne Nachladen angezeigt werden.

Beispiel:

```
<div data-role="page" id="pageone">
  <div data-role="main">
    <a href="#pagetwo">Go to Page Two</a>
  </div>
</div>
```

```
<div data-role="page" id="pagetwo">
  <div data-role="main">
    <a href="#pageone">Go to Page One</a>
  </div>
</div>
```

(http://www.w3schools.com/jquerymobile/jquerymobile_pages.asp)

Die Pages einer App können aber auch auf mehrere HTML-Dokumente aufgeteilt werden, die man dann wie üblich verlinken kann. Wenn der Benutzer auf einen solchen Link klickt,

dann fängt jQuery Mobile diesen Request standardmäßig ab, lädt das HTML-Dokument mit Ajax und legt den Inhalt des Dokuments, eingebettet in einem Page div, im aktuellen DOM ab. Enthält das geladene Dokument bereits Page divs, dann wird der erste gefundene Page div im aktuellen DOM abgelegt. Anschließend wird die nachgeladene Page zur Anzeige gebracht. Will man das Nachladen per Ajax für Dokumente aus derselben Domäne deaktivieren, so kann man dies durch die Angabe von data-ajax="false" auf dem Link-Element erreichen.

Eine Page kann hinsichtlich ihres Layouts in einen

- Header-, einen
- Content- und einen
- Footer-Bereich unterteilt werden.

Im Header werden üblicherweise ein Seitentitel und ein oder zwei Buttons angezeigt, beispielsweise zur „Zurück-Navigation“ oder zum Aktivieren eines Bearbeitungsmodus.

Der Footer kann beispielweise eine Tab Bar zur Navigation in unterschiedliche Bereiche der Applikation aufnehmen.

Beispiel:

```
<body>
<div data-role="page">

  <div data-role="header">
    <h1>Welcome To My Homepage</h1>
  </div>

  <div data-role="main" class="ui-content">
    <p>I Am Now A Mobile Developer!!</p>
  </div>

  <div data-role="footer">
    <h1>Footer Text</h1>
  </div>

</div>
</body>
```

Start unseres Beispiels:

Homepage mit den typischen Elementen:

Sie soll aus zwei Seiten bestehen, wobei die erste Seite eine Liste beinhalten soll. Wenn auf einen Eintrag getippt wird, so wird die zweite Seite aufgerufen. Ein Kopf oben (header) und ein Menü unten (footer) runden die Webseite ab:



1) Erstellen eine kleine Webseite (html5) mit den oben genannten Bereichen: **ohne jQuery**

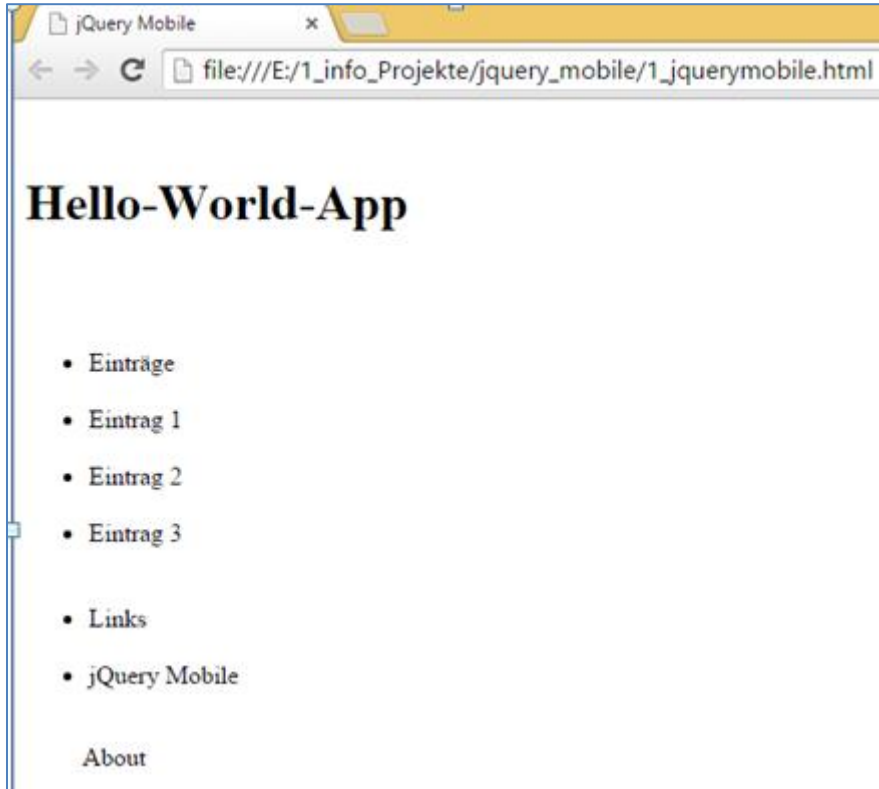
```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width,
6   initial-scale=1">
7 <title>jQuery Mobile</title>
8 </head>
9
10 <body>
11 <div id="welcome">
12   <h1>Hello-World-App</h1>
13   <ul>
14     <li>Einträge</li>
15     <li><a href= "eintrag1.html" >Eintrag 1</a></li>
16     <li><a href= "eintrag2.html" >Eintrag 2</a></li>
17     <li><a href="eintrag3.html" >Eintrag 3</a></li><br>
18   <br>
19     <li>Links</li>
20     <li><a href="http://jquerymobile.com">jQuery Mobile</a></li>
21   </ul>
22   <a href= "about.html">About</a>
23 </div>
```

