

Lecture 1: Getting started...

Olivier Liechti
TWEB



Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Today's agenda

15:45 - 16:05	20'	Intro
16:05 - 16:25	20'	JavaScript 101
16:25 - 16:45	20'	Guidelines for phase 1
16:45 - 17:55	70'	Group in pairs
17:55 - 18:05	10'	Wrap-up



Introduction

Objectives

This course is **not only about coding.**

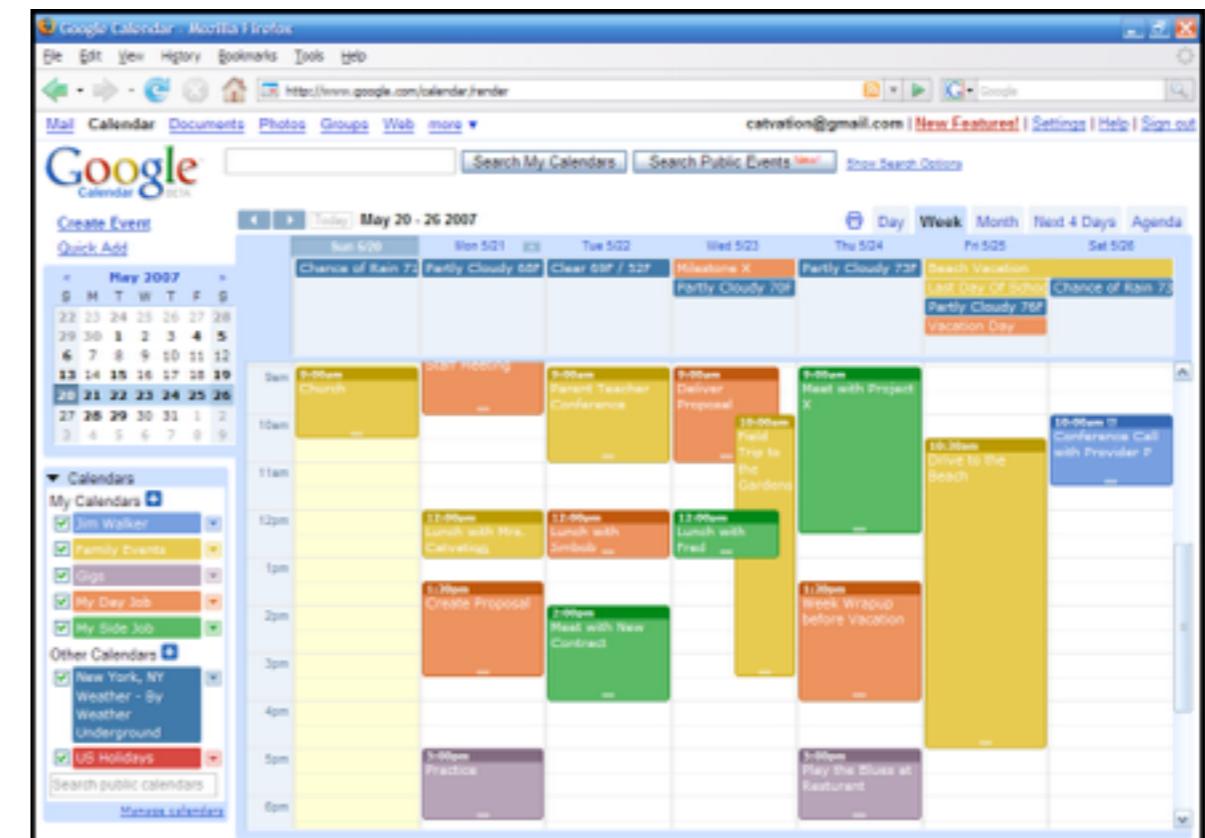
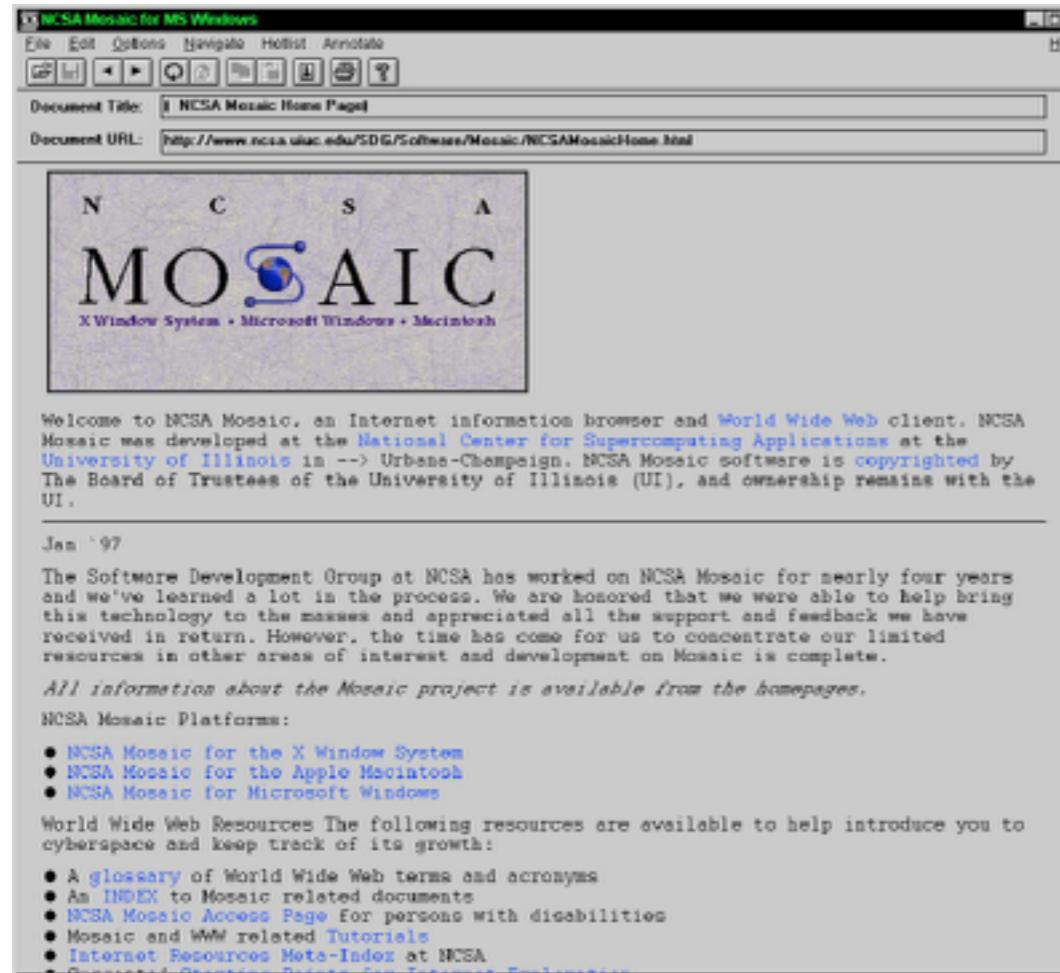
This course is also about designing **usable**,
engaging web apps.

