

Lernmaterial

Clientseitig dynamische Webseiten mit einem Javascript Framework (jQuery) entwickeln

Fachbereich Wirtschaftsinformatik

Stand: 2014

Inhaltsverzeichnis

1	Stichworte.....	3
2	Allgemeines.....	3
3	Einführungsbeispiele DOM-Manipulation mit jQuery.....	5
3.1	Bereitstellung des jQuery-Framework.....	6
3.2	Übung 1: DOM-Manipulation 1.....	8
3.3	Übung 2: DOM-Manipulation 2 (Methoden effizient auslagern).....	15
3.4	Übung 3: DOM-Manipulation 3 (Methoden überladen und verketten).....	20
4	Clientseitige Entwicklung einer Web-App mit JQuery.....	23
4.1	Beispiel 1: Navigation.....	24
4.2	Beispiel 2: Dynamische Auswertung Taschenrechner.....	34
4.3	Beispiel 3: Dynamische Auswertung Währungsrechner.....	44
5	Erweiterung zur Mobile-Web-App mit JQuery-Mobile.....	53
5.1	Am Beispiel des Taschenrechners.....	54
5.2	Am Beispiel des Währungsrechners.....	61
6	Ausblick.....	62
6.1	Anwendungsbeispiel: Android App mit PhoneGap.....	63
6.1.1	Unsere WebApp wird zu einer nativen Android App.....	63
6.1.2	Mit PhoneGap-Build-Webservice eine .apk-Datei erzeugen.....	66
6.1.3	Die Anwendung auf dem virtuellen Android-Gerät (AVD) installieren.....	68
7	Einrichtung und Konfiguration in Eclipse.....	74
7.1	Allgemeines zur Konfiguration der Entwicklungsumgebung Eclipse.....	75
7.2	Syntax-Hilfe mit dem Eclipse-PlugIn-Aptana.....	79
8	Anhang.....	83
8.1	Vorgangsbeschreibung am Beispiel des Taschenrechners.....	84
8.2	Vorgangsbeschreibung am Beispiel des Währungsrechners.....	88
9	Quellen.....	91

1 Stichworte

CSS3, Dop-Down-Menü, DOM-Manipulation, Einführungsbeispiele, Framework, Formulare auswerten, HTML5, Javascript, JQuery, Mobile Web App, Native App, PhoneGap Build, Übungen, Web App, Webprogrammierung.

2 Allgemeines

JQuery ist ein Javascript Framework. Ein Rahmenwerk das im weiten Sinne die Vermittlerfunktion zwischen Browser und Javascript übernimmt und jede Menge Dienstfunktionen zur Verfügung stellt.

Damit sind die Bibliotheken organisiert, online verfügbar und zentral verwaltet.

Das Framework stellt, außer der jQuery Application Programming Interfaces (API), mit jQuery UI u.a. die Komponenten für die Benutzeroberflächen mobiler Endgeräte, deren Zugriffsmethoden und CSS-Klassen, die Erweiterungen für die Gestaltung von Benutzeroberflächen mobiler Endgeräte, online zur Verfügung.

Der umständliche Umgang mit der Document Object Modell (DOM) wird mit Hilfe des Frameworks vereinheitlicht und erleichtert.

Dieses Rahmenwerk besteht aus einem in JavaScript geschriebenen Programm. Es stellt Methoden zur Verfügung die es ermöglichen fortschrittliche Benutzerschnittstellen zu schaffen und aufregende, animierte, animierte, interaktive Websites zu entwerfen.

Einsatzbereiche:

1. Elemente finden
2. Elemente manipulieren
3. Inhalte einfügen
4. Bestandteile einer Seite animieren
5. Event-Handling

Das Framework sorgt u.a. dafür, dass wir uns um die Verwirklichung unserer Ideen kümmern können, anstatt sich mit verschiedenen Bugs und Tücken bestimmter JavaScript-Engines herumzuärgern. [VOL00]



Das Skript schildert den Umgang mit dem Framework anhand von konkreten Beispielen die unter Umständen auch in den Unterricht im Fachbereich Wirtschaftsinformatik respektive im Fachbereich Informatik einbetten lassen.

Aktuelle Versionen des Skriptes selbst und die im Skript behandelten Quellcodes können Sie online herunterladen und testen:

http://www.emotionalspirit.de/speaker2_3_3/index.php/2014/01/27/entwickeln-mit-javascript-framework-jquery-jquery-mobile/

Skript & Sources:

Download: [Eclipse-Projekt \(00_00jQuery.zip\)](#)

Download: [Eclipse-Android-Projekt \(MyApp.zip\)](#)

Online-Lösungen:

[Web-APP](#)

[Mobile-App](#)

Download (PDF): [Arbeitsmaterial JQuery clienseitig final](#)

Download (ODT): [Arbeitsmaterial JQuery clienseitig final](#)



Für alle Inhalte gilt natürlich das Urheberrecht. Ich selber achte auch darauf. Um Details zur Creative-Commons-Lizenz für die von mir selbst verfassten Texte und Quellcodes zu erhalten, klicken Sie links auf das CC-BY-NC-SA-Logo. Für Ergänzungs- und/oder Verbesserungsvorschläge schreiben Sie mir bitte eine E-Mail: cjanischek@gmx.de

Alle Quellangaben wurden nach bestem Gewissen genannt und aufgeführt. Permanent begleitende Literatur waren:

[FRA00]

Franke, Florian, Ippen, Johannes, „Apps mit HTML5 und CSS3“, 978-3-8362-1848-1, 2013, Galileo Computing

[VOL00]

Vollendorf, Maximilian, Bongers, Frank, „jQuery das Praxisbuch“, 978-3-8362-1810-8, 2013, Galileo Computing

[WEN00]

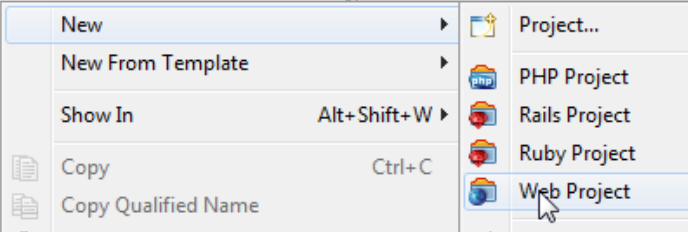
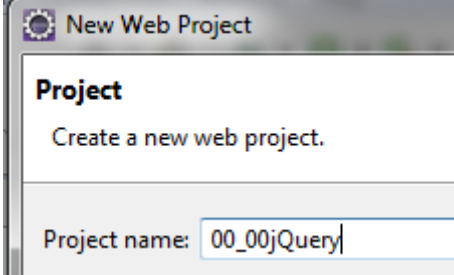
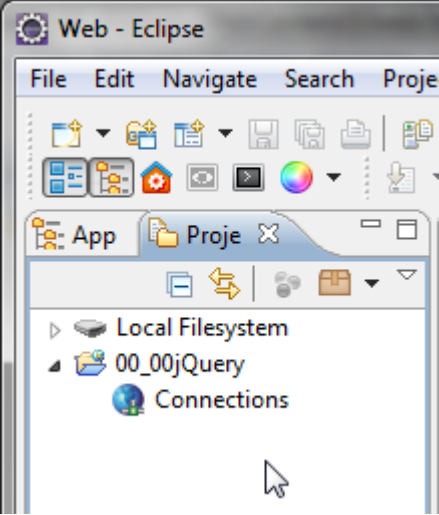
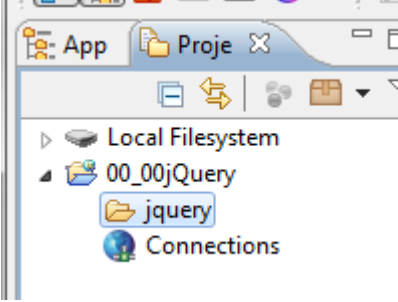

Wenz, Christian, „Javascript“, 978-3-8362-1979-2, 2014, Galileo Computing

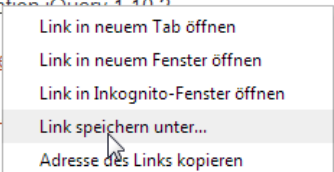
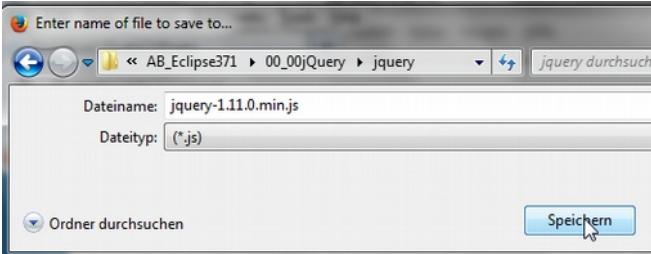
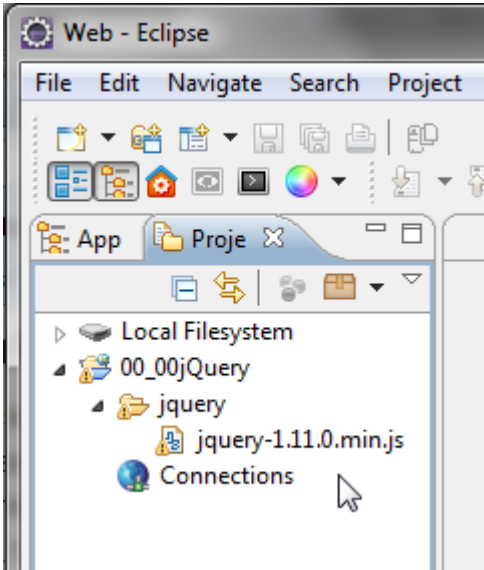
Sowie diverse Internetquellen, die in den Quellenangaben im letzten Kapitel aufgeführt werden.

3 Einführungsbeispiele DOM-Manipulation mit jQuery

Einführungsbeispiele DOM-Manipulation mit jQuery

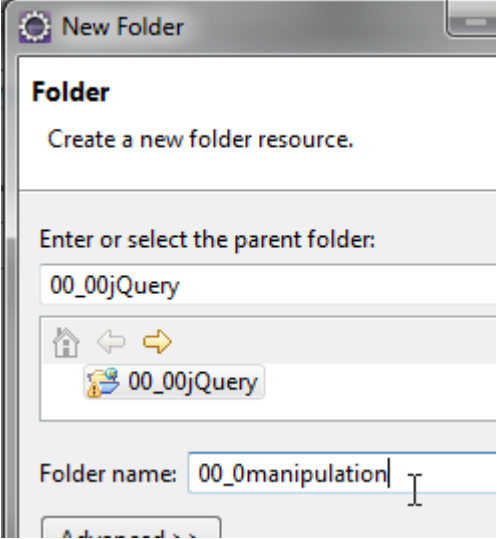
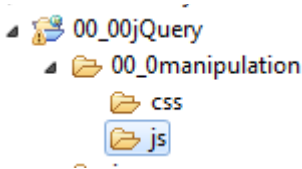
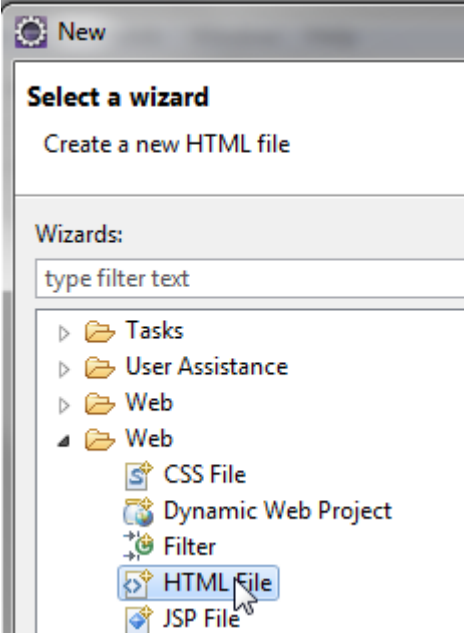
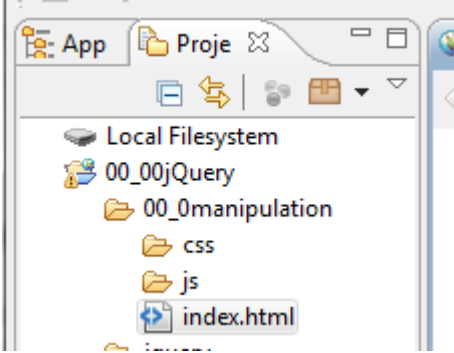
3.1 Bereitstellung des jQuery-Framework

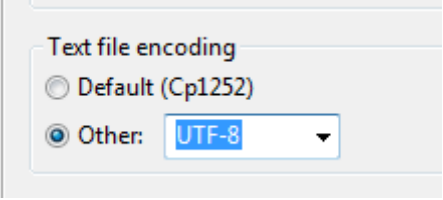
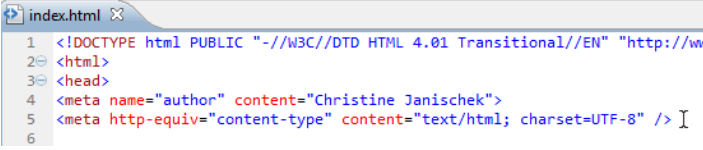
	<p>Starten Sie die Entwicklungsumgebung Eclipse in der WEB-Perspektive und legen Sie ein neues Web Project mit dem Namen >>00_00jQuery<< an:</p> 
	<p>Die jQuery Libs können lokal und online eingebunden werden.</p> <p>Damit Sie jQuery lokal einbinden können, erzeugen Sie im Projektverzeichnis >>00_00jQuery<< ein neues Verzeichnis mit dem Namen >>jquery<<.</p> 
	<p>Laden Sie die minimierte (komprimierte) Produktversion von jQuery herunter.</p> <p>Nutzen Sie dazu die jQueryWebsite: http://jquery.com/</p> <p>Klicken Sie auf die Schaltfläche Download.</p> <p>Für die lokale Einbindung der Mobilen API: http://jquerymobile.com/download/</p>

<p>Download the compressed, production-ready jQuery 1.10.2</p> <p>Download the uncompressed, development-ready jQuery 1.10.2</p> <p>Download the map file for jQuery 1.10.2</p> <p>jQuery 1.10.2 release notes</p> 	<p>Klicken Sie die minimierte Version mit der linken Maustaste an und wählen Sie im Kontext-Menü die Option >>Link speichern unter<<.</p>
	<p>Wählen Sie als Speicher-Ziel das gerade erzeugte Verzeichnis >>jquery<< aus und speichern Sie die Datei</p>
	<p>Wählen Sie in ihrem Projekt im Kontext-Menü die Option >>Refresh<< um die Verzeichnis-Inhalte zu aktualisieren.</p>

Die Einführungsbeispiele wurden an die Anforderungen der Schulung angepasst und erweitert [VOL00].

3.2 Übung 1: DOM-Manipulation 1

	<p>Erzeugen Sie ein neues Verzeichnis:</p> <p>00_0manipulation</p>
	<p>Die Projekte halten sich an das gängige Modell-View-Controller (MVC) Konzept.</p> <p>Legen Sie deshalb zusätzliche Unterverzeichnisse für das Stylesheet (css-verzeichnis) und das Javascript (js-verzeichnis) an.</p>
	<p>Klicken Sie das neu erzeugte Verzeichnis und wählen Sie im Kontextmenü die Option >> New >> Other >> Web >> HTML File</p>  <p>Nennen Sie die Datei >> index.html und klicken Sie auf >> Finish um den Vorgang abzuschließen.</p>

	<p>Hinweis: Ändern Sie in Eclipse >> Window >> General >> Workspace >> Text file encoding auf:</p>  <p>Damit werden die Umlaute in Strings beim öffnen der Datei erkannt und korrekt dargestellt.</p>
 <p>Als HTML5-Dokument:</p> <pre> 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <meta name="author" content="Christine Janischek"> 5 <meta http-equiv="content-type" content="text/html; charset=UTF-8" /> 6 </pre>	<p>Fügen Sie im ersten Schritt die Meta-Informationen hinzu.</p> <p>Die Angaben zum verwendeten Zeichensatz ist u.a. für die Darstellung von Sonderzeichen und Umlauten eine relevante Angabe. Wir wählen dazu das UCS Transformation Format, wobei UCS wiederum Universal Character Set abkürzt, kurz UTF-8.</p> <pre> <meta http-equiv="content-type" content="text/html; charset=UTF-8" /> </pre>
<pre> 7 <title>Einführungsbeispiel: 00_0 Manipulation</title> 8 </pre>	<p>Fügen Sie den Titel ein.</p>
<pre> 9 <!-- Zunächst den import für das Stylesheet: --> 10 <link href="css/styles.css" rel="stylesheet" type="text/css"> 11 </pre>	<p>Die Datei bezieht die Formatangaben aus einem Stylesheet. In der Regel werden diese Angaben ausgelagert in eine separate Datei. Fügen Sie dazu den Link zur Stylesheet Datei >> css/styles.css</p> <pre> <link href="css/styles.css" rel="stylesheet" type="text/css"> </pre>

```

12 <!-- Nun das Framework einbinden: -->
13 <!------- ONLINE Version----->
14
15 <!-- Online Lib JQuery -->
16 <script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
17

```

In den Eingangsbeispielen nutzen wir das online verfügbare Framework von jQuery.

Fügen Sie dazu den Link zur minimierten Version der aktuellen jQuery-Lib ein.

```
<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
```

Alternativ die lokale Lib:

```
<script src="../jquery/jquery-1.11.0.min.js"></script>
```

```

20 <body>
21
22 <!-- ...nun die Box mit den Absätzen: -->
23 <div id="box">
24 <ul>
25 <li>Erster Menüpunkt</li>
26 <li>Zweiter Menüpunkt</li>
27 <li>Dritter Menüpunkt</li>
28 </ul>
29 </div>
30 </body>

```

Ziel ist eine horizontale Navigationsleiste zu schaffen deren Farbgebung sich beeinflussen lässt.

Fügen Sie dazu im Körper des HTML-Dokuments eine Div-Box mit der Liste und den drei Listenpunkten ein.

```

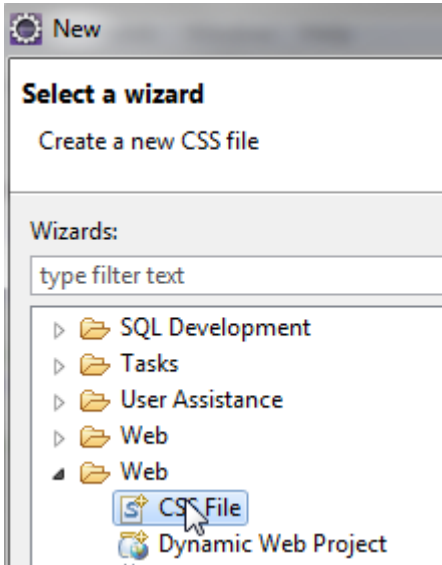
<div id="box">
  <ul>
    <li>Erster Menüpunkt</li>
    <li>Zweiter Menüpunkt</li>
    <li>Dritter Menüpunkt</li>
  </ul>
</div>

```

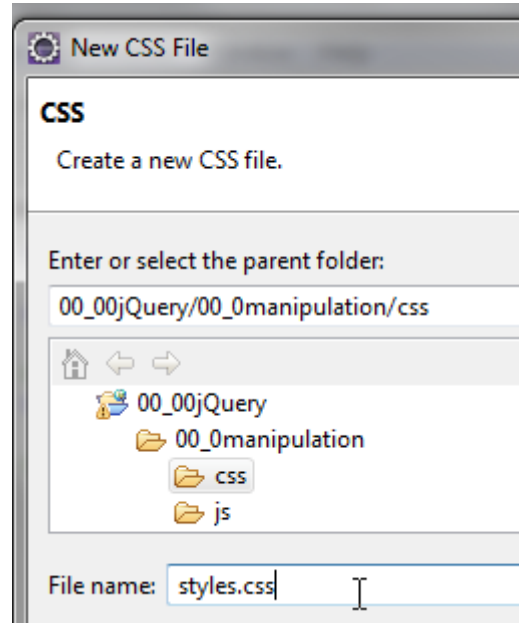
Im Stylesheet styles.css können wir anhand der id Formatangaben definieren. Zum Format gehören u.a. Angaben zur Ausrichtung, Schrift und Farbe.



Im folgenden werden wir nun alle Formatangaben in einem Cascading Style sheet definieren. Dabei werden die in HTML-Auszeichnungen vor- oder selbstdefinierter Art (hier: div-"box", ul, li) genutzt. Das **Box-Modell** wird sowie in CSS, als auch später in JQuery die Basis für die Identifikation der Objekte im Dokument sein. Alle Inhalte im Body stecken in irgendeiner Box!



Nun zu den notwendigen Formatangaben. Erzeugen Sie dazu im css-Verzeichnis eine neue Datei >> styles.css



```

index.html | styles.css
1 @CHARSET "UTF-8";
2
3 body{
4   font-family:Verdana, Arial, Helvetica,sans-serif;
5   font-weight: normal;
6   font-size:100,01%;
7   width: 100%;
8 }

```

Betten Sie zu Beginn, wie üblich die Angaben für den Körper ein. Ändern Sie dazu den Zeichensatz. Definieren Sie dann anschließend die verwendete Schrift-Familie, das Schrift-Gewicht, die relative Angabe der Schriftgröße und die relative Breite des Dokuments.

```
@CHARSET "UTF-8";
```

```

body{
  font-family:Verdana, Arial, Helvetica,sans-serif;
  font-weight: normal;
  font-size:100,01%;
  width: 100%;
}

```

```

10 ul{
11     list-style-type:none;
12 }
13
14 li{
15     display:inline;
16     margin: 0.4em;
17     padding: 0.4em;
18 }
19
20 .green{
21     color:#009933;
22     background-color:#E2FFEC;
23 }

```

Erweitern Sie nun die Formatangaben für die Navigation. Die unsortierte Liste erhält dazu die Angabe keinen List-Style-Typ zu verwenden, die Listenpunkte werden im Ergebnis ausgeblendet:

```
list-style-type:none;
```

Die Listenpunkte selbst sollen nebeneinander liegen, dazu definieren wir diese als inline-Objekte:

```
display:inline;
```

Wir legen im gleichen Zug einen Außen- und Innenabstand fest der sicherstellt, dass die Titel der Listenpunkte nicht zu nah aneinander liegen:

```
margin: 0.4em;
padding: 0.4em;
```

0.4em eine relative Angabe bezogen auf die Schriftgröße des Elements.

```

20 .green{
21     color:#009933;
22     background-color:#E2FFEC;
23 }

```

Eine CSS-Klasse >> .green definiert die Schrift- und Hintergrundfarbe neu.

Die Klasse dient im folgenden dazu, Inhalte mit einem bestimmten Wortlaut mit Hilfe von jQuery zu Manipulieren (zu ändern).

```

.green{
    color:#009933;
    background-color:#E2FFEC;
}

```

Mit dem Aufruf der Klasse soll sich die schriftfarbe

```
color:#009933;
```

Und die Hintergrundfarbe ändern.

```
background-color:#E2FFEC;
```

```

18 <!-- ...und jetzt zu unserem eigenen Code: -->
19 <script type="text/javascript">
20 $(document).ready(function(){
21     $("#box li:contains('Zweiter Menüpunkt')").addClass('green');
22 });
23 </script>

```

Ergänzen Sie nun im Kopfteil der HTML-Datei >> index.html, angrenzend an den Link zur jQuery-Lib, den folgenden Javascript Quellcode.