Code zum Kopieren für den <body>

```
<body>
<div id="welcome">
  <h1>Hello-World-App</h1>
  <ul>
    <li>Einträge</li>
    <li><a href="eintrag1.html" >Eintrag 1</a></li>
    <li><a href="eintrag2.html" >Eintrag 2</a></li>
    <li><a href="eintrag3.html" >Eintrag 3</a></li><br>
  <br>
    <li>Links</li>
    <li><a href="http://jquerymobile.com">jQuery Mobile</a></li>
  </ul>
  <a href= "about.html">About</a>
</div>
</body>
```

Speichere die Site unter „1.jquerymobile.html“

Ergebnis: noch ohne jQuery



Erklärung:

- `<meta name="viewport" content="width=device-width, initial-scale=1">`
Der Viewport ist der Darstellungsbereich. Mit dieser Anweisung wird festgelegt, dass die darstellbare Breite als Content-Breite verwendet wird und der Benutzer nicht skalieren kann. Dies ist bei Web-Apps nicht erwünscht, bei normalen Homepages aber notwendig.

http://www.w3schools.com/jquerymobile/jquerymobile_get_started.asp

Nun müssen das Stylesheet und die JavaScript-Dateien hinzugefügt werden:

1. Zuerst sollen die jQuery Mobile CSS geladen werden.
2. Da jQuery Mobile auf jQuery aufsetzt, muss dann zuerst jQuery geladen werden. Nach dem Ausführen des Codes werden der Hintergrund und die Schrift angepasst angezeigt. **Leider noch nicht im Mobile-Style.**

Daher Code im <head>:

```
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/jquery.mobile-1.4.5.css">
<script src="js/jquery-2.1.4.min.js"></script>
<script src="js/jquery.mobile-1.4.5.js"></script>
<title>erste jQuery Mobile</title>
</head>
```

3. In einer HTML-Datei können mehrere jQM-Seiten enthalten sein. Diese Seiten werden mit <div>-Tags und einem entsprechenden Parameter gekennzeichnet: **<div data-role="page"> </div>**. Daran erkennt jQuery Mobile, dass es sich bei diesem div um das Markup für eine „Page“ im Sinne einer mobilen jQuery-Mobile-App handelt. Die Darstellung dieser Page wird von jQuery Mobile dann entsprechend der Möglichkeiten des darstellenden Browsers optimiert.

Füge dem bestehenden div-Tags „welcome“ das **data-role="page"**-Attribut hinzu. Mit der Rolle »page« teilen wir jQM mit, dass dieser Bereich eine Seite darstellt.

In jeder Page kann es nun weitere div-Elemente mit data-role-Attributen geben.

Der Titel-Bereich

- **header** wird mit <div data-role="header"> gekennzeichnet. Mit **data-position="fixed"** wird weiter angegeben, dass der Titel immer sichtbar sein soll, auch wenn gescrollt wird. Weitere data-role-Parameter sind:
- **content** für den Inhalt der Seite und
- **footer** für die Fußzeile. Es könnten auch die neuen HTML5-Elemente zur Seitenstrukturierung benutzt werden, jedoch muss man trotzdem die role-Attribute für jQM angeben.

