

# jQuery – DIE Javascript-Library

Uwe Dierolf  
INETBIB 2010, 14. April .2010

KIT-BIBLIOTHEK



# Inhalt

- Motivation
- Selektoren
- Events
- Animationen
- DOM Manipulation
- AJAX
- Plugins
- Tipps für Entwickler
- Beispiele

# Motivation

- Rich Clients
  - Beispiele
    - jQuery UI = User Interface <http://jqueryui.com/>
    - Comicstrip Navigation <http://paulbakaus.com/lab/navigation/comicstrip/>
- Einfache Manipulation von Seiteninhalten
  - Beispiel KVK
    - KVK-Suchmaske „Eigene Kataloge“
- Zugriff auf Daten und Ereignisse in Seiten
  - Beispiel KIT-Katalog
    - Loggen von Mausklicks auf externe Links
    - Tastatursteuerung
- Browserunabhängige Erstellung von Javascript-Code

# Unobtrusive Javascript ?

- Bedeutung
  - unaufdringlich, unauffällig
- Idee
  - CSS trennt Layoutinformation vom HTML
  - jQuery trennt Javascript vom HTML
  - Separation von Funktionalität (behaviour layer)
- Einklinken in HTML-Seiten
  - Laden von jQuery und eigenem JS-Programm
    - `$(document).ready(function() {....});`

# Selektoren

- Schnell mal "Objekte" im HTML auffinden
  - IDs, Klassen, Attribute, Attribute mit speziellen Werten etc.
    - `$('#id')` , `$('.class')`, `$(element)`
  - Attribute
    - `[name=value]`
  - Multiple selectors
    - AND `[name=value][name2=value2]`
    - OR `("selector1, selector2, selectorN")`
  - Hierarchisch
    - `$(ancestor descendant)` , `$(parent > child)` )
  - Filter
    - `$(p:first)` , `$(div:visible)` , `$(*:header)` )
  - U.v.a. mehr (s. [jQuery](#))
    - `$(div:has(img.thumbnail[src$=.png ]:not(:hidden)))'` )

# Selektoren II

- Liefern Treffermengen (wrapped set), auf die Aktionen angewandt werden können
- Chaining von Aktionen
  - Jede Methode liefert die Objekte zurück, auf denen Sie arbeitet
  - Beispiel
    - Suche alle Paragraphen und mache darin den Text rot und blende sie dann langsam aus  
`$("#p").css("color","red").hide("slow")`

# Events

- Ereignisse abfangen
  - Events an Objekte binden (bind / unbind)
    - `$('#id').bind('click' , callbackHandler)`
    - `$('#id').bind('click', function() { alert('User clicked on "foo."'); });`
  - Shortcut für `bind('eventname')`
    - `click()`
    - `mouseenter()` etc.
- Eventhandler erhält das Event Object
  - Praktisch, um z.B. das Objekt zu ermitteln, auf das geklickt wurde
- Live-Events
  - Events Objekten zuordnen, die es noch nicht gibt

# Animationen

- Von jQuery UI (= user interface) lernen
  - Comicstrip animation (a la Macintosh)
    - Z.B. von Paul Bakaus
    - Fisheye <http://interface.eyecon.ro/demos/fisheye.html>



# DOM-Manipulation

- Insert / change / replace / remove nodes
- Inside `.append(content)` / `.prepend(content)`
- Outside `.after(content)` / `.before(content)`
- Around `.wrap(content)` / `.wrapAll(content)`
- Einfache Operationen
  - `$('#id').text('der Text')` / `$('#id').html('<p>huhu</p>')`
  - `$('#id').empty()` / `.remove()`

# AJAX

- Laden von "externen" Inhalten
  - VORSICHT: Sicherheitsmodell schränkt ein
    - Extern heißt "auf dem eigenen Server !"
    - Man sieht nicht unbedingt sofort, dass AJAX im Spiel ist
  - `$('#id').load("externalData.html" )`
    - Praktisch: `load()` erlaubt auch die Angabe von Selektoren
      - `load(file #id)`
  - `$.get("http://. . ." , [ data ] , [ callback ] , [ type ] ) ;`
    - Analog POST-Requests
  - `$.getScript("meinJavascript.js" , [ callback ] ) ;`
  - Callback wird nur bei Erfolg aufgerufen !

# Tipps

- Möglichst oft IDs und Klassennamen im HTML einsetzen
  - Sowas kann man einfach per Selektor suchen
  - Man kann mehr als nur einen Klassennamen verwenden
    - Wird oft zum Speichern eines Status benutzt
- Beliebige eigene Attributnamen sind praktisch
  - Kann man mit `attr()` leicht lesen und ändern
    - `$('#id').attr('attributname')`
    - `$('#id').attr('attributname',newValue)`

# Plugins nutzen

- Wer sucht, der findet !
  - Nicht gleich alles selber programmieren
  - Es gibt tausende von Plugins
  - Eigene Plugins zu erstellen, ist nicht (so) schwer