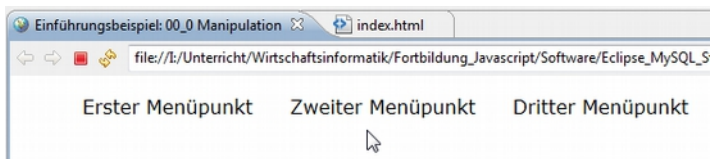
```
<script type="text/javascript">
  $(document).ready(function(){
    $("#box li:contains('Zweiter Menü-
punkt')").addClass('green');
  });
</script>
```

Sobald das Dokument fertig geladen wurde

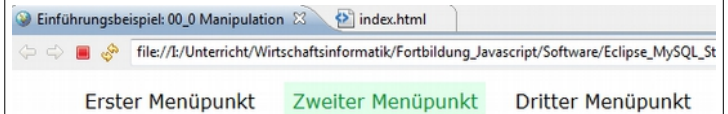
```
$(document).ready(...);
```

Die Funktion ermittelt die im HTML-Dokument enthaltenen Div-Boxen mit der id „box“ und untersucht die Listenpunkte. Die identifizierten Objekten mit dem Wortlaut „Zweiter Menüpunkt“ wird daraufhin die CSS-Klasse „green“ zugewiesen.

```
function(){
  $("#box li:contains('Zweiter Menü-
punkt')").addClass('green');
}
```



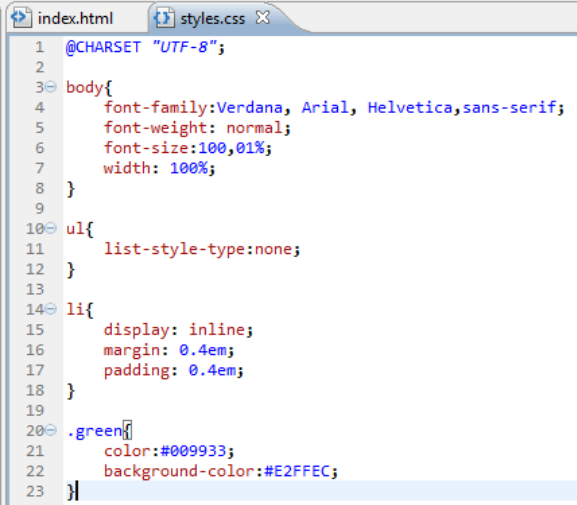
vorher: View



nachher: View

```
index.html | styles.css
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="author" content="Christine Janischek">
5 <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
6
7 <title>Einführungsbeispiel: 00_0 Manipulation</title>
8
9 <!-- Zunächst den import für das Stylesheet: -->
10 <link href="css/styles.css" rel="stylesheet" type="text/css">
11
12 <!-- Nun das Framework einbinden: -->
13 <!------- ONLINE Version----->
14
15 <!-- Online Lib JQuery -->
16 <script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
17
18 <!-- Referenz auf die lokale Lib JQuery -->
19 <script src="../jquery/jquery-1.11.0.min.js"></script>
20
21 <!-- ...und jetzt zu unserem eigenen Code: -->
22 <script type="text/javascript">
23   $(document).ready(function(){
24     $("#box li:contains('Zweiter Menüpunkt')").addClass('green');
25   });
26 </script>
27 </head>
28 <body>
29
30 <!-- ...nun die Box mit den Listenpunkten: -->
31 <div id="box">
32   <ul>
33     <li>Erster Menüpunkt</li>
34     <li>Zweiter Menüpunkt</li>
35     <li>Dritter Menüpunkt</li>
36   </ul>
37 </div>
38 </body>
39 </html>
40
```

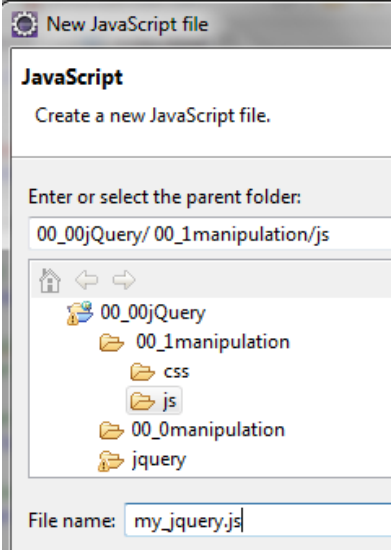
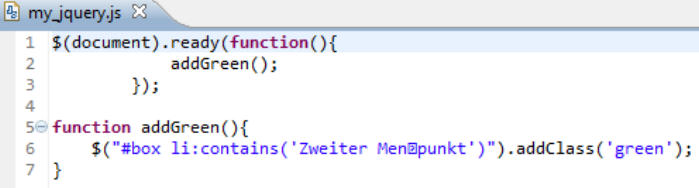
Quellcode: index.html

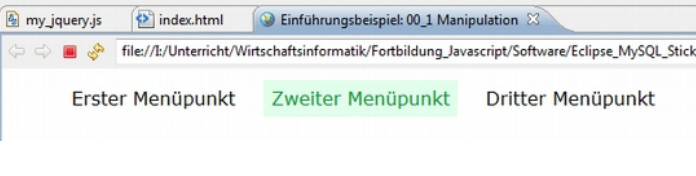
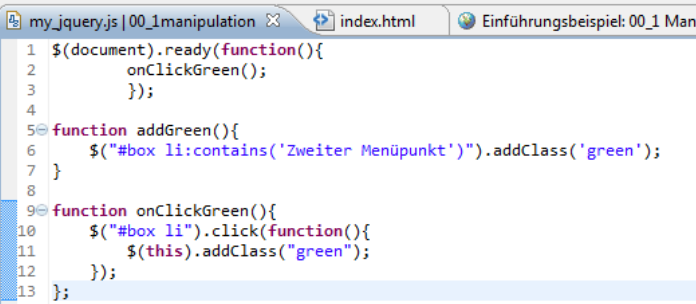
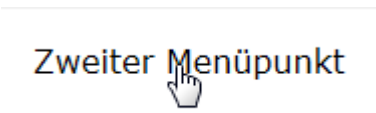


```
1 @CHARSET "UTF-8";
2
3 body{
4     font-family:Verdana, Arial, Helvetica,sans-serif;
5     font-weight: normal;
6     font-size:100,01%;
7     width: 100%;
8 }
9
10 ul{
11     list-style-type:none;
12 }
13
14 li{
15     display: inline;
16     margin: 0.4em;
17     padding: 0.4em;
18 }
19
20 .green{
21     color:#009933;
22     background-color:#E2FFEC;
23 }
```

Quellcode: styles.css

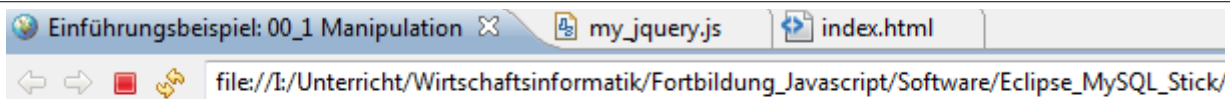
3.3 Übung 2: DOM-Manipulation 2 (Methoden effizient auslagern)

	<p>Wir verwenden das zuvor erzeugte Ergebnis, um es zu erweitern. Kopieren Sie dazu das Verzeichnis >> 00_0manipulation und benennen Sie es >> 00_1manipulation.</p>
	<p>Erzeugen Sie im js-Verzeichnis eine JavaScript Datei. Wählen Sie dazu im Kontextmenü die Option >> New >> Other >> Javascript >> Javascript Source File</p> <p>Nennen Sie die Datei >> my_jquery.js</p> <p>Wir lagern künftig alle eigenen Javascript-Funktionen aus in eine extra Datei. Auf diese Weise vermeiden wir Dopplungen von Quellcode und können die Funktionen problemlos wiederverwenden.</p> <p>Ein weiterer Vorteil ist, dass wir bei Änderung einer Funktionalität einen zentralen Ort haben wo die Veränderung stattfinden muss.</p>
	<p>Lagern Sie den Javascript Quellcode aus der HTML Datei in die Javascript Datei aus.</p> <p>Sobald das Dokument fertig geladen wurde</p> <pre>\$(document).ready(...);</pre> <p>Erfolgt der Methodenaufruf</p> <pre>addGreen();</pre> <p>Die Funktion <code>addGreen()</code> enthält nichts anderes, wie zuvor.</p>

<pre> 20 21 <!-- Eigene local JQuery Lib ----> 22 <script type="text/javascript" src="js/my_jquery.js"></script> 23 </pre>	<p>Ersetzen Sie dann den Javascript Quellcode im Kopf der HTML Datei durch die Referenz auf die ausgelagerte Datei.</p> <pre> <script type="text/javascript" src="js/my_jquery.js"></script> </pre>
	<p>Testen Sie das Ergebnis. An der View sollte sich in diesem Stadium noch nichts geändert haben.</p>
 <pre> 1 \$(document).ready(function(){ 2 onClickGreen(); 3 }); 4 5 function addGreen(){ 6 \$("#box li:contains('Zweiter Menüpunkt)').addClass('green'); 7 } 8 9 function onClickGreen(){ 10 \$("#box li").click(function(){ 11 \$(this).addClass("green"); 12 }); 13 }; </pre>	<p>Nun soll die Klasse „green“ erst dann zugewiesen werden, wenn eine der Listenpunkte angeklickt wird.</p> <p>Ergänzen Sie die Javascript-Datei <code>my_jquery.js</code> um die Funktion <code>onClickGreen()</code>.</p> <p>Die Methode weist nun auf Klick „click()“ dem aktuellen Listenpunkt mit dem Selektor „#box li“ die CSS-Klasse „green“ zu.</p> <p>Ersetzen Sie danach den Methodenaufruf <code>addGreen();</code> durch <code>onClickGreen();</code></p>
<pre> 14 li{ 15 display: inline; 16 margin: 0.4em; 17 padding: 0.4em; 18 cursor: pointer; 19 } 20 21 .green{ 22 color:#009933; 23 background-color:#E2FFEC; 24 cursor:default; 25 } </pre>	<p>Erweitern Sie das Stylesheet um einen Curser-Pointer. Damit kann der Nutzer erkennen, dass er auf die Listenpunkte klicken muss, um eine Veränderung hervorzurufen.</p> <p>Ergänzen Sie dazu die CSS-Anweisung für die Listenpunkte:</p> <pre> cursor: pointer; </pre>  <p>Für den Fall, dass der Nutzer den Listenpunkt bereits angeklickt hat und die Klasse „green“ bereit zugewiesen wurde soll der Curser wieder auf „normal“ zurückgesetzt werden.</p>


```
cursor:default;
```

Zweiter Menüpunkt



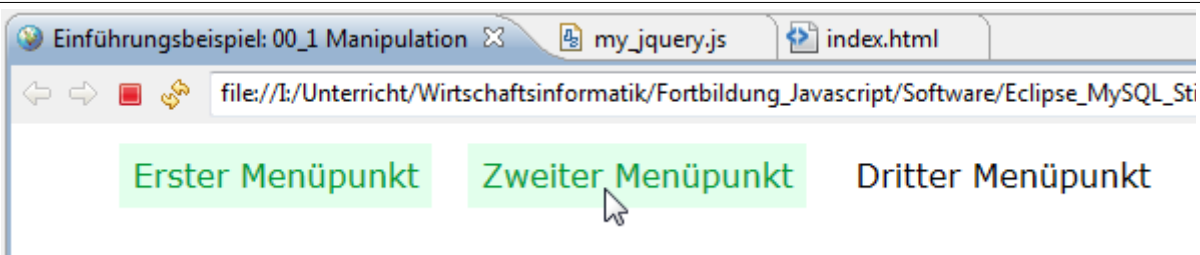
Erster Menüpunkt Zweiter Menüpunkt Dritter Menüpunkt

Ohne Klick...



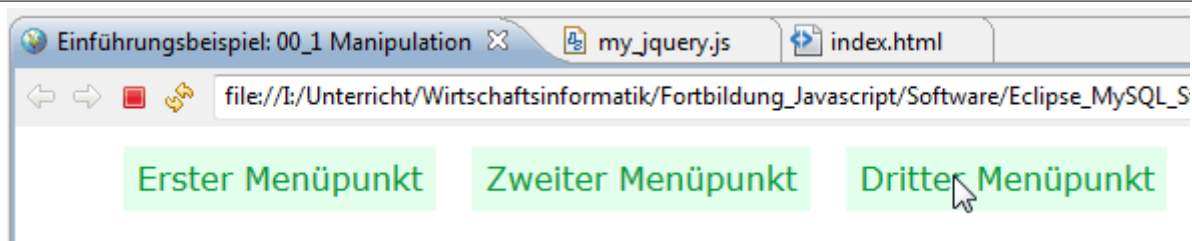
Erster Menüpunkt Zweiter Menüpunkt Dritter Menüpunkt

Auf Klick: View 1



Erster Menüpunkt Zweiter Menüpunkt Dritter Menüpunkt

Auf Klick: View 2



Erster Menüpunkt Zweiter Menüpunkt Dritter Menüpunkt

Auf Klick: View 3

```

index.html x styles.css
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="author" content="Christine Janischek">
5 <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
6
7 <title>Einführungsbeispiel: 00_1 Manipulation</title>
8
9 <!-- Zunächst den import für das Stylesheet: -->
10 <link href="css/styles.css" rel="stylesheet" type="text/css">
11
12 <!-- Nun das Framework einbinden: -->
13 <!------- ONLINE Version----->
14
15 <!-- Online Lib JQuery -->
16 <script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
17
18 <!-- Referenz auf die lokale Lib JQuery -->
19 <script src="../jquery/jquery-1.11.0.min.js"></script>
20
21 <!-- Eigene local JQuery Lib --->
22 <script type="text/javascript" src="js/my_jquery.js"></script>
23
24 </head>
25 <body>
26
27 <!-- ...nun die Box mit den Listenpunkten: -->
28 <div id="box">
29 <ul>
30 <li>Erster Menüpunkt</li>
31 <li>Zweiter Menüpunkt</li>
32 <li>Dritter Menüpunkt</li>
33 </ul>
34 </div>
35 </body>
36 </html>

```

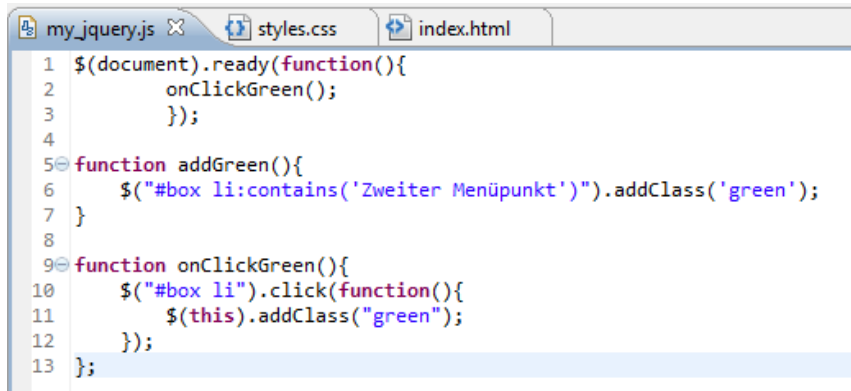
Quellcode: index.html

```

styles.css x index.html Einführungsbispiel: 00_1 Manipulation
1 @CHARSET "UTF-8";
2
3 body{
4 font-family:Verdana, Arial, Helvetica,sans-serif;
5 font-weight: normal;
6 font-size:100,01%;
7 width: 100%;
8 }
9
10 ul{
11 list-style-type:none;
12 }
13
14 li{
15 display: inline;
16 margin: 0.4em;
17 padding: 0.4em;
18 cursor: pointer;
19 }
20
21 .green{
22 color:#009933;
23 background-color:#E2FFEC;
24 cursor:default;
25 }

```

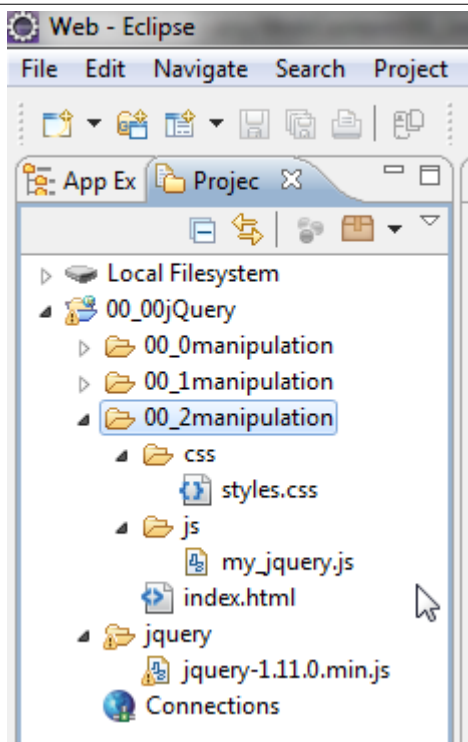
Quellcode: styles.css



```
1 $(document).ready(function(){
2     onClickGreen();
3     });
4
5 function addGreen(){
6     $("#box li:contains('Zweiter Menüpunkt)').addClass('green');
7 }
8
9 function onClickGreen(){
10    $("#box li").click(function(){
11        $(this).addClass("green");
12    });
13 };
```

Quellcode: my_jquery.css

3.4 Übung 3: DOM-Manipulation 3 (Methoden überladen und verketteten)



Wir verwenden das zuvor erzeugte Ergebnis, um es zu erweitern. Kopieren Sie dazu das Verzeichnis >> 00_1manipulation und benennen Sie es >> 00_2manipulation.

```

1 $(document).ready(
2     function(){
3         var mSelector = $("#box li");
4         onClickGreen(mSelector);
5     });
6
7 function onClickGreen(){
8     $("#box li").click(function(){
9         $(this).addClass("green");
10    });
11 };
12
13 function onClickGreen(pSelector){
14 |
15 };

```

Im obigen Beispiel haben wir den CSS-Selektor auf >>#box p<< statisch definiert.

Damit eine wiederverwendbare, flexiblere Lösung entsteht, werden wir den CSS-Selektor gleichbedeutend in eine Objektvariable packen und ihn als Parameter übergeben.

```

var mSelector = $("#box li");
onClickGreen(mSelector);

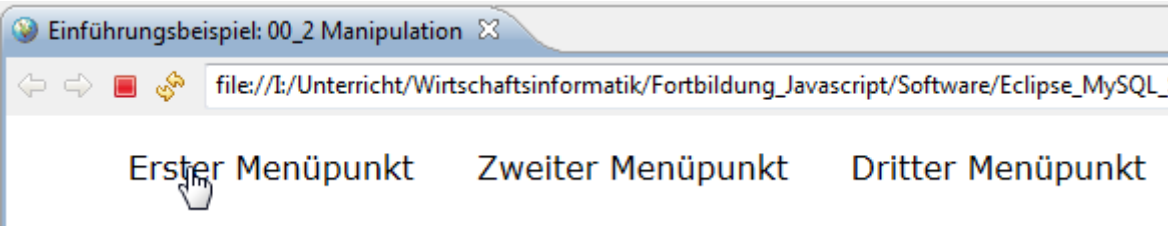
```

Wir überladen dazu die bereits existierende `onClickGreen()` Methode und deklarieren eine parameterbehaftete Version.

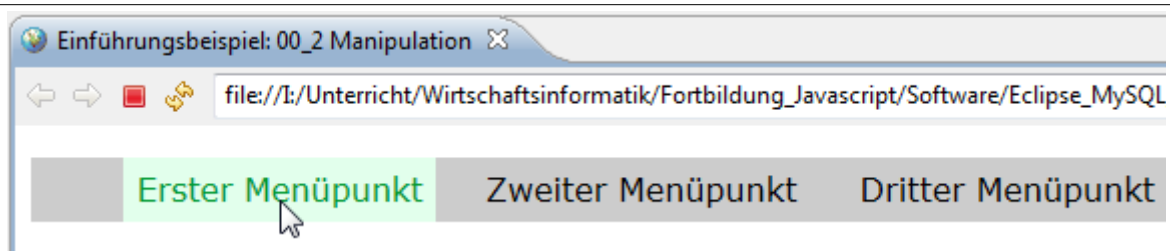
```

function onClickGreen(pSelector){
    ...
};

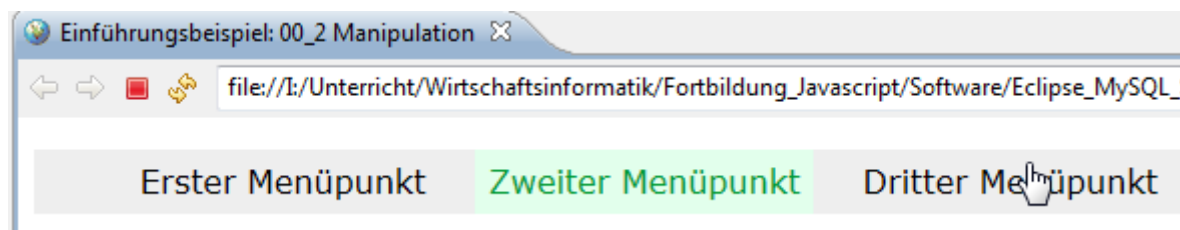
```

<pre> 13 function onClickGreen(pSelector){ 14 var obj = pSelector; 15 obj.click(function(){ 16 var pos = obj.index(this); 17 obj 18 .parent() 19 .removeClass(); 20 \$(this) 21 .addClass("green") 22 .parent() 23 .addClass("boxColor-" + pos); 24 }); 25 }; </pre>	<p>Die Methode soll gleichzeitig an Funktionalität gewinnen.</p> <p>Dazu verketteten wir mehrere Methodenauf-rufe nach folgenden Muster:</p> <pre> \$(selektorausdruck) .methode1() .methode2() .methode3(); </pre> <p>Die Erläuterung der einzelnen Aufrufe erfolgt im folgenden separat.</p>
<pre>var pos = obj.index(this);</pre>	<p>Ermittelt den aktuellen Index (die Position) des angeklickten Listenpunktes und merkt diese Information in einem lokalen Attribut <code>pos</code> vor.</p>
<pre>obj .removeClass() .parent() .removeClass();</pre>	<p>Da wir nicht wissen welche Aktionen der Nutzer zuvor ausgeführt hat stellen wir den Ursprungszustand her. Wir entfernen dazu die Klasse „green“, traversieren zum Elternknoten und entfernen auch dort alle CSS-Klassen (<code>boxColor-0</code> bis <code>2</code>).</p>
<pre>\$(this) .addClass("green") .parent() .addClass("boxColor-" + pos);</pre>	<p>Der angeklickte Listenpunkt erhält dann die Klasse „green“ zugewiesen. Dann traversieren wir zum Elternknoten und fügen die positionsabhängige Klasse hinzu.</p>
<pre> 23 .boxColor-0{ 24 line-height: 2; 25 background-color:#CCCCCC; 26 } 27 28 .boxColor-1{ 29 line-height: 2; 30 background-color:#EEEEEE; 31 } 32 33 .boxColor-2{ 34 line-height: 2; 35 background-color:#999999; 36 } </pre>	<p>Erweitern Sie das Stylesheet <code>styles.css</code>, um die benötigten Klassen <code>boxColor-0</code> bis <code>2</code>.</p>
 <p>The screenshot shows a browser window titled 'Einführungsbeispiel: 00_2 Manipulation'. The address bar shows the file path: <code>file:///I:/Unterricht/Wirtschaftsinformatik/Fortbildung_Javascript/Software/Eclipse_MySQL_...</code>. Below the address bar, there are three buttons: 'Erster Menüpunkt', 'Zweiter Menüpunkt', and 'Dritter Menüpunkt'. A mouse cursor is pointing at the first button.</p>	

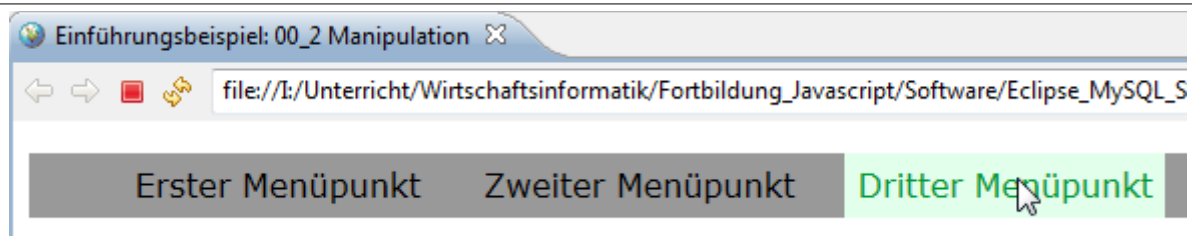
Ohne Klick...



