



# Zusammenfassung

Die Dynamik und Innovationsrate der JavaScript-Welt ist enorm: Grund genug für uns, einen Einstieg zu wagen. Wir zeigen in diesem Vortrag, wie wir als erfahrene Java-Entwickler erste Schritte mit den MEAN-Technologien machten: MongoDB, ExpressJs, AngularJs und NodeJs. Der Vortrag bietet einen Überblick über die Architektur einer MEAN-Applikation sowie über die eingesetzten Frameworks und Tools.

Zudem zeigen wir auf, wie wir unsere Einarbeitung in die JavaScript-Welt als Mini-Projekt vorbereitet und durchgeführt haben. Das Resultat findet sich unter [notes42.herokuapp.com](https://notes42.herokuapp.com), ein Prototyp einer "geeky" Web-App für Links und Notizen.

# Das Ziel unseres «Projekts»

## Mini-Projekt basierend auf MEAN Stack





**STOP**

**Bitte keine Werbung**

# Das Ziel des Vortrags

## Disclaimer

- Wir sehen uns nicht als JavaScript-Evangelisten und JavaScript nicht als «Silver Bullet». Wir denken aber, dass es sich auch für erfahrene JEE-Entwickler lohnt, sich dem JavaScript-Stack auseinanderzusetzen.
- Wir wollen in diesem Vortrag einen Überblick über die eingesetzten Sprachen, Frameworks und Tools geben. Wir zeigen auf, was uns speziell aufgefallen ist.
- Die nachfolgende Demo soll einen Eindruck geben, was wir im Rahmen unseres Weiterbildungs-Projektes entwickelt haben. Es soll keine Werbung für die Applikation sein: Das ist lediglich ein Prototyp.

# Die Idee







## «geeky» Web-App für Links, Notizen

Notes42 [all notes](#) [recently added notes](#) [export all notes](#) [all shared queries](#) Raimond Reichert ▾

? #programming

g+ id **programming** se

### Queried Notes (5)

Tags	Notes	
<a href="#">#g+</a> x + <a href="#">#id</a> x <a href="#">#programming</a> x	<a href="http://code.org/educate/20hr">code.org/educate/20hr</a> K-8 20 Lektionen Einführung ins Programmieren mit Blockly, teilweise in deutsch verfügbar (zB Videos mit Untertiteln). Je nach Altersstufe ("appropriate for kindergartners through 8th graders and beyond") dürfte es aber wohl durchaus etwas anspruchsvoller sein. G+	  
<a href="#">#id</a> x + <a href="#">#programming</a> x	<a href="http://henrikwarne.com/2012/06/02/why-i-love-coding">henrikwarne.com/2012/06/02/why-i-love-coding</a> Summary of Fred Brooks "The Joys of the Craft" in "The Mythical Man-Month": <ul style="list-style-type: none"><li>• The sheer joy of making things.</li><li>• The pleasure of making things that are useful to other people.</li><li>• The fascination of fashioning complex puzzle-like objects of interlocking moving parts, and watching them work in subtle cycles, playing out the consequences of principles built in from the beginning.</li><li>• The joy of always learning, which springs from the nonrepeating nature of the task.</li><li>• The delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of imagination. ... yet the program construct, unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separate from the construct itself</li></ul>	  







# Einfache Query: Notes zu Tag-Kombination

## ? #mean #grunt

? #mean #grunt|

grunt mean

### Queried Notes (2)

Tags	Notes	
#grunt x #mean x	<a href="http://www.npmjs.org/package/grunt-focus">www.npmjs.org/package/grunt-focus</a> um dedizierte Watches zu definieren	  
#grunt x #mean x	<a href="http://www.npmjs.org/package/load-grunt-config">www.npmjs.org/package/load-grunt-config</a> um Gruntfile.js aufzuteilen	  




# Modify: Note hinzufügen

#mean #grunt <http://gruntjs.com/plugins> Grunt Plugins

agile allegro angularjs **ausflug** autor bootstrap cacm coffeescript css eduself ergon familie foo g+ ganztage git grunt gutwetter halbtage html html5 id inbox java javascript kia kino **log** mean migros mongodb nadine&richi nodejs positioning presentations programming published quovadis regexp restaurant robotics schlechtwetter se testing thomas todo tools upcoming wandern windows

## Modified Notes (1)

Cool! We have added the following notes:

Tags	Notes	
#grunt x #mean x +	<a href="http://gruntjs.com/plugins">gruntjs.com/plugins</a> Grunt Plugins	  



# Piped Command «Export»: Notes exportieren

## ? #mean #grunt | export

```
? #mean #grunt | export|
```

grunt mean

### Exported Notes

```
#grunt #mean https://www.npmjs.org/package/grunt-focus um dedizierte Watches zu definieren  
#grunt #mean https://www.npmjs.org/package/load-grunt-config um Gruntfile.js aufzuteilen  
#grunt #mean http://gruntjs.com/plugins Grunt Plugins
```

# Piped Command «Share»: Notes veröffentlichen










## ? #mean #grunt |share

? #mean #grunt |share|

Share with this URL

grunt mean

### Queried Notes (3)

Tags	Notes	
#grunt x #mean x	<a href="http://www.npmjs.org/package/grunt-focus">www.npmjs.org/package/grunt-focus</a> um dedizierte Watches zu definieren	  
#grunt x #mean x	<a href="http://www.npmjs.org/package/load-grunt-config">www.npmjs.org/package/load-grunt-config</a> um Gruntfile.js aufzuteilen	  
#grunt x #mean x	<a href="http://gruntjs.com/plugins">gruntjs.com/plugins</a> Grunt Plugins	  

# Piped Command «Share»: Veröffentlichte Notes

notes42.herokuapp.com/#!/shared/4c54d060-34cc-4720-9424-80c24c15121d

Raimond Reichert shared 3 notes with you

Tags	Notes
#grunt #mean	<a href="http://www.npmjs.org/package/grunt-focus">www.npmjs.org/package/grunt-focus</a> um dedizierte Watches zu definieren
#grunt #mean	<a href="http://www.npmjs.org/package/load-grunt-config">www.npmjs.org/package/load-grunt-config</a> um Gruntfile.js aufzuteilen
#grunt #mean	<a href="http://gruntjs.com/plugins">gruntjs.com/plugins</a> Grunt Plugins

# Weitere Query Operatoren und Kombination von Queries und Piped Commands

Notes mit Tag «css» ausschliessen	? #mean -#css
Notes, die in den letzten 7h erstellt oder verändert wurden	? :7h
Sortieren nach zum Beispiel URL	?   sort by url asc
Sortiert exportieren	?   sort by created desc   export
Sortiert veröffentlichen	?   sort by urlVisitedCount desc   share

# Die Vorbereitung Planung

<b>April</b>	<b>Ideen prüfen, Stories schreiben</b>
<b>Mai</b>	<b>Lesen, lesen, lesen</b>
<b>Juni</b>	<b>Setup von VM sowie Entwicklungs- und Produktionsumgebung</b>
<b>Juli</b>	<b>10 Tage Projekt</b>
<b>Seit August</b>	<b>Sporadische Weiterentwicklung</b>

# Die Vorbereitung Stories für «minimal viable product» auf github.com

The screenshot shows the GitHub interface for the repository 'samzurcher / eduself'. The repository is private and has 2 watchers, 0 stars, and 0 forks. The 'Issues' tab is active, displaying a list of 10 open issues and 47 closed issues. The issues are sorted by creation date. Each issue includes a title, a description, labels (epic, links, ready, story), and a comment count.

