

# jQuery - die Javascript-Bibliothek

Rene Schickbauer

2012-04-28

## Überblick

- Über mich
- Web 2.0?

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

# Überblick

# Über mich

## Überblick

- **Über mich**
- Web 2.0?

## Die Ausgangssituation

## jQuery

## Beispiel: Tabs

## Plugins

## Selektoren und Chaining

## Abschluss

- Name: Rene Schickbauer
- Beruf: Systemadministrator und Softwareentwickler in der Produktionsnahen IT
- Arbeitgeber: Magna Powertrain, Ilz
- Alter: 35
- Hobbies: Programmieren, Radfahren, Bücher schreiben, Zeitraffer-Aufnahmen für YouTube machen, ...
- Familienstand: Einsam (interessierte Frauen melden sich bitte nach dem Vortrag ;-)

# Web 2.0?

## Überblick

- Über mich
- **Web 2.0?**

## Die Ausgangssituation

## jQuery

## Beispiel: Tabs

## Plugins

## Selektoren und Chaining

## Abschluss

- Immer mehr webbasierte Anwendungen (alles in der Wolke bzw. “Cloud”)
- Anwender erwarten sich, daß die Webanwendungen wie lokale Applikationen statt wie statische Webseiten funktionieren
- HTML/CSS/Javascript ist komplex. HTML5 noch mehr.
- jQuery ist eine Javascript-Bibliothek, die viele Web2.0-Aufgaben start erleichtert.

Überblick

---

Die Ausgangssituation

---

- Web 1.0
- Web 1.5
- Web 2.0

jQuery

---

Beispiel: Tabs

---

Plugins

---

Selektoren und Chaining

---

Abschluss

---

# Die Ausgangssituation

# Web 1.0

Überblick

Die Ausgangssituation

● Web 1.0

● Web 1.5

● Web 2.0

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

- Statische Seiten werden vom Server erzeugt
- Forms werden an den Server gesendet, der wiederum statische Seiten erzeugt
- Seiten werden von Hand oder mit Toolkits als statisches HTML+CSS erzeugt und ggf. als Template abgelegt.
- Layout werden per HTML und CSS gemacht

# Web 1.5

Überblick

Die Ausgangssituation

- Web 1.0
- **Web 1.5**
- Web 2.0

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

- Bessere Interaktivität durch Javascript und AJAX
- Mehrfache CSS-dateien erlauben Theming-Support.
- Die meisten Javascripts sind “Homebrew” und projektspezifisch.
- Veränderung des Layouts ist komplex.
- Die 20+ verschiedenen Browser sind alle kaputt. Da hilft nur browserspezifisches Coden.

# Web 2.0

Überblick

Die Ausgangssituation

- Web 1.0
- Web 1.5
- **Web 2.0**

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

- Aufkommen der großen Webapplikationen
- HTML5 hält langsam aber sicher Einzug
- Seiten-Layout wird immer mehr erst auf Client-Seite durch Javascript “fertiggestellt”
- Aufgrund der steigenden Komplexität bevorzugen immer mehr Webentwickler fertige Javascript-Toolkits.
- Die mittlerweile 50+ (durchweg kaputten) Browser sind immer noch ein Problem. Aber viele Javascript-Toolkits können damit transparent umgehen...

Überblick

---

Die Ausgangssituation

---

**jQuery**

---

- Was ist jQuery
- Theme generieren
- Seite vorbereiten
- jQuery-Funktionen nach dem Laden der Seite
- Elemente umbauen

Beispiel: Tabs

---

Plugins

---

Selektoren und Chaining

---

Abschluss

---

**jQuery**

# Was ist jQuery

Überblick

Die Ausgangssituation

jQuery

● Was ist jQuery

- Theme generieren
- Seite vorbereiten
- jQuery-Funktionen nach dem Laden der Seite
- Elemente umbauen

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

- Umfangreiche Javascript-Bibliothek zur DOM-Manipulation und Eventhandling
- jQueryUI bietet Theme-Support und einen Online-Konfigurator
- Riesige Auswahl an Plugins
- Operationen können über Selektoren an vielen Elementen gleichzeitig durchgeführt werden
- jQuery unterstützt eine große Anzahl Browser

# Theme generieren

Überblick

Die Ausgangssituation

jQuery

- Was ist jQuery
- **Theme generieren**
- Seite vorbereiten
- jQuery-Funktionen nach dem Laden der Seite
- Elemente umbauen

