jQuery 2013

bit.ly/devMedia - Interaktive Version der Präsentation!

JohannesHoppe.de

bit.ly/devMedia - Interaktive Version der Präsentation!

jQuery Core

getElementsByClassName

```
<div class="postit">
    <h2>Hello World</h2>
</div>

<script>
var divs = document.getElementsByClassName('postit');

for (i = 0; i < divs.length; i++) {
    divs[i].style.backgroundColor = '#F0F0A6';
    divs[i].style.padding = '10px';
}
</script>
```

» Demo

```
<script>
if (!document.getElementsByClassName) {
    document.getElementsByClassName = function (className) {
        return this.querySelectorAll("." + className);
    };
    Element.prototype.getElementsByClassName
        = document.getElementsByClassName;
}
</script>
```

» Demo

jQuery!

```
<script>
$(".postit").css({
    backgroundColor: '#F0F0A6',
    padding: '10px'
});
</script>
```

» Demo

The jQuery way

1. Select some HTML (Sizzle Selector Engine)
2. Do something with it

jQuery core

DOM manipulation ✓

event handling ✓

animations

AJAX ✓

plugins ✓ (next webinar)


```
$("#myId")         // selects (one) element with the id "myId"
$(".myClass")      // selects all elements with the class name "myClass"
$("div")           // selects all elements of type "div"
$("div:first")     // selects the first element of type "div"
$("div:odd")       // selects odd "div" elements, zero-indexed
$("input:not(:checked)") // equals to
$("input").not(":checked")
```

Selector Performance

Use IDs where possible
Avoid selecting by class only
KISS
Increase Specificity from Left to Right
Avoid touching the DOM

Do something with it

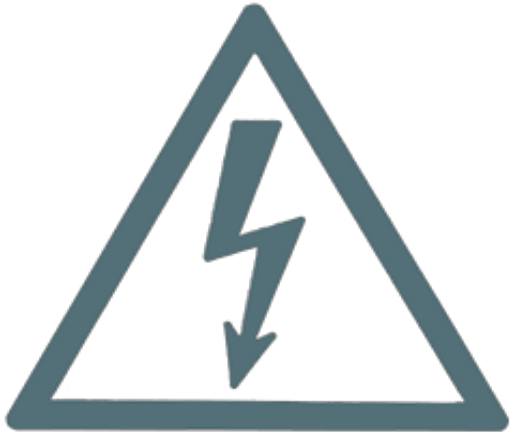
Manipulation of elements
Change of element attributes
Traversing
Events
Effects
AJAX