Auf Klick: View 1



Auf Klick: View 2

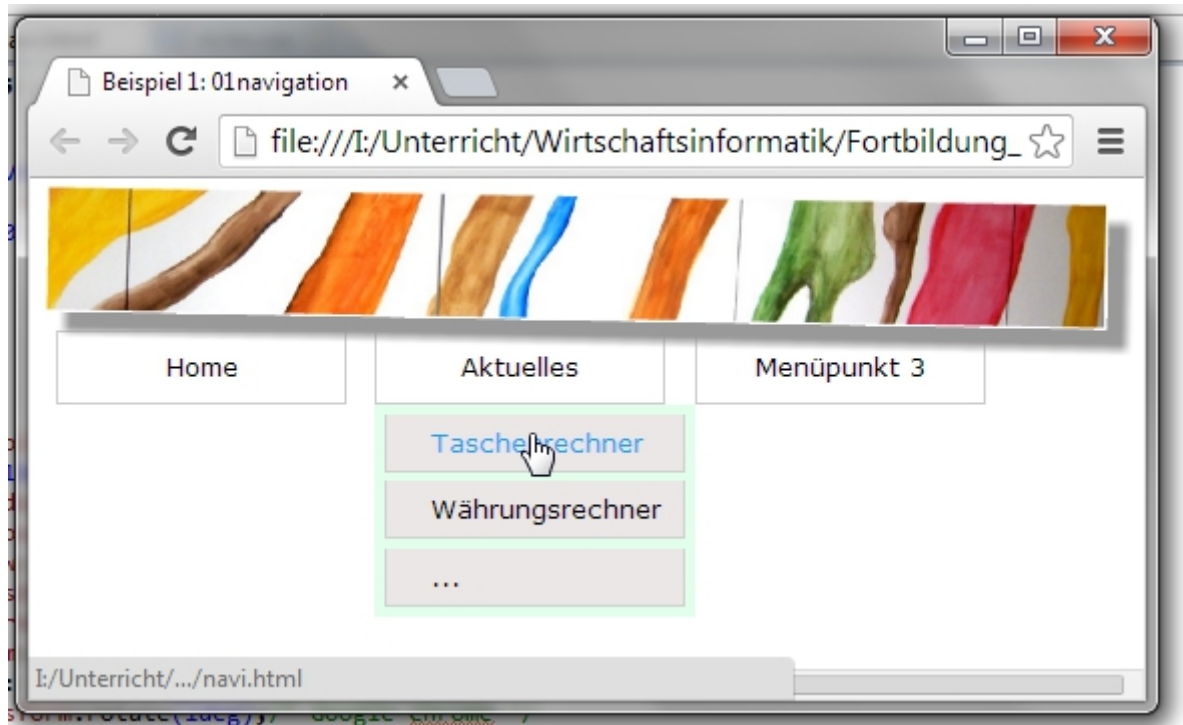


Auf Klick: View 3

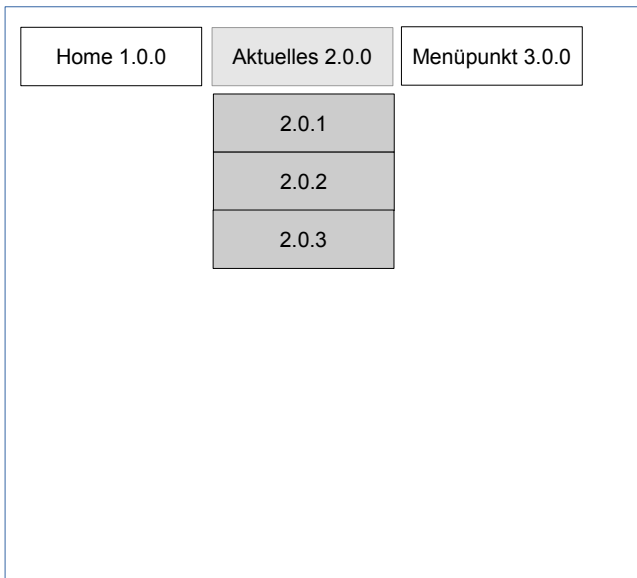
4 Clientseitige Entwicklung einer Web-App mit JQuery

Clientseitige Entwicklung einer Web-App mit JQuery

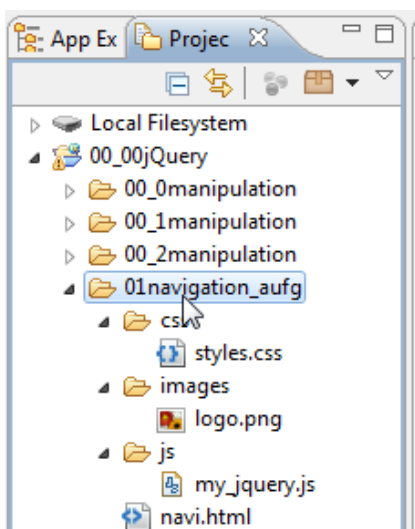
4.1 Beispiel 1: Navigation



View der Navigation: navi.html



Ein erster Entwurf der Benutzeroberfläche mit dem Textverarbeitungsprogramm:
Einfaches Drop-Down-Menü



Wir verwenden das Verzeichnis `01navigation_aufg`, um es zu erweitern.

```
<div class="page">
  
</div>
```

Öffnen Sie die HTML-Datei `navi.html`

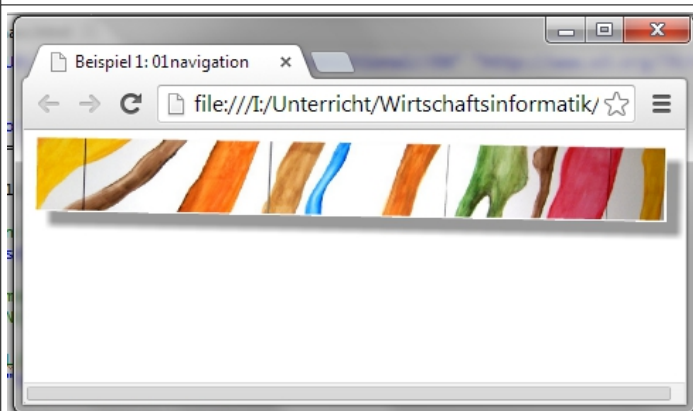
Ergänzen Sie den Quellcode für die Div-Box `page`.

Fügen Sie die Bildreferenz für die Logo-Datei `logo.png` ein.

Das Kunstwerk stammt aus dem Fundus der Gewerbliche, Kaufmännische, Landwirtschaftliche Schule Wangen und wurde von Klaus Merk abfotografiert.

Künstler: Hans-Georg Ziegler

Die in den Quellcodebeispielen eingebundene Grafik `logo.png` ist selbst erstellt.



Um die Grafik in der Box dynamisch anzupassen und den Schatten einzufügen benötigt das Stylesheet `styles.css` die CSS-Klasse `.schatten`

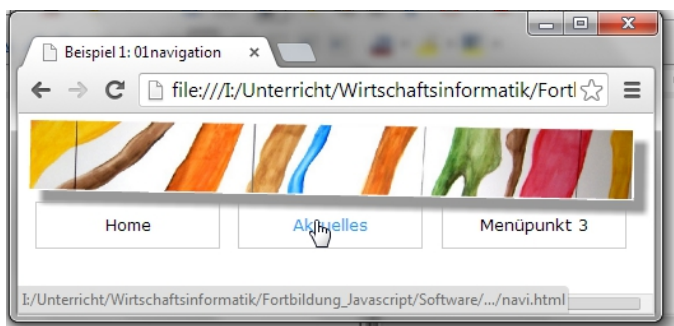
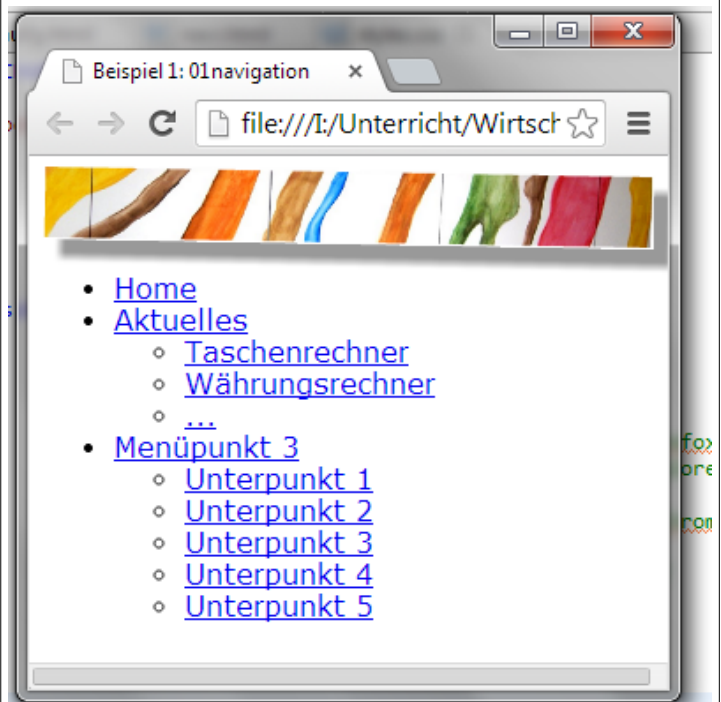
Den Schatten erzeugt die Anweisung `box-shadow` dazu erfolgt die Angabe der horizontalen und vertikalen Platzierung, dem Versatz und der Farbe.

```
box-shadow: none|h-shadow v-shadow blur
spread color |inset|initial|inherit;
```

Optionale Angaben	Erklärung
-------------------	-----------

```
.schatten{
  width:95%;
  height:auto;
  border:1px solid #fff;
```

<pre> /* Allgemein */ box-shadow: 3px 2px 10px #999; /* Mozilla Firefox */ -moz-box-shadow: 3px 2px 10px #999; /* Internet Explorer */ -ms-box-shadow: 3px 2px 10px #999; /* Opera */ -o-box-shadow: 3px 2px 10px #999; /* Google Chrome */ -webkit-box-shadow: 3px 2px 10px #999; /* Allgemein */ transform: rotate(1deg); /* Mozilla Firefox */ -moz-transform: rotate(1deg); /* Internet Explorer */ -ms-transform: rotate(1deg); /* Opera */ -o-transform: rotate(1deg); /* Google Chrome */ -webkit-transform: rotate(1deg); } </pre>	<table border="1" data-bbox="804 190 1519 443"> <tr> <td>blur</td> <td>unscharfe Distanzangabe</td> </tr> <tr> <td>spread</td> <td>die Größe des Schattens</td> </tr> <tr> <td>inset</td> <td>erzeugt einen „inner shadow“</td> </tr> <tr> <td>initial</td> <td>setzt die default Werte</td> </tr> </table> <p>Die Präfixe für den Browser dienen dem Kompatibilitätsabgleich. Die Rotation erzeugt die Anweisung <code>transform: rotate(angle)</code></p> <p>transform: rotate(1deg);</p> <p>Ergänzen Sie den Quellcode für den Schatten in dem Stylesheet <code>styles.css</code> und testen Sie das Ergebnis.</p>	blur	unscharfe Distanzangabe	spread	die Größe des Schattens	inset	erzeugt einen „inner shadow“	initial	setzt die default Werte
blur	unscharfe Distanzangabe								
spread	die Größe des Schattens								
inset	erzeugt einen „inner shadow“								
initial	setzt die default Werte								
<pre> <div id="nav" > </div> </pre>	<p>Wir erweitern die HTML-Datei <code>navi.html</code>. Platzieren Sie im Anschluss an die Bildreferenz die Div-Box <code>nav</code> für die Navigation.</p>								
<pre> Home Aktuelles <ul class="leaf_col-2"> Taschenrechner Währungsrechner ... Menüpunkt 3 <ul class="leaf_col-3"> Unterpunkt 1 Unterpunkt 2 Unterpunkt 3 Unterpunkt 4 Unterpunkt 5 </pre>	<p>Fügen Sie darin die Listen für die Navigation ein.</p> <p>Dazu werden unsortierte Listen geschachtelt. Die unteren Ebenen werden dem Hauptmenü untergeordnet.</p>								



Wir widmen uns der Listenformatierung im Stylesheet `styles.css`.

Um das nebenstehende Ergebnis zu erzeugen folgen Sie den Erläuterungen der einzelnen CSS-Anweisungen in den kommenden Schritten.

```

39 .leaf_col-2{
40     background-color:#E2FFEC;
41 }
42
43 .leaf_col-3{
44     background-color:#CDD8E0;
45 }
46

```

Die `leaf`-Klassen dienen später dazu die Hintergrundfarbe der `ul`-Box zu manipulieren.

```

47 #nav ul{
48     list-style-type:none;
49     width: auto;
50     line-height: 200%;
51     margin: 0 auto;
52     padding:0;
53 }

```

Für alle Listenelemente der unsortierten Liste innerhalb der `div`-Box `nav` werden die Listenpunkte ausgeblendet.

```
list-style-type:none;
```

Die Breite der `ul`-Box orientiert sich an den Inhalten automatisch.

	<pre>width: auto;</pre> <p>Die Angabe der Zeilenhöhe erfolgt prozentual doppelt so hoch.</p> <pre>line-height: 200%;</pre> <p>Der Außenabstand wird für oben und unten auf 0 gesetzt, die restlichen (links und rechts) sollen sich automatisch anpassen.</p> <pre>margin: 0 auto;</pre> <p>Der Innenabstand wird für alle Seiten „genullt“.</p> <pre>padding:0;</pre>
<pre>55 #nav li { 56 width:150px; 57 float:left; 58 padding:0; 59 margin:0.3em; 60 }</pre>	<p>Für jeden Listenelement wird eine Breite von 150 Pixel fixiert.</p> <pre>width:150px;</pre> <p>Platziert man mehrere Listenelemente mit der gleichen Ausrichtung <code>float: left</code> erfolgt eine Aneinanderreihung auf der Horizontalen.</p> <pre>float:left;</pre> <p>Die Abstände (innen und außen) werden in Kurzform definiert.</p> <pre>padding:0; margin:0.3em;</pre>
<pre>62 #nav li a { 63 display:block; 64 text-align:center; 65 text-decoration:none; 66 color:#000; 67 height:35px; 68 font-size:0.8em; 69 border:1px solid #CCC; 70 } 71</pre>	<p>Mit der Anweisung <code>display:block</code> orientiert sich die Breite den Listenelementfeldes nicht variabel am Inhalt, sondern an der Box selbst.</p> <pre>display:block;</pre> <p>Der Text innerhalb der <code>li</code>-Box soll zentriert-mittig ausgerichtet sein und die standardmäßige Unterstreichung von Hy-</p>

	<p>perlinks soll unterlassen werden.</p> <pre>text-align:center; text-decoration:none;</pre> <p>Als Schriftfarbe legen wir schwarz fest.</p> <pre>color:#000;</pre> <p>Die Höhe wird einheitlich auf mindestens 35 Pixel festgelegt.</p> <pre>height:35px;</pre> <p>Die Schriftgröße erfolgt relativ, abhängig von der Angabe im Körper mit 80 Prozent.</p> <pre>font-size:0.8em;</pre> <p>Jeder Listenpunkt soll für sich einen Rahmen erhalten. 1 Pixel breit, durchgehend und in der Farbe Antrazit.</p> <pre>border:1px solid #CCC;</pre>
<pre>72 #nav li a:hover { 73 color:#18A2FA; 74 } 75</pre>	<p>Der Mouseover-Effekt soll durch einen Farbwechsel der Schriftfarbe offensichtlich werden. Ein Hellblau wird festgelegt.</p> <pre>color:#18A2FA;</pre>
<pre>78 #nav ul ul { 79 display:none; 80 position:absolute; 81 } 82</pre>	<p>Alle Listenpunkte im Untermenü werden zu Beginn ausgeblendet und ansonsten am Elternknoten ausgerichtet positioniert. Augenscheinlich erfolgt die Ausrichtung dann zentriert. Aufgeklappt werden die Untermenüs erst bei einem Mouseover.</p> <pre>display:none; position:absolute;</pre>
<pre>49 #nav li li { 50 float:none; 51 } 52</pre>	<p>Die Anweisung <code>float: none</code> erzeugt hier den Effekt, dass die Listenpunkte des Untermenüs untereinander statt nebeneinander angezeigt werden.</p> <pre>float:none;</pre>

```

87 #nav li li a {
88     background:#EBE7E6!important;
89     text-align:left;
90     height:auto;
91     width:auto;
92     line-height:1.0em;
93     padding:8px 20px;
94 }

```

Die Hintergrundfarbe der Listenpunkte im Untermenü wird mit dem Zusatz `wichtig` deklariert. Damit wird sichergestellt dass die Farbgebung für die `li`-Box dominiert. Die Wichtigkeit wird hochgestuft!

```
background:#EBE7E6!important;
```

Die Angabe der Textausrichtung sorgt dafür, dass mehrzeilige Angaben nicht zentriert sondern linksbündig ausgerichtet werden.

```
text-align:left;
```

Die Angabe der Höhe und Breite der `li`-Box erfolgt jeweils automatisch.

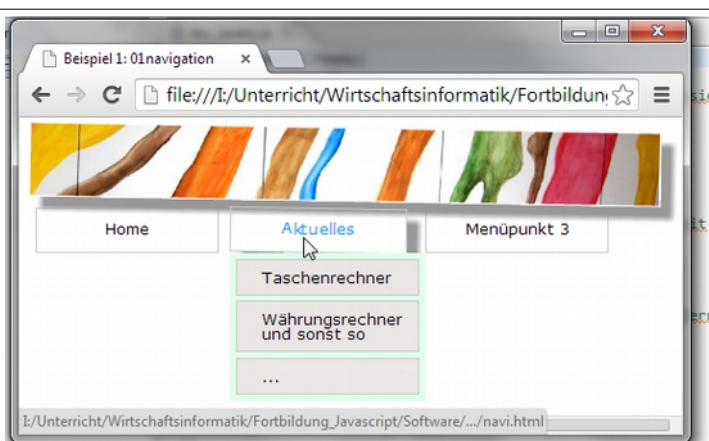
```
height:auto;
width:auto;
```

Die Zeilenhöhe wird relativ, abhängig von der Angabe in der `ul`-Box halbiert.

```
line-height:1.0em;
```

Der Innenabstand wird für oben und unten mit 8 Pixel, für links und recht mit 20 Pixel absolut festgelegt.

```
padding:8px 20px;
```

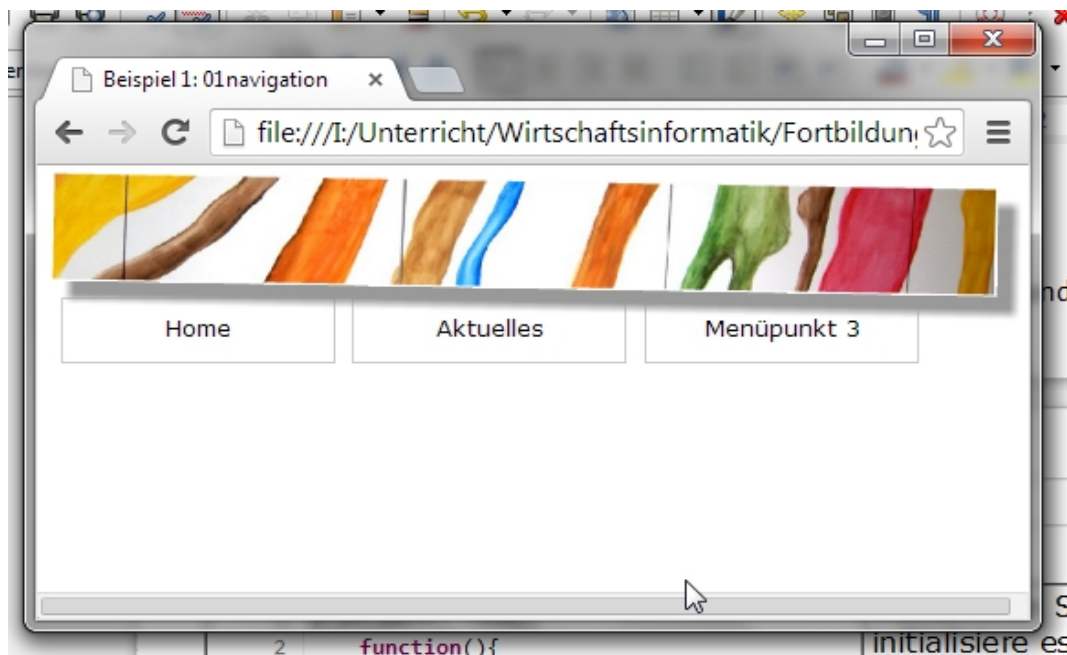


Damit die zuvor definierten Formatangaben für die Untermenüs auch ersichtlich werden muss nun die interaktive Ansprache der Listenelemente durch das Javascript implementiert werden.

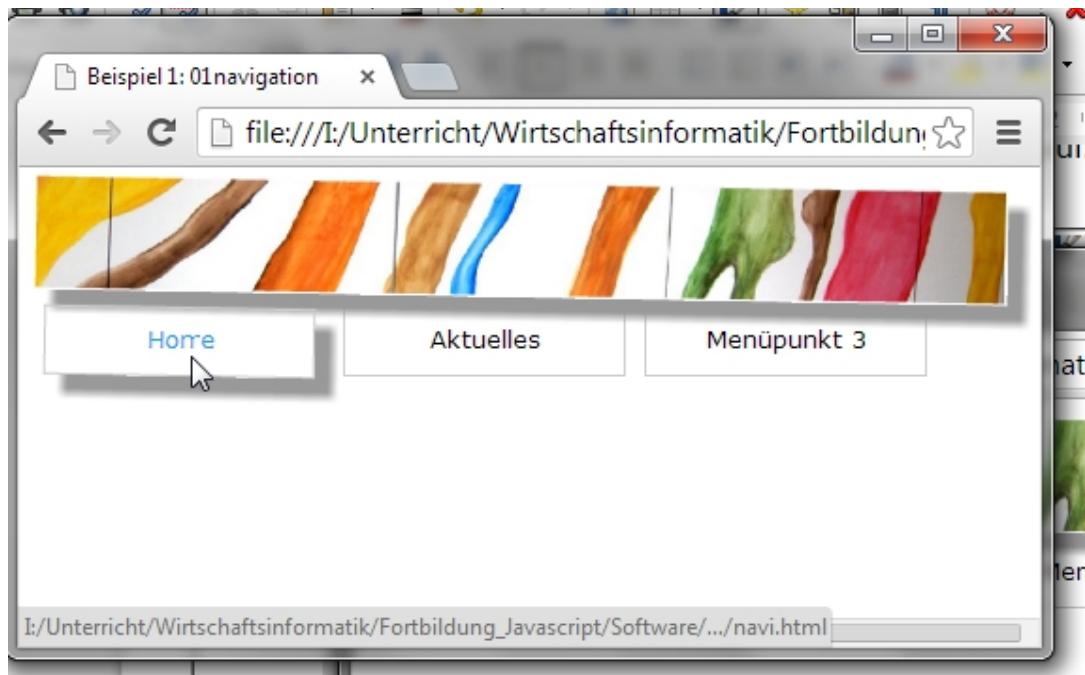
Dazu werden zwei Methoden in die Javascript-Datei `my_jquery.js` eingebettet.

Die Methode `onClickColor()` und die Methode `onClickNavigate()`

<pre> 8 function onClickColor(pSelector){ 9 var obj = pSelector; 10 obj.click(function(){ 11 var pos = obj.index(this); 12 obj 13 .removeClass() 14 .parent() 15 .removeClass(); 16 \$(this) 17 .addClass("schatten") 18 .parent() 19 .addClass("leaf_col-" + pos); 20 }); 21 }; </pre>	<p>Fügen Sie für die Methode <code>onClickColor()</code> den fehlenden Quellcode an den dafür vorgesehenen Stellen ein. Sie finden an entsprechender Stelle im Quellcode eine Erklärung in Form eines Kommentars.</p>
<pre> 23 function onClickNavigate(){ 24 var obj = \$("#nav li:has(ul)"); 25 obj.hover(function(){ 26 \$(this).find("ul").slideDown("slow"); 27 }, function(){ 28 \$(this).find("ul").hide(); 29 }); 30 }; 31 </pre>	<p>Fügen Sie für die Methode <code>onClickNavigate()</code> den fehlenden Quellcode an den dafür vorgesehenen Stellen ein. Sie finden an entsprechender Stelle im Quellcode eine Erklärung in Form eines Kommentars.</p>
<pre> 1 \$(document).ready(2 function(){ 3 var mSelector = \$("#nav a"); 4 onClickNavigate(); 5 onClickColor(mSelector); 6 }); 7 </pre>	<p>Deklarieren Sie ein Attribut <code>mSelector</code> und initialisiere es mit <code>\$("#nav a")</code></p> <p>Rufen Sie danach die beiden deklarierten Methoden auf.</p>



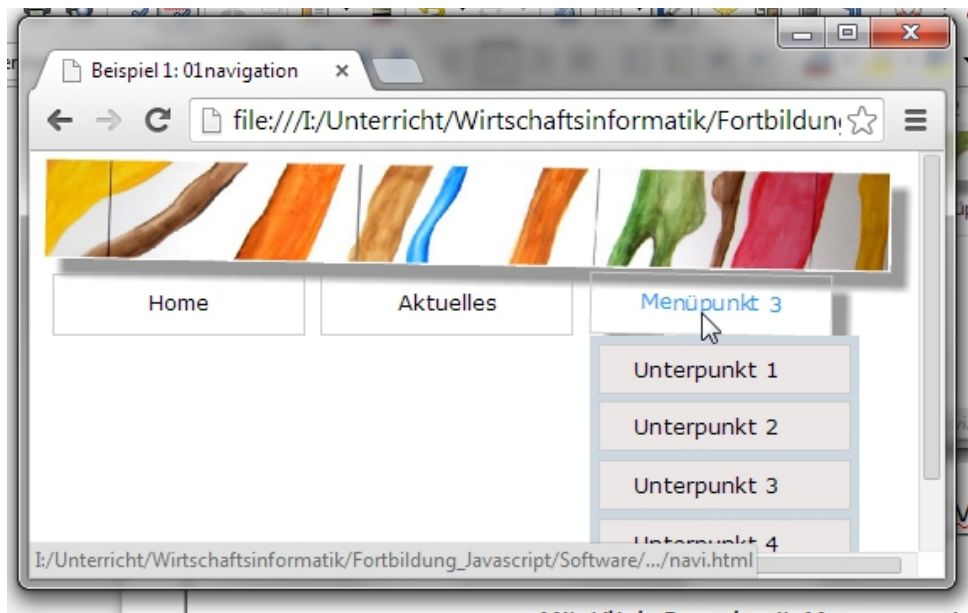
Ohne Klick und ohne Mousover



Mit Klick 1 und mit Mousover: View 1

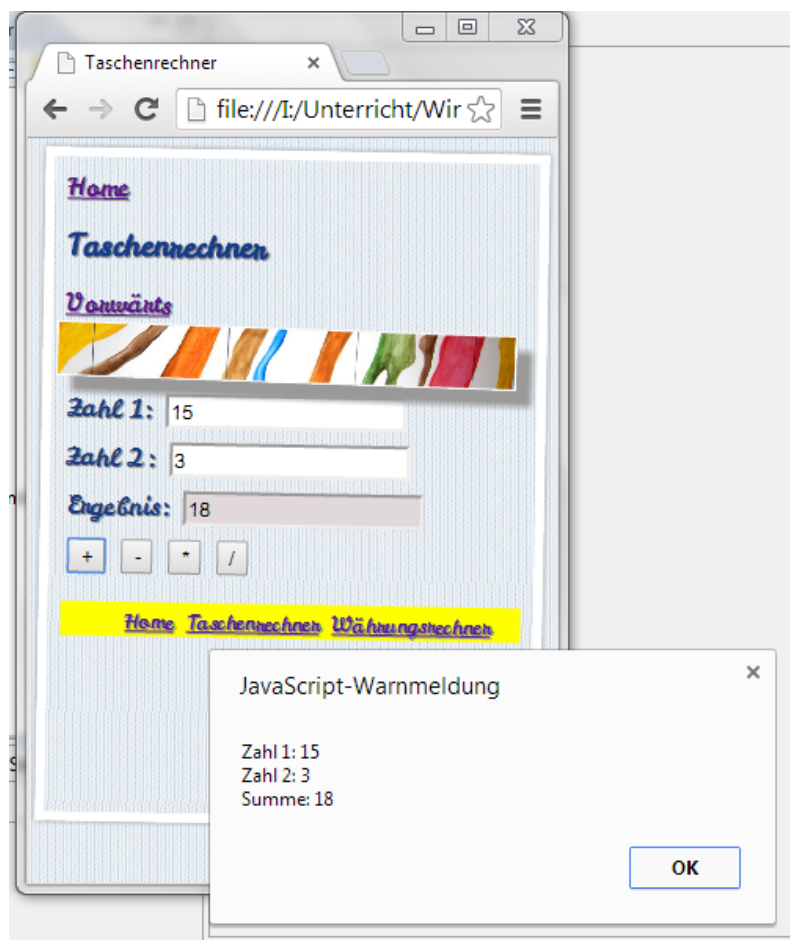


Mit Klick 2 und mit Mousover: View 2



Mit Klick 3 und mit Mousover: View 3

4.2 Beispiel 2: Dynamische Auswertung Taschenrechner



Taschenrechner

Zahl 1:

Zahl 2:

Ergebnis:

+

*

-

/

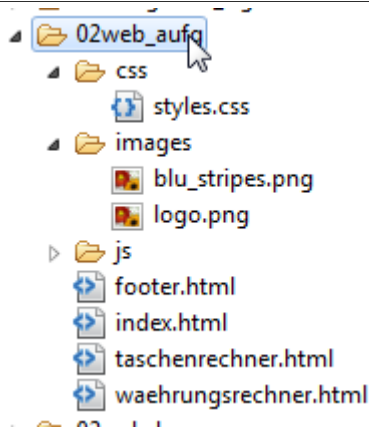
Home | Taschenrechner

Ergebnis:

XX.XX

Storyboard

Erster Taschenrechner-Entwurf der Benutzeroberfläche mit dem Textverarbeitungsprogramm.