Repository: samzurcher / eduself PRIVATE

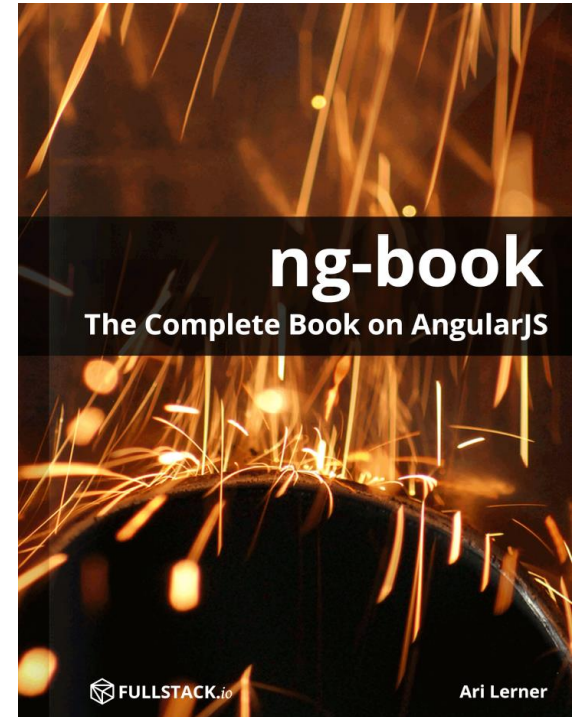
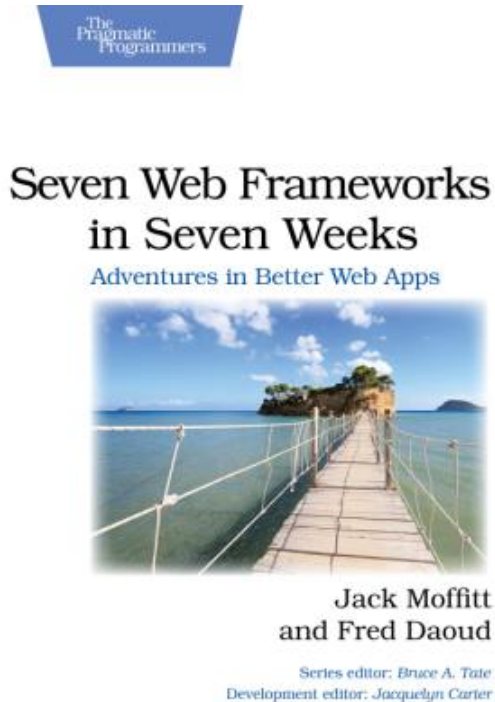
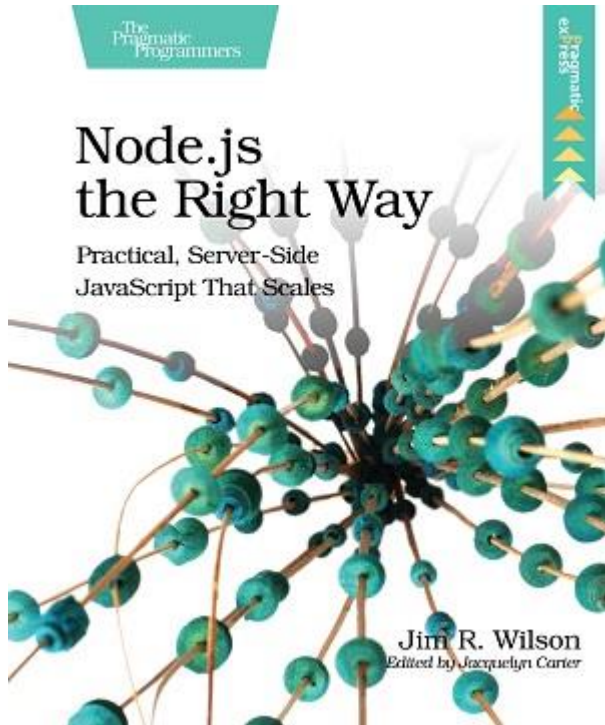
Unwatch 2 | Star 0 | Fork 0

Filters: is:issue is:closed sort:created-asc | New issue

Clear current search query, filters, and sorts

Issue ID	Title	Labels	Created	Assignee	Sort	Comments
#1	Einfache Applikation mit Google Authentisierung	epic: infrastructure, ready, story	opened on 30 Apr by raimondreichert	Sprint 1		19
#2	Setup DA project	epic: infrastructure, story	opened on 30 Apr by raimondreichert			1
#3	WebUI: Alle Tags anzeigen	epic: links, ready, story	opened on 30 Apr by raimondreichert	Sprint 3		1
#4	JSON API: Link erfassen	epic: links, ready, story	opened on 30 Apr by raimondreichert	Sprint 2		7
#5	WebUI: Link erfassen	epic: links, ready, story	opened on 30 Apr by raimondreichert	Sprint 2		4
#6	JSON API: Alle Tags abfragen	epic: links, ready, story	opened on 30 Apr by raimondreichert	Sprint 3		6

# Die Vorbereitung angularjs.org, nodejs.org, ...



# Die Vorbereitung

## Scripted Ubuntu VM Setup

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10  
echo 'deb http://downloads-distro.mongodb.org/repo/debian-sysvinit dist 10gen' |  
sudo tee /etc/apt/sources.list.d/mongodb.list  
sudo apt-get update  
sudo apt-get install mongodb-10gen
```

```
sudo apt-get update  
sudo apt-get -y install python-software-properties python g++ make  
sudo add-apt-repository ppa:chris-lea/node.js  
sudo apt-get update  
sudo apt-get -y install nodejs  
sudo npm install -g grunt-cli bower
```

# ...





## Die Vorbereitung MEAN-Projekt-Scaffolding mit mean.io

- > sudo npm install -g meanio@latest
- > mean init notes42
- > cd notes42 && npm install
- > grunt

... und schon hat man eine einfache CRUD-Applikation  
inklusive Authentifizierung.

*Inzwischen gibt es mit [meanjs.org](http://meanjs.org) einen Fork davon...*



# Die Vorbereitung Infrastruktur-Setup abgeschlossen

- ✓ «Setup VM und Entwicklungsumgebung»
- ✓ «Setup Projekt»
- ✓ «Einfache Applikation mit Google Authentisierung»
- ✓ «Setup Produktionsumgebung»

# Die Umsetzung 5 Sprints à 2 Tage

April	Ideen prüfen, Stories schreiben
Mai	Lesen, lesen, lesen
Juni	VM Setup, Development und Production Environment Setup
<b>Juli</b>	<b>10 Tage Projekt</b>
Seit August	Sporadische Weiterentwicklung

# Der Arbeitsplatz Setup für «Extreme pair programming»





# Komponenten der Applikation

## Kleine, fokussierte Plugins für so ziemlich alles



### «Client» AngularJs

notes42  
module



angular-markdown

angular-spinner

ng-table

lodash

«dependency  
management»  
**Bower**

### «Server» Nodejs

notes42  
module



express (request routing)

baucis (REST services)

passport (authentication)

lodash (utilities)

mongoose (db connector)

«dependency  
management»  
**Node  
Package  
Manager  
(NPM)**

«Data Base» **MongoDB**

# Produktionsumgebung: «git push heroku master»

«Runtime environment»



«Browser» Chrome

«Client» AngularJs

notes42  
module



angular-markdown

angular-spinner

ng-table

lodash

«Server» NodeJs

notes42  
module



express (request routing)

baucis (REST services)

passport (authentication)

lodash (utilities)

mongoose (db connector)

«automation»  
GruntJs

coffee

less

jshint

coffee\_jshint

cssmin

uglify

«Data Base» MongoDB







**GRUNT**

# Development GruntJs: CoffeeScript und LESS «on save» übersetzen

«Zero Turnaround» Live Reload

«Browser» Chrome

«Debugging» Batarang

«Client» AngularJs

notes42  
module



angular-markdown

angular-spinner

ng-table

lodash

«Server» NodeJs

notes42  
module



express (request routing)

baucis (REST services)

passport (authentication)

lodash (utilities)

mongoose (db connector)

«Data Base» MongoDB

«automation»  
GruntJs

coffee

less

jshint

coffee\_jshint





# Development

## GruntJs: Unit Tests «on save» ausführen

«Browser» Chrome

«Debugging» Batarang

«Client» AngularJs

notes42  
module



angular-markdown

angular-spinner

ng-table

lodash

«Server» NodeJs

notes42  
module



express (request routing)

baucis (REST services)

passport (authentication)

lodash (utilities)

mongoose (db connector)

«Data Base» MongoDB

«automation»  
GruntJs

mocha

jasmine

protractor

## Development

### JavaScript: Funktionaler Programmierstil

«functions are first class citizens»

Closures und (haufenweise) Callbacks zentral.

Gewöhnungsbedürftig: Kein Compiler, der vor Scope-Problemen in Closures warnt.