Example

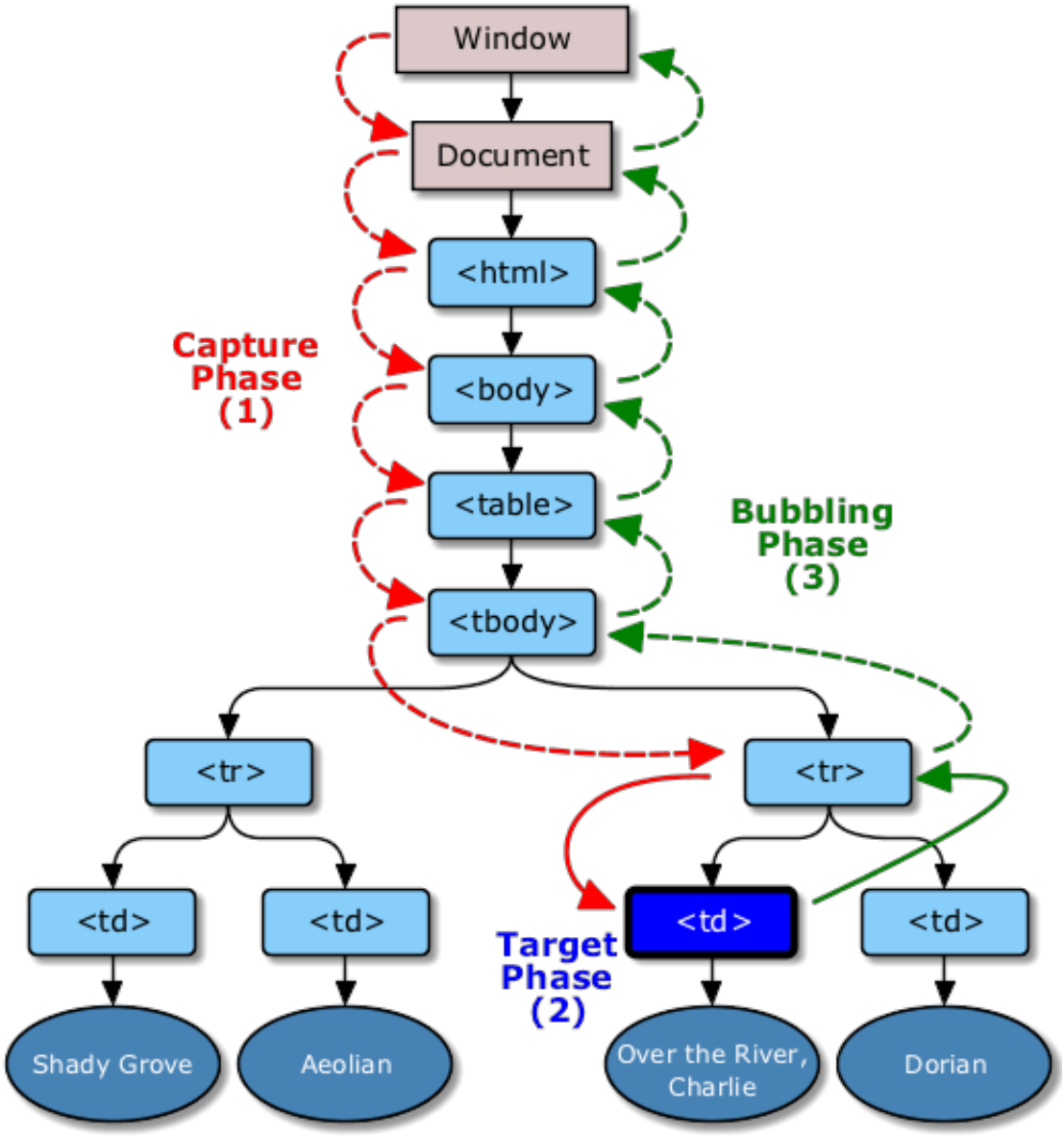
```
<script>
$('<h1>')
    .prependTo("body")
    .text(":-(")
    .css({
        position: "absolute",
        top: "50%",
        left: "50%",
        zIndex: "999"
    })
    .click(function () {
        $(this).fadeOut(function() {
            $(this).load("/examples/smilie.html").fadeIn();
        });
    });
</script>
```