It is also about **advertising** these apps to an
external audience.

The Web as an Application Platform

heig-vd

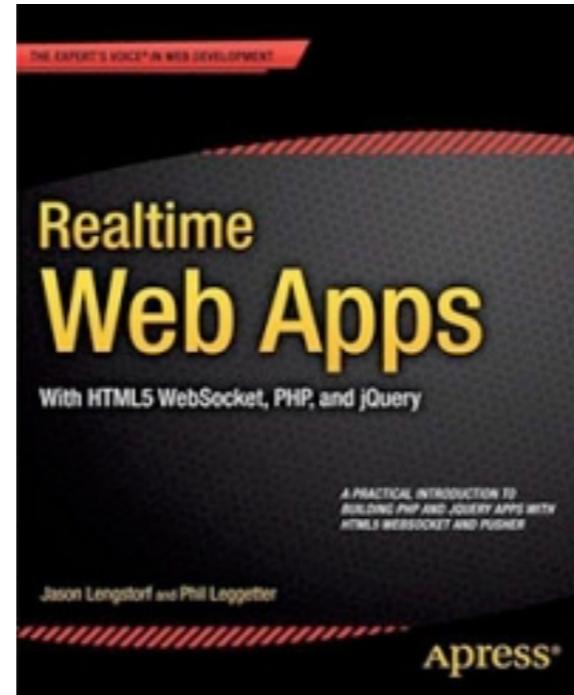
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