```
// returns list of books with at least 'threshold' copies sold
function bestSellingBooks(threshold) {
  return bookList.filter(
    function (book) { return book.sales >= threshold; }
  );
}
```

## Development

### JavaScript: Prototypen statt Klassen

```
// Prototyp-Definition
```

```
var Car = function Car (make, model, year) { // Konstruktor  
  this.make = make;  
  this.model = model;  
  this.year = year;  
}
```

```
Car.prototype.toString = function() { // Instanz-Methode  
  return this.make + ' ' + this.model + ' (' + this.year + ')';  
}
```

```
console.log(new Car("Eagle", "Talon TSi", 1993).toString());  
// prints "Eagle Talon Tsi (1193)"
```

## Development

### JavaScript: Prototypen statt Klassen

**// Beliebigen existierenden Objekt-Prototyp erweitern**

```
String.prototype.dasherize = function() {  
  return this.replace(/_/g, "-");  
};
```

```
console.log("no_more_underscores".dasherize());  
// prints "no-more-underscores"
```

## Development

### JavaScript: Duck Typing statt Interfaces

- JavaScript ist dynamisch typisiert:  
Duck Typing statt Interfaces
- Nachteile der dynamischen Typisierung:
  - Objekt-Datenstruktur ist im Wesentlichen ein assoziativer Array.
  - Automatisierte Refactoring trotz IDE-Unterstützung schwierig.

## Development

### JavaScript: Duck Typing statt Interfaces

- JavaScript ist dynamisch typisiert – Duck Typing statt Interfaces:  
«When I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck.»
- Gerade im Zusammenhang mit Callbacks angenehm, da man nicht überall Ein-Methoden-Interfaces definieren muss.
- Nachteile der dynamischen Typisierung:
  - Objekt-Datenstruktur ist im Wesentlichen ein assoziativer Array. Es gibt kein «Design by Contract» in Form von Klassen oder Interfaces, um Objekte verbindlich zu definieren.
  - Automatisierte Refactorings trotz IDE-Unterstützung schwierig.

## Development

### JavaScript: «Sicher» entwickeln

Software Engineering «Best Practices» werden zur zwingenden Voraussetzung, um in der dynamisch-typisierten JavaScript-Welt sicher entwickeln zu können.

- **Testing: Automatisierte Tests auf allen Ebenen: Unit, Integration, Application – gutes Tooling!**
- **Design: «Law of Demeter» einhalten: «Objekte interagieren nur mit ihrer unmittelbaren Umgebung»**
- **Static Analysis: Tools wie JsHint, «a program that flags suspicious usage in programs written in JavaScript»**

## Development

# CoffeeScript: «A little language that compiles into JS»

	CoffeeScript	JavaScript
Functions	<code>square = (x) -&gt; x * x</code>	<code>square = function(x) {   return x * x; };</code>
Array comprehension	<code>cubes = (math.cube num for num in list)</code>	<code>cubes = (function() {   var _i, _len, _results;   _results = [];   for (_i = 0, _len = list.length; _i &lt; _len;   _i++) {     num = list[_i];     _results.push(math.cube(num));   }   return _results; })();</code>



## Development

### CoffeeScript: «syntactic sugar» für OOP

```
class Animal
```

```
  constructor: (@name) ->
```

```
    move: (meters) ->
```

```
      alert @name + " moved #{meters}m."
```

```
class Snake extends Animal
```

```
  move: ->
```

```
    alert "Slithering..."
```

```
    super 5
```

```
sammy = new Snake "Sammy the Python"
```

```
sammy.move()
```

## Development

### CoffeeScript: Nachteile

- Build-Infrastruktur wird komplexer, da CoffeeScript kompiliert werden muss.
- JavaScript-Fehlermeldungen zur Laufzeit nicht immer einfach nach CoffeeScript rückzuübersetzen, im Zweifelsfall muss generiertes JavaScript angeschaut werden.
- Tools wie Coffee-JsHint haben noch Kinderkrankheiten wie unbrauchbare Fehlermeldungen.



# Development

## lodash: Unverzichtbare Utilities

`_removeTags: (command, tagArray) ->`  
`tagsToRemove = _.intersection command.excludedTags, tagArray`  
`_.xor tagArray, tagsToRemove`

`_addTags: (command, tagArray) ->`  
`_.union tagArray, command.tags`

`_getTagStringValue: (notes) ->`  
`tagStrings = _.map notes, (note) -> note.tagString`  
`sortedTagString = _.sortBy tagStrings, (tagString) -> tagString`  
`tagStringValue = _.reduce tagStrings, (result, value) -> result + ' ' + value`

# ... und vieles mehr!

## Development

# AngularJs : Data Binding – HTML + Expressions + Scopes

```
<html ng-app="phonecatApp"> <!-- head-Element weggelassen -->
<body ng-controller="PhoneListCtrl">
  <ul>
    <li ng-repeat="phone in phones">{{phone.name}}: {{phone.snippet}}</li>
  </ul>
</body>
</html>
```

```
var phonecatApp = angular.module('phonecatApp', []);
phonecatApp.controller('PhoneListCtrl', function ($scope) {
  $scope.phones = [
    {'name': 'Nexus S', 'snippet': 'Fast just got faster with Nexus S.'},
    {'name': 'Motorola XOOM', 'snippet': 'The Next, Next Generation tablet.'} ];
});
```

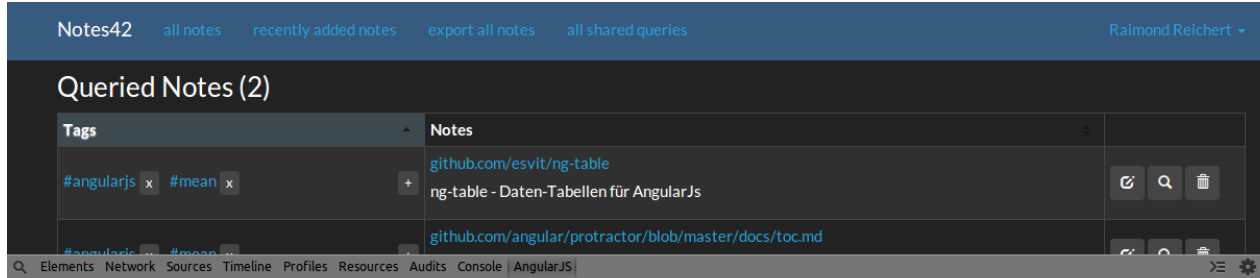
## Development

### AngularJs : Testing

```
describe 'PhoneListCtrl', () ->
  it 'should create "phones" model with 2 phones', () ->
    scope = {}
    ctrl = new PhoneListCtrl scope
    expect(scope.phones.length).toBe 2
```

# Development

## AngularJs : Hierarchische Scopes – Debugging in Chrome



Models Performance Dependencies Options Help  Enable

### Scopes