» Demo

Event Model



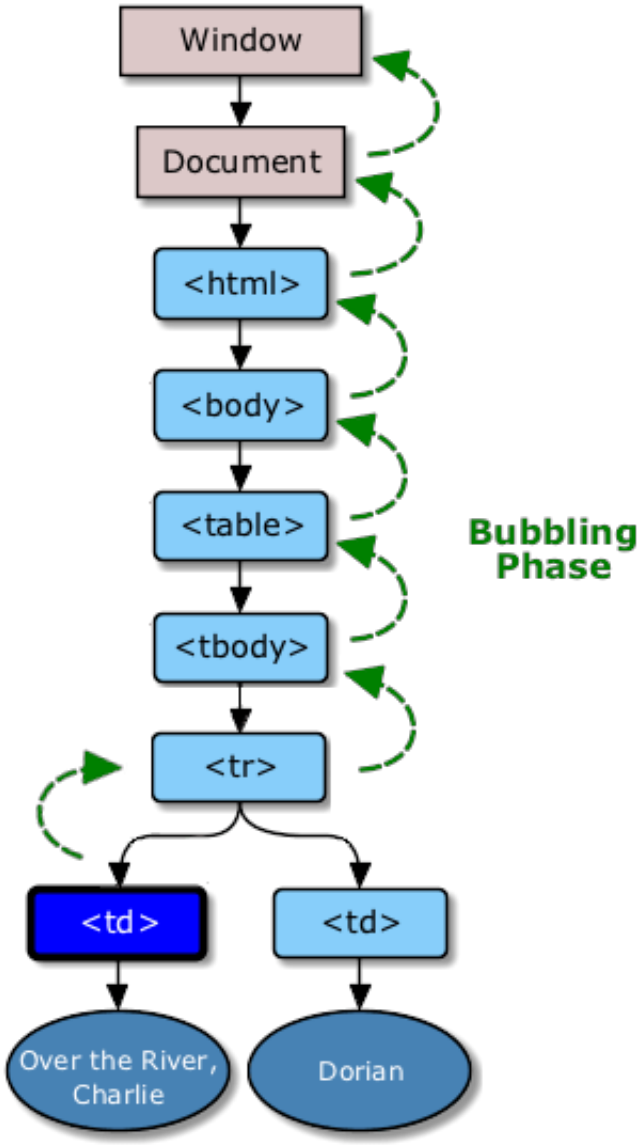
DOM Level 3 Event Model


Window
Document
<html>
<body>
<table>
<tbody>
<tr>
<tr>
<td>
<td>
<td>
<td>
Shady Grove
Aeolian
Over the River, Charlie
Dorian
Capture Phase (1)
Target Phase (2)
Bubbling Phase (3)

DOM Level 3 Event Model

```
document.getElementsByTagName('a')[0]
    .addEventListener("click", function() {
        alert(this.text);
});
```

jQuery Event Model


Window
Document
<html>
<body>
<table>
<tbody>
<tr>
<td>
<td>
Over the River, Charlie
Dorian
Bubbling Phase

jQuery Event Model

Demo Link

```
$('a').on('click', function() {
    alert(this.text);
});
```

```
$('a').click(function() {
    alert(this.text);
});
```

Some Ajax

```
$(function() {
    $('#example').dataTable({
        sAjaxSource: "/api/Note/"
    });
});
```

First try

```
$(function() {
    $('td').click(function() {
        alert($(this).text());
    });
});
```

Delegated events

```
$(function() {
    $('table').on('click', 'td', function() {
        alert($(this).text());
    });
});
```

API changes!

```
$(selector).live(events, data, handler);                // jQuery 1.3+
$(document).delegate(selector, events, data, handler); // jQuery 1.4.3+
$(document).on(events, selector, data, handler);        // jQuery 1.7+
```

Asynchronous Communication

before jQuery 1.5

```
$.ajax({
    url: "/examples/webinar.json",
    success: function(result) {
        $.each(result, function(index, value) {
          console.log(value.Title);
        });
    }
});
```

Deferred object

```
$.ajax("/examples/webinar.json")
.done(function(result) {
    $.each(result, function(i, value) {
        console.log(value.Title);
    });
});
```

Deferred object - Promisses

```
$.when(
    $.ajax("/examples/webinar.json"),
    $.ajax("/examples/webinar.json"))
.done(function(result1, result2) {
    var bothResults =  result1[0].concat(result2[0]);
    $.each(bothResults, function(i, value) {
        console.log(value.Title);
    });
});
```

Best Practices

Modul loaders

use AMD (require.js)

```
define('myFirstModule', ['jquery'], function() {

    return {
        saySomething : function() { alert("hello!"); }
    }
});

require(['myFirstModule'], function(t) {
    t.saySomething();
});
```

Own Events

Publish/Subscribe Pattern

```
var $events = $({});

$events.bind('somethingHappens', function() {
    alert("Something happened!");
});

$events.trigger('somethingHappens');
```

ASP.NET MVC

Bundling and Minification

Plugins

1. Utility functions

similar to global functions

```
(function($) {
    $.say = function(what) {
        alert('I say ' + what);
    };
})(window.jQuery);
```

2. Wrapper methods

operate on a jQuery wrapped set

```
(function($){
    $.fn.changeColor = function() {
        return this.css('color', 'green');
    };
})(window.jQuery);
```

Default options

```
(function($){
    $.fn.changeColor = function(options) {

        var settings = $.extend({
            color: "green"
        }, options );

        return this.css('color', settings.color);
    };
})(window.jQuery);
```

Chaining!

