

jQuery

DOM-Scripting für ~~Programmierer~~ Webdesigner

Bibliotheken und Frameworks

- Dojo
- MochiKit
- Yahoo UI
- Prototype
- script.aculo.us
- mootools
- moo.fx
- [und viele mehr](#)

Noch'n Javascript-
Framework?

Ja.

Prototype vs. jQuery

	<i>Prototype</i>	<i>jQuery</i>
<i>Größe komplett</i>	48 KB	48 KB
<i>Größe minimal</i>	3 KB	16 KB
<i>Last stable</i>	1.4 (9/05)	1.0.1 (8/06)
<i>Zusatzmodule</i>	sehr viele	einige
<i>Dokumentation</i>	schäbig	engagiert
<i>Syntax</i>	wie Ruby	wie CSS+JS

Warum dann jQuery? KSD!

- Konzentration
- Schnell zu lernen
- Designerkompatibilität

Klasse hinzufügen

```
$('#navi li:nth-child(2)').addClass('aktiv');
```

Jede Aktion beginnt mit der Auswahl der zu behandelnden DOM-Elemente.

Javascript-Objekt oder CSS3-Ausdruck

= Das zweite Listenelement innerhalb vom Element mit der ID "navi" bekommt die Klasse "aktiv" angehängt.

Event ankleben

```
        $('ul').click(  
function(){ $(this).find('li').css('color', 'red')}  
        );
```

Der click-Befehl klebt seine definierte Funktion als EventListener an das/die ausgewählte(n) Element(e).

`$(this)` nimmt Bezug auf das außen definierte Objekt.

= Bei Klick auf ein beliebiges ul werden alle in dem ul enthaltenen li-Elemente in roter Schrift dargestellt.

Funktionsketten

```
$( '#body h2' ).addClass( 'inner-h2' ).next()  
    .css( 'color', 'red' ).click(  
    function(){ alert( 'gelickt!' )}  
    ).previous().slide( 'slow' );
```

= An jede Funktion können per Punkt beliebig viele weitere Funktionen angehängt werden.

Prototype-Syntax vs. jQuery-Syntax

```
$$('.element').each(function(node) {  
  Element.addClassName(node, 'className');  
})
```

```
$('.element').addClass('className');
```

Bestandteile von jQuery

- DOM-Manipulation
- Events hinzufügen
- Visuelle Effekte
- Ajax-Funktionalität