```
< Scope (001)
  < Scope (002)
    < Scope (004)
      < Scope (005)
        < Scope (006)
      < Scope (007)
        < Scope (008)
          < Scope (009)
            < Scope (00A)
              < Scope (00B)
                < Scope (00F)
                  < Scope (00G)
                    < Scope (00H)
                      < Scope (00I)
                        < Scope (00C)
                          < Scope (00J)
                            < Scope (00K)
                              < Scope (010)
                                < Scope (01F)
                                  < Scope (01G)
                                    < Scope (01H)
                                      < Scope (01I)
                                        < Scope (01J)
                                  < Scope (00N)
                                    < Scope (00S)
                                      < Scope (00Y)
                                        < Scope (00T)
```

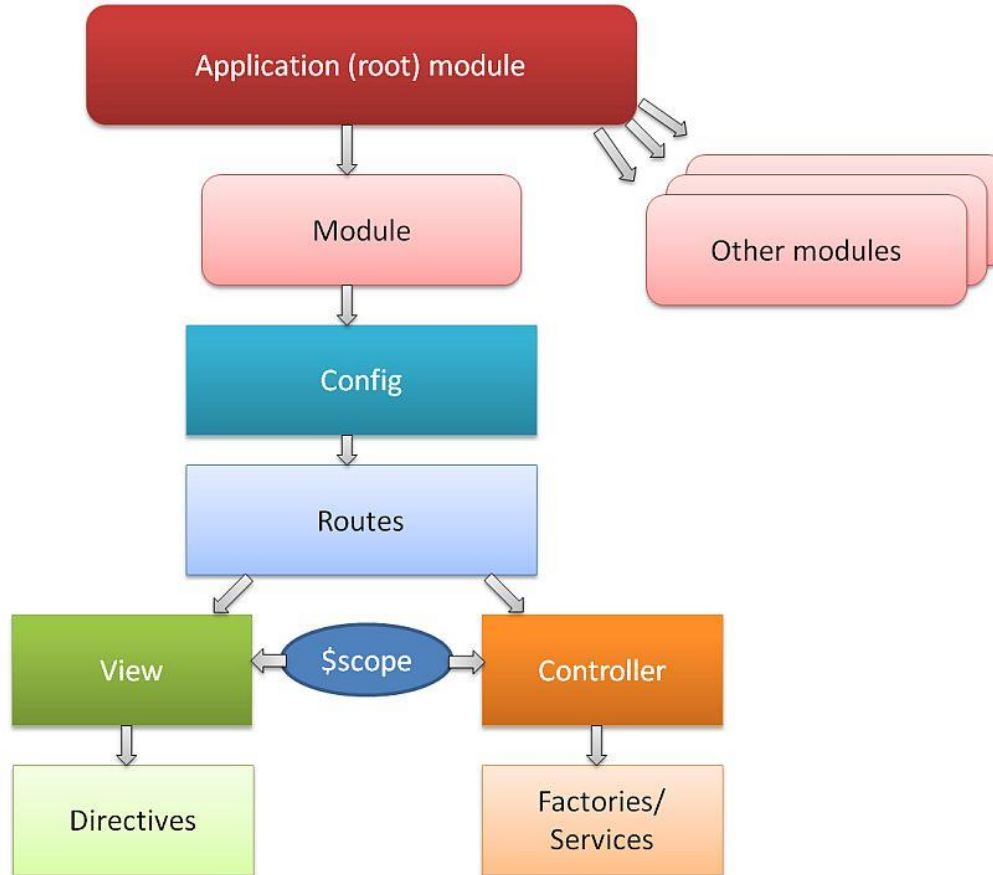
### Models for (008)

```
{
  global: {
    user: {
      name: Raimond Reichert
      _id: 53b460ec3db20d0800001f80
      username: raimond.reichert@gmail.com
      roles:
        [ authenticated ]
    }
    authenticated: 1
    isAdmin: 0
  }
  command: {
    state: notInitial
    value: ? #angularjs
    error:
  }
  guiMode: {
    value: expert
  }
  result: {
    value: {
      commandMode: query
      onlyQueryCommand: {
        mode: query
        tags:
          [ angularjs ]
        excludedTags:

```

# Development

## Angularjs : Gibt App-Architektur vor – flache Lernkurve



### Konzepte

- Template
- Directive
- Model
- Scope
- Expression
- Compiler
- Filter
- View
- Data Binding
- Controller
- Dependency Injection
- Module
- Service

## Development

# AngularJs: Google als Sponsor, starke Community

### Google als Sponsor

“Misko Hevery [...] managed to recreate a web application that consisted of 17 thousand lines of code and took 6 months to develop in a mere 3 weeks using just GetAngular. Reducing the size of the application to just about 1,000 lines of code convinced Google to start sponsoring the project, turning it into the open-source AngularJS we know today.”

### Starke Community

AngularJs hat über 800 Module auf [ngmodules.org](http://ngmodules.org).

[www.airpair.com/js/javascript-framework-comparison](http://www.airpair.com/js/javascript-framework-comparison)



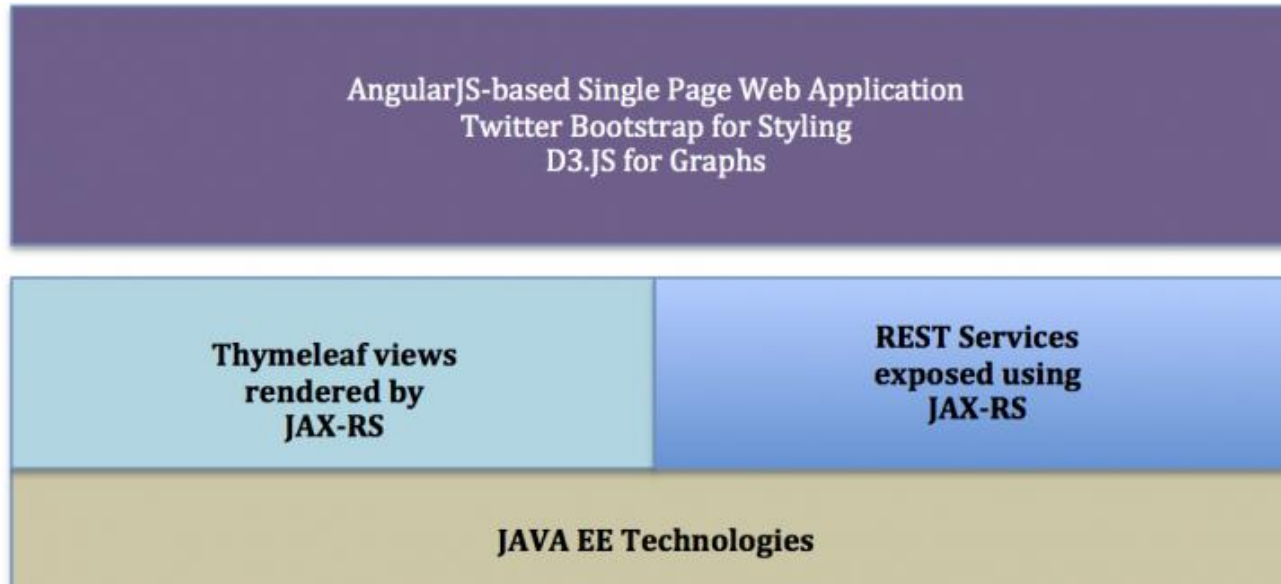
# Development AngularJs: Verbreitung

Interest over time. Web Search. Worldwide, 2004 - present.



## Development

# AngularJS: «Java EE 7 Real World Experience»



...

JEE7 – AngularJS als empfohlene Frontend-Technologie:

[blog.arungupta.me/2014/09/](http://blog.arungupta.me/2014/09/)

[log-your-miles-and-community-runs-java-ee-7-real-world-experience/](http://log-your-miles-and-community-runs-java-ee-7-real-world-experience/)

## Development

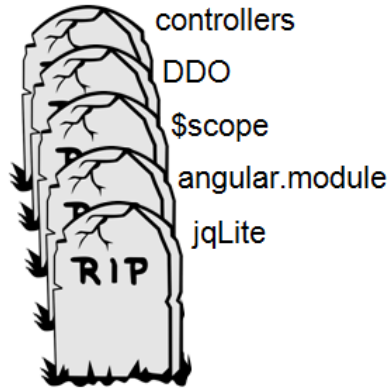
# AngularJs: P.S. Browser Kompatibilität...

Unterstützt: Safari, Chrome, Firefox, Opera, IE9/10/11 und Mobile Browser (Android, Chrome Mobile, iOS Safari).

Nicht unterstützt: IE8. Da Microsoft im April 2014 offiziell den Support für Windows XP einstellte und Internet Explorer 8 vor allem darauf lief, unterstützt AngularJs ab Version 1.3 IE 8 nicht mehr.

# AngularJs 2

## Grundlegende Änderungen – kompletter Rewrite von 1.3

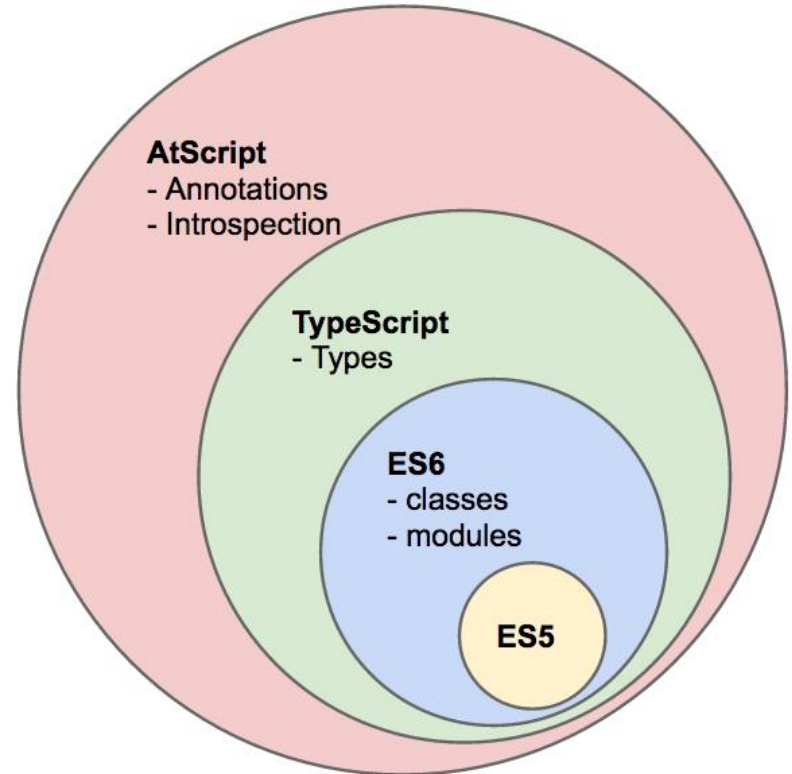


+ generic binding syntax

+ DI query mechanism

+ benchpress

+ WTF instrumentation





# Development

## Nodejs: Angenehm steile Lernkurve

### example.js

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```

### > node example.js

Server running at http://127.0.0.1:8124/



## Development

### NodeJs + ExpressJs: Routing einfach gemacht

