



# AJAX IN ACTION

Konferenz für Web Development, Design & Technology

## MooTools

Eine Übersicht

## Wer bin ich?

- Jan Kassens
- 21 Jahre
- 2. Semester Informatik an der TU Darmstadt
  
- MooTools Comitter / Contributor
  
- [jan@kassens.net](mailto:jan@kassens.net)
- <http://twitter.com/kassens>

## Warum ein Framework?

- Verbesserter Workflow
- Browserunabhängig



etc.

## Was ist MooTools?

- Moo.fx
  - minimale Effektbibliothek auf Prototype.js
- MooTools
  - Prototype.js ersetzt
  - Zusätzliche Features

## Ziele von MooTools

- Erweitert native Objekte von JavaScript
- Pseudo-klassisches Klassenmodell
- Einfache Erweiterbarkeit
- Modular
- Wiederverwendbarkeit
- Vollständige Basis für UI Entwicklung

## Feature Übersicht

- Programmier-Framework
- DOM-Helfer
- Ajax, JSON, Cookies
- Effekte
- Sortables, Drag'n'Drop, Slider, Accordion, ...

## Core: \$type

```
$type('foo');           // 'string'  
$type({});             // 'object'  
$type$(id);           // 'element'  
$type(function(){});  // 'function'  
$type(1);             // 'number'  
$type(/foo/);         // 'regexp'  
// etc.
```

```
String.type('foo');    // true  
Object.type('foo');   // false  
// usw.
```

## Core: \$extend

```
var events = {
  focus: fn1,
  blur: fn2
};

$extend(events, {
  mouseenter: fn3,
  mouseout: fn4
});

// Ergebnis:
events = {
  focus: fn1,
  blur: fn2,
  mouseenter: fn3,
  mouseout: fn4
};
```

## Core: \$merge

```
var cheapCars = {
  ford: {escort: 16000},
  vw: {golf: 14000}
};
var expansiveCars = {
  vw: {phaeton: 120000},
  porsche: {'911': 200000}
};
var cars = $merge(cheapCars, expansiveCars);

// Ergebnis:
cars = {
  ford: {escort: 8000},
  vw: {golf: 14000, phaeton: 120000},
  porsche: {'911': 200000}
};
```

## Natives

- Veränderung der Prototypen
  - `(5.6).round();`
- Generics:
  - `String.toUpperCase('foo');`
- Vorteile:
  - elegante, kompakte Syntax
  - wie die built-in Methoden
- Nachteil:
  - Inkompatibel mit anderen Frameworks, die die Prototypen verändern, insb. Prototypejs

## Native: String

```
' foo '.trim(); // 'foo'  
'foo'.capitalize(); // 'Foo'  
'1'.toInt(); // 1  
'1.11'.toFloat(); // 1.11  
'Ich mag {was}!'.substitute({was: 'Kekse'}); // 'Ich mag Kekse!'
```

## Native: Array

```
var arr = ['javascript', 'ruby', 'java'];
```

```
arr.each(function(what){  
    alert("I like " + what);  
});
```

```
arr.indexOf('ruby'); // 1  
arr.contains('php'); // false  
arr.include('php');  
arr.getRandom(); // javascript
```

```
var upper = arr.map(function(value, index){  
    return value.toUpperCase();  
});
```

```
var upper = arr.map(String.toUpperCase);  
// etc.
```

## Native: Hash

Niemals: Object.prototype verändern!

```
var hash = new Hash({
  'age': 21,
  'first': 'Jan',
  'last': 'Kassens'
});
hash.each(function(value, key){
  console.log(key, value);
});
```

```
Hash.each(obj, fn);
```

filter, some, every, map,

getKeys, getValues, toQueryString, ...

## Native: Erweitern

```
String.implement({  
  alert: function(){  
    alert(this);  
  }  
});
```

```
'hi'.alert();
```

### Ebenfalls Generics:

```
String.alert('hi');
```

## DOM: Finden

```
$(id);  
$$(selector);  
element.getElements(selector);  
...
```

```
$$('div');  
$$('.someClass');  
$$('ul > li');  
$$('div.someClass, span.someClass');  
$$('input[name="email"]');  
$$('img[src$=png]');  
nth-child, disabled, empty, contains...
```

## DOM: Erstellen

```
new Element('img', {src: 'foo.jpg'}).inject(target);
```

```
var myDiv = new Element('div', {  
  'class': 'foo',  
  styles: {'margin': 10, 'padding-bottom': 5},  
  events: {  
    click: function(event){  
      alert('click');  
    }  
  },  
  text: 'some Text'  
});
```

```
myDiv.replaces(myOtherElement);
```

## DOM: Eigenschaften

```
element.setStyle('margin', 10);
```

```
element.get('tag');
```

```
element.get('src');
```

```
element.set('href');
```

```
element.getSize().x;
```

```
element.getScroll().y;
```

```
element.hasClass('active');
```

```
element.addClass('active');
```

## DOM: Storage

Niemals Objekte in Elementen speichern.

Lösung:

```
$(id).store('key', value);
```

```
var sections = $$('div.section');  
var tabs = $$('div.tab');  
tabs.each(function(tab, index){  
    tab.store('section', sections[index]);  
});
```

```
// later...  
tab.retrieve('section');
```

## Effekte

- Fx
  - Fx.Tween, Fx.Morph, Fx.Scroll, ...
- Transitions
  - bounce:out, elastic:in:out
- Events
  - start, complete, cancel



## Request.HTML

- Request
  - Request, Request.HTML, Request.JSON
- Events
  - start, complete, success, failure, ...



## Class: Basics

```
var Animal = new Class({
  initialize: function(name){
    this.name = name;
  },

  eat: function(food){
    console.log(this.name + ' sagt: Mhh, ' + food);
  },

  toString: function(){
    return this.name;
  }
});
```

```
var tom = new Animal('Tom');
var jerry = new Animal('Jerry');
tom.eat(jerry); // Mhh, Jerry
```

## Class: Inheritance

```
var Cow = new Class({
  Extends: Animal,

  eat: function(food){
    if (food instanceof Animal){
      console.log("I don't eat animals.");
    } else {
      this.parent(food);
    }
  }
});
```

```
var erika = new Cow('Erika');
var berta = new Cow('Berta');
berta.eat(erika); // I don't eat animals.
berta.eat('Gras'); // Berta sagt: Mhh, Gras
```

## Class: Events

```
var Animal = new Class({
  Implements: Events,
  initialize: function(name){ this.name = name; },
  eat: function(food){
    console.log(this.name + ' sagt: Mhh, ' + food);
    this.fireEvent('eat', [food]);
  },
  toString: function(){ return this.name; }
});
```

```
var tom = new Animal('Tom');
tom.addEvent('eat', function(food){
  console.log(this.name + ' ate ' + food);
});
```

```
var jerry = new Animal('Jerry');
tom.eat(jerry); // Tom sagt: Mhh, Jerry
                // Tom ate Jerry
```

## Class: Options

- Mehr als 3 Argumente unübersichtlich.
- Options
  - optional
  - benannt
  - Reihenfolge beliebig

```
new User('James', {  
  admin: true,  
  url: 'http://james.example.com'  
});
```



## Class: Request.Form

- Strukturierung
- Wiederverwendbarkeit
- Erweiterbarkeit
- Wartbarkeit



<http://kassens.net>

<http://mootools.net>

<irc://irc.freenode.net/#mootools>

<http://groups.google.com/group/mootools-users>

<http://mootorial.com>

Buch:

MooTools Essentials

von Aaron Newton

im Apress Verlag

