

13 Frameworks

In diesem Kapitel erfahren Sie

- ✓ was Frameworks sind
- ✓ wie Sie das jQuery-Framework einsetzen

Voraussetzungen

- ✓ Gute JavaScript-Kenntnisse

13.1 Was sind Frameworks?

RIAs mit ihren reichhaltigen Möglichkeiten verändern die Art der Nutzung des WWW. Ebenso wird die Bedeutung von klassischen Desktop-Applikationen neu positioniert. Viele früher nur als Desktop-Applikation genutzte Programmtypen finden sich plötzlich im Web und werden mit dem Browser ausgeführt. Seien es persönliche Kalender, vollständige Office-Programme, Spiele, Routenplaner, ganz integrierte Entwicklungsumgebungen oder Kommunikationsprogramme. Auch mobile Webseiten oder Anwendungen für Handys oder Smartphones basieren mehr und mehr auf Web-Technologie. Dies verändert nicht zuletzt das Anwenderverhalten wie auch die Anwendererwartung bei Internet-Applikationen im Allgemeinen sowie der Verfügbarkeit von Leistungen. RIAs stehen auf der einen Seite als klassische Web-Applikationen (aber mit einem gewissen Mehrwert) immer zur Verfügung, wenn man einen halbwegs schnellen Internetzugang und einen modernen Browser hat. Andererseits sind sie teilweise von der Bedienung, der Performance wie auch Optik mittlerweile fast gar nicht mehr von klassischen Desktop- oder Mobil-Applikationen zu unterscheiden.

Frameworks und Toolkits

Die Erstellung von RIAs ist kaum ohne den Einsatz von Web-Frameworks oder Toolkits zu bewerkstelligen, zumindest nicht mit einem adäquaten Aufwand. Bedenken Sie, dass Sie sich beim Einsatz eines Frameworks oder Toolkits in Abhängigkeit von einem Hersteller respektive einem Projekt begeben. Ebenso erfordert der Einsatz von Frameworks/Toolkits die Einarbeitung in die jeweiligen Funktionsbibliotheken und Arbeitsweisen dieses Systems.

Was genau ein Framework ist und wie es sich von einem Toolkit unterscheidet, ist nicht standardisiert. Sowohl eine verbindliche Definition als auch eine Abgrenzung ist nicht einfach. Allgemein versteht man unter einem Framework ein Programmiergerüst, das bestimmte Funktionalitäten bereitstellt. Ein Framework ist selbst kein fertiges Programm, sondern stellt den Rahmen (Frame) zur Verfügung, innerhalb dessen ein oder mehrere Programmierer eine Anwendung erstellen. Ein Framework beinhaltet in der Regel eine Bibliothek mit nützlichen vorgegebenen Codestrukturen (meist auf Basis einer spezifischen Sprache wie JavaScript), legt aber – im Gegensatz zu einer reinen Bibliothek – auch eine Steuerung von Verhaltensweisen bei der Verwendung fest (z. B. eine Syntax bzw. Grammatik). Frameworks können auf verschiedene Weise arbeiten. Beliebte sind JavaScript-Frameworks, die im Browser dem Programmierer Unterstützung bei JavaScript-Aktionen liefern.

Bei einem Toolkit steht dagegen mehr die Sammlung an Programmen (Tools) im Fokus, die aber durchaus auch auf spezifischen Bibliotheken und einem Syntaxkonzept aufsetzen können. Sowohl ein Framework, aber vor allen Dingen Toolkits stellen vielfach sogenannte **Widgets** beziehungsweise Komponenten zur Verfügung. Dabei handelt es sich in der Regel um Elemente, aus denen eine grafische Benutzerschnittstelle (UI – User Interface oder GUI – Graphical User Interface) zusammengesetzt wird.

13.2 Der Einsatz von Frameworks anhand von jQuery

jQuery ist ein beliebtes JavaScript-Framework. Das Kern-Framework selbst stellt komfortable Funktionalitäten zur Verfügung. Zum Beispiel:

- ✓ sichere und einfache Techniken zur Manipulation des DOM,
- ✓ erweiterte Ajax-Funktionalität,
- ✓ Effekte und Animationen,
- ✓ universelle CSS-Unterstützung und
- ✓ eine erweiterte Ereignisbehandlung.

Es gibt auch eine einfache Funktionserweiterung durch frei verfügbare Plug-ins, beispielsweise jQuery UI. jQuery ist die Basis, die beim Quellcode und der Programmierung unterstützt, während jQuery UI darauf als eigenständiges Framework aufbaut und optisch ausgereifte Oberflächenkomponenten und ein CSS-Themeframework zur Verfügung stellt. Das Mobile Framework, das wie jQuery UI auf nativem jQuery unmittelbar aufsetzt, stellt Unterstützung für mobile Webseiten bereit.