```
var express = require('express');
```

```
var app = express();
```

```
app.get('/hello.txt', function(req, res) {  
  res.send('Hello World');  
});
```

```
var server = app.listen(3000, function() {  
  console.log('Listening on port %d', server.address().port);  
});
```

## Development

# NodeJs + ExpressJs: Server-side Rendering mit Jade

```
app.engine('jade', require('jade').__express);  
app.get('/', function (req, res) {  
  res.render('index', {  
    title: 'Hey',  
    message: 'Hello there!'  
  });  
})
```

*index.jade*

```
html  
  head  
    title!= title  
  body  
    h1!= message
```

## Development

### NodeJS: Testing

```
describe 'CommandNoteMapper', () ->
  it 'no change for empty modify command', () ->
    command = new Command Command.MODIFY
    note = new Note()
    mapper.update command, note
    note.isModified().should.be.false
```





## Development

# Nodejs: Leichtgewichtiger Container, einfach erweiterbar

- Im Vergleich zu JEE-Containern
  - ist Node **extrem leichtgewichtig** und startet in Sekundenbruchteilen,
  - gibt es keine Standards, aber eine sehr aktive Community.
- In Nodejs sind nur eine **überschaubare Anzahl APIs integriert**. Das wiederum hält die Lernkurve steil.
  - Für die Entwicklung von Nodejs-Webapplikationen kommt sehr oft auch **Expressjs** (Web-App-Framework) zum Einsatz.
- Mit dem **Node Package Manager**. Via **www.npmjs.org** steht eine riesige Anzahl von Modulen zur Auswahl (Stand Sept 2014: 92'000).
  - “The **ecosystem** around JavaScript as a serious application platform continues to evolve. Many interesting new tools for testing, building, and managing dependencies in both server- and client-side JavaScript applications have emerged recently.” (Thoughtworks Tech Radar 2014)

## Development

# Node.js: Sinnvolle Szenarien für Einsatz von Node.js

“Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.“ (nodejs.org)

Gut geeignet für ...	Nicht geeignet für ...
<ul style="list-style-type: none"><li>▪ Single-Page-Applikationen Backends / JSON-basierte REST-Services</li><li>▪ Web-Realtime-Applikationen</li><li>▪ Streaming</li></ul>	<ul style="list-style-type: none"><li>▪ rechenintensive Anwendungen</li></ul>

[heise.de/developer/artikel/2x-Nein-4x-Ja-Szenarien-fuer-Node-js-2111050.html](https://heise.de/developer/artikel/2x-Nein-4x-Ja-Szenarien-fuer-Node-js-2111050.html)



## Development

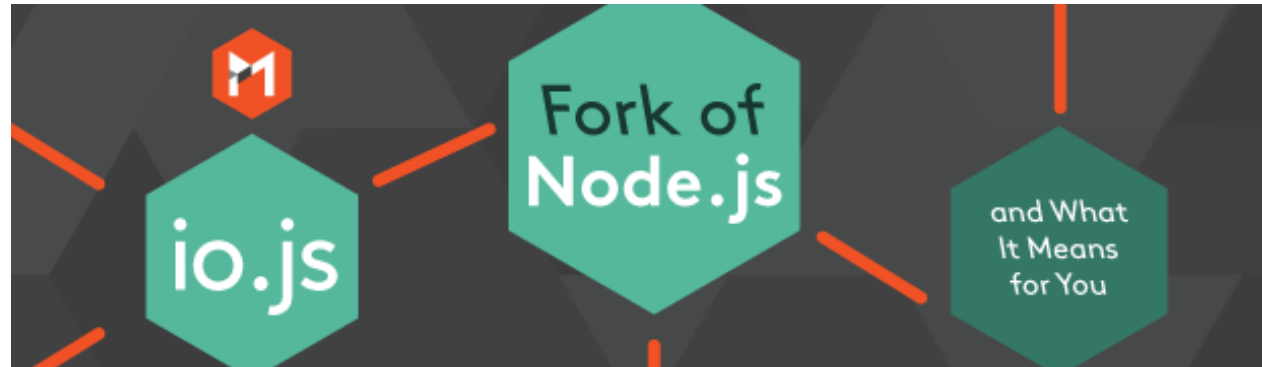
# Nodejs: Produktiv einsetzbar und eingesetzt

- **LinkedIn** hat von Ruby on Rails auf Nodejs umgestellt, siehe etwa [cacm.acm.org/magazines/2014/2/171684-node-at-linkedin](http://cacm.acm.org/magazines/2014/2/171684-node-at-linkedin).
  - Nodejs als Server-seitiger Integrationslayer für diverse Backend-Systeme, die grösstenteils in JEE entwickelt wurden.
  - Beeindruckende Performance: «we discovered that Node was roughly 20 times faster than what we had been using and its memory footprint was smaller as well.»
- Microsoft, VMWare, Ebay, Yahoo und viele andere setzen Nodejs ein.

## Development

### Node.js: Fork io.js – praktische Relevanz noch unklar

- io.js is a collaborative fork of joyent/node. io.js is created by Fedor Indutny, a long-time active node core team member responsible for some of the most important parts of the Node.js runtime.
- io.js continues the work that was previously being done by Node Forward in the node-forward/node repository. We hope to merge with the original Node.js project at some point in the future.
- <http://blog.izs.me/post/104685388058/io-js>



## Development

# MongoDB: Dokument-basierte Datenbank

*Eigenwerbung von mongodb.org*

MongoDB (from "humongous") is an open-source document database, and the leading NoSQL database. Written in C++, MongoDB features:

- **Document-Oriented Storage:** JSON-style documents with dynamic schemas offer simplicity and power.
- **Full Index Support:** Index on any attribute, just like you're used to.
- **Querying:** Rich, document-based queries.
- Map/Reduce, GridFs und vieles mehr

## Development

### mongoosejs: «elegant mongodb object modeling»

```
mongoose = require 'mongoose'
```

```
Schema = mongoose.Schema
```

```
timestamps = require 'mongoose-  
times'
```

```
NoteSchema = new Schema {
```

```
  text: String
```

```
  tagArray: [
```

```
    type: String
```

```
    index: true
```

```
  ]
```

```
  tagString:
```

```
    type: String
```

```
    default: ""
```

```
    trim: true
```

```
  url:
```

```
    type: String
```

```
    default: ""
```

```
    index: true
```

```
  ownedBy:
```

```
    type: Schema.Types.ObjectId
```

```
    ref: 'User'
```

```
    index: true
```

```
  }
```

```
NoteSchema.plugin timestamps
```

## Development

# MongoDb + mongoosejs: Aggregation Pipeline-Beispiel