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

- Den jQueryUI Themeroller im Browser aufrufen:  
<http://jqueryui.com/themeroller/>
- Theme konfigurieren
- Theme downloaden
- File entpacken und zum Webespace hochladen

# Seite vorbereiten

Überblick

Die Ausgangssituation

jQuery

- Was ist jQuery
- Theme generieren
- **Seite vorbereiten**
- jQuery-Funktionen nach dem Laden der Seite
- Elemente umbauen

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

- Folgende Zeilen in die Webseite einfügen um jQueryUI zu laden (Pfad und Versionsnummern anpassen!):

```
<link type="text/css"
href="css/custom-theme/jquery-ui-1.8.19.custom.css"
rel="stylesheet" />
```

```
<script type="text/javascript"
src="js/jquery-1.7.2.min.js"></script>
```

```
<script type="text/javascript"
src="js/jquery-ui-1.8.19.custom.min.js"></script>
```

# jQuery-Funktionen nach dem Laden der Seite

Überblick

Die Ausgangssituation

jQuery

- Was ist jQuery
- Theme generieren
- Seite vorbereiten
- **jQuery-Funktionen nach dem Laden der Seite**
- Elemente umbauen

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

- Um jQuery nach dem Laden der Seite auf Elemente anzuwenden, brauchen wir eine passende Funktion

```
<script type="text/javascript">  
    $(document).ready(function(){  
        });  
</script>
```

# Elemente umbauen

Überblick

Die Ausgangssituation

jQuery

- Was ist jQuery
- Theme generieren
- Seite vorbereiten
- jQuery-Funktionen nach dem Laden der Seite
- **Elemente umbauen**

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

- In diese Funktion können wir beliebige jQuery-Aufrufe einbauen
- z.B. möchten wir alle Buttons und Form-Buttons als jQuery-Buttons darstellen:

```
$( "button, input:submit, input:reset, input:button").button();
```

- Ausserdem möchten wir alle Elemente mit der ID “datepicker” zu jQuery-Datepickern machen (Datumfelder mit klickbaren Kalendern):

```
$('#datepicker').datepicker({  
    inline: true  
});
```

Überblick

Die Ausgangssituation

jQuery

**Beispiel: Tabs**

- HTML code
- Javascript code

Plugins

Selektoren und Chaining

Abschluss

# Beispiel: Tabs

## HTML code

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

● **HTML code**

● Javascript code

Plugins

Selektoren und Chaining

Abschluss

Wir möchten auf unserer Seite Tabs darstellen. Kein Problem.

- Als erstes erstellen wir das entsprechende HTML mit unserem Inhalt:

```
<div id="tabs">
  <ul>
    <li><a href="#tabs-1">First</a></li>
    <li><a href="#tabs-2">Second</a></li>
    <li><a href="#tabs-3">Third</a></li>
  </ul>
  <div id="tabs-1">Das ist das erste Tab</div>
  <div id="tabs-2"></div>
  <div id="tabs-3"><input type="button" value="Drittes
  Tab"></div>
</div>
```

# Javascript code

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

● HTML code

● **Javascript code**

Plugins

Selektoren und Chaining

Abschluss

- Javascript Code einfügen um die div-Elemente in echte Tabs umzuwandeln:

```
$('#tabs').tabs();
```

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

**Plugins**

- Zentrales Archiv
- Beispiel: Tabellen 1/2
- Beispiel: Tabellen 2/2

Selektoren und Chaining

Abschluss

# Plugins

# Zentrales Archiv

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

● **Zentrales Archiv**

● Beispiel: Tabellen 1/2

● Beispiel: Tabellen 2/2

Selektoren und Chaining

Abschluss

- Bis vor kurzem auf <http://plugins.jquery.com>
- Das große Uuups: *“Later in the week, while I was attempting to delete four spam items, I was left completely horrified when the results of the operation reported that the remaining 10% — every single plugin remaining in the database — had been purged. All that remained was a year-old backup.”*
- Unvollständiges Archiv auf <http://archive.plugins.jquery.com/>
- Komplettes neues System in Arbeit
- Fast alle Plugins haben ihre eigenen Webseiten und sind über Google findbar
- Suchtipp *“Top jquery plugins for `problem_description`”*