Herunterladen können Sie das kostenlose Framework unter <http://www.jquery.com>. Dieses besteht aus einer JavaScript-Datei, die Sie in einem Verzeichnis ablegen und über das Skript-Tag im Kopfbereich Ihrer Webseite einbinden:



```
<script type="text/javascript" src="jquery-2.1.1.min.js"></script>
```

In diesem Beispiel handelt es sich um die Version 2.1.1. Die Angabe von *min* im Dateinamen weist darauf hin, dass dies eine minimierte Version von jQuery ist, in der alle überflüssigen Zeichen entfernt worden sind. Die Datei ist daher sehr kompakt und auf die schnelle Übertragung optimiert, aber in einem Texteditor nur schwer zu lesen.

Wenn Sie mit jQuery arbeiten, können Sie im JavaScript die nachfolgende Funktion verwenden.

```
$(document).ready(function() {
});
```

Mit diesen Zeilen geben Sie an, dass das darin auszuführende JavaScript erst gestartet werden soll, wenn das komplette HTML-Dokument geladen wurde und der DOM-Baum fertig ist. Von Hand wurde das im Buch immer mit `window.onload=function() { ... } erledigt`, das Framework kümmert sich jedoch im Hintergrund noch um diverse Korrekturen und Anpassungen.

Selektoren

Die grundlegende Schreibweise in jQuery sieht folgendermaßen aus:

```
$(Selektor).Methode
```

Das `$`-Zeichen kennzeichnet den jQuery-Namensraum (engl.: namespace) zum Ansprechen der Elemente. Über den Selektor geben Sie das Element an, das angesprochen werden soll. Mit der Methode legen Sie fest, was mit dem Element passieren soll.

Die Selektoren in jQuery entsprechen im Wesentlichen denen von CSS3. Sie können dort beispielsweise den Namen eines Elements, die ID oder den Klassennamen angeben. Sie können aber auch Elemente über die von CSS bekannten Selektoren ansprechen (Nachfahre-Selektor, Kind-Selektor, Attribut-Selektor).

Selektion	Schreibweise
Ein HTML-Element	<code>\$('div')</code>
Mehrere HTML-Elemente	<code>\$('div, h1, p')</code>

Selektion	Schreibweise
Ein mit <code>id</code> definiertes HTML-Element	<code>\$('#id')</code>
Ein HTML-Element mit bestimmtem CSS-Klassenamen	<code>\$('div.Klassenname')</code>
Nachfahre eines Elements	<code>\$('div a')</code>
Kind des Elements	<code>\$('h1 > div')</code>
Attribut eines Elements	<code>\$('[align]')</code>
Attributswert	<code>\$('[align=right]')</code>
HTML-Element und Attribut	<code>\$('p[align=right]')</code>

Ereignisbehandlung

Nach der Selektion eines Elements in einer Webseite können Sie mit jQuery angeben, auf welches Ereignis reagiert werden soll. Dazu gibt es diverse Methoden, die an den klassischen Eventhandlern orientiert sind.

```
$(Selektor).click( function(){
});
```

