

JavaScript: Von einfachen Scripten zu komplexen Anwendungen

MMT 28

3. Dezember 2011

Guten Tag

Mathias Schäfer (molily)

molily.de

Software-Entwickler bei 9elements.com

Spezialgebiet JavaScript

Worum geht es?

Überblick über bewährte Techniken, um
JavaScript zu strukturieren

Insbesondere größere clientseitige
Anwendungen

Themen

- I. Hintergrund
- II. Einfache Scripte mit jQuery und Co.
- III. Erste Strukturierungen
- IV. Vier Ansätze für komplexe Anwendungen

I.

JavaScript ist toll!

Was ist das Besondere an JavaScript? (technisch)

Unkonventionell, da dynamisch und funktional

Mächtig und ausdrucksstark

Muss verstanden und beherrscht werden

Fallstricke und Designfehler

Was ist das Besondere an JavaScript? (sozial)

Viele Anfänger und Nicht-Fachleute schreiben
JavaScript

The World's Most Misunderstood Programming
Language (Crockford 2001)

... Has Become the World's Most Popular
Programming Language (Crockford 2008)

II.

Einfache Scripte

jQuery

Führende Bibliothek für Scripting im Browser

Benutzereingaben verarbeiten

Das Dokument verändern

Daten empfangen und senden

jQuery

```
jQuery(document).ready(function ($) {  
    $("#loadmore").click(function () {  
        var href = $(this).attr("href");  
        $("#content").load(href);  
    });  
});
```

Was jQuery gut kann

Schneller Einstieg

Knapper und verständlicher Code

Häufige Aufgaben einfach lösen

Ökosystem an Plugins und Fertigsripten

Was jQuery nicht leistet

jQuery deckt nur einen kleinen Bereich ab

jQuery-Code skaliert nicht,
wird schlecht lesbar und wartbar

jQuery bietet (fast) nichts zur Strukturierung

III.

Erste Strukturierung

Unstrukturierter Code

```
wert = 0;
```

```
document.getElementById('addieren')  
  .addEventListener('click', erhöhe, false);
```

```
function erhöhe () {  
  wert += 1;  
  document.getElementById('ausgabe').innerHTML = wert;  
}
```

Objekt-Literal

```
var Zähler = {  
  
    buttonId : 'addieren',  
    ausgabeId : 'ausgabe',  
  
    wert : 0,  
  
    init : function () {  
        var button = document.getElementById(this.buttonId);  
        button.addEventListener('click', this.erhöhe, false);  
    },  
  
    erhöhe: function () {  
        Zähler.wert += 1;  
        document.getElementById(this.ausgabeId).innerHTML =  
            Zähler.wert;  
    }  
  
};
```

Module

```
var Zähler = (function () {  
  
    var buttonId = 'addieren',  
        ausgabeId = 'ausgabe',  
        wert = 0;  
  
    function init () {  
        var button = document.getElementById(buttonId);  
        button.addEventListener('click', erhöhe, false);  
    }  
  
    function erhöhe () {  
        wert += 1;  
        document.getElementById(ausgabeId).innerHTML = wert;  
    }  
  
    init();  
  
    return { erhöhe: erhöhe };  
  
})();  
  
Zähler.erhöhe();
```


Pseudoklassen

```
function Auto (name) {  
  this.name = name;  
}
```

```
Auto.prototype.fahre = function () {  
  alert(this.name + ' macht brumm!');  
};
```

```
var käfer = new Auto('Käfer');  
käfer.fahre();
```

50% sind geschafft

Behaviour Layer (Koch 2004, Weakley 2005)

Unobtrusive JavaScript (Heilmann 2005)

Objekt-Literal (Heilmann 2006)

Module-Pattern (Heilmann 2007)

Pseudoklassen und prototypische Vererbung

IV.

**JavaScript-
Anwendungen**

Problemstellung

Ist JavaScript für größere
Webanwendungen geeignet?

Nicht ohne weiteres.



Marinus Lübbe

News Stream

Edit Subscriptions

27 Subscriptions

- Legendary Directors 14
- Arrested Developm... 1
- R.I.P.D. 1
- I Melt With You 1
- Resident Evil: Retr... 1
- Frank or Francis 1
- Seven Psychopaths 1
- Hugo 1
- Shame 3



Mirror Mirror – indieWIRE's The Playlist – 2 days ago

Edit

Tarsem's Snow White Movie Now Officially Titled 'Mirror Mirror'



With a release date about five months away, it's about time Tarsem's Snow White movie got a title. First known in early stages as *The Brothers Grimm: Snow White*, then simply being that untitled movie that was irritating the hell out of Universal, Relativity have announced today that the film will now be called: *Mirror Mirror*.

Read the full story at indieWIRE's The Playlist:

http://blogs.indiewire.com/theplaylist/archives/tarsems_snow_white_movie_now_officially_titled_mirror_mirror1/

Like Send Be the first of your friends to like this.