## Beispiel: Tabellen 1/2

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

- Zentrales Archiv
- **Beispiel: Tabellen 1/2**
- Beispiel: Tabellen 2/2

Selektoren und Chaining

Abschluss

- Eine Datentabelle soll mit mit Javascript-Sortierung und Filterung ausgestattet werden.
- Suche nach “*top jquery plugins for tables*” ergibt u.a. einen Link auf das wsblogz
- Erstes Plugin: DataTables
- Passendes CSS in die Webseite dazuladen (Pfade anpassen):

```
<style type="text/css" title="currentStyle">  
    @import "/foo/css/demo_page.css";  
    @import "/foo/css/demo_table.css";  
</style>
```

## Beispiel: Tabellen 2/2

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

- Zentrales Archiv
- Beispiel: Tabellen 1/2
- **Beispiel: Tabellen 2/2**

Selektoren und Chaining

Abschluss

- Nach dem jQuery-Javascript das Plugin-Script dazuladen (Pfad anpassen):

```
<script type="text/javascript" language="javascript"
src="/foo/js/jquery.dataTables.js"></script>
```

- Die Datentabelle mit der ID “mydatatable” umbauen (lassen):

```
$('#mydatatable').dataTable();
```

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

**Selektoren und Chaining**

- Selektoren
- Chaining
- Komplexes Beispiel
- Filtern

Abschluss

# Selektoren und Chaining

# Selektoren

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

● Selektoren

● Chaining

● Komplexes Beispiel

● Filtern

Abschluss

- Selektoren können nach verschiedenen Kriterien existierende Elemente auswählen und per chaining Funktionen darauf aufrufen

- Select nach ID und src-Attribut ändern:

```
$("#" + cimageid).attr("src", options.image_checked);
```

- Nach Typ auswählen (mehrere Typen gleichzeitig):

```
$( "button, input:submit, input:reset, input:button").button();
```

- Neues <img> Element aus freier Luft erfinden und nach einem vorher gemerkten Element einfügen:

```
$( '' ).insertAfter( mymarker );
```

# Chaining

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

- Selektoren
- **Chaining**
- Komplexes Beispiel
- Filtern

Abschluss

- Fast alle jQuery-Methoden (Funktionen) geben ein (oder mehrere) Objekte zurück, normalerweise das Objekt, auf dem die Methode angewendet wurde.
- Damit lassen sich auf einen Selektor gleichzeitig mehrere Methoden anwenden:

```
$("#mylogo").fadeOut('fast') .addClass("active")  
.css('marginRight', '10px');
```

- Den Text “Linuxtage” zu jedem <p> und <div> Element hinzufügen:

```
$("<strong>Linuxtage</strong>") .appendTo("div")  
.appendTo("p");
```

# Komplexes Beispiel

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

- Selektoren
- Chaining
- Komplexes Beispiel
- Filtern

Abschluss

- Wir haben viele Checkboxes mit dazugehörigen `<span>`-Elementen:

```
<div> <input type="checkbox" name="a" />
<span>Mary</span> </div>
```

```
<div> <input type="checkbox" name="b" />
<span>Icm</span></div>
```

```
<div> <input type="checkbox" name="c" checked="checked"
/><span>Peter</span> </div>
```

- Finde *alle* nicht angekreuzten Checkboxes und mache dazugehörige `<span>`-Elemente gelb:

```
$("#input:not(:checked) + span").css("background-color",
"yellow");
```

# Filtern

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

- Selektoren
- Chaining
- Komplexes Beispiel

● **Filtern**

Abschluss

- jQuery kann im Chaining auch weiter filtern.
- Alle Listenelemente mit der Klasse “even” rot einfärben

```
$('li').filter(':even').css('background-color', 'red');
```

- Das funktioniert auch mit einer (anonymen) Funktion.
- Alle Listenelemente rot einfärben, die genau ein einzelnes `<strong>`-Tag enthalten:

```
$('li').filter(  
  function(index) {  
    return $('strong', this).length == 1;  
  }).css('background-color', 'red');
```

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

**Abschluss**

- Fragen und Antworten
- Ende

# Abschluss

# Fragen und Antworten

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

● **Fragen und Antworten**

● Ende

“Hier werden Sie geholfen”

# Ende

Danke fürs zuhören!

Überblick

Die Ausgangssituation

jQuery

Beispiel: Tabs

Plugins

Selektoren und Chaining

Abschluss

● Fragen und Antworten

● **Ende**