Beispiel: kap13/jquery-start.html

In diesem Beispiel sollen die Absätze eines HTML-Dokuments über drei verschiedene Schaltflächen angesprochen werden. Das Aussehen der Absätze soll entsprechend per CSS verändert werden.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Mein erstes jQuery Dokument</title>
  ① <script type="text/javascript" src="jquery/jquery-2.1.1.min.js"></script>
</head>
<body>
  <h3>Ansprechen der HTML-Elemente</h3>
  ② <p>Absatz ohne Attribut und ID: Fortune plango vulnera stilantibus ...</p>
    <p align="right">rechtsbündiger Absatz: Fortune plango vulnera ...</p>
    <p id="absatz3">Absatz mit der ID "absatz3": Fortune plango vulnera ...</p>
    <div id="buttons">
      ③ <button id="btn1">Alle Absätze grün einfärben</button>
        <button id="btn2">nur rechtsbündige Absätze rot einfärben</button>
        <button id="btn3">Absatz mit ID rechtsbündig ausrichten</button>
    </div>
    <script type="text/javascript">
      ④ $(document).ready(function() {
        ⑤ $('#btn1').click(function(){
          ⑥ $('p').css('background-color', '#47b300' );
        });
        ⑦ $('#btn2').click(function(){
          $('p[align=right]').css('background-color', '#b30047' );
        });
        ⑧ $('#btn3').click(function(){
          $('#absatz3').css('text-align', 'right' );
        });
      });
    </script>
  </body>
</html>

```

- ① Als Erstes binden Sie die jQuery-Bibliothek in die Webseite ein. In diesem Fall befindet sich die Bibliotheksversion 2.1.1 im Unterverzeichnis *jquery*.
- ② Im nächsten Schritt geben Sie drei Absätze an, wobei der zweite rechts ausgerichtet wird und der dritte Absatz die ID `absatz3` erhält.
- ③ Über die drei Schaltflächen, denen jeweils eine andere ID zugewiesen wird, sollen die drei Absätze verändert werden. Das Ereignis `onclick` wird hier nicht angegeben.
- ④ Mit `$(document).ready(function() { });` leiten Sie die Funktion zum Ansprechen der jQuery-Funktionen ein. Diese sollen überwacht und ausgeführt werden, sobald das Dokument komplett geladen wurde.
- ⑤ In dieser Zeile wird das HTML-Element mit der ID `btn1` selektiert. Mit der Methode `click()` leiten Sie die Methode ein, die aufgerufen werden soll, wenn auf das Element mit der ID `btn1` geklickt wird.
- ⑥ Ist dies der Fall, werden über `$('p')` alle Absätze selektiert und ihnen wird über die Methode `css()` die entsprechende CSS-Formatierung zugewiesen. Hier wird z. B. die Hintergrundfarbe Grün gesetzt.
- ⑦ Wird auf das Element mit der ID `btn2` geklickt, werden mit `$('p[align=right]')` nur die rechtsbündig ausgerichteten Absätze selektiert und der Hintergrund wird entsprechend rot eingefärbt.
- ⑧ Hier legen Sie das Klickereignis für Element `btn3` fest, bei dem der Absatz mit der ID `absatz3` rechtsbündig ausgerichtet werden soll.

Wie Sie im Quelltext sehen können, werden die Ereignisse, die abgefragt werden sollen, nicht direkt am Element angegeben, sondern über den Selektor innerhalb der jQuery-Routine definiert.

Effekte und Animationen

Mit jQuery können Sie Elemente animiert ein- und ausblenden lassen. Bekannt sind diese Effekte z. B. in FAQs, in denen man auf eine Frage klickt und die Antwort animiert ein- und ausgeblendet wird. Dies ist in jQuery über die Methode `slideToggle()` zu realisieren.

Beispiel: *kap13/jquery-toggle.html*

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Absätze einblenden</title>
  ① <script type="text/javascript" src="jquery/jquery-2.1.1.min.js"></script>
  <style type="text/css">
    p { border: 3px dashed green; padding: 5px }
  </style>
</head>
<body>
  <h3>Antworten ein- und ausblenden</h3>
  ② <div id="fragen">
    <div>
      <a href="#">Meine Frage Nummer 1 </a>
      <p>Antwort zur Frage Nummer 1: Fortune plango vulnera...</p>
    </div>
    <div>
      <a href="#">Meine Frage Nummer 2 </a>
      <p>Antwort zur Frage Nummer 2: Fortune plango vulnera ...</p>
    </div>
    <div>
      <a href="#">Meine Frage Nummer 3 </a>
      <p>Antwort zur Frage Nummer 3: Fortune plango vulnera ...</p>
    </div>
  </div>

```

```

③ <script type="text/javascript">
④   $(document).ready(function() {
      $('#fragen div a').click(function() {
⑤       $(this).next('p').slideToggle("slow");

      $('#fragen div p').hide();
    });
  });
</script>
</body>
</html>

```

- ① Als Erstes binden Sie die jQuery-Bibliothek in die Webseite ein.
- ② Im nächsten Schritt legen Sie einen Container mit der ID `fragen` fest. Darin befinden sich weitere Container ohne ID mit jeweils einem Link als Frage, gefolgt von dem Absatz mit der Antwort.
- ③ Innerhalb von `$(document).ready(function() { });` definieren Sie das Klick-Ereignis für das HTML-Element `a`, das ein Kind von `div` sein muss. Dieses wiederum muss ein Kind des Elements mit der ID `fragen` sein.
- ④ Wird dieses Ereignis ausgelöst, ermitteln Sie über die Angabe von `$(this)`, welches Element angeklickt wurde. Mit der Methode `next('p')` greifen Sie ausgehend von dem Element auf den nächstgelegenen Absatz zu. Diesen Absatz, der die Antwort auf die Frage beinhaltet, blenden Sie mit der Methode `slideToggle()` ein und aus.
- ⑤ Damit beim Laden der Webseite erst einmal keine Antwort sichtbar ist, blenden Sie am Anfang mit `hide()` alle im Container `fragen` befindlichen Absätze aus.