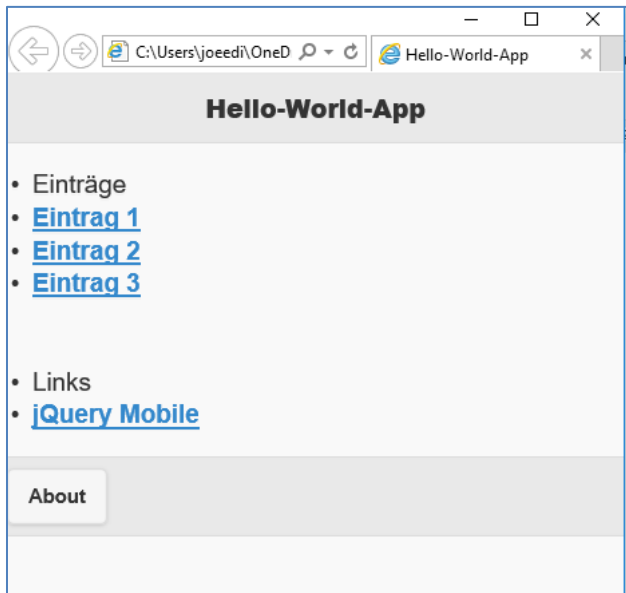
Page-Struktur:

```
<div data-role="page">
  <div data-role="header">...</div>
  <div data-role="content">...</div>
  <div data-role="footer">...</div>
</div>
```

Code:

```
<body>
<div id="welcome" data-role="page">
  <div data-role="header" data-position="fixed">
    <h1>Hello-World-App</h1>
  </div>
  <div data-role="content">
    <ul>
      <li>Einträge</li>
      <li><a href="eintrag1.html" >Eintrag 1</a></li>
      <li><a href="eintrag2.html" >Eintrag 2</a></li>
      <li><a href="eintrag3.html" >Eintrag 3</a></li>
      <br><br>
      <li>Links</li>
      <li><a href="http://jquerymobile.com">jQuery Mobile</a></li>
    </ul>
  </div>
  <div data-role="footer">
    <a href= "about.html">About</a>
  </div>
</div>
</body>
```

Ergebnis (Bildschirm verkleinert):



listview

Jetzt fehlt uns noch die typische Liste, die auf Mobiles anzutreffen ist. Auch zu diesem Zweck gibt es eine Rolle. Das ul-Element wird mit der Rolle **listview** versehen:

```
<ul data-role="listview" data-inset="true">
```

Das data-inset-Attribut steuert das Aussehen der Liste. Bei true wird ein Rand freigelassen, ansonsten füllt sie die Seitenbreite.

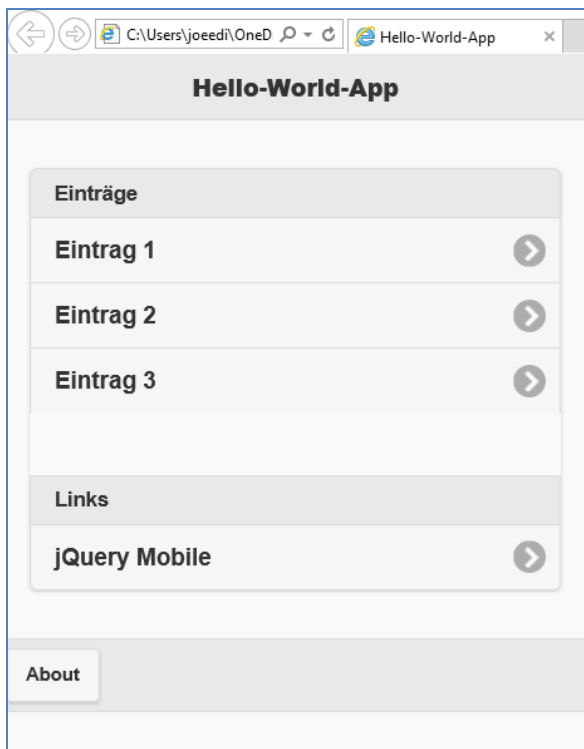
„list-divider“

Um Titelzeilen, wie zum Beispiel »Links« hervorzuheben, kann das entsprechende li-element mit der Rolle data-role="list-divider" versehen werden:

Beispiel für „Einträge“ und „Links“:

```
<div data-role="content">
  <ul data-role="listview" data-inset="true">
    <li data-role="list-divider">Einträge</li>
    <li><a href="eintrag1.html">Eintrag 1</a></li>
    <li><a href="eintrag2.html">Eintrag 2</a></li>
    <li><a href="eintrag3.html">Eintrag 3</a></li>
    <br><br>
    <li data-role="list-divider">Links</li>
    <li><a href="http://jquerymobile.com" >jQuery Mobile</a></li>
  </ul>
</div>
```

Jetzt sieht auch die Liste aus, wie man sie auf einem Mobile Device erwarten würde:

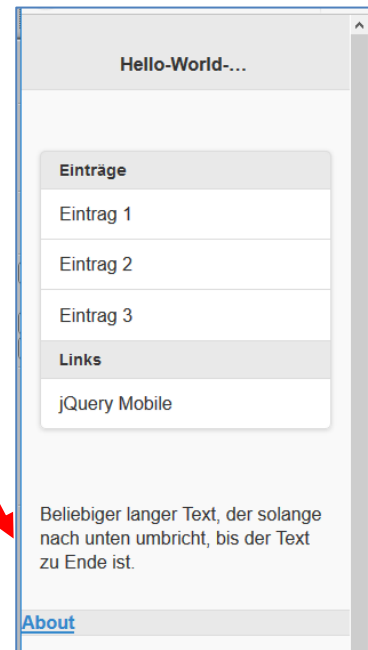


Normalen Text einfügen

der automatisch umgebrochen wird. Eine „listview“ und Elemente in der sind immer nur einzeilig.

Erstelle einfach ein neues
<div data-role="content"> außerhalb der
 von oben.

Eventuell müsste man mit <margin> arbeiten,
um links und rechts einen schönen Abstand
zwischen Text und Box zu erhalten.



```
16 </div>
17 <div data-role="content">
18   <ul data-role="listview" data-inset="true">
19     <li data-role="list-divider">Einträge</li>
20     <li><a href="eintrag1.html" >Eintrag 1</a></li>
21     <li><a href="eintrag2.html" >Eintrag 2</a></li>
22     <li><a href="eintrag3.html" >Eintrag 3</a></li>
23     <li data-role="list-divider">Links</li>
24     <li><a href="http://jquerymobile.com" >jQuery Mobile</a></li>
25   </ul>
26 </div>
27 <div data-role="content">
28 <p>Beliebiger langer Text, der solange nach unten umbricht,
29 bis der Text zu Ende ist.</p>
30 </div>
31 <div data-role="footer">
32   <li><a href="about.html">About</a></li>
33 </div>
34 </div>
```

rel="external"