Wir verwenden das Verzeichnis 02web_aufg, um es zu erweitern.



Wir widmen uns im Ersten Schritt dem Taschenrechnerformular. Öffnen Sie dazu die HTML-Datei `taschenrechner.html`

Ergänzen Sie im `form`-Container den HTML-Quellcode für die Texteingabefelder für `zahl1` und `zahl2`

```
<div class = "container">
  <label for="zahl1">Zahl 1:</label>
  <input type="text" name="zahl1" id="zahl1"
value="" />
</div>

<div class = "container">
  <label for="zahl2">Zahl 2:</label>
  <input type="text" name="zahl2" id="zahl2"
value="" />
</div>
```



Ergänzen Sie im `form`-Container den HTML-Quellcode für das Ergebnisfeld. Da in diesem Feld ausschließlich Ausgaben erfolgen soll deklarieren wir das Feld als schreibgeschützt. Wir nutzen dazu das Komponenten-Flag

```
readonly="readonly"
```

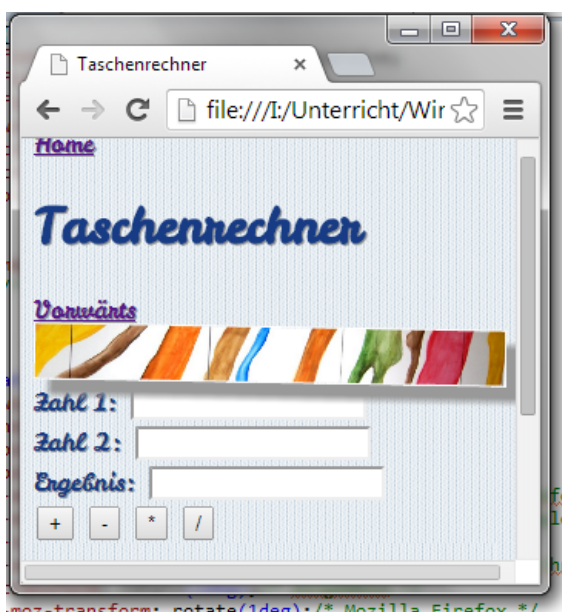
Quellcode für das Ergebnisfeld:

```
<div class = "container">
  <label for="ergebnis">Ergebnis:</label>
  <input type="text" name="ergebnis"
id="ergebnis" class = "readonly" value="" rea-
donly="readonly"/>
</div>
```



Ergänzen Sie im form-Container den HTML-Quellcode für die Schaltflächen der Rechnoperationen.

```
<div class = "container">
  <button type="button" id="addition">
      +
  </button>
  <button type="button" id="subtraktion">
      -
  </button>
  <button type="button" id="multiplikation">
      *
  </button>
  <button type="button" id="division">
      /
  </button>
</div>
```



Wir widmen uns nun den noch zu ergänzenden Formatangaben in der Stylesheet-Datei `styles.css`.

Für den body-Container definieren wir die gängigen Formatangaben für Webdokumente.

Zur Abwechslung nutzen wir die Google Schrift „Leckerli One“ in der kursiven Variante mit einem normalen Schriftgewicht. Im Kopfteil des HTML-Dokuments wurde dazu bereits auf die Googleapi-Schriftquelle verwiesen.

```
font-family: 'Leckerli One', cursive;
font-weight: normal;
```

Die Textfarbe ist mit einem dunklen Blau-ton an das Hintergrundbild „blue stripes“ angepasst. Der Text bekommt einen leicht horizontal und vertikal verschobenen Schatten in Antrazit.

```
color:#173b84;
text-shadow: 1px 1px 0px #999;
Background-image:
url(../images/blu_stripes.png);
```

Beispielgebend war ein Hintergrundbild

aus dem online Repertoire der Subtle Patterns [SUB00].

Die in den Quellcodebeispielen eingebundene Grafik `blu_stripes.png` ist selbst erstellt.



Definieren Sie im nächsten Schritt die Formatangaben für den Page-Container. Fügen Sie dazu erst die CSS-Anweisungen für die `.page`-Klasse ein.

Wir definieren einen 6 Pixel breiten weißen, durchgehenden Rahmen

```
border:6px solid #fff;
```

Wie zuvor in der `.schatten`-Klasse definieren wir die `.page`-Klasse einen Schatten und die Rotation für den ganzen Page-Container.

```
/* Allgemein */
box-shadow: 3px 2px 10px #999;
/* Mozilla Firefox */
-moz-box-shadow: 3px 2px 10px #999;
/* Google Chrome */
-webkit-box-shadow: 3px 2px 10px #999;
/* Internet Explorer */
-ms-box-shadow: 3px 2px 10px #999;
/* Opera */
-o-box-shadow: 3px 2px 10px #999;
/* Allgemein */
transform: rotate(1deg);
/* Mozilla Firefox */
-moz-transform: rotate(1deg);
/* Internet Explorer */
-ms-transform: rotate(1deg);
/* Opera */
-o-transform: rotate(1deg);
/* Google Chrome */
-webkit-transform: rotate(1deg);
```

```
8  iframe[seamless]{
9      background-color: transparent;
10     border: 0 none transparent;
11     padding: 0;
12     overflow: hidden;
13     width:100%
14 }
```

Die Formatierung des iFrames ist zwingend erforderlich, um das bisher noch sichtbare iFrame-Fenster auszublenden.

```
overflow: hidden;
```

Wird definiert für den Fall, dass der In-

halt über den Rand laufen würde. Mit dem Wert `hidden` der übergroße Inhalt abgeschnitten und nicht angezeigt.

Das Attribut gibt an, dass der `iframe` ohne sichtbaren Rahmen in die Seite eingebettet werden soll. Das funktioniert auf diese Weise auch beim IE.



Auch die Fußzeile >> `footer.html` ist Bestandteil des Page-Containers. Sie ist als `iFrame` in die restlichen HTML-Dokumente eingebunden und enthält eine Liste mit Verweisen. Für die im Page-Container enthaltene Listenpunkte `.page li` definieren wir zusätzlich die bereits in den vorhergehenden Beispielen besprochenen Formatangaben.

```
list-style-type:none;
color:#173b84;
font-size:1.2em;
line-height:1em;
text-align:left;
```

Für den `footer`-Container wird die Hintergrundfarbe gelb definiert.

```
background-color:yellow;
```

Da die Listenpunkte im Footer `footer li` auf der Horizontalen ausgerichtet werden sollen, müssen wir einige zuvor gemachte Angaben erweitern bzw. sogar überschreiben.

```
display:inline;
list-style-type:none;
padding:0.1em;
font-size:0.8em;
line-height:1em;
text-align:left;
```

Definieren Sie im `navbar`-Container des HTML-Dokuments `footer.html` die Listenpunkte mit den Verweisen der Fußzeile ein.

```
<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="taschenrechner.html">Taschenrechner</a></li>
  <li><a href="waehrungsrechner.html">Währungsrechner</a></li>
</ul>
```



vorher



nachher

Nun zu den Feinheiten. Der Innenabstand von oben und von links wird reguliert.

```
70 .header{
71     padding-top:0.5em;
72     padding-left:0.5em;
73 }
```

Taschenrechner

vorher

Taschenrechner

nachher

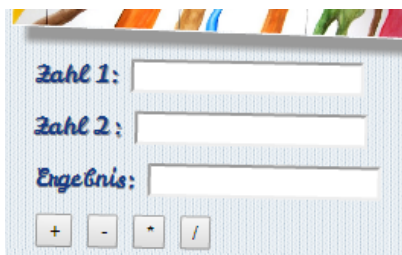
Die Regulierung der Schriftgröße für Überschriften vom Typ `h1` im Header.

```
77 .header h1{
78     font-size:1.2em;
79 }
```

Der Innenabstand von oben und von links wird auch für den `content`, das `form` und den `container` reguliert.




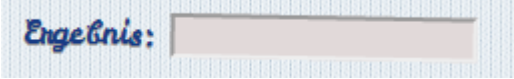
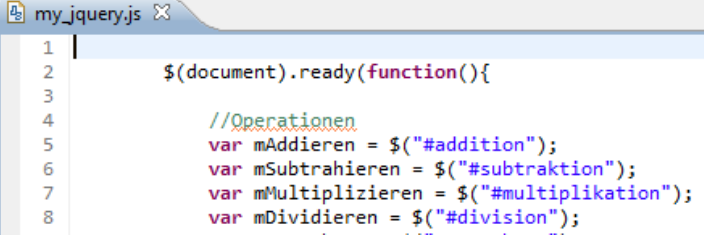
vorher

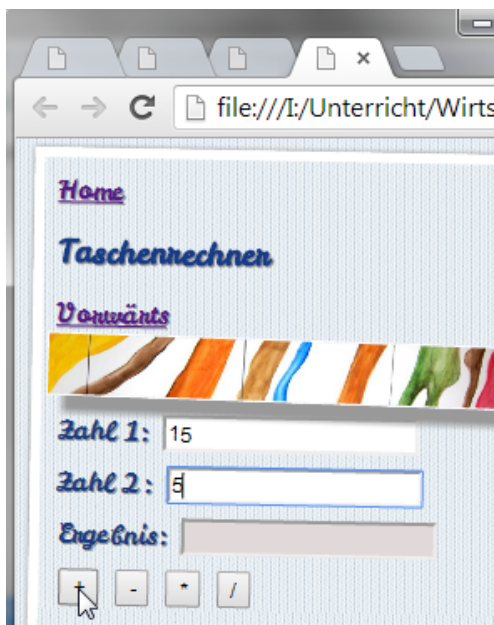


nachher

```
79 .content {
80     padding-top:0.5em;
81     padding-left:0.5em;
82 }
```

```
84 .form {
85     padding-top:0.5em;
86     padding-left:0.5em;
87 }
88
```

	<pre>89 .container{ 90 padding-top:0.5em; 91 padding-left:0.5em; 92 }</pre>
 <p style="text-align: center;">vorher</p>	<pre>96 .readonly{ 97 background-color:#E1D9D9; 98 }</pre>
 <p style="text-align: center;">nachher</p>	
 <pre>1 \$(document).ready(function(){ 2 3 4 //Operationen 5 var mAddieren = \$("#addition"); 6 var mSubtrahieren = \$("#subtraktion"); 7 var mMultiplizieren = \$("#multiplikation"); 8 var mDividieren = \$("#division");</pre>	<p>Bisher haben wir uns rein um das Layout gekümmert. Nun wollen wir uns der Formelarauswertung widmen.</p> <p>Anhand der im Formular festgelegten ids für die Schaltflächen können wir die click-Events identifizieren und Aktionen auslösen. Wir übernehmen dazu die ids in die lokalen Attribute mAddieren, mSubtrahieren, mMultiplizieren und mDividieren.</p> <p>Übernehmen Sie die Anweisungen, an der dafür ausgewiesenen Stelle, in der Javascript-Datei my_jquery.js.</p> <pre>var mAddieren = \$("#addition"); var mSubtrahieren = \$("#subtraktion"); var mMultiplizieren = \$("#multiplikation"); var mDividieren = \$("#division");</pre>
<pre>11 //Textfelder für Eingaben 12 var mZahl1 = \$("#zahl1"); 13 var mZahl2 = \$("#zahl2");</pre>	<p>Im Anschluss daran deklarieren und initialisieren wir Attribute für die Eingabewerte.</p> <p>Für alle Textfelder wurden dazu im Formular ids festgelegt. Damit im vorliegenden Fall die Eingaben zahl1 und zahl2 des Benutzers gelesen und verarbeitet werden können, übernehmen wir die ids in die lokalen Attribute mZahl1 und mZahl2.</p>



```
var mZahl1 = $("#zahl1");
var mZahl2 = $("#zahl2");
```

Für die Operation Addieren deklarieren wir eine Methode:

```
function onClickAddieren(pOperation, pZahl1,
pZahl2){
};
```

Die Operation und Eingabewerte werden als Parameter übergeben.

In der Methode selbst passiert folgendes:

Wir merken uns die Operation in einem lokalen Attribut `operation` vor. Außerdem deklarieren wir ein Attribut für das `ergebnis` und initialisieren es mit dem Standardwert 0.

```
var operation = pOperation;
var ergebnis = 0;
```

Für den Fall, dass der Benutzer einen Wert für `zahl1` und `zahl2` eingibt und auf die Schaltfläche `addieren` klickt sollen diese Werte übernommen werden.

Es folgt die `click()`-Methode. Diese wird am Objekt der `operation` aufgerufen und übernimmt dabei die Funktion eines Listeners. Das funktioniert wie ein Fühler, der an der Oberfläche lauscht und registriert wenn die Schaltfläche angeklickt wird.

```
operation.click(function(){
});
```

Innerhalb dieser Funktion werden als nächstes die Attribute für die Eingabewerte deklariert und initialisiert.

Damit wir mit den eingegebenen Werten Berechnungen durchführen können, müssen sie analysiert (geparst) und gewandelt werden.

```
var zahl1 = parseFloat(pZahl1.val());
var zahl2 = parseFloat(pZahl2.val());
```

Es folgt gleich im Anschluss eine Kontrolle ob die Werte auch numerischer Natur sind.

Wir nutzen dazu den Methodenaufruf der booleschen jQuery-Methode `jQuery.isNumeric(wert)`. Sie liefert uns den Rückgabewert `true` für numerische Werte und `false` für nicht numerische Werte. Beide Fälle werden mit der Kontrollstruktur `if()-else{}` abgefangen und behandelt.

```
if(jQuery.isNumeric(zahl1) && jQuery.isNumeric(zahl2)){
    //JA-Fall
}else{
    //NEIN-Fall
}
```

Im JA-Fall (IF):

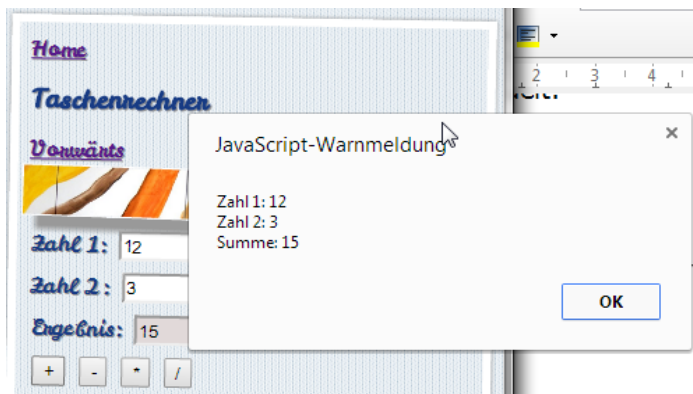
Wird das Ergebnis berechnet.
`ergebnis = zahl1 + zahl2;`

Die `zahl1`, `zahl2` und das `ergebnis` werden in einem Dialogfenster (`alert`) angezeigt. Mit dem Ausdruck „`\n`“ werden Zeilensprünge erzeugt. Die Verkettung von Strings (Zeichenketten) und Attributwerten erfolgt durch den „`+`“ Operator.

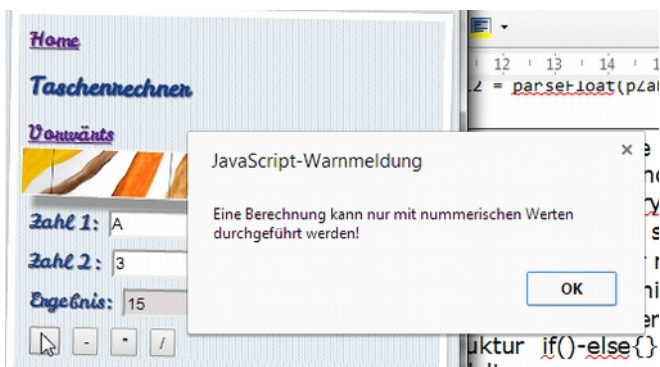
```
alert("Zahl 1: " + zahl1 + "\n" + "Zahl 2: " + zahl2 + "\n" + "Summe: " + ergebnis );
```

Mit einem Klick auf `ok` im Dialogfenster erfolgt dann abschließend die Ausgabe des `ergebnis` im Textfeld mit der `id` `ergebnis`.

```
$("#ergebnis").val(ergebnis);
```



JA-Fall (IF)



NEIN-Fall (ELSE)

```

15 /*Taschenrechner*/
16 function onClickAddieren(pOperation, pZahl1, pZahl2){
17     var operation = pOperation;
18     var ergebnis = 0;
19     operation.click(function(){
20         var zahl1 = parseFloat(pZahl1.val());
21         var zahl2 = parseFloat(pZahl2.val());
22
23         if(jQuery.isNumeric(zahl1) && jQuery.isNumeric(zahl2)){
24             ergebnis = zahl1 + zahl2;
25             alert("Zahl 1: " + zahl1+ "\n" + "Zahl 2: " + zahl2+ "\n" + "Summe: " + ergebnis );
26             $("#ergebnis").val(ergebnis);
27         }else{
28             alert("Eine Berechnung kann nur mit numerischen Werten durchgeführt werden!");
29         }
30     });
31 };
32

```

Zur Kontrolle

```

1
2     $(document).ready(function(){
3
4         //Operationen
5         var mAddieren = $("#addition");
6
7         //Textfelder für Eingaben
8         var mZahl1 = $("#zahl1");
9         var mZahl2 = $("#zahl2");
10
11         //Methoden: Verhaltensweisen
12         onClickAddieren(mAddieren,mZahl1,mZahl2);
13     });

```

Im NEIN-Fall (ELSE):
Erfolgt lediglich die Ausgabe der Meldung "Eine Berechnung kann nur mit numerischen Werten durchgeführt werden!" im Dialogfenster.

```

alert("Eine Berechnung kann nur mit numerischen Werten durchgeführt werden!");

```

Um die implementierte Methode testen zu können muss der Methodenaufruf erfolgen.

Ergänzen Sie die ready()-Methode um den Methodenaufruf:

```

onClickAddieren(mAddieren,mZahl1,mZahl2);

```

Testen Sie die Anwendung.

```

onClickSubtrahieren(mSubtrahieren,mZahl1,mZahl2);
onClickMultiplizieren(mMultiplizieren,mZahl1,mZahl2);
onClickDividieren(mDividieren,mZahl1,mZahl2);

```

Aufgabe:

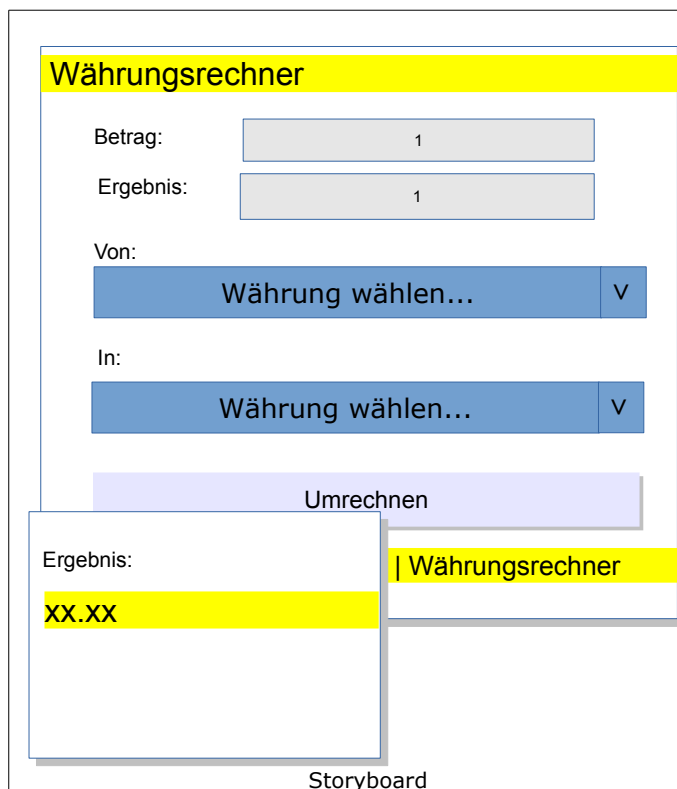
Erweitern Sie die Javascript-Datei `my_jquery.js`.

Implementieren Sie mit ihren erworbenen Kenntnissen den fehlenden Quellcode für die übrigen Rechenoperationen subtrahieren, multiplizieren und dividieren.

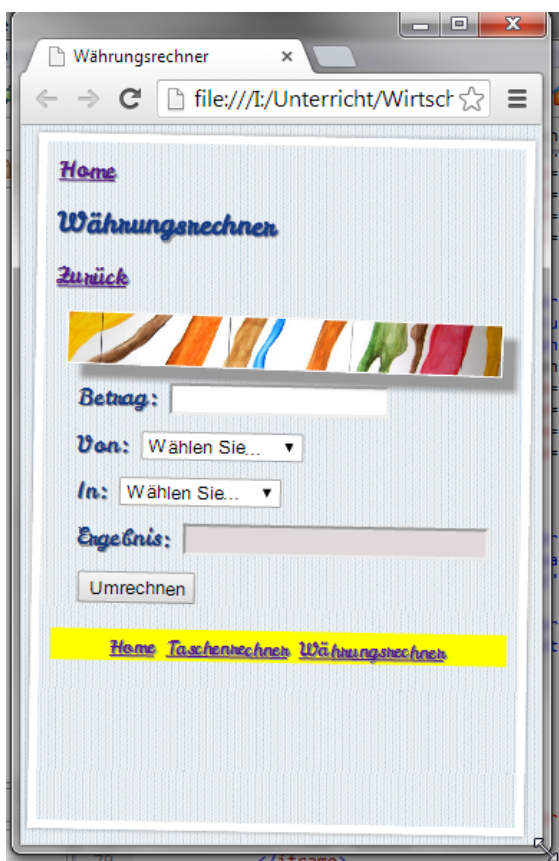
Zusatzaufgabe:

Stellen Sie sicher, dass der Benutzer bei einer Division durch null von seinem Tun unterrichtet wird. Melden Sie in diesem Fall: „Eine Division durch 0 ist nicht möglich!“

4.3 Beispiel 3: Dynamische Auswertung Währungsrechner



Erster Währungsrechner-Entwurf der Benutzeroberfläche mit dem Textverarbeitungsprogramm. Ein einfacheres Beispiel bietet die Literatur [WEN00]



Im ersten Schritt ergänzen wir das Formular für den Währungsrechner. Öffnen Sie dazu das im Projekt bereits enthaltene HTML-Dokument `waeahrungsrechner.html` mit dem HTML-Editor.