```
class TagsFinder
```

```
  # constructor not shown
```

```
  find: (command, successCallback, failureCallback) ->
```

```
    Note.aggregate([
```

```
      {$match: @matchConditionFactory.queryMatchCondition(command)}
```

```
      {$unwind: '$tagArray'}
```

```
      {$group: {_id: '$tagArray', number: {$sum: 1}}}
```

```
      {$sort: {_id: 1}}
```

```
    ]).exec (err, tags) ->
```

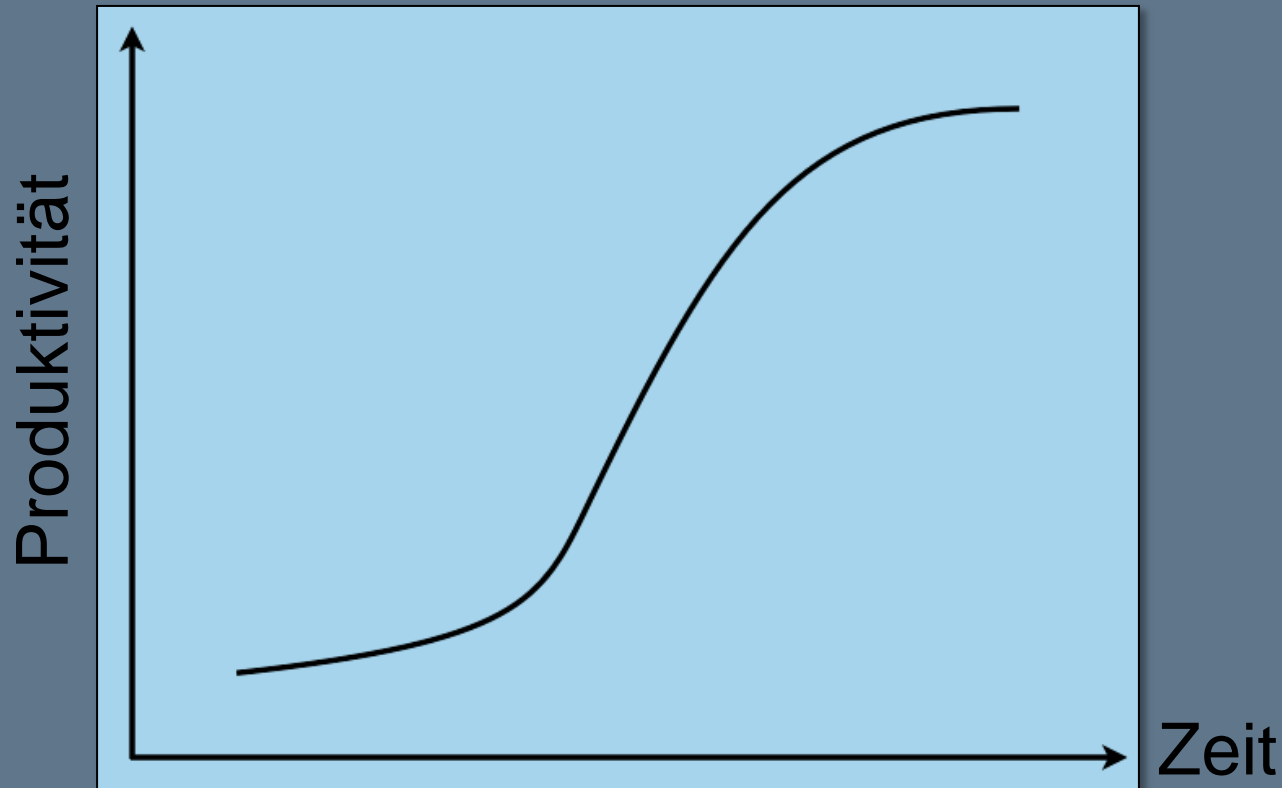
```
      mappedTags = _.map tags, (tag) -> { value: tag._id, count: tag.number }
```