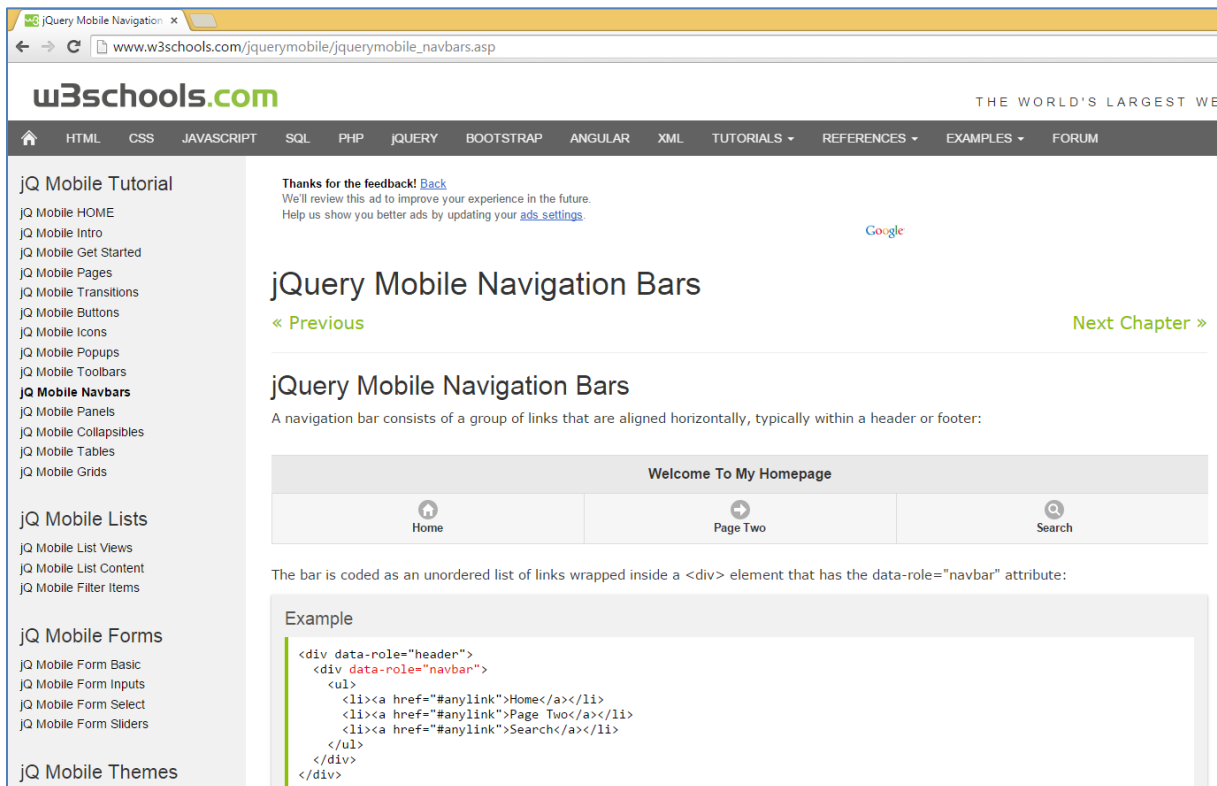
Wird auf einen der Hyperlinks geklickt, so wird die Seite per Ajax nachgeladen. Deshalb ist zwischen **externen Seiten und eigenen Seiten zu unterscheiden**. Bei externen Seiten muss dem a-Element das Attribut rel="external" hinzugefügt werden. Die Listendefinition sieht nun folgendermaßen aus:

```
<li data-role="list-divider">Links</li>
<li><a href="jquerymobile.com" rel="external">jQuery Mobile</a></li>
</ul>
```

Data-role="navbar"

Erzeugt horizontale Buttons

Info: http://www.w3schools.com/jquerymobile/jquerymobile_navbars.asp



The screenshot shows the w3schools.com website with a navigation bar at the top. The main content area is titled "jQuery Mobile Navigation Bars" and includes a "Previous" link and a "Next Chapter" link. Below the title, there is a navigation bar with three buttons: "Home", "Page Two", and "Search". The text below the navigation bar states: "A navigation bar consists of a group of links that are aligned horizontally, typically within a header or footer:". Below this text, there is an "Example" section with the following HTML code:

```
<div data-role="header">
  <div data-role="navbar">
    <ul>
      <li><a href="#anylink">Home</a></li>
      <li><a href="#anylink">Page Two</a></li>
      <li><a href="#anylink">Search</a></li>
    </ul>
  </div>
</div>
```

Änderung im „footer“ für ein kleines Menü:

Es soll neben „about“ ein zweiter Link eingefügt werden, in Form von Buttons, inkl. Symbol:

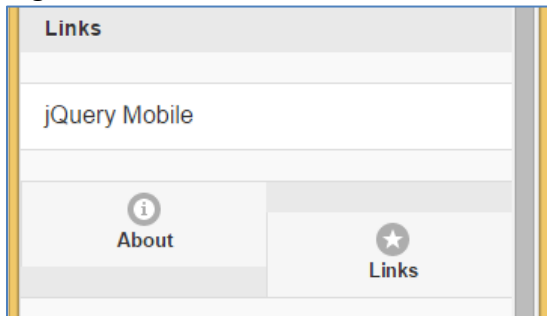
```
<div data-role="footer">
  <div data-role="navbar">
    <ul>
      <li><a data-icon="info" href="about.html">About</a></li>
      <li><a data-icon="star" href="details.html">Links</a></li>
    </ul>
  </div>
</div>
```

```
<div data-role="footer">
  <div data-role="navbar" ">
    <ul>
      <li><a data-icon="info" href="about.html">About</a></li>
      <li><a data-icon="star" href="details.html">Links</a></li>
    </ul>
  </div>
</div>
```

Die Liste der Buttons wird über eine HTML-Liste in Form von Links umgesetzt.