Wir möchten mit den folgenden Schritten das nebenstehende Ergebnis erzeugen.

<pre><div class ="container"> <label for="betrag">Betrag:</label> <input type="text" name="betrag" id="betrag" value="" /> </div></pre>	<p>Das Dokument ist gleich aufgebaut, wie das HTML-Dokument für den Taschenrechner. Es fehlen die Formular-Komponenten.</p> <p>Erzeugen Sie, wie bereits im Taschenrechnerformular praktiziert einen Container der Klasse <code>.container</code> und implementieren Sie darin das Label und das Texteingabefeld für den Betrag.</p>
<pre><div class ="container"> <label for="waehrung1">Von:</label> <select name="waehrung1" id="waehrung1"> <option>Wählen Sie...</option> <option value="1">Euro</option> <option value="2">Britische Pfund</option> <option value="3">US Dollar</option> <option value="4">Yen</option> </select> </div></pre>	<p>Außer den bereits bekannten Labels, Textfelder und Schaltflächen nutzen wir für die Auswahl der Ausgangs- und Zielwährung die allgemein üblichen Drop-Down-Menüs.</p> <p>Der absolute Vorteil dieser Komponente ist, dass der Benutzer quasi keine Fehler bei der Eingabe machen kann.</p> <p>In HTML wird das Drop-Down-Menü durch einen <code>Select</code>-Container realisiert. Dieser Container enthält die Auswahlfelder sogenannte <code>options</code>. Der <code>Select</code>-Container bekommt für die Ausgangswährung <code>id waehrung1</code>. Die <code>optionen</code> erhalten ein Wert <code>value</code> 1 bis 4.</p> <p>Fügen Sie im Anschluss an den Betrag den Container <code>container</code> für die Ausgangswährung (Von) ein. Kopieren und verwenden Sie den Quellcode dann für die Zielwährung (In). Benennen Sie den <code>Select</code>-Container für die Zielwährung mit der <code>id waehrung2</code>, die Auswahlfelder <code>optionen</code> bleiben die selben.</p>
<pre><div class ="container"> <label for="zielwaehrung">Ergebnis:</label> <input type="text" name="zielwaehrung" id="zielwaehrung" class ="readonly" value="" readonly="readonly" size="30"/> </div></pre>	<p>Fügen Sie den Quellcode für das Ergebnisfeld mit der <code>id ergebnis</code> ein. Wie zuvor im Taschenrechnerformular sollen hier nur Ausgaben erfolgen. Implementieren Sie deshalb das Komponentenflag „<code>readonly</code>“ und nutzen Sie die bereits existierende CSS-Klasse <code>.readonly</code>, um die Hintergrundfarbe in Grau farblich abzusetzen.</p> <pre>class ="readonly" value="" readonly="readonly"</pre>

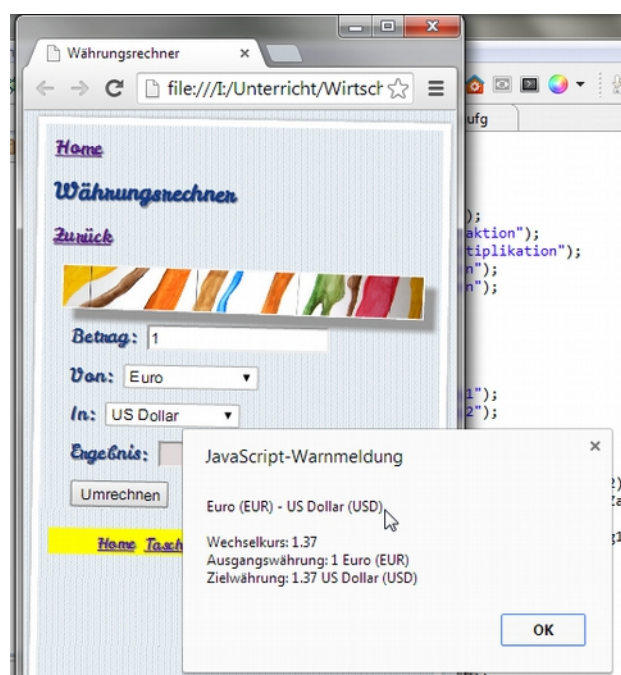
	<p>Legen Sie für die Größe des Textfeldes 30 Zeichen fest.</p> <p><code>size="30"</code></p>
<pre><div class ="container"> <button type="button" id="umrechnen"> Umrechnen </button> </div></pre>	<p>Definieren Sie die Schaltfläche für die Operation umrechnen.</p>

Da der Währungsrechner auf das gleiche Stylesheet `styles.css` zurückgreift, wie unsere Taschenrechner-Anwendung werden wir die CSS-Klassen und -Boxen einfach wiederverwenden.

Wir können uns also gleich der Erweiterung der eigentlichen Funktionalität widmen.

Öffnen Sie dazu die Javascript-Datei `my_jquery.js`.

Zum Verständnis



Ein Beispiel. Alle Wechselkurse beziehen sich auf eine Einheit der Ausgangswährung.

Die Angaben

US Dollar (USD) = 1.37

Britische Pfund (GBP) = 0.82

Japanischer Yen (JPY) = 140.71

bedeuten, dass **1 Euro** 1.37706 USD bzw. 140,71 JPY usw. entspricht.

Wir lassen bei unserer Betrachtung erst einmal außer acht, dass für den Ankauf und Verkauf von Devisen unterschiedliche Wechselkurse existieren. Allerdings sollte es natürlich möglich sein die Anwendung nachträglich zu erweitern. [OAN00]

```
$(document).ready(function(){
    //Operationen
    var mAddieren = $("#addition");
    var mSubtrahieren = $("#subtraktion");
    var mMultiplizieren = $("#multiplikation");
    var mDividieren = $("#division");
    var mUmrechnen = $("#umrechnen");

    //Textfelder für Eingaben
    var mZahl1 = $("#zahl1");
    var mZahl2 = $("#zahl2");
    var mBetrag = $("#betrag");
    var mWaehrung1 = $("#waehrung1");
    var mWaehrung2 = $("#waehrung2");

    //Methoden: Verhaltensweisen
    onClickAddieren(mAddieren,mZahl1,mZahl2);
    onClickSubtrahieren(mSubtrahieren,mZahl1,mZahl2);
    onClickMultiplizieren(mMultiplizieren,mZahl1,mZahl2);
    onClickDividieren(mDividieren,mZahl1,mZahl2);
    onClickUmrechnen(mUmrechnen, mBetrag, mWaehrung1, mWaehrung2);
});
```

Der Benutzer hat vier Angaben zu tätigen

1. den Betrag `betrag`
2. die Ausgangswährung `waehrung1`
3. die Zielwährung `waehrung2`
4. die Operation `umrechnen`

Alle Angaben macht er auf der Benutzeroberfläche, sodass wir in der Lage sind die Informationen auszulesen und zu verarbeiten.

Wir erweitern nun das Skript nach dem gleichen Schema wie das zuvor beim Taschenrechner der Fall war. Da wir vier statt drei Eingabewerte haben wird die Methode `onClickUmrechnen()` mit vier Parametern ausgestattet (siehe 1 – 4).

Nun zur Implementation der Methode selbst. Es gibt unzählige Möglichkeiten einen Währungsrechner zu programmieren.

Im Rahmen dieses Skriptes werden wir uns auf eine Lösungsmöglichkeit beschränken.

```
function onClickUmrechnen(pOperation, pBetrag,
    pWaehrung1, pWaehrung2){
};
```

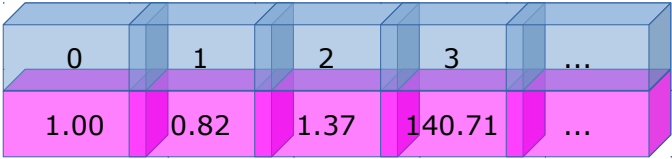
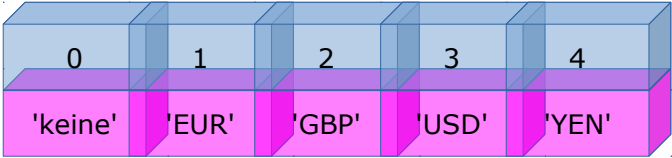
Wir deklarieren darin Attribute die wir im Rahmen der Verarbeitung benötigen werden.

```
var operation = pOperation;
var zielwaehrung = 0;
```

Das Attribut `meldung` erhält später Meldungen die im Dialogfenster für den Benutzer erscheinen sollen.

```
var meldung = null;
```

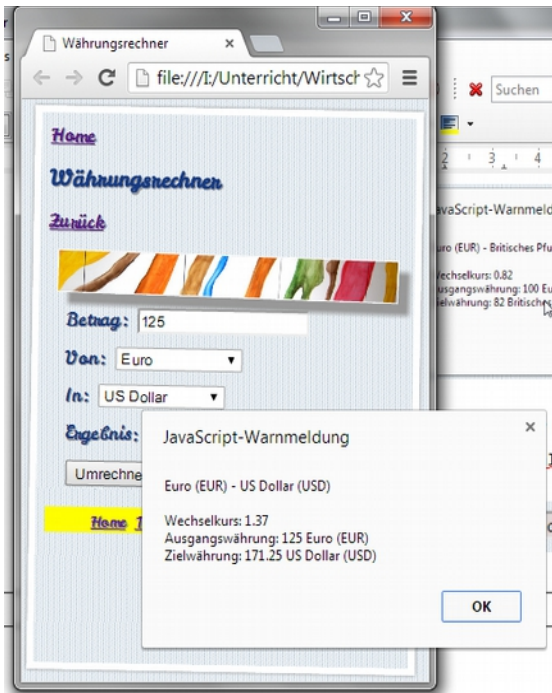
Das Attribut `wk_index` erhält im Rahmen der Auswertung den „Index im Daten-Container“ für den Wechselkurs, der abhängig von der ausgewählten Kombination an Währungen (Ausgangswährung und

	<p>Zielwahrung) festgelegt wurde.</p> <pre>var wk_index = 0;</pre>
 <p style="text-align: center;">wechselkurs</p>  <p style="text-align: center;">waehrung</p>	<p>Wir nutzen zwei nummerierte Listen als Daten-Container (Arrays):</p> <ol style="list-style-type: none"> den Wechselkurs-Container <code>wechselkurs</code> <pre>var wechselkurs = new Array();</pre> den Wahrungs-Container <code>waehrung</code> <pre>var waehrung = new Array();</pre>
<pre>//Kurse vom 4.3.2014 var wechselkurs = new Array(); wechselkurs[0] = 1.00; // w1 == w2 wechselkurs[1] = 0.82; //Euro (EUR) - Britisches Pfund (GBP) wechselkurs[2] = 1.37; //Euro (EUR) - US Dollar (USD) wechselkurs[3] = 140.71; //Euro (EUR) - Japanischer Yen (JPY) wechselkurs[4] = 1.21; //Britisches Pfund (GBP) - Euro (EUR) wechselkurs[5] = 1.67; //Britisches Pfund (GBP) - US Dollar (USD) wechselkurs[6] = 169.55; //Britisches Pfund (GBP) - Japanischer Yen (JPY) wechselkurs[7] = 0.73; //US Dollar (USD) - Euro (EUR) wechselkurs[8] = 0.60; //US Dollar (USD) - Britisches Pfund (GBP) wechselkurs[9] = 101.39; //US Dollar (USD) - Japanischer Yen (JPY) wechselkurs[10] = 0.01; //Japanischer Yen (JPY) - Euro (EUR) wechselkurs[11] = 0.01; //Japanischer Yen (JPY) - Britisches Pfund (GBP) wechselkurs[12] = 0.01; //Japanischer Yen (JPY) - US Dollar (USD)</pre>	<p>Der Wechselkurs-Container wird mit den Wechselkursen initialisiert die abhangig von der Kombination aus Ausgangs- und Zielwahrung ermittelt werden konnen.</p> <p>Zur Aktualisierung konnen Sie den Tagesaktuellen Kursrechner von Oanda http://www.oanda.com/currency/converter/ [OAN00] nutzen.</p>
<pre>var waehrung = new Array(); waehrung[0] = "Keine Wahrung"; waehrung[1] = "Euro (EUR)"; waehrung[2] = "Britisches Pfund (GBP)"; waehrung[3] = "US Dollar (USD)"; waehrung[4] = "Japanischer Yen (JPY)";</pre>	<p>Der Wahrungs-Container enthalt die Detailbezeichnungen fur die verwendeten Wahrungen. Der Index stimmt mit den Werten <code>values</code> aus dem <code>Select</code>-Container, des im Formular der HTML-Datei <code>waehrungsrechner.html</code> eingebetteten Drop-Down-Menus, uberein.</p>
<p>Werte lesen:</p> <pre>operation.click(function(){ var ausgangswaehrung = parseFloat(pBetrag.val().replace(',','.')); var waehrung1 = parseFloat(pwaehrung1.val()); var waehrung2 = parseFloat(pwaehrung2.val());</pre>	<p>Die Klick-Methode enthalt nun alle restlichen Anweisungen</p> <ol style="list-style-type: none"> Werte lesen Betrag, Ausgangswahrung, Zielwahrung Werte verarbeiten Prufung, Berechnung Werte ausgeben Dialogfenster, Ergebnisfeld <p>Das klassische Prinzip der Informatik: Eingabe, Verarbeitung und Ausgabe (kurz</p>

	<p>EVA-Prinzip). Ausgelöst wird der Vorgang durch den Klick auf die Schaltfläche umrechnen.</p> <pre>operation.click(function(){ });</pre> <p>Innerhalb der Methode werden anfangs die Werte gelesen. Besonderheit hier ist der Methodenaufruf der jQuery-Methode <code>replace('alt','neu')</code>.</p> <p>Sie stellt hier sicher, dass der Nutzer einen Dezimalwert als Betrag eingeben kann, ohne dass fehlerhafte Werte übernommen werden. Die Methode ersetzt das Komma ggf. durch einen Punkt, bevor der Wert geparkt (analysiert & gewandelt) wird.</p> <pre>var ausgangswaehrung = parseFloat(pBetrag.val().replace(',', '.'));</pre> <p>Lesen der Währungsnummern</p> <pre>values var waehrung1 = parseFloat(pWaehrung1.val()); var waehrung2 = parseFloat(pWaehrung2.val());</pre>
<p>Werte verarbeiten: Prüfung</p> <pre>if(waehrung1 == waehrung2){ //Euro (EUR) - Euro (EUR) //Britisches Pfund (GBP) - Britisches Pfund (GBP) //US Dollar (USD) - US Dollar (USD) //Japanischer Yen (JPY) - Japanischer Yen (JPY) wk_index = 0; }else if(waehrung1 == 1 && waehrung2 == 2){ //Euro (EUR) - Britisches Pfund (GBP) wk_index = 1; }else if(waehrung1 == 1 && waehrung2 == 3){ //Euro (EUR) - US Dollar (USD) wk_index = 2; }else if(waehrung1 == 1 && waehrung2 == 4){ //Euro (EUR) - Japanischer Yen (JPY) wk_index = 3;</pre>	<p>Wir nutzen die Kontrollstruktur <code>if(){}else if(){} </code> um die insgesamt 16 möglichen Kombinationen zu prüfen und um den Index für den Wechselkurs festzulegen. Der ist relevant für die darauf folgende Berechnung.</p> <p>Übernehmen Sie den Quellcode und ergänzen Sie selbstständig die fehlenden Fälle.</p>
<p>Werte verarbeiten: Berechnung</p> <pre>zielwaehrung = Math.round((ausgangswaehrung * wechsellkurs[wk_index])*100) / 100 + " " + waehrung[waehrung2]; ausgangswaehrung = ausgangswaehrung + " " + waehrung[waehrung1]; meldung = waehrung[waehrung1] + " - " + waehrung[waehrung2];</pre>	<p>Zur Erläuterung. Wir nutzen die Umrechnungsformel</p> <pre>zielwaehrung = ausgangswaehrung * wechsellkurs</pre> <p>Um kaufmännisch, auf 2 Dezimalstellen zu Runden erweitert man den Term und nutzt einen Methodenaufruf der jQuery-Methode <code>Math.round(wert)</code>.</p>

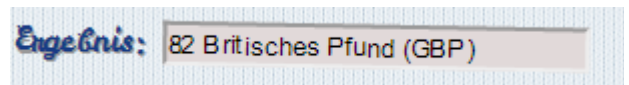
	<p>Die Methode rundet allerdings kaufmännisch auf Ganze Zahlen.</p> <p>Ein kleiner Trick und wir können die Methode zum kaufmännischen Runden auf zwei Dezimalstellen trimmen [WEB01]:</p> <pre>Math.round((ausgangswaehrung * wechselkurs) * 100) / 100</pre> <p>Der Wechselkurs muss abhängig vom ermittelten <code>wk_index</code> dem Daten-Container <code>wechselkurs</code> entnommen werden.</p> <pre>wechselkurs[wk_index]</pre> <p>Dem berechneten Wert für die <code>zielwaehrung</code> wird die Detailbezeichnung angehängt. Dazu wird die Detailbezeichnung abhängig von dem Wert <code>value</code> für die <code>waehrung2</code> aus dem Daten-Container <code>waehrung</code> ermittelt.</p> <pre>waehrung[waehrung2]</pre> <p>Dem Betrag (<code>ausgangswaehrung</code>) wird für die Ausgabe ebenfalls die Detailbezeichnung angehängt.</p> <pre>waehrung[waehrung1]</pre> <p>Abschließen wird das Attribut <code>meldung</code> mit den Detailbezeichnungen erzeugt.</p> <pre>meldung = waehrung[waehrung1] + " - " + waehrung[waehrung2];</pre>
Werte ausgeben:	<p>Die Ausgabe im Dialogfenster:</p> <pre>alert(meldung + "\n\n" + "Wechselkurs: " + wechselkurs[wk_index] + "\n" + "Ausgangswaehrung: " + ausgangswaehrung + "\n" + "Zielwaehrung: " + zielwaehrung);</pre>

```
//Ausgabe
alert(meldung + "\n\n" + "Wechselkurs: " + wechsellkurs[wk_index]
+ "\n" + "Ausgangswährung: " + ausgangswaehrung + "\n"
+ "Zielwährung: " + zielwaehrung);
$("#zielwaehrung").val(zielwaehrung);
```



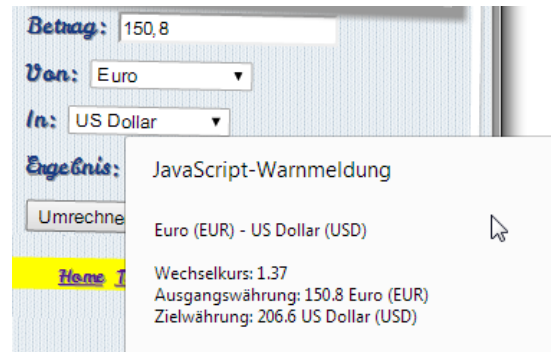
Die Ausgabe in der im Ergebnisfeld:

`$("#zielwaehrung").val(zielwaehrung);`



Testen Sie auch die Grenzfälle.

Eingabe einer Dezimalzahl:



Zusatzaufgabe:

Erweitern Sie die Rechnung um die Kontrolle des eingegebenen Betrages. Die Kontrolle soll sicherstellen, dass der Wert auch numerischer Natur ist (siehe Taschenrechner-Beispiel)

Im JA-Fall (IF):

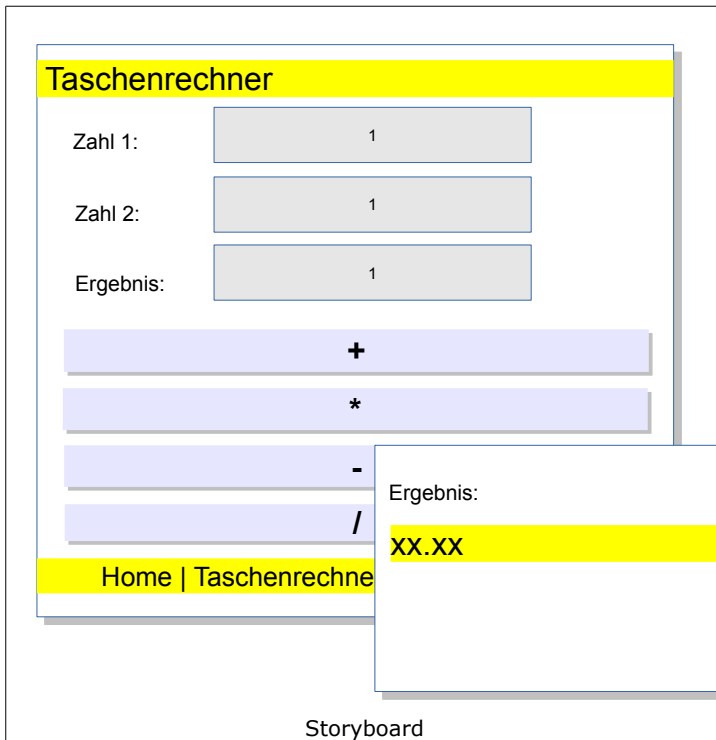
Erfolgt die Prüfung, Berechnung und Ausgabe des Ergebnisses.

	<p>Im NEIN-Fall (ELSE): Erfolgt lediglich die Ausgabe der Meldung "Eine Berechnung kann nur mit einem numerischen Wert durchgeführt werden!" im Dialogfenster.</p>
--	--

5 Erweiterung zur Mobile-Web-App mit JQuery-Mobile

Erweiterung zur Mobile-Web-App mit JQuery-Mobile

5.1 Am Beispiel des Taschenrechners



Ausgehend vom letzten Zwischenergebnis für den Taschenrechner, sind bei mobilen Anwendungen vor allen Dingen stilistische Faktoren von Bedeutung. Die Business-Logik (jQuery-Quellcode) bleibt vielfach unangetastet.

Dinge die zur Anpassung gehören können:

1. Mobile-APIs einbinden
2. Typografie und Bilder anpassen
3. Kopf- und Fußzeilen anpassen
4. Navigationsleisten anpassen
5. ggf. Browser-Optimierung

Wir verwenden das Verzeichnis `03web_aufg`, um es zu erweitern.



Wir müssen als aller erstes die Bibliotheken (.js und .css-Datei) von JQuery und JQuery-Mobile einbinden.

Dazu werden in der Kopfzeile des HTML-Dokuments die entsprechende Referenzen eingefügt.

Im Hinblick darauf, dass die Mobile App auch ohne Verknüpfung mit dem Internet darstellbar und funktionstüchtig sein soll, binden wir in diesem Schritt alle Bibliotheken offline ein. **Achten Sie dabei auf die Reihenfolge!**

Kontrollieren Sie ob sich das jQuery-Stylesheet in das CSS-Verzeichnis befindet und fügen Sie den Verweis im Head des HTML-Dokuments `taschenrechner.html` ein.

```
<link rel="stylesheet" href="css/jquery.mobile-1.4.2.min.css" />
```

Dann kontrollieren Sie ob sich die JQuery-Javascript-Datei im js-Verzeichnis befindet

und ob im des HTML-Dokuments `taschenrechner.html` den Verweis darauf ein.

```
<script type="text/javascript" src="js/jquery-1.11.0.min.js"></script>
```

Kontrollieren Sie dann ob sich die JQuery-Mobile-Javascript-Datei im `js`-Verzeichnis befindet und fügen Sie in den Head des HTML-Dokuments `taschenrechner.html` den Verweis ein.

```
<script type="text/javascript" src="js/jquery.-mobile-1.4.2.min.js"></script>
```

Der Verweis auf die Javascript-Datei `PhoneGap.js` wird erst im kommenden Kapitel relevant, nämlich dann wenn wir den Webservice `PhoneGap-Build` nutzen um aus unserer web App eine native App zu erzeugen. Fügen Sie den Verweis im Hinblick darauf ein.

```
<script type="text/javascript" src="PhoneGap.js" charset="utf-8"></script>
```

Fügen Sie danach den Verweis auf unser eigenes Stylesheet `styles.css` ein.

```
<link rel="stylesheet" href="css/styles.css" type="text/css" />
```

Abschließend folgt noch der übliche Verweis auf unsere Javascript-Datei `my_jquery.js`.

```
<script type="text/javascript" src="js/my_jquery.js"></script>
```

Das `Cache-Manifest` werden wir erst im kommenden Kapitel ansprechen.

Wir widmen uns nun im kommenden Schritt der für die Formatierung notwendigen Anpassungen im HTML-Dokument `taschenrechner.html` und im Stylesheet `styles.css`.