```
      successCallback mappedTags
```

```
  # error handling not shown
```

# Der Durchbruch

Nach 7 Tagen gefühlt sehr produktiv





# Der Rückblick



## Lessons Learned

### Methodisch: Weiterbildung als Projekt

- Weiterbildung als **Projekt** sorgt für Motivation und Fokus
- **Vorbereitung**, um «produktiv» entwickeln zu können
- Durchführung in «**Extrem pair programming**»-Modus

# Lessons Learned

## Methodisch

- **Weiterbildung als Projekt sorgt für Motivation und Fokus**
  - Der Lerneffekt beim Entwickeln einer Applikation ist deutlich höher als beim Lesen von Büchern
  - Geeignet für «Autodidakten» oder mit entsprechender Anleitung durch Coach
  - Im Idealfall entsteht eine Applikation der Kategorie «eat your own dogfood»
  - Eine der grössten Herausforderungen war, eine geeignete Idee zu entwickeln
- **Vorbereitung notwendig, um «produktiv» entwickeln zu können**
  - Unverzichtbar, um in der kurzen Zeit produktiv entwickeln zu können
  - Zeit für Infrastruktur-Aufbau: Der Aufwand zahlt sich rasch wieder aus
  - Einlesen in die Technologien: Spezialisierung hilfreich
  - Stories auf Github halfen, Projekt zu strukturieren und uns zu fokussieren
- **Durchführung in «Extrem pair programming»-Modus**
  - «Extrem pair programming»: Deutlich effizienter gelernt als alleine
  - Kurze Feedback-Zyklen: abends «lessons learned» notieren, morgens umsetzen
  - Time boxing: Manchmal müssen pragmatische Hacks genügen
  - «Continuous Deployment»: Von Anfang an regelmässige Verifikation auf herokuapp.com
  - Das Erfolgserlebnis: Nach 7 Tagen auch individuell produktiv

# Lessons Learned

## Was noch fehlt für «produktives Projekt»

### ▪ **Continuous Integration**

- Zum Beispiel mit Jenkins: [blog.dylants.com/2013/06/21/jenkins-and-node](http://blog.dylants.com/2013/06/21/jenkins-and-node)
- Strider ([stridercd.com](http://stridercd.com)) für Selbstbetrieb; auch als Hosted Service zu haben
- Travis-Ci ([travis-ci.org](http://travis-ci.org)) als Hosted Service für Github-Projekte
- CodeShip ([www.codeship.io](http://www.codeship.io)) als Hosted Service

### ▪ **Betrieb von NodeJs-Server**

- Einzelne NodeJs-Server: Sehr einfach mit Forever ([github.com/nodejitsu/forever](https://github.com/nodejitsu/forever))
- Clusters zum Beispiel mit Strongloop ([strongloop.com/node-js/controller](http://strongloop.com/node-js/controller))

# Lessons Learned

## Was noch fehlt für «produktives Projekt»

- **Code Coverage Tools**

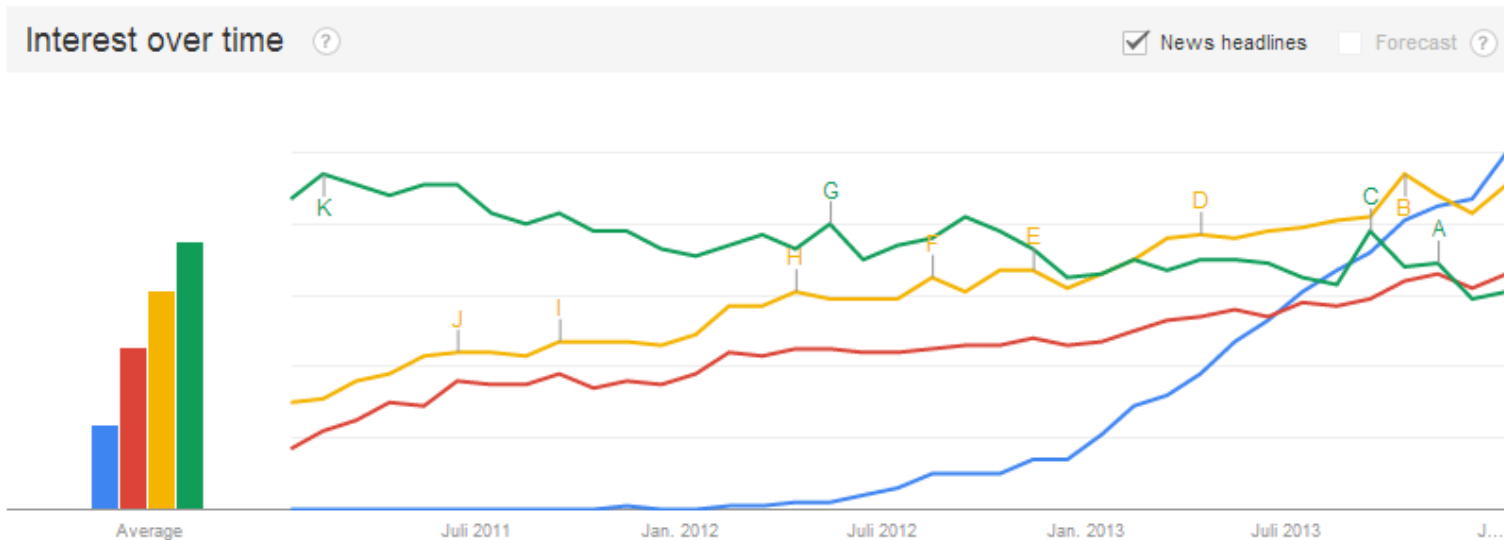
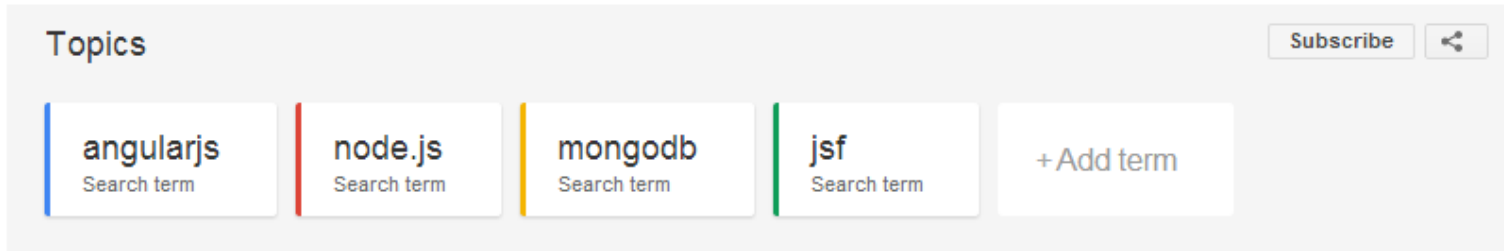
- Istanbul ([github.com/gotwarlost/istanbul](https://github.com/gotwarlost/istanbul)) wird von Karma Test Runner verwendet
- BlanketJs ([blanketjs.org](https://blanketjs.org))
- JsCover ([tntim96.github.io/JScover](https://tntim96.github.io/JScover))

- **Performanz-Analyse-Tools**

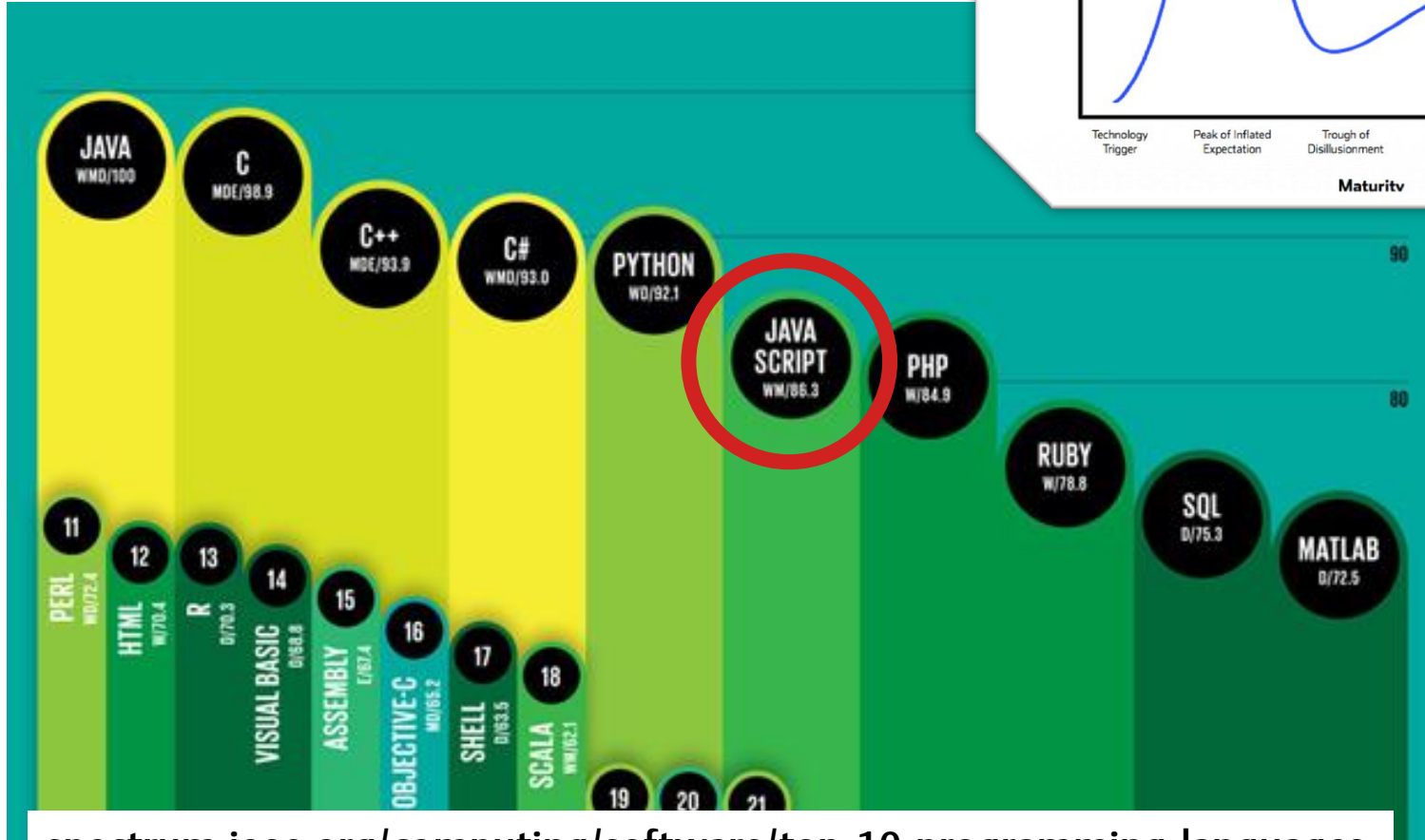
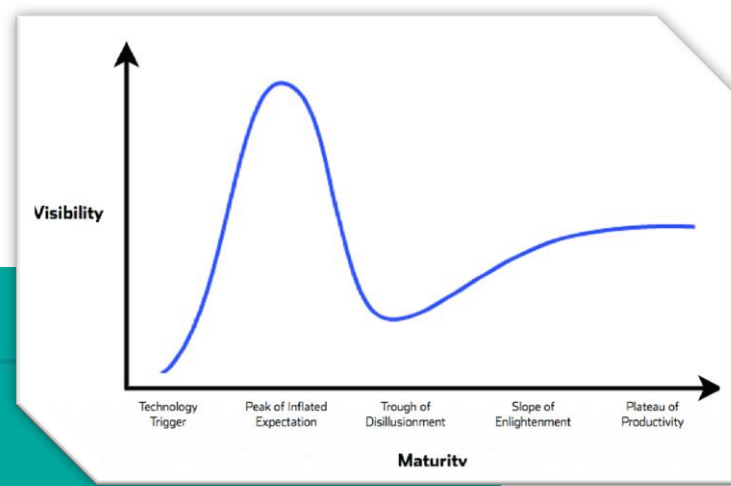