# "Meine" Plugins

- `jquery.cookies.2.2.0.js`
  - Cookie-Management
- `jquery.hotkeys.js`
- `jquery.json-2.2.js`
  - JSON-Behandlung
- `jquery.values.js`
  - Verwaltung von Formularwerten
- `jquery.sync-load.js` (Jakob Westhoff)
  - Synchrone `load()`-function zum Laden von HTML in Variablen
- `jtip.js` (modified)
  - Tooltips von Cody Lindley

# Fallstricke beim Entwickeln

- Fehler (frühzeitig) erkennen
  - Beispiel
    - Include eines jQuery-Plugins vergessen aber Plugin-function aufgerufen
    - Ready-Handler bricht mitten drin ab !
- Firefox-Fehlerkonsole öffnen
- Firebug-Konsole benutzen
  - Unten am Rand aufs rote Kreuz achten

# Fallstricke bei event-Handlern I

- Man kann einen Event mehrmals an ein Objekt binden
  - vorher unbind() verwenden
  - Hilfreich: "namespaces"
    - `$('#div').bind('click.mynamespace' , function () { /* ... */ });`
    - `$('#div').unbind('click.mynamespace');`
    - Alles entfernen:  
`$('#div').unbind('. mynamespace');`
- Event-Modell hängt **LEIDER DOCH** vom Browser ab
  - Zum Glück nur selten !
  - Bsp.: change-Event kommt beim IE erst nach blur
  - Bubbling von unten nach oben
  - Ggfs. stopPropagation verwenden

## Fallstricke bei event-Handlern II

- Callback ohne Argumente
  - `$.get('myhtmlpage.html', myCallBack);`
- Callback mit Argumenten
  - **Falsch:**  
`$.get('myhtmlpage.html',  
myCallBack(param1, param2));`
  - **Richtig**  
`$.get('myhtmlpage.html',  
function(){ myCallBack(param1, param2); });`



# Javascript-Tipps

- Rückgabewerte
  - Sind manchmal *null* oder *undefined* oder der String "*undefined*"
- Rückgabewerte sind manchmal Objekte, manchmal Strings
  - Beispiel JSON
    - Firefox ab 3.5 behandelt JSON intern als Objekt, IE noch nicht
    - Führt zu JSON.parse-Fehlern
- Code schützen
  - Google Closure Compiler
    - <http://code.google.com/intl/de-DE/closure/compiler/>
  - JS Obfuscator
    - <http://javascript-obfuscator.software.informer.com/>

# VORSICHT Cookies

- Cookies als Gedächtnis
  - Kann ins Auge gehen, da zu viel zum Server übertragen wird
    - Error 500
  - Abhilfe: localStorage
  - Nachteil: Kennen noch nicht alle Browsern

# jQuery lernen

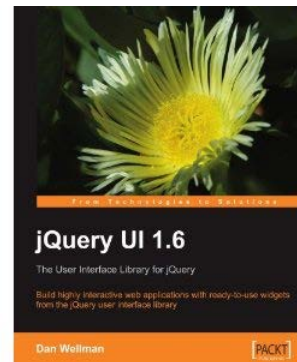
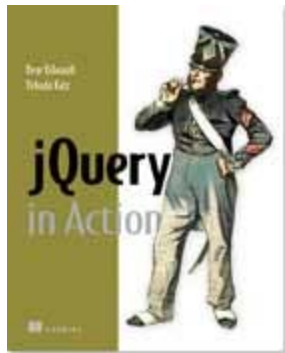
- jQuery in Action lesen
  - Herumspielen mit den diversen Labs
    - <http://www.ubka.uni-karlsruhe.de/jquery/jquery-in-action-DEMO/chapter1/document.ready.html>
- Tools für Firefox
  - Firebug
  - FireQuery
  - Eventbug
  - Livehttpheaders

# Fazit

- Wer es nicht benutzt, ist selber schuld
  - jQuery ist klein
  - Ziemlich gut getestet
  - Sehr performant
  - Kompatibel mit anderen JS-Libraries (noConflict)
- jQuery 1.4 ist noch neu
  - Z.T. um Faktoren schneller
  - Vorsicht bei Verwendung alter Plugins oder UI

# Literatur

- *jQuery in Action*, Bear Bibeault und Yehuda Katz, engl.
- *jQuery*, (Ralph Steyer), dt.,
- *Learning jQuery 1.3*, PACKT Publishing, engl.
- *jQuery Reference Guide*, PACKT Publishing, engl.
- *jQuery UI 1.6*, PACKT Publishing, engl. (neu 1.7)



# Links

- [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page)
- <http://www.ubka.uni-karlsruhe.de/jquery/jquery-in-action-DEMO/>
- [http://en.wikipedia.org/wiki/Unobtrusive\\_JavaScript](http://en.wikipedia.org/wiki/Unobtrusive_JavaScript)
- <http://westhoffswelt.de/blog.html>
- [http://westhoffswelt.de/data/portfolio/webtechcon\\_2009\\_bubbles\\_and\\_trees\\_with\\_jquery.pdf](http://westhoffswelt.de/data/portfolio/webtechcon_2009_bubbles_and_trees_with_jquery.pdf)