Always return this

```
(function($){
    $.fn.someNewMethod = function() {
        return this.each(function(){


        });
    };
})(jQuery);
```

TDD with Jasmine

Why Jasmine?

BDD-style similar to JSpec or RSpec,
created by authors of jsUnit and Screw.Unit

independent from any browser, DOM,
framework or host language

integrates into continuous build systems

Jasmine Bootstrap

```
<!DOCTYPE html>
<html>
<head>
    <title>Jasmine Spec Runner</title>

    <link rel="stylesheet" href="lib/jasmine-1.3.1/jasmine.css" />
    <script src="lib/jasmine-1.3.1/jasmine.js"></script>
    <script src="lib/jasmine-1.3.1/jasmine-html.js"></script>

    <!-- include source files here... -->
    <script src="src/Player.js"></script>
    <script src="src/Song.js"></script>

    <!-- include spec files here... -->
    <script src="spec/SpecHelper.js"></script>
    <script src="spec/PlayerSpec.js"></script>

    <script>

        (function () {

            var htmlReporter = new jasmine.HtmlReporter();
            var jasmineEnv = jasmine.getEnv();

            jasmineEnv.addReporter(htmlReporter);
            jasmineEnv.specFilter = function (spec) {
                return htmlReporter.specFilter(spec);
            };

            var currentWindowOnload = window.onload;

            window.onload = function () {
```

Output

```
Jasmine 1.3.1 revision 1354556913                    finished in 0.075s

• • • • •

Passing 5 specs                                      No try/catch ☐

Player
  should be able to play a Song

  when song has been paused
    should indicate that the song is currently paused
    should be possible to resume
  tells the current song if the user has made it a favorite

  #resume
    should throw an exception if song is already playing
```

Hello World

```
var helloWorld = function() {
    return "Hello World!";
};

describe('helloWorld', function() {
    it('says hello', function() {

        expect(helloWorld()).toEqual("Hello World!");
    });
});

jasmine.getEnv().execute();
```

hint: press F12 and paste this code!

Test-Driven Development

1. Write your tests
2. Watch them fail
3. Make them pass
4. Refactor
5. Repeat

see Growing Object-Oriented Software, Guided by Tests, page 6
see Working Effectively with Legacy Code, page 62 or many other

1. Write your tests

```
/// <reference path="../jquery-2.0.3.js" />
/// <reference path="saveFormat.js" />

describe("saveFormat_fail", function () {

    var original = '{0} - {1} - {2}';

    it("should replace placeholders", function () {
        var expected = 'boo!';
        var formated = $.saveFormat(original, 'A', 'B', 'C');
        expect(formated).toEqual(expected);
    });

    it("should encode injected content", function () {
        var expected = 'A - &lt;b&gt;TEST&lt;/b&gt; - C';
        var formated = $.saveFormat(original, 'A', '<b>TEST</b>', 'C'
);
        expect(formated).toEqual(expected);
```

2. Watch them fail

```
(function($){

    $.saveFormat = function () {
        return "boo!";
    };

})(window.jQuery);
```

jasmine.getEnv().execute(); Demo

3. Make them pass

```
(function ($) {

    var htmlEncode = function(input) {
        return ($('<div/>').text(input).html());
    };

    $.saveFormat = function () {

        var args = Array.prototype.slice.call(arguments);
        var txt = args.shift();

        $(arguments).each(function (i, item) {

            item = htmlEncode(item);
            txt = txt.replace("{" + (i - 1) + "}", item);

        });
        return txt;
```

jasmine.getEnv().execute(); Demo

4. Refactor

```
(function($) {

    var htmlEncode = function(input) {
        return ($('<div/>').text(input).html());
    };

    $.saveFormat = function (txt) {
        $.each(arguments, function (i, item) {
            if (i > 0) {
                item = htmlEncode(item);
                txt = txt.replace("{" + (i - 1) + "}", item);
            }
        });
        return txt;
    };
})(window.jQuery);
```