Ersetzen Sie...

```
<div class="page" id="p2" >
```

Wir beginnen mit dem Page-Container im

<p>durch... <code><div data-role="page" id="p3"></code></p>	<p>HTML-Dokument <code>taschenrechner.html</code> data-role: Generell werden in jQuery-Mobile-Framework alle Definitionen und Inhaltstypen über das <code>data-role</code> Attribut festgelegt. Dazu wird das Attribut im <code>div</code>-Container definiert. Für alle Inhaltstypen existiert jeweils ein Wert. [FRA00]</p> <ul style="list-style-type: none"> • <code>page</code> (Seiten-Box) • <code>header</code> (Kopfzeilen-Box) • <code>content</code> (Inhalts-Box) • <code>footer</code> (Fußzeilen-Box) • <code>fieldcontain</code> (Komponenten-Box) • <code>button</code> (Schaltflächen-Box) • ... <p>Einige dieser Inhaltstypen werden wir für unsere App nutzen.</p> <p>Ändern Sie die Angaben im <code>Page</code>-Container ab, wie nebenstehend angezeigt. Wir ergänzen dazu im folgenden die noch fehlenden <code>data-role</code>-Attribute an den entsprechenden Stellen.</p>
<p>Ersetzen Sie... <code><div class="header"></code></p> <p>durch... <code><div data-role="header" class="header" data-tap-toggle="false" data-theme="b"></code></p>	<p>data-role: Wir nutzen das <code>data-role</code>-Attribut mit dem Wert <code>header</code>.</p> <p>data-tap-toggle: Mit dem Wert „false“ wird erreicht, dass die Kopf- und Fußzeile bei einer Berührung oder auf Klick nicht ausgeblendet wird. [FRA00]</p> <p>data-theme: Mit dem Wert „b“ nutzen wir eines der vom jQuery-Mobile-Framework zur Verfügung gestellten Themes (Layouts). Eine dunkle Variante mit blauen Highlight-Effekten. [JQU01]</p> <p>Ändern Sie die Angaben im <code>Header</code>-Container ab, wie nebenstehend angezeigt.</p>
<p>Ersetzen Sie...</p>	<p>data-role:</p>

<pre>Dieser Text wird nicht angezeigt</pre> <p>durch...</p> <pre>Dieser Text wird nicht angezeigt</pre>	<p>Wir nutzen das data-role-Attribut mit dem Wert <code>button</code>.</p> <p>data-icon: Im jQuery-Mobile-Framework existieren eine Reihe von selbsterklärenden, vordefinierten Schaltflächen die wir im folgenden nutzen werden. Das <code>data-icon home</code> erzeugt ein kleines Haus-icon das in der Regel auf die Startseite einer Anwendung verweist. [JQU00]</p> <p>Data-iconpos: Mit dem Wert <code>"notext"</code> versteckt das PlugIn (die JQuery-Mobile-Erweiterung) den Text. Andere mögliche Werte sind u.a. <code>"right"</code>, <code>"left"</code>, <code>"top"</code>, <code>"bottom"</code>.</p> <p>data-transition: Das jQuery-Mobile-Framework beinhaltet ein CSS-basierte Auswahl an Übergangseffekten. Diese können wahlweise in Verweise und Formularelementen angewendet werden. <code>Fade</code> ist der Standardwert (default-Wert). Die Effekte können wir auf den Seiten der W3Schools http://www.w3schools.com/jquerymobile/jquerymobile_transitions.asp [W3S01]</p> <p>target: Das HTML-Tag kennen Sie bereits. In Verweisen werden mit entsprechenden Werten das Ziel sichergestellt. Der Wert <code>"_parent"</code> öffnet den Inhalt des Verweises im aktuellen Fenster.</p> <p>Ändern Sie den in der <code>header</code>-Box enthaltenen <code>Home</code>-Verweis ab, wie nebenstehend angezeigt.</p>
<p>Ersetzen Sie...</p> <pre>Dieser Text wird nicht angezeigt</pre> <p>durch...</p> <pre><a href="waehrungsrechner.html" id="forward" data-role="button" data-icon="arrow-r" data-</pre>	<p>Der einzige unterschied ist die <code>data-icon</code>-Angabe. Wir verwenden den Pfeil nach rechts.</p> <p>Data-icon: Eine weitere vordefinierte Schaltfläche ist</p>

```
iconpos="notext" data-transition="fade" target="_parent">Dieser Text wird nicht angezeigt</a>
```

der Pfeil nach rechts, das data-icon attribute

data-iconpos:
Erhält den Wert notext!

data-transition:
Erhält den Wert fade!

Ändern Sie den in der header-Box enthaltenen Verweis auf die waehrungsrechner.html ab, wie nebenstehend angezeigt.

```
[data-role=page]{
  font-family:Verdana, Arial, Helvetica,sans-serif;
  font-weight: normal;
  font-size:100,01%;
  background-image:
url("../images/blu_stripes.png");
}

[data-role=header] {
  padding-top:0.5em;
  padding-left:0.5em;
}

[data-role=header] h1{
  text-shadow: 0.5px 0.5px 0px rgba(255, 255, 255, .7);
  font-size:1.2em;
}
```

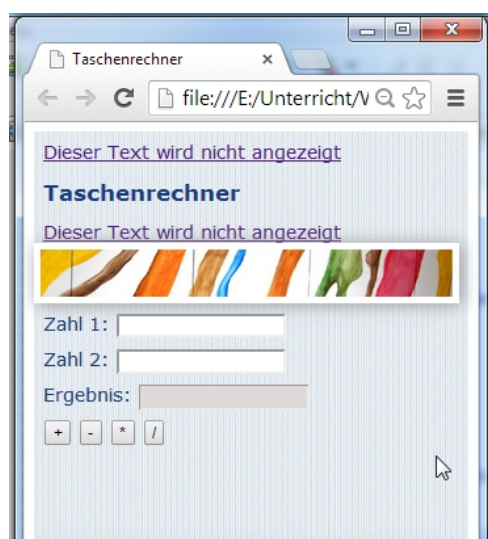
Ergänzen Sie die fehlenden Angaben für das data-role-Attribut:

1. page
2. header

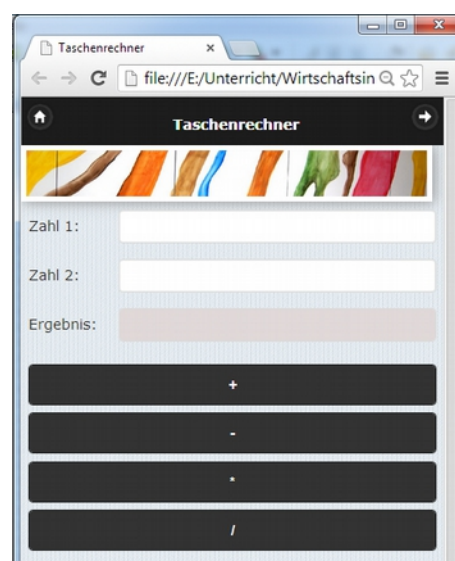
in der Stylesheet-Datei styles.css.

RGBA-Farbwert, der eine gewisse Transparenz zuweisen kann. Man registriert den Unterschied mit bloßem Auge nur, wenn man das tag abwechselnd:

entfernt - testet - einfügt - testet



Ohne jQuery-Mobile-Framework



Mit jQuery-Mobile-Framework

Allgemeiner Hinweis zum data-role-Attribute und Aptana

HTML Validation warnings for the data-role attribute

„This had bugged me for months. ... What did work for me was turning off HTML Tidy Validator in Aptana 3.2.2.“

Forumseintrag, Mirko, 24.03.2013

<http://stackoverflow.com/questions/7847596/aptana-studio-3-warnings-when-using-data-role-tags-of-html-5>

Lösung:

Preferences >> Aptana Studio >> Validation >> HTML

Beide Optionen deaktivieren HTML Tidy Validator `build` und `reconciliation`.

Ersetzen Sie...

```
<div class="content" id="taschenrecher-
inhalt">
```

durch...

```
<div data-role="content" class="content"
id="taschenrecher-inhalt">
```

Fahren Sie mit der Implementierung der `data-role-Attribute` fort.

Ersetzen Sie...

```
<div class="container">
```

durch...

```
<div data-role="fieldcontain" class="contai-
ner">
```

Alle Komponenten die zusammengehören, wie beispielsweise das Label zum zugehörigen Textfeld werden in `fieldcontain`-Boxen gepackt.

Ändern Sie alle `div`-Boxen der Klasse „`container`“ ab, wie nebenstehend angezeigt.

```
data-role="button"
```

```
data-transition="fade"
```

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

Fügen Sie für alle Schaltflächen (`buttons`) die nebenstehenden Attribute und Werte ein.

```
[data-role=content]{
    width: 95%;
}

[data-role=fieldcontain]{
    margin-top: 0.5em;
    padding: 0em 0.5em;
}
```

Ergänzen Sie die fehlenden Angaben für das `data-role-Attribut`:

1. `content`
2. `fieldcontain`

in der Stylesheet-Datei `styles.css`.

```

1
2     $(document).ready(function(){
3         //Navigationsschaltflächen
4         var mHome = $("#home");
5         var mForward = $("#forward");
6         var mBackward = $("#backward");
7         var mTaschenrechner= $("#taschenrechner");
8         var mWaehrungsrechner = $("#waehrungsrechner");
9
10        //Operationen
11        var mAddieren = $("#addition");
12        var mSubtrahieren = $("#subtraktion");
13        var mMultiplizieren = $("#multiplikation");
14        var mDividieren = $("#division");
15        var mUmrechnen = $("#umrechnen");
16
17        //Textfelder für Eingaben
18        var mZahl1 = $("#zahl1");
19        var mZahl2 = $("#zahl2");
20        var mBetrag = $("#betrag");
21        var mWaehrung1 = $("#waehrung1");
22        var mWaehrung2 = $("#waehrung2");
23
24        //Methoden: Verhaltensweisen
25        reloadFooterFor(mHome);
26        reloadFooterFor(mForward);
27        reloadFooterFor(mBackward);
28        reloadFooterFor(mTaschenrechner);
29        reloadFooterFor(mWaehrungsrechner);
30        addieren(mAddieren,mZahl1,mZahl2);
31        subtrahieren(mSubtrahieren,mZahl1,mZahl2);
32        multiplizieren(mMultiplizieren,mZahl1,mZahl2);
33        dividieren(mDividieren,mZahl1,mZahl2);
34        umrechnen(mUmrechnen, mBetrag, mWaehrung1, mWaehrung2);
35
36    });
37

```

Eine notwendige Erweiterung der Javascript-Datei, fällt erst auf wenn wir die Anwendung mit dem Firefox-Browser testen.

Die Tatsache, dass bei Aufrufen der einzelnen Seiten über die Header-, Footer- oder das Menü auf der Startseite, der iFrame erst beim `reload()` korrekt angezeigt wird, ist auf einen Bug des Browsers zurückzuführen.

Wir werden diesen Bug mit Hilfe von jQuery eliminieren.

Öffnen Sie dazu die Javascript-Datei `my_jquery.js`.

Ergänzen Sie den Quellcode an den gekennzeichneten Stellen.

Den Schaltflächen haben wir vorsorglich ja bereits eine eindeutige `id` gegeben. Anhand dieser `id` werden wir nun die den Klick durch den Benutzer erkennen und das erneute laden der iFrame-Datei forcieren.

```

68 function reloadFooterFor(pOperation){
69     var operation = pOperation;
70     operation.click(function(){
71         $('#iframe').contentWindow.location.reload(true);
72     });
73 };
74

```

Ergänzen Sie danach die Funktion für das automatische Laden des iFrames (in unserem Fall die Seite `footer.html` als Inhalt des iFrames).

Wenn künftig eine der Navigationsflächen angeklickt wird, erfolgt mit dem Seitenaufruf, das erneute laden des iFrames. Damit wird sichergestellt, dass die Fußzeile im korrekten Layout bereitgestellt wird.

5.2 Am Beispiel des Währungsrechners

Währungsrechner

Betrag:

Ergebnis:

Von: Währung wählen... v

In: Währung wählen... v

Umrechnen

Ergebnis: Währungsrechner

XX.XX

Storyboard

Aufgabe:

Orientieren Sie sich an den bereits implementierten Erweiterungen im HTML-Quellcode für den Taschenrechner und ergänzen Sie den Quellcode für den Währungsrechner selbstständig.

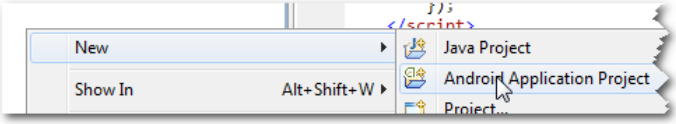
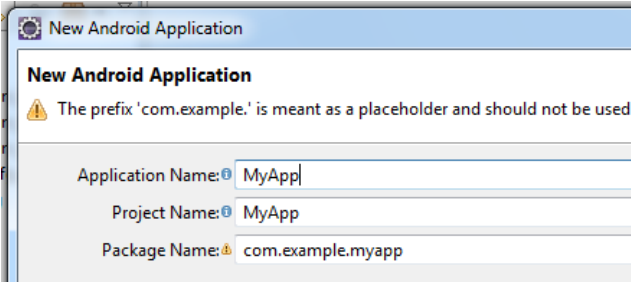
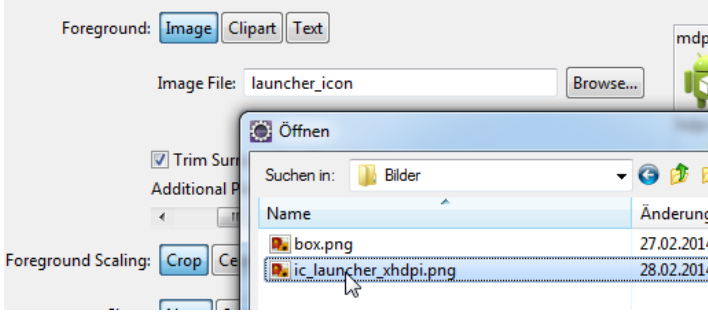
Testen und korrigieren Sie gegebenenfalls die Anwendung ausgiebig (alle Funktionen, alle Browser).

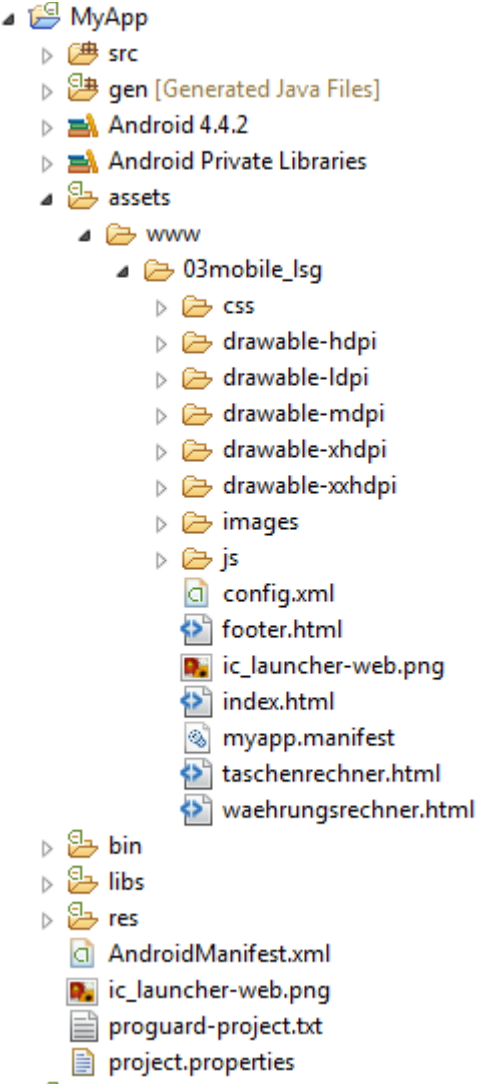
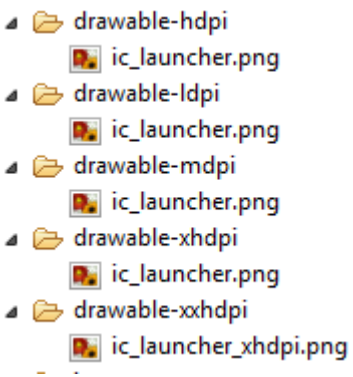
6 Ausblick

Ausblick

6.1 Anwendungsbeispiel: Android App mit PhoneGap

6.1.1 Unsere WebApp wird zu einer nativen Android App

	<p>Erzeugen Sie ein Android-Projekt.</p> <p>Wir benötigen diese nur dazu, um die für Android-Apps notwendige Projektstruktur zu verwenden.</p>
	<p>Sie können dazu den App-Name MyApp verwenden und dem Assistenten einfach folgen.</p>
<p>Configure Launcher Icon Configure the attributes of the icon set</p> 	<p>Belassen Sie die Angaben und klicken Sie 2-malig >> NEXT. Bei der Konfiguration des App-Icons können Sie die bereitgestellte Grafik <code>ic_launcher_hdpi.png</code> verwenden.</p> <p>Schließen Sie den Vorgang mit ab: NEXT >> NEXT >> Finish</p>

 <ul style="list-style-type: none"> MyApp <ul style="list-style-type: none"> src gen [Generated Java Files] Android 4.4.2 Android Private Libraries assets <ul style="list-style-type: none"> www <ul style="list-style-type: none"> 03mobile_lsg <ul style="list-style-type: none"> css drawable-hdpi drawable-ldpi drawable-mdpi drawable-xhdpi drawable-xxhdpi images js <ul style="list-style-type: none"> config.xml footer.html ic_launcher-web.png index.html myapp.manifest taschenrechner.html waehrungsrechner.html bin libs res AndroidManifest.xml ic_launcher-web.png proguard-project.txt project.properties 	<p>Erzeugen Sie im <code>assets</code>-Verzeichnis ein <code>www</code>-Verzeichnis und kopieren Sie das Ergebnis aus Kapitel 4 in dieses Verzeichnis.</p>
 <ul style="list-style-type: none"> drawable-hdpi <ul style="list-style-type: none"> ic_launcher.png drawable-ldpi <ul style="list-style-type: none"> ic_launcher.png drawable-mdpi <ul style="list-style-type: none"> ic_launcher.png drawable-xhdpi <ul style="list-style-type: none"> ic_launcher.png drawable-xxhdpi <ul style="list-style-type: none"> ic_launcher_xhdpi.png 	<p>Grafikverzeichnisse. Hier werden die Grafiken in unterschiedlicher Qualität zur Verfügung gestellt.</p> <p>Ein Überblick zu den Auflösungen finden Sie hier:</p> <p>http://developer.android.com/design/style/iconography.html</p> <p>[DEV00]</p>


```

myapp.manifest - Editor
Datei Bearbeiten Format Ansicht ?
CACHE MANIFEST

# HTML-Dateien
index.html
taschenrechner.html
waehrungsrechner.html
footer.html

# JavaScript-Dateien
js/jquery-1.11.0.mins.js
js/jquery.mobile-1.4.0.mins.js
js/my_jquery.js

#CSS-Dateien
css/styles.css
css/jquery.mobile-1.4.0.min.css

# Bilder
images/logo.png
images/blue_stripes.png
images/ajax-loader.png
ic_launcher-web.png

http://www.emotionalspirit.de/mobile

```

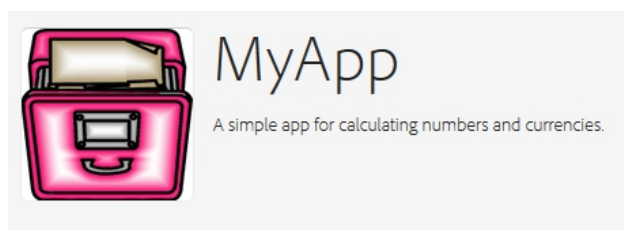
Cache-Manifest erstellen. Damit wird bei einer Offline-App sichergestellt welche Dateien gespeichert werden sollen.

```

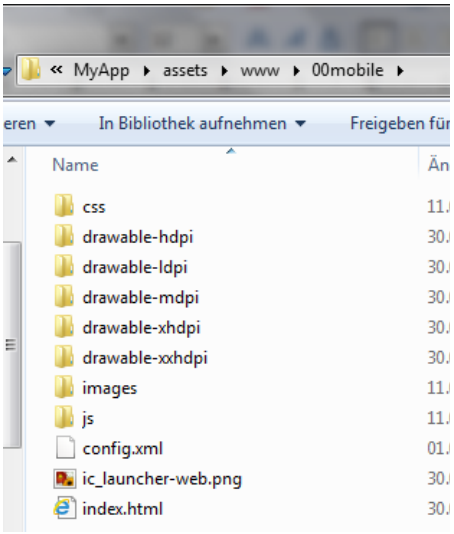
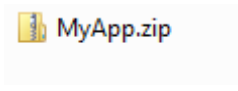
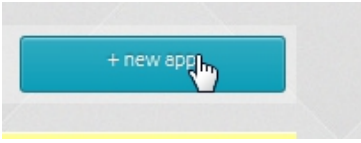
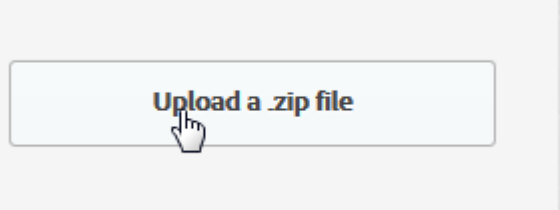
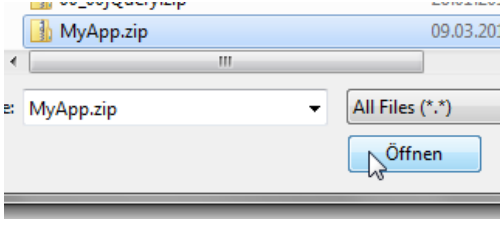
<name>MyApp</name>
<icon src="ic_launcher-web.png" />
<description>
  A simple app for calculating numbers and currencies.
</description>
<author href="http://www.emotionalspirit.de/mobile" email="cjanischek@gmx.de">
  Christine Janischek
</author>

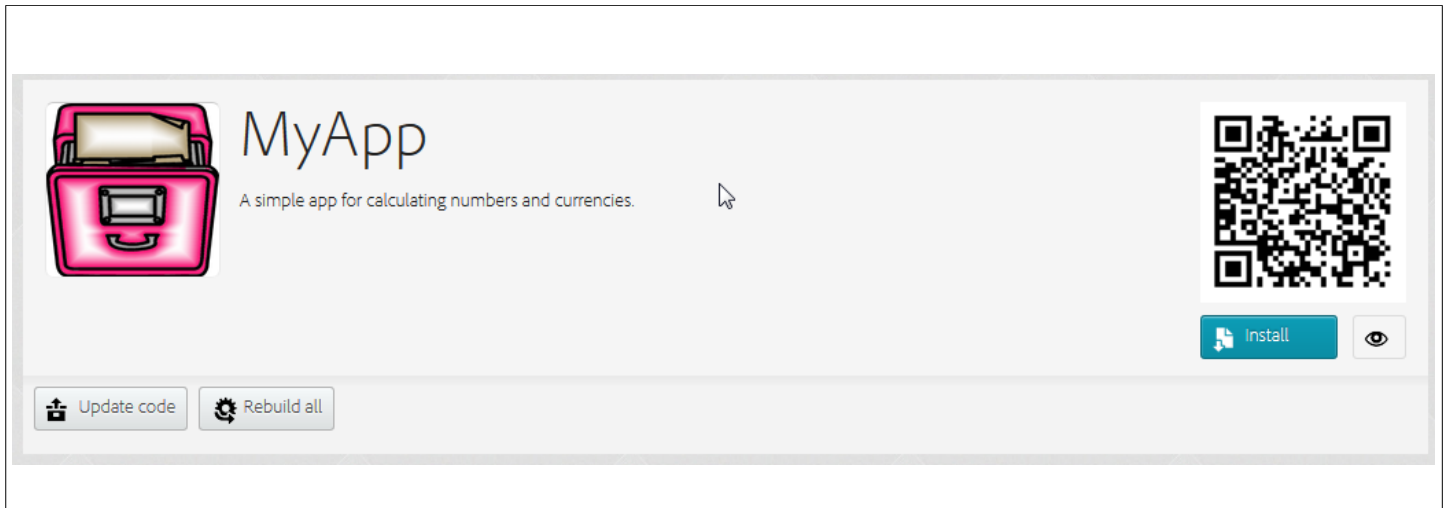
```

Die Datei `config.xml` enthält wichtige Meta-Daten der Anwendung. Ergänzen Sie den Titel (name), das App-Icon, die Kurzbeschreibung und die Angaben zum Urheber.



