

Metaprogrammierung

Metaprogrammierung

Themenfelder in Programmiersprachen

Daten notieren, speichern und verwenden

- Variablen, Zuweisungen, Ausdrücke, Operatoren

Ablauf und Reihenfolge von Instruktionen

- Befehle,
Bedingte Anweisungen, Fallunterscheidungen, Schleifen

Programmteile gruppieren, Komplexität abstrahieren

- Funktionen, Parameter, Argumente, Rückgabewerte

Abstraktion von konkreten Daten, Modellierung

- Objekte, Methoden, Attribute, Klassen


jQuery

JavaScript Bibliothek

klar definiertes Problem:

- Browser sind / waren unfreundlich
 - UI im Browser als Document Object Model gespeichert
 - Manipulation des UI == Identifikation & Manipulation der Elemente
 - getElementById(„mein_elem_23“) holt ein einziges Element
- sehr „geniale“ Nutzungsweise
 - eleganter, sprechender Syntax
 - method chaining !!
 - es sieht sooo einfach aus...
- Browser-Kompatibilität (Abstraktion der Hersteller-„Fehler“)

jQuery Funktion



[Download](#) [API Documentation](#) [Blog](#) [Plugins](#) [Browser Support](#)

- Ajax
 - Global Ajax Event Handlers
- Helper Functions
- Low-Level Interface
- Shorthand Methods
- Attributes
- Callbacks Object
- Core
- CSS
- Data
- Deferred Object
- Deprecated
 - Deprecated 1.3
 - Deprecated 1.7
 - Deprecated 1.8
 - Deprecated 1.10
- Dimensions
- Effects
 - Basics
 - Custom
 - Fading
 - Sliding
- Events
 - Browser Events
 - Document Loading
 - Event Handler Attachment
 - Event Object
 - Form Events
 - Keyboard Events
 - Mouse Events
- Forms
- Internals
- Manipulation
 - Class Attribute
 - Copying
 - DOM Insertion, Around
 - DOM Insertion, Inside
 - DOM Insertion, Outside
 - DOM Removal
 - DOM Replacement

jQuery API

.add() Traversing > Miscellaneous Traversing

Add elements to the set of matched elements.

.addBack() Traversing > Miscellaneous Traversing

Add the previous set of elements on the stack to the current set, optionally filtered by a selector.

.addClass() Attributes | Manipulation > Class Attribute | CSS

Adds the specified class(es) to each of the set of matched elements.

.after() Manipulation > DOM Insertion, Outside

Insert content, specified by the parameter, after each element in the set of matched elements.

.ajaxComplete() Ajax > Global Ajax Event Handlers

Register a handler to be called when Ajax requests complete. This is an AjaxEvent.

.ajaxError() Ajax > Global Ajax Event Handlers

Register a handler to be called when Ajax requests complete with an error. This is an Ajax Event.

.ajaxSend() Ajax > Global Ajax Event Handlers

Attach a function to be executed before an Ajax request is sent. This is an Ajax Event.

.ajaxStart() Ajax > Global Ajax Event Handlers

Register a handler to be called when the first Ajax request begins. This is an Ajax Event.

.ajaxStop() Ajax > Global Ajax Event Handlers

Register a handler to be called when all Ajax requests have completed. This is an Ajax Event.

.ajaxSuccess() Ajax > Global Ajax Event Handlers

Attach a function to be executed whenever an Ajax request completes successfully. This is an Ajax Event.

All Selector ("*") Selectors > Basic

Selects all elements.

.andSelf() Deprecated > Deprecated 1.8 | Traversing > Miscellaneous Traversing

Add the previous set of elements on the stack to the current set.

.animate() Effects > Custom

Perform a custom animation of a set of CSS properties.

jQuery Funktionalität

Elemente selektieren

- einzeln
- Gruppen
- nach Kriterien

Elemente manipulieren

- Sichtbarkeit, Aussehen, Positionierung
- Existenz, Interaktivität

Datenverkehr zu Servern

- Http Requests GET, POST (Seitenaufrufe)
- AJAX (asynchroner Datenaufruf hinter den Kulissen)

Event Handling

- Anbinden, Auswahl, eigene Events





jQuery „Aktivierungsmuster“

DOLLAR Selektion PUNKT tuWas(mitDenDaten)

„\$“ entspricht „jQuery“ und ist der Funktionsname

`$(„div“)` `$(„.tasten“)` `$(„#student32“)`

- selektiert eine Anzahl Elemente aus dem DOM-Baum
- packt sie in ein spezielles jQuery-Konstrukt

`$(selektion).funktionsname()`

- `hide()`, `show()`, `addClass()`, `animate()`, `replace()`, `children()`
- Manipulieren
- Zugriff auf Attributwerte, lesen oder setzen

```
1 $( ".section.article.p.halb" ).addClass( 'passiv' ).css( { border: 1px solid red, } ).slideUp();
```


jQuery „Aktivierungsmuster“

DOLLAR Selektion PUNKT tuWas(mitDenDaten)

„\$“ entspricht „jQuery“ und ist der Funktionsname

\$(„div“) \$(„.tasten“) \$(„#student32“)

- selektiert eine Anzahl Elemente aus dem DOM-Baum
- packt sie in ein spezielles jQuery-Konstrukt

\$(selektion) funktionenname()

– hide

– Ma

– Zu

```
1 $( ".section.article.p.halb" )  
2     .addClass( 'passiv' )  
3     .css( "border", "1px solid red" )  
4     .slideUp()  
5     .click( function( e ) {  
6         console.log( 'Klick!' );  
7     } );
```

```
slideUp();
```




Aufgabe 01: Eloquent JavaScript

es gibt kaum eine schönere und besser geschriebene Art JavaScript zu lernen, wie mit den Geschichten von Marijn Haverbeke -- <http://eloquentjavascript.net/2ndedition/preview>

ELOQUENT JAVASCRIPT SECOND EDITION

A modern introduction to JavaScript, programming, and the wonders of the digital.

[[new cover here]]

Written by Marijn Haverbeke.

Licensed under a [Creative Commons attribution-noncommercial license](#). All code in this book may also be considered licensed under an [MIT license](#).

Aufgabe 01: Eloquent JavaScript

es gibt kaum eine schönere und besser geschriebene Art
JavaScript zu lernen, wie mit den Geschichten von Marijn
Haverbeke -- <http://eloquentjavascript.net/2ndedition/preview>

ELOQUENT JAVASCRIPT SECOND EDITION

A modern introduction to JavaScript
digital.

[[new cover here]]

Written by Marijn Haverbeke.

Licensed under a Creative Commons
code in this book may also be c

CONTENTS

Introduction

1. Values, Types, and Operators (Part 1: Language)
2. Program Structure
3. Functions
4. Data Structures: Objects and Arrays
5. Higher-order Functions
6. The Secret Life of Objects
7. Practical: Electronic Life
8. Bugs and Error Handling
9. Regular Expressions
10. Modules
11. Practical: A Programming Language
12. JavaScript and the Browser (Part 2: Browser)
13. The Document Object Model
14. Handling Events
15. Practical: A Platform Game