Antworten ein- und ausblenden

[Meine Frage Nummer 1](#)

[Meine Frage Nummer 2](#)

Antwort zur Frage Nummer 2: Fortune plango vulnera stilitibus oculis. Mihi ad enarrandum hoc argumentum comit, si ad ausculandum vostra erit benignitas. Qui autem auscultare nolet, exurgat foras, ut sit, ubi sedeat ile qui auscultare vult. Nunc qua ad enarrandum hoc argumentums comit, si ad auscultandum vostra erit benignitas.

[Meine Frage Nummer 3](#)

Antwort zur Frage Nummer 3: Fortune plango vulnera stilitibus oculis. Mihi ad enarrandum hoc argumentum comit, si ad ausculandum vostra erit benignitas. Qui autem auscultare nolet, exurgat foras, ut sit, ubi sedeat ile qui auscultare vult. Nunc qua ad enarrandum hoc argumentums comit, si ad auscultandum vostra erit benignitas.

Die Fragen 2 und 3 wurden angeklickt

Daten laden mit Ajax

Ein weiteres Feature von jQuery ist die Integration der Ajax-Funktionalität. Hierfür wird in der JavaScript-Bibliothek eine erweiterte Form des XMLHttpRequest-Objekts zur Verfügung gestellt, das jQuery XMLHttpRequest, kurz `jqXHR`.

```
$.ajax()
```

Die Methode `$.ajax()` erwartet in Klammern die für die Kommunikation mit dem Server notwendigen Parameter. Die wichtigsten Parameter finden Sie in der nachfolgenden Tabelle:

Parameter	Erläuterung
url	Dies ist die Adresse, die vom Skript aufgerufen werden soll.
type	Mit <code>type</code> geben Sie die Methode der Datenübertragung an, meist <code>POST</code> oder <code>GET</code> . Letzteres ist der Vorgabewert.
async	Hier wird ein boolescher Wert erwartet. Dieser Parameter gibt an, wie die Datenanforderung erfolgen soll. Der Standardwert <code>true</code> besagt, dass der Browser nicht auf den Rückgabewert des Servers warten soll. Dies soll verhindern, dass der Browser den Anwender blockiert, wenn keine Rückmeldung vom Server erfolgt.
data	Mit diesem Parameter können Sie dem aufgerufenen Skript Werte übergeben, die in Form eines Query-Strings gesendet werden. Diese Werte sind in der Form <code>&variable1=wert1&variable2=wert2</code> anzugeben.
datatype	Der Server wird eine Antwort in einer festgelegten Struktur zurückliefern. Damit der Browser weiß, wie er damit umgehen soll, geben Sie den Datentyp an. Verfügbare Typen sind <code>xml</code> , <code>html</code> , <code>script</code> , <code>json</code> , <code>jsonp</code> und <code>text</code> .
success(data, textStatus, jqXHR)	Hier hinterlegen Sie eine sogenannte Callback-Funktion, die im Erfolgsfall aufgerufen werden soll. Der aufgerufenen Funktion können Sie dann drei optionale Rückgabewerte übergeben. <code>data</code> enthält die Daten vom Server und <code>textStatus</code> ist eine Zeichenkette, die den Rückgabestatus beschreibt. Der dritte Parameter ist das <code>jqXHR</code> -Objekt, das den Aufruf der Ajax-Abfrage durchgeführt hat.

Eine komplette Liste der möglichen `$.ajax()`-Parameter können Sie in der offiziellen, englischsprachigen jQuery-API-Dokumentation unter <http://api.jquery.com/jquery.ajax/> nachlesen.



Beispiel: kap13/jquery_ajax.html

Das Beispiel *ajax-get.html* aus diesem Ajax-Kapitel soll als jQuery-Variante ausgeführt werden.