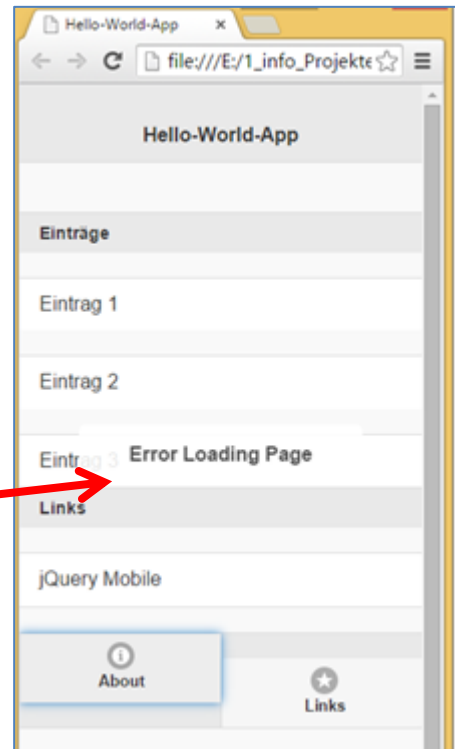
Mit dem Attribut **data-icon** kann eines der vordefinierten Icons verwendet werden.

Ergebnis:



Verlinkung der „about.html“

Wenn auf den Menüpunkt *About* geklickt wird, versucht jQM per Ajax die Datei *about.html* nachzuladen. Dies scheitert, und entsprechend wird eine Fehlermeldung ausgegeben:



Übung:

Erstellung der About-Seite:

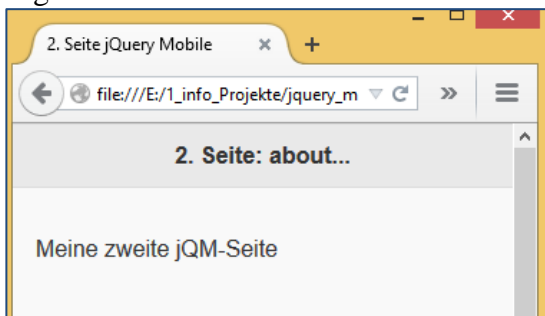
```
<!DOCTYPE HTML>
<html>
<head> <meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>2. Seite jQuery Mobile</title>
<link rel="stylesheet" href="lib/jquery.mobile-1.4.5.min.css">
<script src="lib/jquery-2.1.4.min.js"></script>
<script src="lib/jquery.mobile-1.4.5.min.js"></script>
</head>
<body>
<!-- Page About -->
<div data-role="page" id="about">
  <div data-role="header">
    <h1>2. Seite: about...</h1>
  </div>
<!-- Content -->
  <div data-role="content">
    <p>Meine zweite jQM-Seite</p>
  </div>
</div>
</body>
</html>
```

Speichere diese HTML Seite unter „about.html“

Name	Änderungsdatum	Typ	Größe
lib	12.08.2015 08:13	Dateiordner	
1_jquerymobile	14.08.2015 17:47	HTML-Datei	2 KB
about	14.08.2015 18:02	HTML-Datei	1 KB

Tipp: Wenn aus Dreamweaver heraus diese 2. Seite nicht gezeigt wird, öffne diese aus dem Windows-Explorer heraus.

Ergebnis:



Aber wie kommt der Benutzer nun zurück?

Button - Toolbars

Erzeuge einen Button z.B. im Header:

http://www.w3schools.com/jquerymobile/jquerymobile_toolbars.asp

Header Bars

The header is located at the top of the page and usually contain a page title/logo or one or two buttons (typically home, options or search).

You can add buttons to the left and/or to the right side in the header.