Demo

5. Repeat

```
(function($) {
    var htmlEncode = function(input) {
        return ($('<div/>').text(input).html());
    };

    $.saveFormat = function () {
        var args = Array.prototype.slice.call(arguments);
        var txt = args.shift();

        $.each(args, function (i, item) {
            item = htmlEncode(item);
            txt = txt.replace("{" + i + "}", item);
        });
        return txt;
    };
})(window.jQuery);
```

Demo

Testing HTML

Jasmine is DOM agnostic
comes without tools to set up HTML fixtures

Definition:
A test fixture is a fixed state of a set of
objects used as a baseline for running tests.

First Solution

in memory fixture with jQuery

```
describe('trivial jQuery plugin', function () {

    var fixture;
    beforeEach(function () {
        fixture = $('<div>some HTML code here</div>');
    });

    it('should do something', function () {
        fixture.myPlugin();
        expect(fixture).toHaveClass("newClass");
    });
});
jasmine.getEnv().execute();
```

... only works for trivial plugins!

Clumsy Solution

directly append to/remove from DOM

```
describe('my jQuery plugin', function () {

    beforeEach(function () {
        $('#fixture').remove();
        $('body').append('<div id="fixture">HTML</div>');
    });

    it('should do something', function () {
        $('#fixture').myPlugin();
        expect($('#fixture')).toHaveClass("newClass");
    });
});
jasmine.getEnv().execute();
```

jasmine-jquery

custom matchers, HTML/style/JSON fixtures, event spies

```
describe('my jQuery plugin', function () {

    beforeEach(function() {
        jasmine.getFixtures().fixturesPath='js/5_jasmine-demo_jquery';
        jasmine.getFixtures().load('jquery.myPlugin.spec.html');
    });

    it('should do something', function() {

        var $div = $('#helloWorld').myPlugin();
        expect($div).toHaveClass("newClass");
    });
});
jasmine.getEnv().execute();
```

Demo

Mobile Apps

Mobile Apps

1. Native apps
2. Mobile optimized Websites
3. Hybrid apps


Carrier
9:12 PM
Bootstrap-Ko
The following list was rendered with **jQuery** & **Knockout** and styled via the Slate **Bootstrap** theme.
Require.js & Prinzipien für modulares JavaScript - Teil 2
02.10.2013
Require.js & Prinzipien für modulares JavaScript - Teil 1
19.09.2013
jQuery 2013 - Teil 2
26.09.2013
jQuery 2013 - Teil 1
12.09.2013
ASP.NET Web API - Teil 2
27.08.2013
ASP.NET Web API - Teil 1
13.08.2013
Knockout.js für .NET
11.06.2013


Carrier
9:12 PM
Home
jQuery Mobile
The following list was rendered with jQM, a framework on top of jQuery.
Require.js & Prinzipien für modu…
Require.js & Prinzipien für modu…
jQuery 2013 - Teil 2
jQuery 2013 - Teil 1
ASP.NET Web API - Teil 2
ASP.NET Web API - Teil 1


Carrier
9:12 PM
Home
Kendo UI Mobile
The following list was rendered with Kendo UI Mobile, a MVVM framework on top of jQuery.
Require.js & Prinzipien für modulares JavaScript - Teil 2
Require.js & Prinzipien für modulares JavaScript - Teil 1
jQuery 2013 - Teil 2
jQuery 2013 - Teil 1
ASP.NET Web API - Teil 2
Home
Details

Downloads

bit.ly/devMedia
bit.ly/devMediaCode

02.10.

Webinar: require.js & modulares JS - Teil 2

50% Wiesn-Rabatt!

Code: Requirejs_JohannesHoppe

Danke!