- Strongloop ([strongloop.com/node-js/monitoring/](https://strongloop.com/node-js/monitoring/))
- Weitere Möglichkeiten siehe [www.clock.co.uk/blog/easy-cpu-profiling-in-node-js](http://www.clock.co.uk/blog/easy-cpu-profiling-in-node-js)

# Lessons Learned

## JavaScript macht Spass und ist voll im Trend ...



# Lessons Learned ... der noch sehr jung ist!

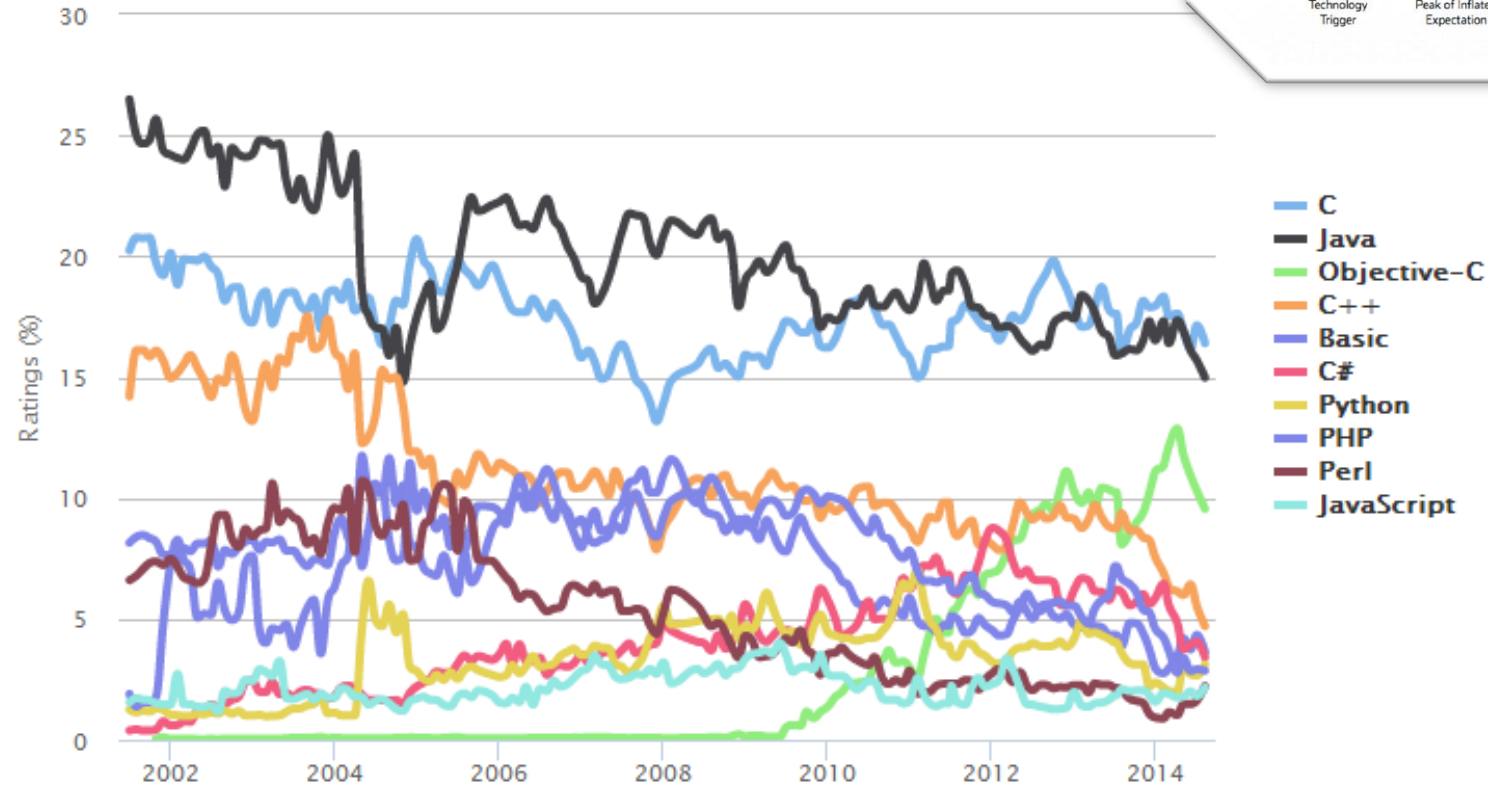
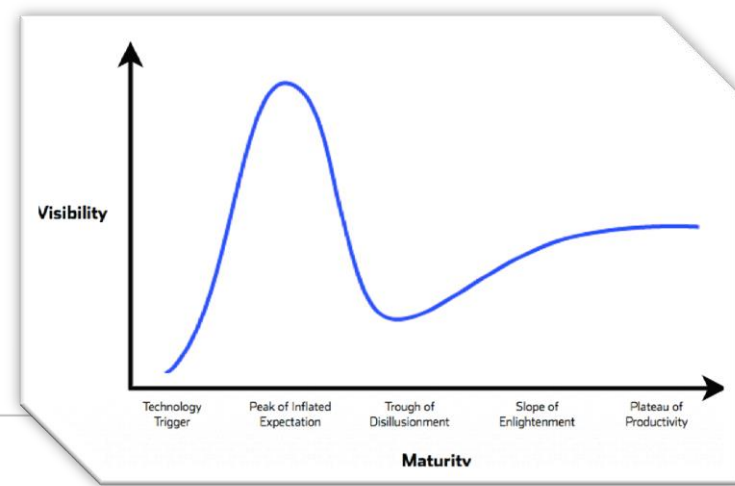


[spectrum.ieee.org/computing/software/top-10-programming-languages](http://spectrum.ieee.org/computing/software/top-10-programming-languages)

# Lessons Learned ... der noch sehr jung ist!

## TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)





**notes42.herokuapp.com**

**«notes for two» oder «notes fortytwo»?**

**Raimond Reichert – raimond@ergon.ch**

**Sam Zürcher – sam@ergon.ch**

**Ergon Informatik AG**

## Development

### JavaScript: Aber was ist mit der Wartung?

Stefan Hanenberg et al. **An empirical study on the impact of static typing on software maintainability**. Empirical Software Engineering, Oct. 2014, Vol. 19, No. 5, pp 1335-1382.

- Static type systems help use a new set of classes – an effective form of documentation
- **We believe the most important result is that the static type systems showed a clear tendency in class identification tasks, and that we found a first indicator that this is caused by a reduced navigation effort.**
- Static type systems make it easier for humans and reduce the effort to fix type errors.
- For fixing semantic errors, we observed no difference with respect to development times, and static type systems may not be helpful in preventing semantic errors.

Development

JavaScript: Best Practices noch relevanter

## 急がばまわれ

Isogaba maware – «Wenn Du es eilig hast, mach einen Umweg»

- Testing: Automatisierte Tests auf allen Ebenen
- Sauberes Design: «Law of Demeter» einhalten
- Abhängigkeiten: Explizit deklarieren
- Dokumentation: APIs sauber dokumentieren
- Continuous Integration, ...

# Development

## AngularJS: Aber die Performanz...?

### Is AngularJS Fast Enough?

1. Is it possible to get a near instantaneous initial page load?  
**Yes, use server pre-rendering.**
2. Do simple pages operate within the near instantaneous bucket without much effort? **Yes.**
3. When performance issues do come up, is it possible to optimize any issues down to at most a small perceptible delay? **Yes.** These issues do not come up often, they are coming up even less frequently as more optimizations are added to Angular and there are a number of ways to resolve any issue that does come up.
4. Does your app perform on mobile devices? **Yes, use Ionic.**

**So, is AngularJS fast enough? Hell, yes.**

[medium.com/@jeffwhelpley/is-angularjs-fast-enough-98dcf96406c8](https://medium.com/@jeffwhelpley/is-angularjs-fast-enough-98dcf96406c8)

# Ideen? Nehmen wir gerne entgegen!



**Raimond Reichert**  
**raimond@ergon.ch**

**Sam Zürcher**  
**sam@ergon.ch**

**Ergon Informatik AG**