The code below, will add a "Home" button to the left and a "Search" button to the right of the header title text:

Example

```
<div data-role="header">
  <a href="#" class="ui-btn ui-icon-home ui-btn-icon-left">Home</a>
  <h1>Welcome To My Homepage</h1>
  <a href="#" class="ui-btn ui-icon-search ui-btn-icon-left">Search</a>
</div>
```

Übung:

Erstelle einen Home-Button in der „about.html“ im Header (linke Seite).

```
<a href="1_jquerymobile.html" class="ui-btn ui-icon-home ui-btn-icon-left">Home</a>
```

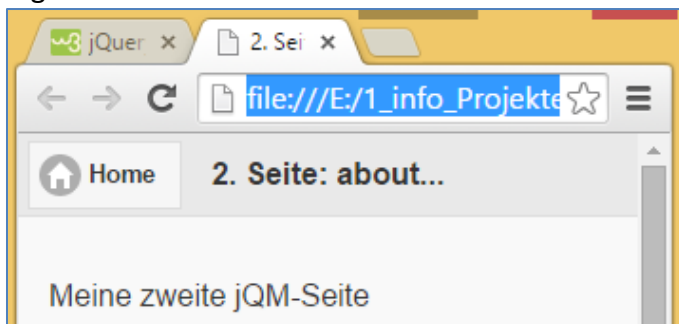
```

14 <body>
15 <!-- Page About -->
16 <div id="about" data-role="page" >
17   <div data-role="header" data-position="fixed">
18     <a href="1_jquerymobile.html" class="ui-btn ui-icon-home ui-btn-icon-left">Home</a>
19     <h1>2. Seite: about...</h1>
20   </div>
21

```

- class="" eine Klasse
- ui-btn-icon-left: auf der linken Seite darstellen
- ui-icon-home: verwendet das Häuschen als Icon-Symbol

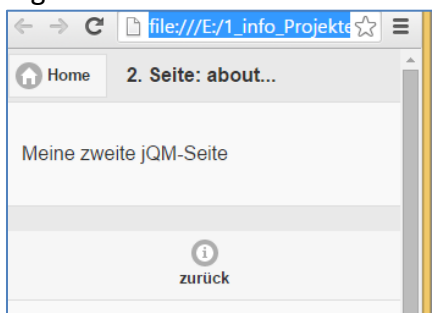
Ergebnis:



Übung:

Erstelle noch einen einzelnen Button in Form einer „Navbar“ unterhalb. Dieser soll ebenfalls auf die Home-Site verlinken. Symbol = „Info“, Text soll „zurück“ sein.

Ergebnis:



Code:

```

26   <div data-role="footer">
27     <div data-role="navbar">
28       <ul>
29         <li><a data-icon="info" href="1_jquerymobile.html">zurück</a></li>
30       </ul>
31     </div>
32   </div>

```

Zu betrachten unter: http://5ckprojekt.bplaced.net/jqm/1_jquerymobile.html

Übung: Login-Page

- Titel: Login Page
- Innerhalb der Page `<div id="loginPage">` werden zwei weitere divs als `<data-role="header">` und `<data-role="content">` ausgezeichnet.
- Im Content div werden zwei Paare von Label- und Input-Element als jeweils zusammengehörig ausgezeichnet.
Dies funktioniert mit „fieldcontain“:

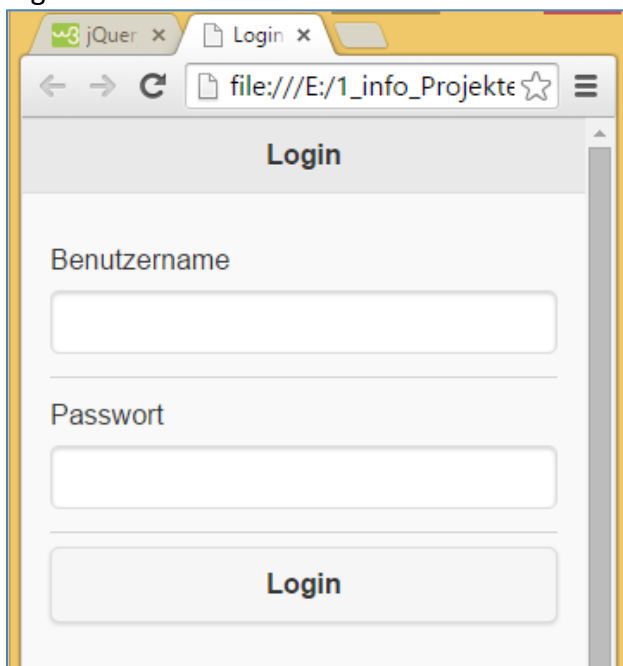
```
<div data-role="fieldcontain">  
<label for="username">Benutzername</label>  
<input type="text" id="username">  
</div>
```