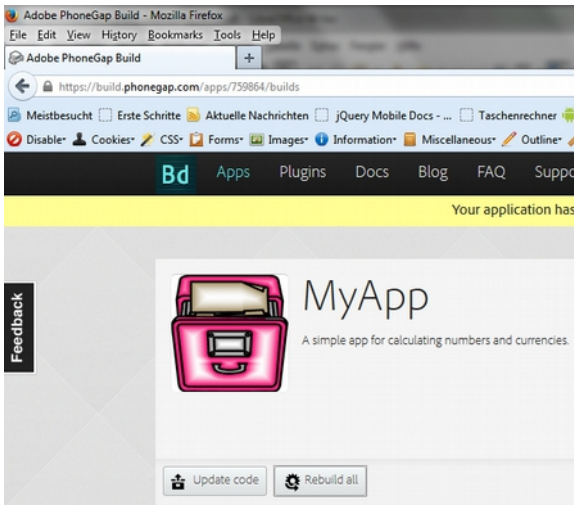
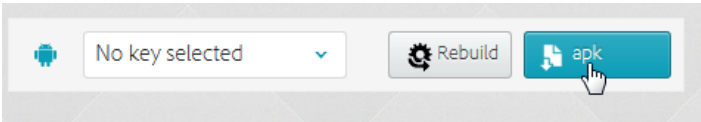
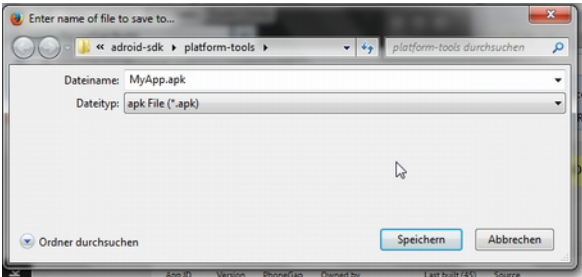
6.1.2 Mit PhoneGap-Build-Webservice eine .apk-Datei erzeugen

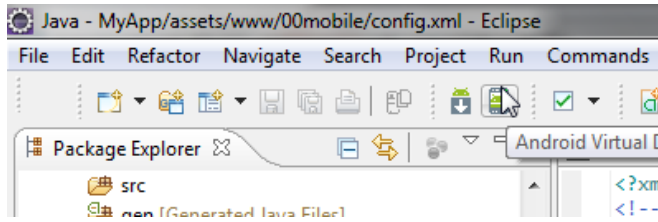
	<p>Das MyApp Verzeichnis muss dazu gezippt werden.</p> 
<p>https://build.phonegap.com/apps sign in with github...</p>	<p>Sie benötigen ein Profil bei Github oder alternativ bei Adobe PhoneGap.</p> <p>Mit dem angelegten Benutzernamen und Passwort können Sie sich dann einloggen.</p>
	<p>Die zip-Datei können Sie mit einem Klick auf die Schaltfläche APPS >> New app >> private.</p>
	<p>Nutzen Sie die Schaltfläche Upload a zip.-file >> Browse um die zip-Datei von Ihrem Rechner auszuwählen.</p>
	<p>zip-Datei auswählen und öffnen.</p>



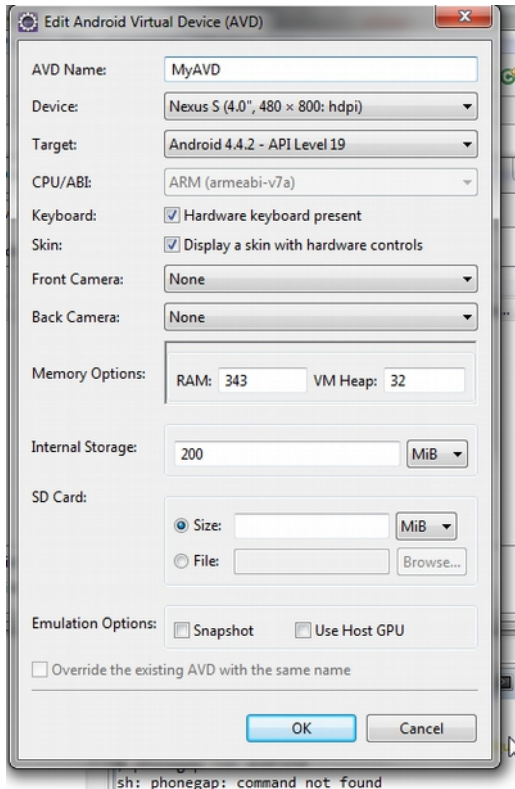
The screenshot shows a mobile application interface for "MyApp". On the left is a pink suitcase icon with a white document inside. To its right, the text "MyApp" is displayed in a large, black, sans-serif font. Below the title, a subtitle reads "A simple app for calculating numbers and currencies." A mouse cursor is positioned over the text. On the right side of the interface is a square QR code. Below the QR code are two buttons: a blue button with a white download icon and the text "Install", and a white button with a black eye icon. At the bottom of the interface are two buttons: "Update code" with a refresh icon and "Rebuild all" with a gear icon.

6.1.3 Die Anwendung auf dem virtuellen Android-Gerät (AVD) installieren

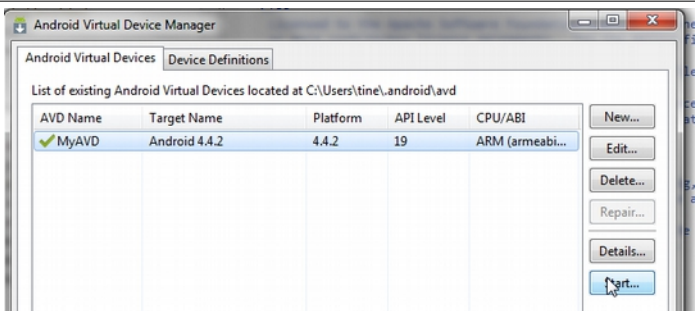
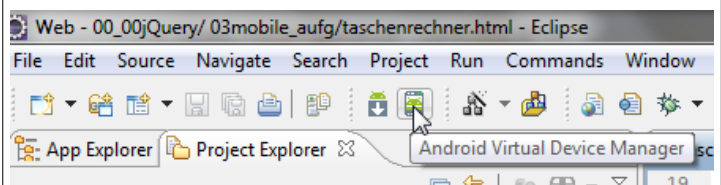
	<p>https://build.phonegap.com/apps</p> <p>Der PhoneGap Webservice</p>
	<p>Android Version mit einem Klick auf die Schaltfläche „apk“ herunterladen.</p>
	<p>Die Datei muss im Verzeichnis ihrer Eclipse Installation in das android-sdk-Verzeichnis gespeichert werden.</p> <pre>eclipse >> android-sdk >> platform-tools</pre> <p>Wechseln Sie um fortzufahren wieder in die Entwicklungsumgebung Eclipse.</p>



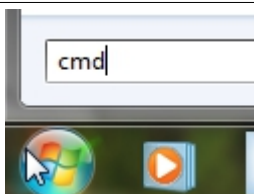
Der AVD-Manager (Android Virtual Device Manager) muss für eine Installation auf dem virtuellen Gerät gestartet sein.



Klicken Sie auf den AVD-Manager, um Ein virtuelles Gerät zu konfigurieren und mit OK zu bestätigen.



AVD starten >> Klick auf Start... >> launch



Den Installationsprozess veranlassen wir über die Konsole. Dazu die Konsole starten. Geben Sie dazu im Start-Menü den Befehl cmd ein.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\tine>I:
I:\>cd Digibag2012\Programme\eclipse\android\platform-tools
I:\Digibag2012\Programme\eclipse\android\platform-tools>adb install MyApp.apk
64 KB/s (591933 bytes in 8.908s)
 pkg: /data/local/tmp/MyApp.apk
Success

```

Wechseln Sie mit den entsprechenden Konsolen-Befehlen in das `platform-tools`-Verzeichnis.

Dazu den Laufwerksbuchstaben ihres externen Datenträgers, auf dem sich die Eclipse-Installation befindet, mit einem Doppelpunkt eingeben und mit `Enter` bestätigen.

I:

Wechseln Sie dann mit dem `cd`-Befehl und der Pfadangabe in das `platform-tools`-Verzeichnis und bestätigen sie die Eingabe wieder mit `Enter`.

```

cd
Eclipse_MySQL_Stick\Programme\eclipse\android-sdk\platform-tools

```

Geben Sie dann den Installationsbefehl an und bestätigen sie nochmals mit `Enter` Ihre Eingabe.

```
adb install MyApp.apk
```

Die Installation sollte dann erfolgreich vollzogen werden.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

I:\Digibag2012\Programme\eclipse\android-sdk\platform-tools>adb install MyApp.apk
1060 KB/s (3890558 bytes in 3.583s)
 pkg: /data/local/tmp/MyApp.apk
Success

```

```

I:\Digibag2012\Programme\eclipse\android\platform-tools>adb install MyApp.apk
87 KB/s (591933 bytes in 6.591s)
 pkg: /data/local/tmp/MyApp.apk
Success

```

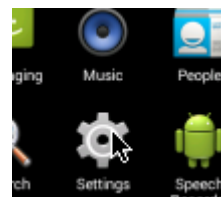
Dazu muss der Emulator gestartet sein!

Testinstallation auf der AVD des Emulators

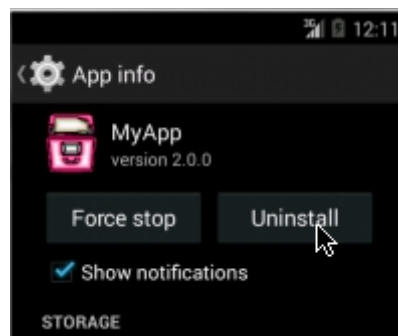
Hinweis:

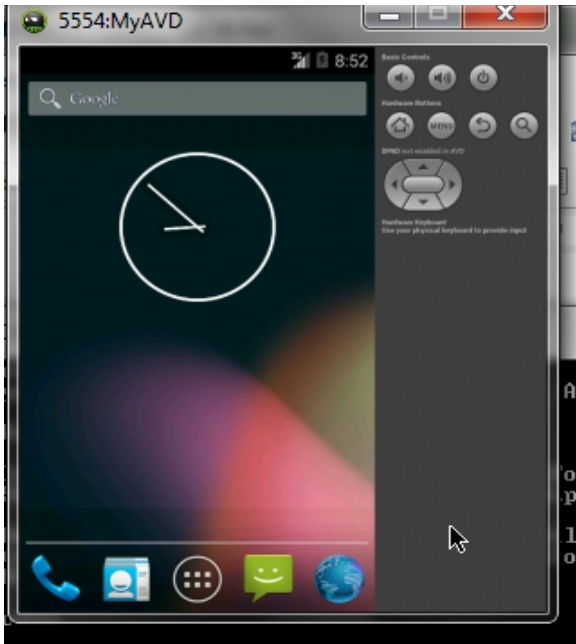
Falls Sie die Installation wiederholen möchten ist es zwingend notwendig die schon installierte Version zuvor zu deinstallieren.

Klicken Sie dazu in der AVD auf die Schaltfläche



Settings >> Apps >> MyApp >> Uninstall





Wechsel Sie dann auf das virtuelle Gerät.

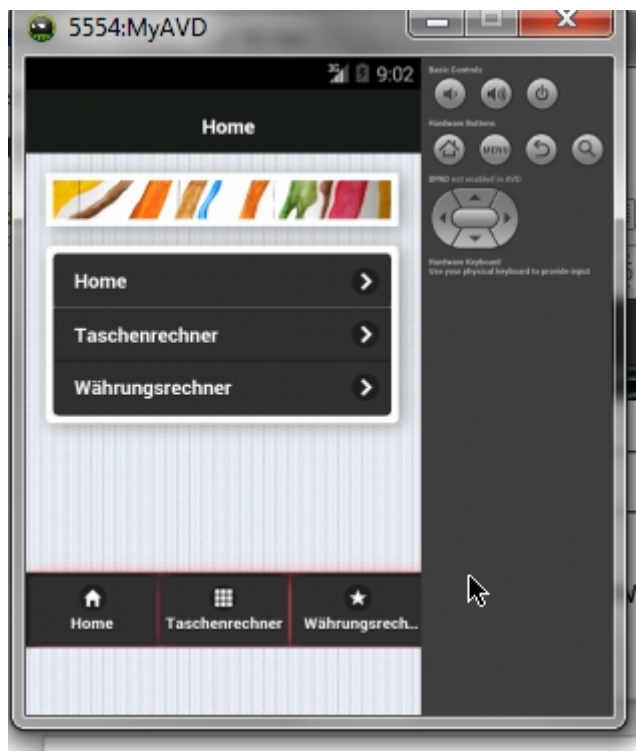
Schieben Sie das Schloss mit dem Mauszeiger nach rechts, um auf die virtuelle Benutzeroberfläche des Mobiltelefons zu gelangen.

Klicken Sie auf die Schaltfläche



Im Apps-Menü finden Sie nun das Icon unserer Anwendung MyApp.

Mit einem Doppelklick auf die Schaltfläche öffnen sich die Anwendung.



Herzlichen Glückwunsch die Anwendung ist nun einsatzfähig. Testen Sie alle Funktionalitäten. **Im Anhang finden Sie dazu mögliche Vorgangsbeschreibungen!**

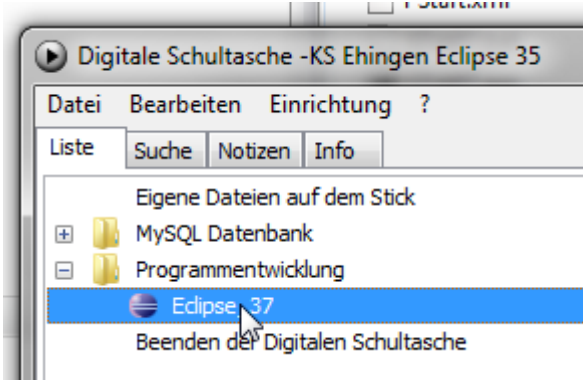
7 Einrichtung und Konfiguration in Eclipse

Einrichtung und Konfiguration in Eclipse

7.1 Allgemeines zur Konfiguration der Entwicklungsumgebung Eclipse

Die Installationen erfolgen auf Basis der Stick-Lösung 3 des LS:

<http://www.schule-bw.de/unterricht/faecher/informatik/tools/digitale-schultaschen/>



Version Eclipse Indigo, basierend auf der Eclipse Classic Version 3.7.2

Es empfiehlt sich nach dem Herunterladen ein Update der installierten Erweiterungen: Help >> Check for Updates

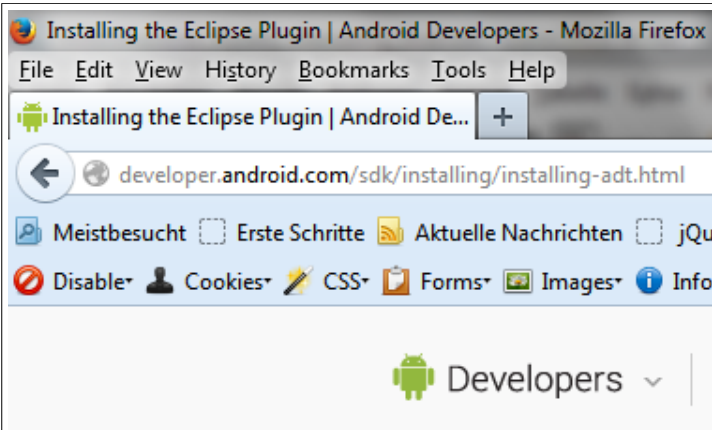
Name	Location
<input checked="" type="checkbox"/> Android PlugIn	https://dl-ssl.google.com/android/eclipse/
<input checked="" type="checkbox"/> Aptana	http://download.aptana.com/studio3/plugin/install
<input type="checkbox"/> Azzuri Clay	http://www.azzurri.co.jp/eclipse/plugins/
<input checked="" type="checkbox"/> eUML	http://www.soyatec.com/update
<input type="checkbox"/> eUML2	jar:file:/H:/Archiv/Eclipse/Eclipse37_Indigo/eUML2-free-Edition-3.7.1
<input checked="" type="checkbox"/> Graphical Editing Framework (GEF)	http://download.eclipse.org/tools/gef/update-site/releases/
<input checked="" type="checkbox"/> Indigo	http://download.eclipse.org/releases/indigo
<input checked="" type="checkbox"/> jQueryWTP	http://svn.codespot.com/a/eclipselabs.org/jsdt-jquery/updatesite
<input checked="" type="checkbox"/> Mylyn for Eclipse Indigo	http://download.eclipse.org/mylyn/releases/indigo
<input checked="" type="checkbox"/> SQLExplorer	http://eclipse-sql.sourceforge.net/
<input checked="" type="checkbox"/> The Eclipse Project Updates	http://download.eclipse.org/eclipse/updates/3.7
<input checked="" type="checkbox"/> The Eclipse Web Tools Platform (W...)	http://download.eclipse.org/webtools/repository/indigo
<input checked="" type="checkbox"/> Windows Builder	http://dl.google.com/eclipse/inst/d2wbpro/latest/3.7

Hier ein Überblick genutzter Repositories

Window >> Preferences >> Install / Update

Hinweis:

Falls Sie eine manuelle Installation durchführen möchten beginnen Sie mit dem WTP-PlugIn (Web Tools Plattform), da einige der sonst aufgeführten Erweiterungen auf Teile dieses PlugIns zugreifen.



Die ADT setzt einige Erweiterungen voraus u.a. die Eclipse Webtools (WTP / WST):
<http://download.eclipse.org/releases/indigo/>

Erweitert durch die Android Developer Tools (ADT):

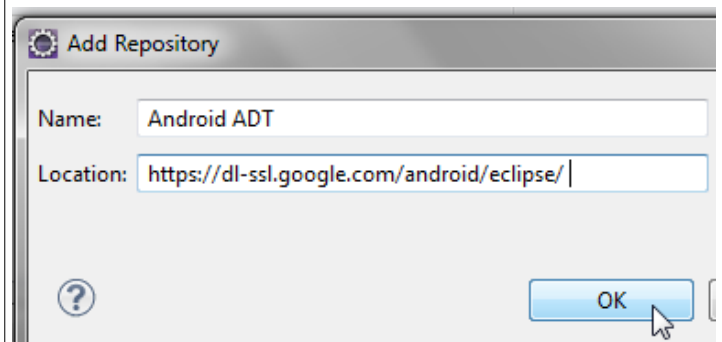
Help >> Install Software >> Add

Repositor: Android-ADT-PlugIn

<https://dl-ssl.google.com/android/eclipse/>

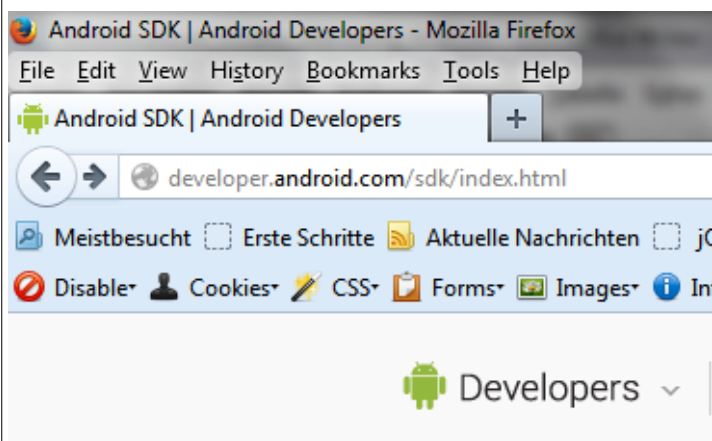
Alternativ:

<http://dl-ssl.google.com/android/eclipse/>



Im Netzwerk mit Proxy sollten in Eclipse die Netzwerkeinstellungen erweitert werden:

Window >> Preferences >> General >> Network Connections



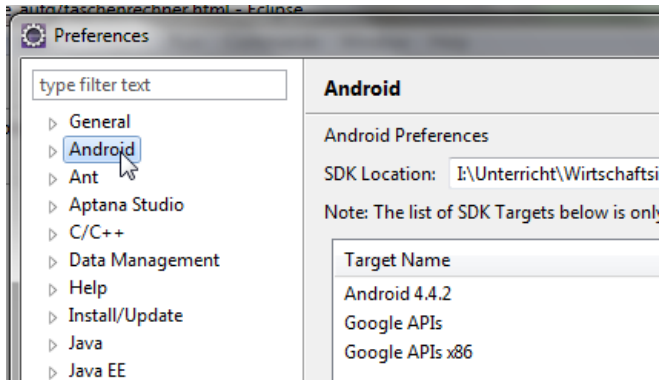
Erweitert durch den Software Development Kit (SDK).

Nur die SDK:

http://dl.google.com/android/installer_r22.3-windows.exe

Für das ganze Bundle:

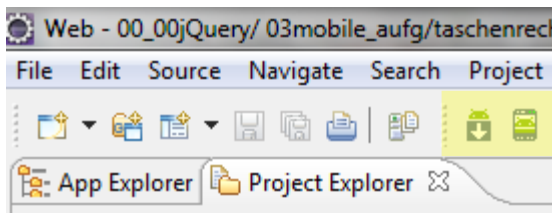
<http://developer.android.com/sdk/index.html#windows-bundle>



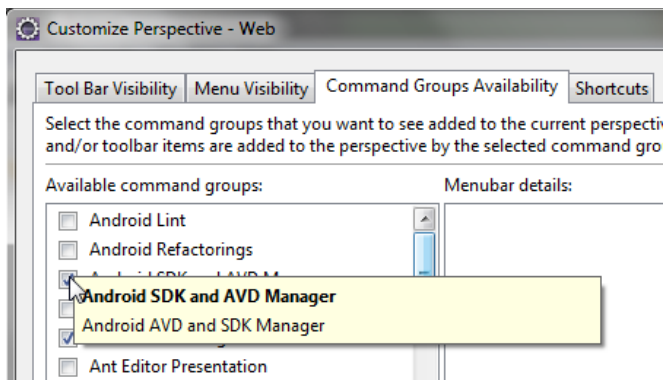
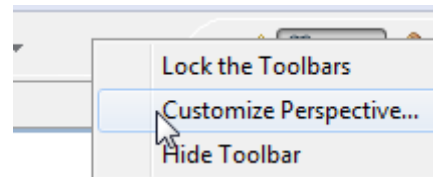
Installieren Sie die android-sdk am besten direkt in ihr Eclipse-Programmverzeichnis.

Definieren Sie den Pfad in den Einstellungen. Wählen Sie dazu in der Menü-Leiste die Option >> Window >> Preferences >> Android >> SDK Location

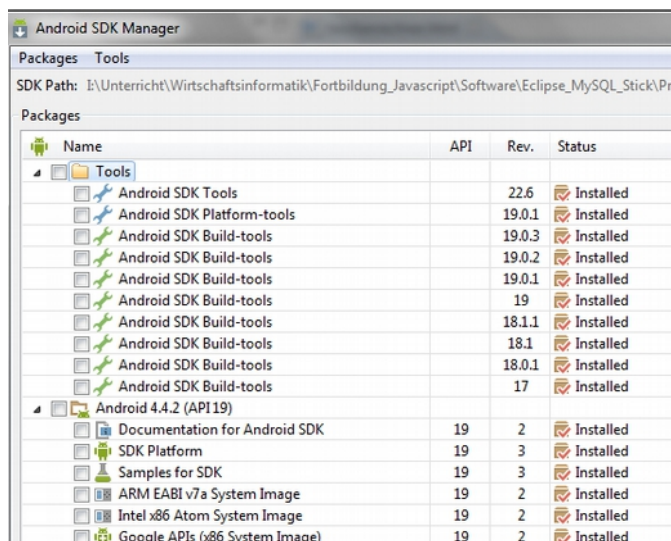
Geben Sie über die Schaltfläche Browse den Pfad zum android-sdk-Verzeichnis an.



Falls nach dem Update und einem Neustart die Icons für den Android SDK-Manager und AVD-Manager nicht in der Symbolleiste in Eclipse ersichtlich sind, klicken Sie rechts neben der Symbolleiste auf die freie Fläche und wählen die Option im Kontext-Menü >> Customize Perspective



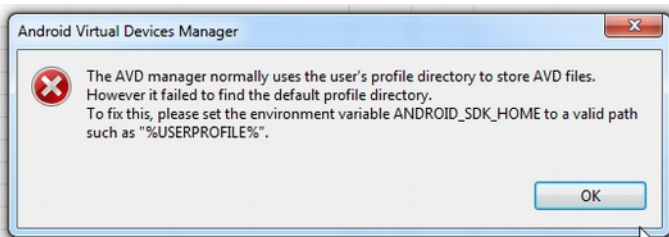
Stellen Sie sicher, dass auf dem Reiter Command Groups Availability die Option Android SDK and AVD Manager aktiviert ist.



Starten Sie die Android SDK und installieren Sie die Tools und Android API (aktuell 19). Vorsicht das kann dauern, man kann den Installationsvorgang auch im Hintergrund ablaufen lassen.

Im Feld Status können Sie nachsehen welche Pakete installiert wurden.

Für den Unterricht empfehle ich den Schülern die Digitale Tasche mit Eclipse und der aktualisierten SDK ins Tauschverzeichnis zu spielen. Das kopieren dauert im Netzwerk allerdings auch seine Zeit!