①	<pre> <!DOCTYPE html> <html> <head> <meta charset="utf-8"> <title>Mit jQuery Text nachträglich laden</title> <script type="text/javascript" src="jquery/jquery-2.1.1.min.js"></script> <script type="text/javascript"> \$(document).ready(function() { </pre>
②	<pre> \$.ajax({ type: 'GET', url: 'text.txt', datatype: 'text', </pre>
③	<pre> success: function(data, statusCode) { </pre>
④	<pre> \$.ajax({ type: 'GET', url: 'text.txt', datatype: 'text', <h2>AJAX - Text laden</h2> <p>Der nachfolgende Text wird dynamisch nach dem Laden der Webseite geladen und angezeigt.</p> <div id="textstelle">Dieser Text wird ausgetauscht...</div> <div id="status">Statusausgabe</div> }) } } </pre>
	<pre> } } } </body> </html> </pre>

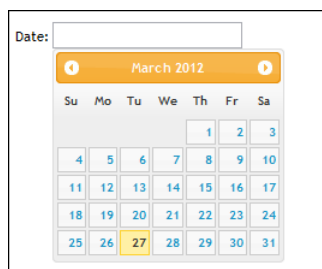
- ① Die JavaScript-Bibliothek wird zur weiteren Verwendung in die Webseite eingebunden.
- ② Nachdem das Laden der Webseite beendet ist, soll über `$.ajax()` im Hintergrund eine entsprechende Server-Anfrage durchgeführt werden. Die Anfrage soll über die Methode `GET` erfolgen und die Datei `text.txt` laden. Die Server-Antwort wird als Text erwartet. Die `$.ajax()`-Methode erhält die Parameter als Name-Wert-Paare der Form `({name1: 'wert1', name2: 'wert2', ...})`.
- ③ Nach der erfolgreichen Server-Rückmeldung soll die nachfolgende Funktion ausgeführt werden. Als Variablen werden die Serverdaten (`data`) und der Status in Textform (`statusCode`) übergeben.
- ④ An dem Element mit der ID `textstelle` soll der Text und beim Element `status` der Rückgabestatus in Textform ausgegeben werden.

jQuery-Erweiterungen

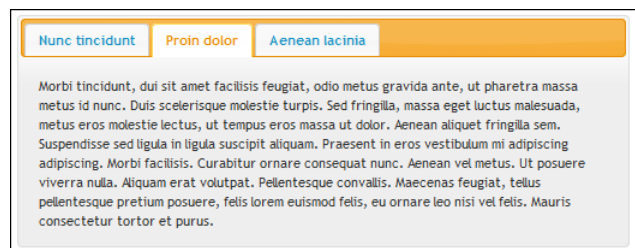
jQuery ist so aufgebaut, dass Erweiterungen erstellt und in eigene Skripte eingebunden werden können. Das bekannteste Plug-In ist das jQuery User Interface, kurz jQuery UI. Neben einer Reihe von Interaktionen, wie Drag & Drop und Sortieren, oder erweiterten Effekten, wie Farbanimationen und Überblendeeffekten, stellt das jQuery UI auch eine Reihe visueller Steuerelemente zur Verfügung, die unter dem Begriff Widgets zusammengefasst sind.

Neben dem Autocomplete-Widget, das automatisch Text bei Benutzereingaben vervollständigt, gibt es auch den DatePicker, einen einfach zu bedienenden Kalender mit Datumsauswahl. Des Weiteren können Sie mit dem Widget Dialog verschiedene Dialogfenster erstellen, mit Progressbar eine Fortschrittsanzeige realisieren, mit Slider verschiedene Schieberegler anbieten, mit Tabs Registerkarten erstellen und mit Accordion, einer Art Ziehharmonikakomponente, bestimmte Elemente automatisch ein- und ausklappen.

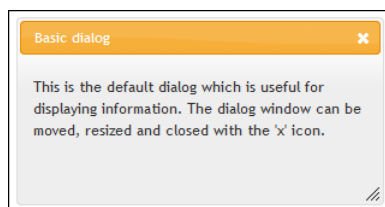
Das jQuery UI, dessen Widgets auch einzeln verwendet werden können, ist unter <http://jqueryui.com/> zu finden. Dort finden Sie auch eine ausführliche Anleitung und verschiedene Beispiele.



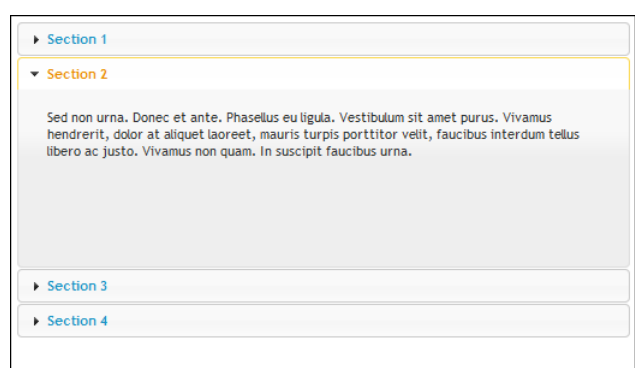
Der Kalender mit Datumsauswahl



Inhalte einer Webseite sind in Register aufgeteilt



Variable Dialogfenster



Der Ziehharmonikaeffekt