Software Craftsmanship

bit.ly/jsCraftsmanship
Interaktive Version der Präsentation!

Created by Johannes Hoppe

JohannesHoppe.de

bit.ly/jsCraftsmanship
Interaktive Version der Präsentation!

Prinzipien

Wissen
Werkzeuge
Wiederholung

Wissen

Know the pitfalls

Implied globals

Forgetting var

```
var foo = function() {
    bar = 1;
};
```

Boolean type conversion

To Truthy or to Falsy. That is the only question!

```
var el = document.getElementById('does_not_exist');

if(el == false) {
    alert("shouldn't we see this message?!");
}
```

Trailing comma

works on my machine!

```
var foo = {
    bar: "bar",
    baz: "baz",
};
```

Return undefined

señor developers wear mustaches

```
var foo = function() {
    return
    {
        x : "looks like C# now!"
    };
}
```

Associative arrays

they don't exist

```
var x = [];
x['foo'] = "bar";
```

try .. catch .. finally

who cares about the reason?

```
var foo = function() {
    try {
       doCrazyStuff;
    } catch (e) {
        return false;
    }
    return true;
};
```

Hoisting

declare upfront all variables

```
var foo = "global";

var bar = function() {
   alert(foo);
   var foo = "local";
   alert(foo);
};
```

Eval

... and the job is done

```
function poorMansJsonParser(text) {
    return eval("(" + text + ")");
}

var text = ' { "hello" : "world" } ';
var json = poorMansJsonParser(text);
```

Eval is evil!

Never ever!

```
var text = ' function() { alert("hacked!"); })( ';
```

Globals

the mother of all antipatterns

```
function foo() {
    return "bar";
}

console.log(this['foo']());
```

Every time you clutter the global namespace,
somewhere in the world a helpless kitten dies!

Wissen

Pretty Code

Globals

reduce, minimize, delete or kill them

```
(function() { "wtf?" })();
```

The switch-case syndrome

a functional language wants functions!

```
switch (something) {

    case 1:
        doFirst();
        break;

    case 2:
        doSecond();
        break;

    case 3:
        doThird();
        break;
}
```

Lookup tables

avoid the switch-case syndrome

```
var methods = {
    1: doFirst,
    2: doSecond,
    3: doThird
};
if (methods[something]) {
    methods[something]();
}
```

Revealing Module Pattern

```
var myRevealingModule = function () {

    var _name = "Johannes";

    function greetings() {
        console.log("Hello " + _name);
    }

    function setName(name) {
        _name = name;
    }

    return {
        setName: setName,
        greetings: greetings
    };
}();
```

» Documentation

Modul loaders

use AMD (require.js)

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

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

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

Events

Publish/Subscribe Pattern

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

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

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

Werkzeuge

Visual Studio 2010/2012
JScript Editor Extensions
Resharper 7.1
JSHint
Chutzpah
Firebug / F12

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 0.05s

• • • • •

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!

Wiederholung

形

“If I would have had time...”

You will never have time!

Learn TDD
Improve by self reflection
Improve by feedback of others
Help others to improve
Teach TDD
Learn a new language

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 test

```
describe("saveFormat", function () {

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

    it("should replace placeholders", function () {
        var expected = 'A - B - C';
        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

```
var saveFormat = function() {
    return "boo!";
};
jasmine.getEnv().execute();
```

Demo

3. Make them pass

```
var saveFormat = function(txt) {

    $(arguments).each(function (i, item) {
        if (i > 0) {
            item = ($('<div/>').text(item).html());
            txt = txt.replace("{" + (i - 1) + "}", item);
        }
    });
    return txt;
};
jasmine.getEnv().execute();
```

Demo

4. Refactor

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

var saveFormat = function(txt) {

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

Demo

5. Repeat

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

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

Demo

Deep practice

codekata.pragprog.com
katas.softwarecraftsmanship.org

Danke!

blog.johanneshoppe.de