- Die typischen Elemente für das Formular sind: `<label>` und `<input type>`
- **data-role="button"**
Anschließend erhält ein Link die Rolle „Button“. Dies veranlasst jQuery Mobile, diesen Link wie einen Button darzustellen. Die verlinkte Site „welcomePage“ wird der Einfachheit halber im Listing nicht gezeigt.

```
<a href="welcomePage.html" data-role="button">Login</a>
```

- Speichere die Seite unter „login.html“

Ergebnis:



Code:

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width,
6     initial-scale=1">
7 <title>Login Page</title>
8 <link rel="stylesheet" href="lib/jquery.mobile-1.4.5.min.css">
9 <script src="lib/jquery-2.1.4.min.js"></script>
10 <script src="lib/jquery.mobile-1.4.5.min.js"></script>
11 </head>
12
13 <body>
14 <div id="loginPage" data-role="page">
15     <div data-role="header">
16         <h1>Login</h1>
17     </div>
18     <div data-role="content">
19         <div data-role="fieldcontain">
20             <label for="username">Benutzername</label>
21             <input type="text" id="username">
22         </div>
23         <div data-role="fieldcontain">
24             <label for="password">Passwort</label>
25             <input type="password" id="password">
26         </div>
27
28         <a href="welcomePage.html" data-role="button">Login</a>
29     </div>
30 </div>
31 </body>
32 </html>
```

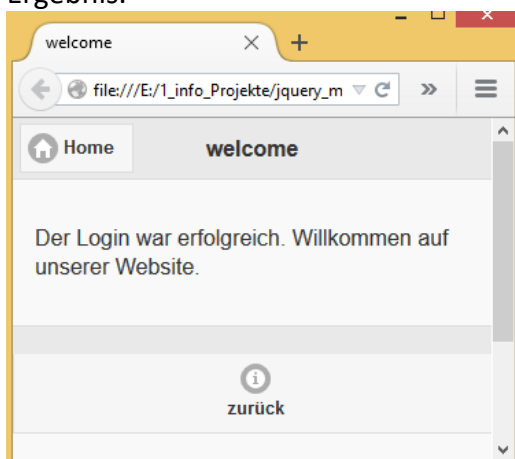
Aufgabe:

Erstelle die zweite Seite „welcomePage.html“.

Diese soll den Titel „welcome“ haben und den Text „Der Login war erfolgreich. Willkommen auf unserer Website.“

Es sollen beide Buttons vorhanden sein, ganz oben links und unten.

Ergebnis:



Ergebnisse live am Handy ansehen:

Auf www.twr.bplaced.net

Beispiel: www.twr.bplaced.net/1_jquerymobile.html

Unter folgenden Dateien:

language	20110727 00:00	2048
lib	20150817 10:44	2048
libraries	20110727 00:00	2048
live access	20121106 00:00	2048
logs	20111113 00:00	2048
media	20140711 00:00	2048
modules	20110727 00:00	2048
MySQLadmin	20110727 00:00	2048
PgSQLadmin	20110727 00:00	2048
plugins	20110727 00:00	2048
templates	20120322 00:00	2048
tmp	20140711 00:00	2048
.ftpquota	20150817 10:44	15
1_jquerymobile.html	20150817 10:44	1381
about.html	20150817 10:44	323
ball_form.html	20150619 10:29	2191
configuration.php	20130206 00:00	1907
htaccess.txt	20110727 00:00	3189
index.php	20110727 00:00	1389
LICENSE.txt	20110727 00:00	17816
login.html	20150817 10:44	866
logo.jpg	20150619 10:29	53910
README.txt	20110727 00:00	4493
robots.txt	20110727 00:00	849
staender.psd	20120909 00:00	57175809
web.config.txt	20110727 00:00	1811
welcomePage.html	20150817 10:44	934

JQM

Eingabehilfen, Formulare...:

Maurice: moderne Webseiten S. 499ff

Web-Apps mit jQuery Mobile – S. 201ff

Quellen:

Florence Maurice, in: Mobile Webseiten, Strategien, Techniken, DOS und DON'TS für Webentwickler, Hanser Verlag München, 2012

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

Bosch, Scheidt, Winterberg, in: Mobile Web-Apps mit JavaScript, Leitfaden für die professionelle Entwicklung, Verlag: entwickler.press, Frankfurt, 2012