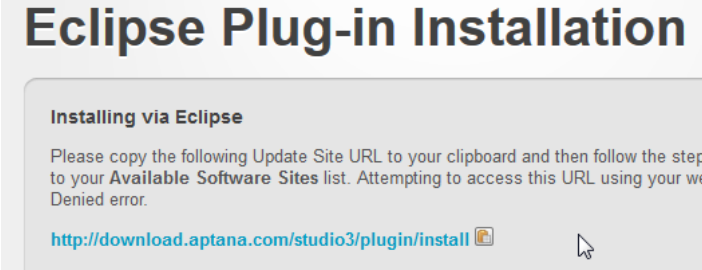
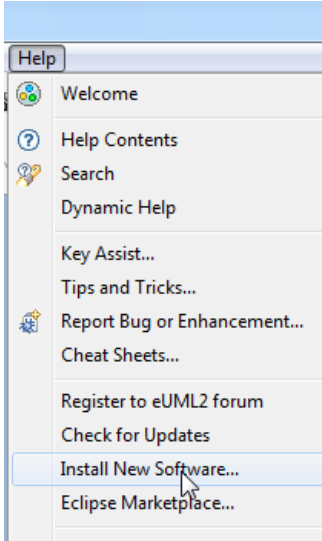
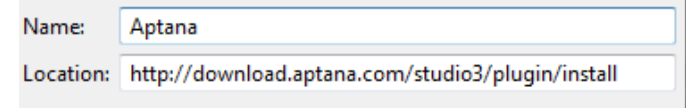
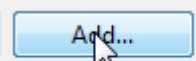


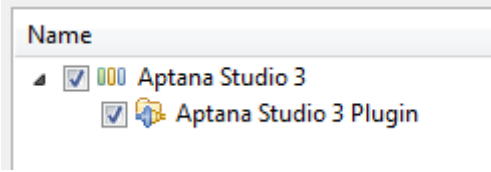
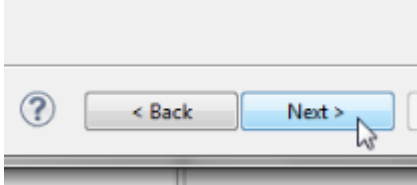
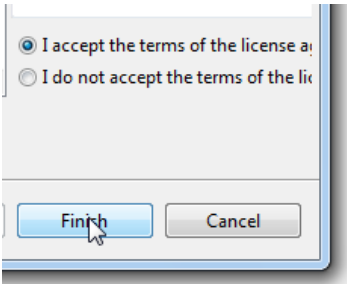
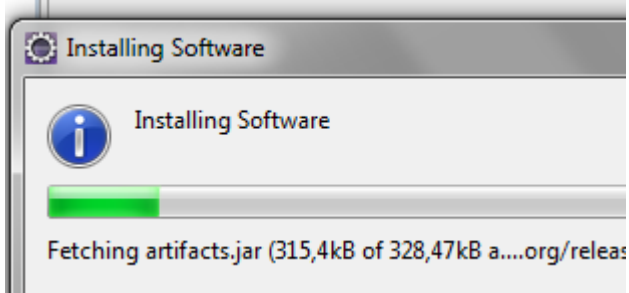
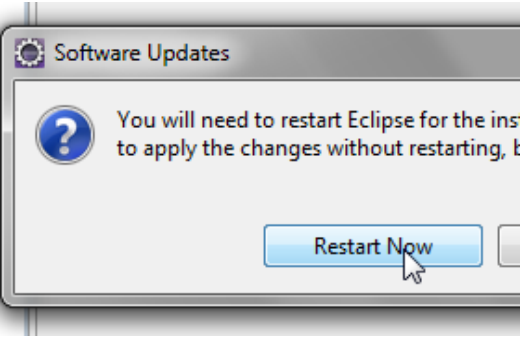
Für den Fall, dass beim öffnen des Android SDK Manager der nebenstehende Fehler auftaucht müssen Sie den Administrator des Systems fragen ob er sicherstellen kann, dass es möglich ist, für die Profile im Netzwerk eine solche Variable zu setzen.

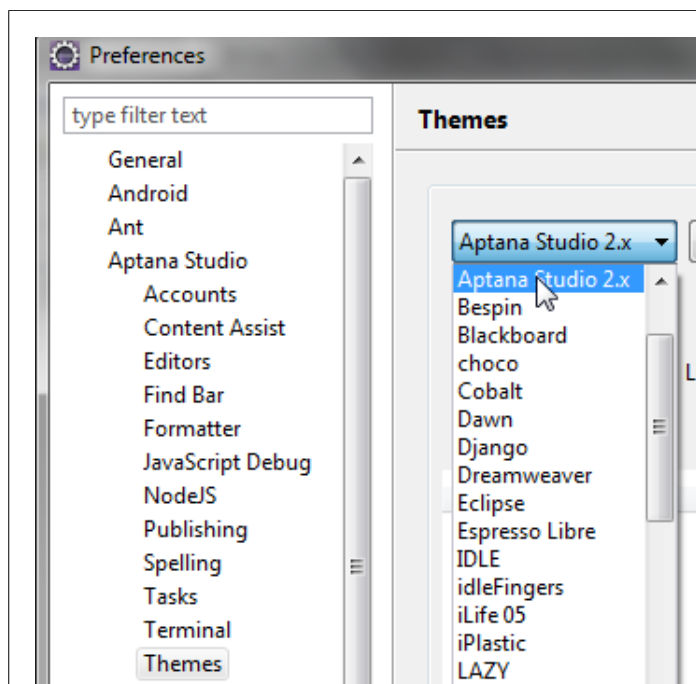
Hilfe aus dem Netz:

http://wiki.answers.com/Q/How_do_you_fix_this_The_AVD_manager_normally_uses_the_user's_profile_directory_to_store_AVD_files_However_it_failed_to_find_the_default_profile_directory#slide=5&article=How_do_you_fix_this_The_AVD_manager_normally_uses_the_user's_profile_directory_to_store_AVD_files_However_it_failed_to_find_the_default_profile_directory

7.2 Syntax-Hilfe mit dem Eclipse-PlugIn-Aptana

	<p>http://www.aptana.com/products/studio3/download [APT00]</p> <p>Wählen Sie die Option >> Eclipse PlugIn Version. Sie können ohne weitere Angaben fortschreiten</p> 
	<p>Das Repository: http://download.aptana.com/studio3/plugin/install</p> <p>Das ist die aktuelle Quelle für das PlugIn. Laut Herstellerangaben besteht eine Kompatibilität für die Eclipse Versionen 3.6 oder besser.</p> <p>Kopieren Sie die o. g. Internet-Adresse.</p>
	<p>Öffnen Sie die Entwicklungsumgebung Eclipse und wählen Sie in der Menü-Leiste die Option: Help >> Install New Software</p>
	<p>Klicken Sie auf die Schaltfläche  "Sites" preferences. und geben Sie Name und Location an.</p>

	<p>Kontrollieren Sie die Auswahl und wählen Sie die Schaltfläche >> Next, um im Installationsprozess fortzuschreiten.</p> 
	<p>Wählen Sie im Fenster Review Licences die Option: I accept the terms of the licence...</p> <p>Installieren sie das plugIn mit einem Klick auf die Schaltfläche >> Finisch.</p> 
	<p>Starten Sie ihre Entwicklungsumgebung erneut.</p>

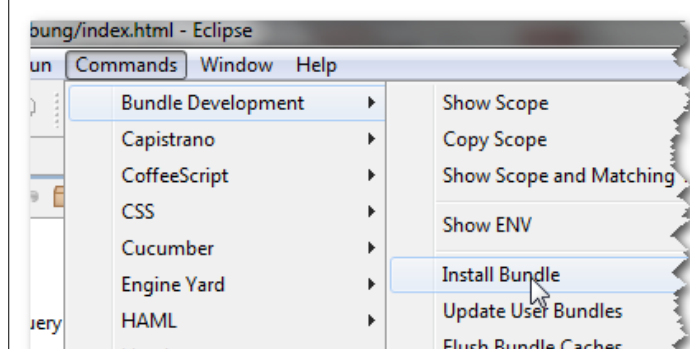


Aptana verwendet Standardmäßig die Editor-Einstellung: Weiße Schrift auf Schwarzem Hintergrund. Das stört!

Ändern Sie diese Einstellung: Windows >> Preferences >> Aptana Studio >> Themes

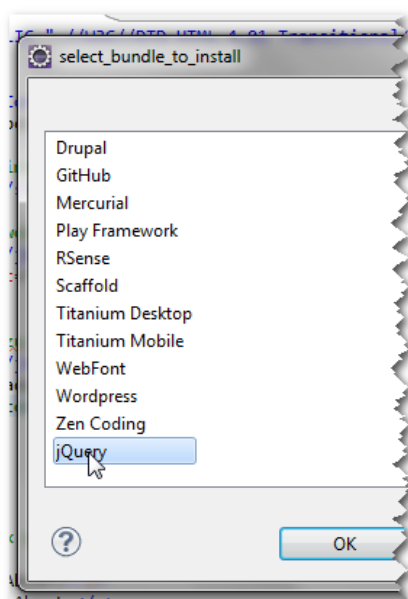
Theme "Aptana Studio 2.x" einstellen und mit einem Klick auf die Schaltfläche „OK“ die Veränderung bestätigen.

Fenster schließen!



Syntax-Hilfe einbinden:

Menü >> Commands >> Bundle Development >> Install Bundle



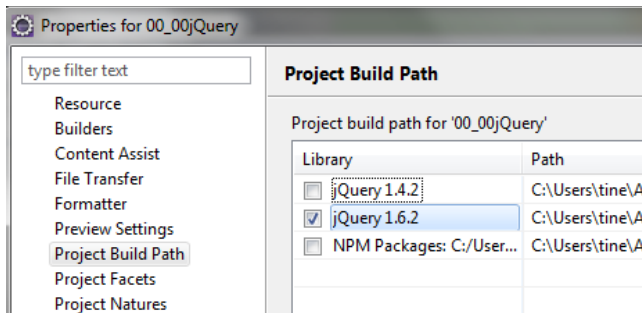
jQuery auswählen und OK klicken.

```

tine@MYTHINK ~/Aptana Rubles
$ git clone git://github.com/aptana/javascript-jquery.ruble.git
Cloning into javascript-jquery.ruble...
remote: Reusing existing pack: 137, done.
remote: Total 137 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (137/137), 322.06 KiB | 206 KiB/s, done.
Resolving deltas: 100% (62/62), done.

```

Auf der Konsole werden folgende Befehle ausgeführt.



Um die Syntaxhilfe für ein Projekt zu aktivieren, sollten Sie später auf das Projekt klicken und im Kontext-Menü die Option >> Properties >> Project Build Path wählen. Im rechten Fenster können Sie dann die verfügbare „Library“ mit einem Klick in der Checkbox aktivieren.

```

<script type="text/javascript">
$(document).ready(function(){
$("#box p:contains('Zweiter Absatz')").ad;
});
</script>
</head>

```

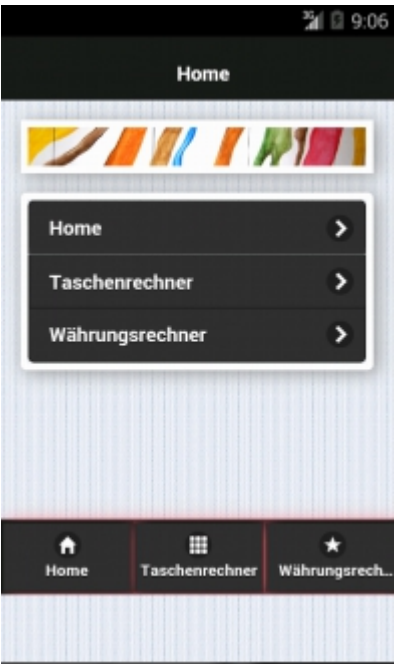
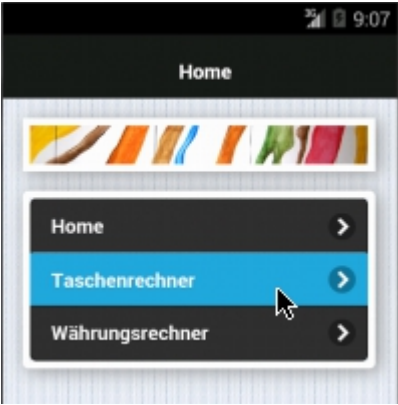
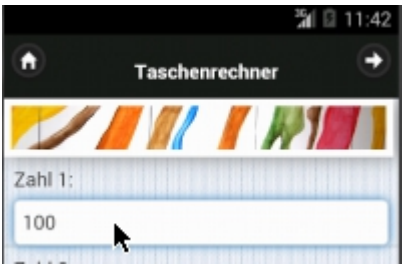
The context menu is open, showing options: 'add' (highlighted), 'addClass', and 'addClass'.

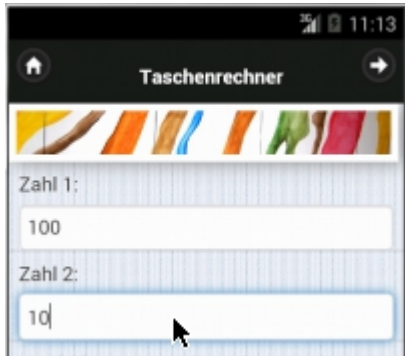
Spätestens nach einem Neustart kann man die Syntax-Hilfe mit der Taste STRG + LEERTASTE bedienen.

8 Anhang

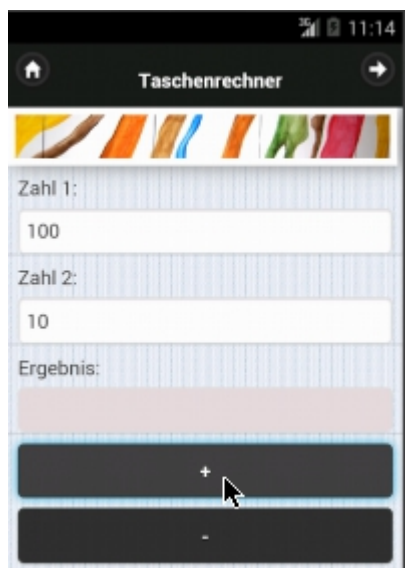
Anhang

8.1 Vorgangsbeschreibung am Beispiel des Taschenrechners

 <p>The screenshot shows the Home screen of an Android device. At the top, the status bar displays the time as 9:06. Below the status bar, the word 'Home' is centered. A colorful, abstract graphic is visible. A dark grey app drawer is open, showing three items: 'Home', 'Taschenrechner', and 'Währungsrechner'. The 'Taschenrechner' item is highlighted with a blue background. At the bottom, the dock contains three icons: a home icon, an app drawer icon, and a star icon.</p>	<p>App öffnen...</p>
 <p>This screenshot is similar to the previous one, but the 'Taschenrechner' item in the app drawer is now highlighted with a blue background, and a mouse cursor is pointing at it. The time in the status bar is now 9:07.</p>	<p>Die Schaltfläche >>Taschenrechner<< anklicken...</p>
 <p>The screenshot shows the 'Taschenrechner' app interface. The title bar at the top says 'Taschenrechner'. Below the title bar, there is a colorful graphic. The main area contains a text input field labeled 'Zahl 1:' with the number '100' entered. A mouse cursor is pointing at the input field. The time in the status bar is 11:42.</p>	<p>Zahl1 eingeben...</p>



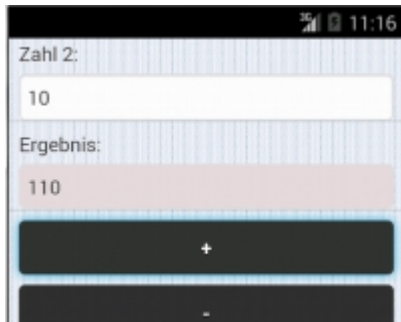
Zahl2 eingeben...



Die Schaltfläche für die gewünschte Rechenoperation anklicken...



Ausgabe im Dialogfenster mit OK bestätigen...

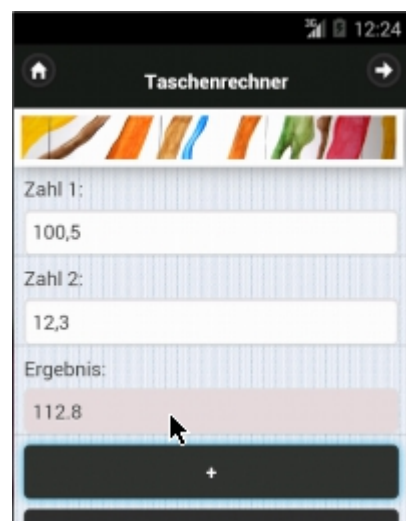


Ergebnis wird ins das Ergebnisfeld übernommen und angezeigt...

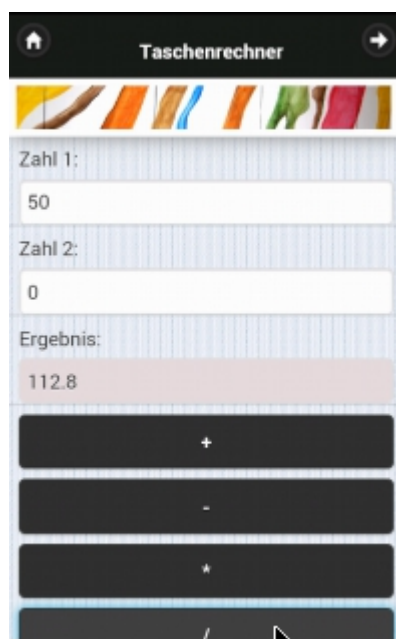


Eingabe

1. Grenzfall:
Eingabe von Dezimalzahlen



Erwartetes Ergebnis



Eingabe

2. Grenzfall:
Division durch Null



Erwartetes Ergebnis



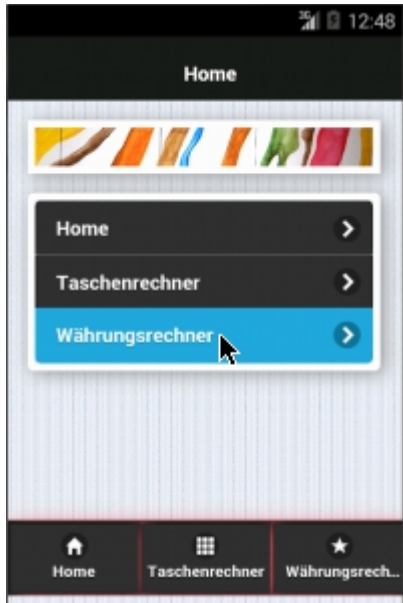
Eingabe

3. Grenzfall: Eingabe nicht numerischer Werte

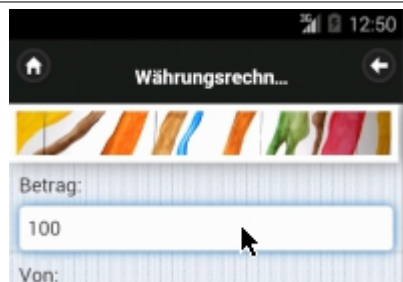


Erwartetes Ergebnis

8.2 Vorgangsbeschreibung am Beispiel des Währungsrechners



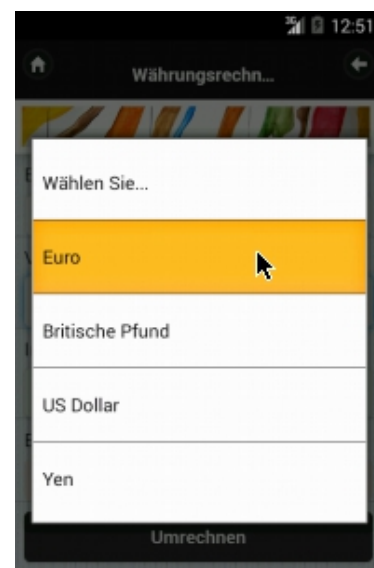
App öffnen und auf die Schaltfläche >> Währungsrechner klicken.

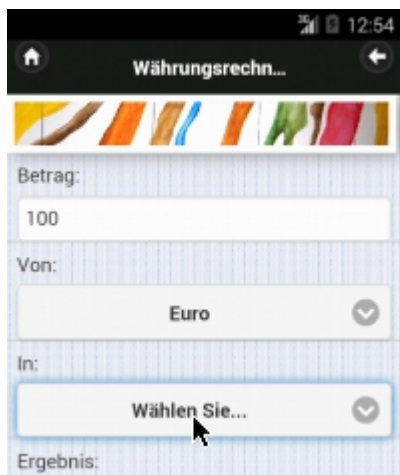


Betrag eingeben...

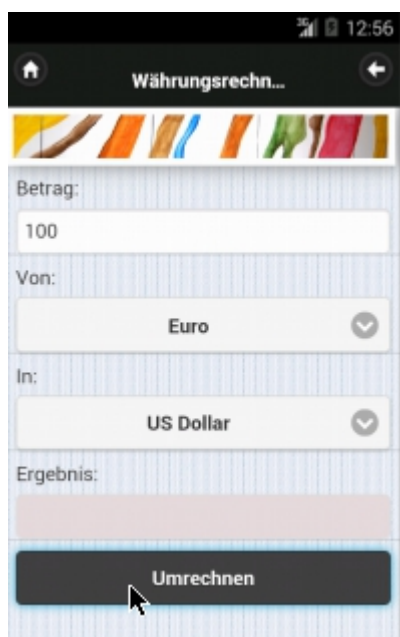
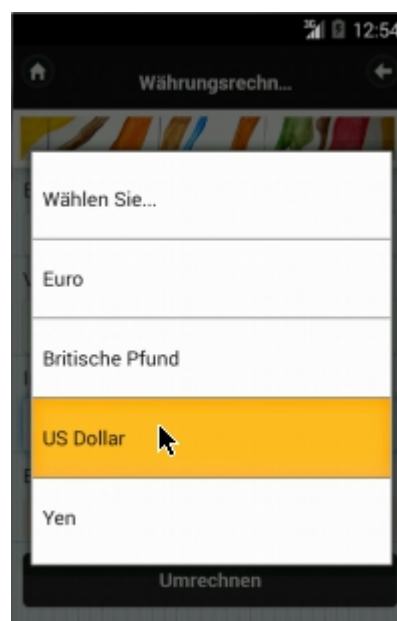


Ausgangswährung auswählen...

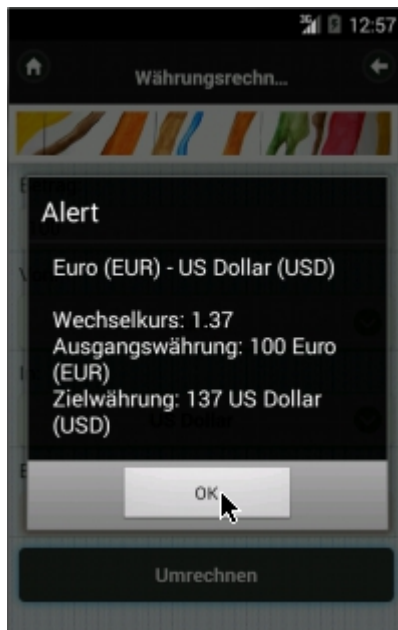




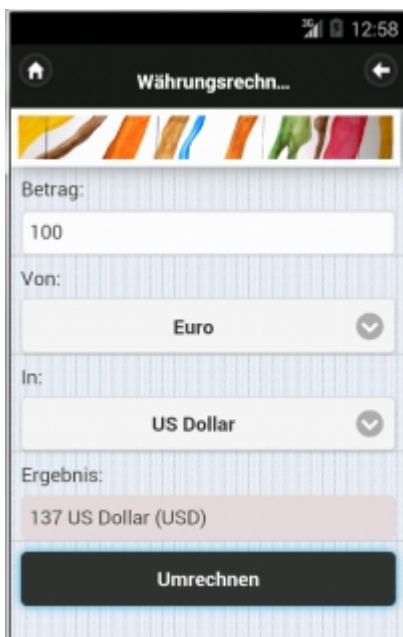
Zielwährung auswählen...



Schaltfläche >>Umrechnen<< anklicken...



Anzeige der Ausgabe im Dialogfenster und die Ausgabe mit OK bestätigen...



Übernahme des Ergebniswertes und anzeige des Ergebnis im Ergebnisfeld ...

Hinweis:
Formulieren Sie auch hier Testfälle, um die Grenzfälle zu testen!

9 Quellen

Quellen

Literaturverzeichnis

- VOL00: Vollendorf, Maximilian, Bongers, Frank, jQuery das Praxisbuch, 2013
FRA00: Franke, Florian, Ippen, Johannes, Apps mit HTML5 und CSS3, 2013
WEN00: Wenz, Christian, Javascript, 2014
SUB00: Seb Jachec, Blu Stripes, 2012, <http://subtlepatterns.com>
OAN00: OANDA Corporation, Currency Converter, 1996-2014, <http://www.oanda.com/currency/converter/>
WEB01: Wilhelm Jansen, Mathematik - Werte runden, 2003-2014, <http://www.web-toolbox.net/webtoolbox/>
JQU01: The jQuery Foundation Developers, Themes framework, 2014, demos.jquerymobile.com/1.2.0/docs/api/
JQU00: The jQuery Foundation Developers, List of Icons, 2014, <http://api.jquerymobile.com/icons/>
W3S01: W3Schools Developers, jQuery Mobile Transitions, 1999-2014, <http://www.w3schools.com/jquerymobile>
DEV00: Android Developers, Iconography, , <http://developer.android.com/design/>
APT00: Appcelerator, Inc, Download - Aptana Studio 3.4.2, 2005, <http://www.aptana.com/products/studio3/>