Add a comment...

Posting as Marinus Lübbe (Not you?) Comment

Post to Facebook

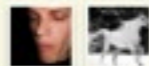
Facebook social plugin

Is Relativity's 'Mirror Mirror' a better title than Universal's 'Snow White and the Huntsman'?

Absolutely



Absolutely not



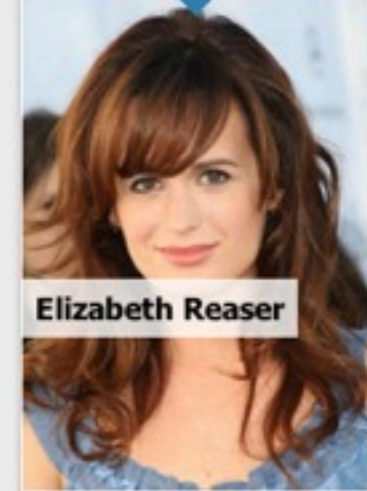
2 picked this

What about you?

Post to Facebook

Nobody picked this

Do you like ...



Elizabeth Reaser

like next

Moviepilot.com

Englischsprachiger Ableger von Moviepilot.de

Film-Nachrichten basierend auf dem
persönlichen Filmgeschmack

Kommende Filmprojekte, Schauspieler,
Regisseure und Themen verfolgen

Frontend-Technik

Backend: Ruby on Rails

Frontend: Haml, Sass, CoffeeScript

Bibliotheken: jQuery, Underscore, Backbone
und Handlebars

Models, Views und Controller (MVC)

Eine API liefert JSON-Daten

I.

Konventionen und Standards

Coding-Guidelines

JavaScript: The Good Parts
(Crockford 2008)

Qualität durch
Selbstbeschränkung

Fallstricke vermeiden



JSLint und JSHint

Code-Analyse anhand der »Good Parts«

Fehler und Warnungen beim Regelverstoß

Eingebettet in Editoren und Build-Werkzeugen

ECMAScript-Standards

ECMAScript 5 (Dezember 2009)

Korrekturen, Ergänzungen und Neuerungen für robustere Programme

ECMAScript 6 (nahe Zukunft)

Eine bessere Sprache zur Entwicklung komplexer Anwendungen und Bibliotheken

2.

Metasprachen und Compiler

Google Closure Compiler

Optimierung und Komprimierung

Überprüfen der Google-Coding-Guidelines

Anmerkungen zu Datentypen, Parametern,
Zugriffsrechten, Klassen usw.

Code-Analyse und Warnungen z.B. bei
Typänderungen

CoffeeScript

Metasprache, die nach JavaScript übersetzt wird

Bessere Lesbarkeit und Syntaxzucker

Verbessert die Produktivität

Keine Antwort auf sämtliche JS-Probleme

CoffeeScript

```
jQuery(document).ready ($) ->  
  $("#loadmore").click ->  
    href = $(@).attr "href"  
    $("#content").load href
```

Google Web Toolkit

Komplettlösung für große Anwendungen

Klassenbibliothek mit vorgefertigten UI-Elementen

Entwicklung in Java in einer IDE wie Eclipse

3.

Design-Pattern

Model View Controller

Bewährtes Pattern für grafische
Benutzeroberflächen

Model: Rohdaten und deren Logik

View: Darstellung der Daten

Controller: Benutzeraktionen auswerten

MVC mit Backbone

Model: Rohdaten als JSON empfangen,
validieren und senden

View: Daten mittels Templates rendern,
Model beobachten und verändern (Binding)

Controller: nicht vorgesehen

Publish/Subscribe

Auf ein Ereignis in einem Teil der Anwendung sollen verschiedene andere Teile reagieren

Ereignis-basierte lose Kopplung:

```
PubSub.publish('login', user)
```

```
PubSub.subscribe('login', callbackFunction)
```

4.

Modularisierung und Building

Build-Werkzeuge

Entwicklung:

Lesbar und verständlich

Produktion:

Komprimiert, zusammengefasst, performant

YUI Compressor, Google Closure Compiler

Module mit RequireJS

Asynchronous Module Definition (AMD)

basisModul.js:

```
define({ f: 1 });
```

aufbauModul.js:

```
require(['basisModul'], function (basisModul) {  
    alert(basisModul.f); // 1  
})
```



Erkenntnisse

JavaScript macht Spaß und lohnt sich

JavaScript-Anwendungen sind nicht einfach

Man muss sich viel selbst erarbeiten

Best Practises für große Anwendungen fehlen

Sprache, Bibliotheken und Tools müssen besser werden

Empfehlungen

Ein komfortables Abstraktionslevel schaffen

Strukturen und Konventionen geben Halt

Vorhandene Werkzeuge ausschöpfen

Metasprachen wie CoffeeScript ausprobieren

Gute MVC-Bibliotheken wählen

Know-How übertragen und weiterentwickeln

Danke!

**Folien und Links unter
molily.de/mmt28**