Some Trends



SPA



Server Push



Automated Development Workflows



Mobile & RWD



Web of Things

Languages, Platforms, Communities

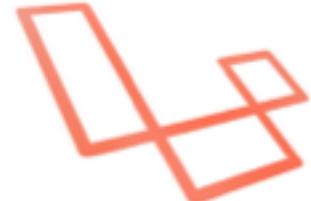


Client

Server



Languages, Platforms, Communities



laravel



JavaScript - End to End



Client

express



node.js™

socket
IO

Server

Pour ce projet, vous travaillez en équipes de 2 étudiant(e)s



	19.09.2016	Bootcamp Javascript 101, squelette e2e, heroku	As a user, I can visit the product landing page, transition to the web app and back to the landing page.
	26.09.2016	Introduction à AngularJS Coding guidelines, controllers, services, directives	As a user, I can navigate between several pages. As a user, I can see a graph (pie chart)
	03.10.2016	Programmation asynchrone Callbacks, promesses, async.js	As a user, I can give provide a GitHub repo, user or organization and see "some" analysis (TBD)
	10.10.2016	Persistence NoSQL MongoDB, mongoose	As a user, I can see the history of my previous queries (towards GitHub) As a user, I can select one query in the history and see the results again (data comes from DB)
	17.10.2016	Applications interactives Socket.IO	As a user, I can see the history of my previous queries (towards GitHub)

Vendredi 28.10.2016 (23h59)

Remise projet "Mining data in GitHub"

Mercredi 02.11.2016
Travail écrit 1

	31.10.2016	Itération 1 Sondages interactifs simples Implémentation de la fonctionnalité de base	As a presenter, I can ask a question to the audience As an auditor, I can answer the questions As a participant, I can see the answers of the audience
	07.11.2016		
	14.11.2016	Itération 2 Gestion des comptes Authentification et autorisation	As a guest, I can register and create an account As a registered user, I can log in and out of my account As a registered user, I have access to specific features (TBD)
	21.11.2016		



	28.11.2016	"Excursion" Analyse de données en JavaScript Introduction à Elasticsearch, D3.js et dockerode	As an app, I can send a stream of user events via the /events endpoint As an app owner, I can define a rule to award a badge when some conditions are met (via the /rules endpoint)
	05.12.2016		

Vendredi 09.12.2016 (23h59)

Remise "Interactive polls" 1

Lundi 16.01.2017
Travail écrit 2
Vendredi 20.01 (23h59)
Remise finale "Interactive polls"

	12.12.2016	Itération 4 Fonctionnalités "avant le cours" Que peut-on faire avant une session?	TBD
	19.12.2016		
	09.01.2016	Itération 5 Fonctionnalités "après le cours" Que peut-on faire après une session?	TBD
	16.01.2016		

Lundi 23.01 et mercredi 25.01
Présentations & démos

	23.01.2016	Wrap-up
--	------------	----------------

Project 1: Mining GitHub data

- **GitHub is a social and collaborative space, with a rich domain model:**
 - Organizations, users, repositories
 - Commits, tags, releases
 - Issues, pull requests, comments
- **GitHub provides a REST API to access this sea of data:**
 - Example 1: GET the repositories from an organization
 - Example 2: GET the contributions made by users on a repo
- **Developers can use the API to extract information and answer questions:**
 - Which are the main contributors on a given project?
 - Are there repositories in the organization which are slowly dying?

Commits | GitHub Developer X Olivier

https://developer.github.com/v3/repos/commits/ ⌂

GitHub Developer API Blog Early Access Support Search...

API Reference Webhooks Guides Libraries

API

Commits

- i. [List commits on a repository](#)
- ii. [Get a single commit](#)
- iii. [Get the SHA-1 of a commit reference](#)
- iv. [Compare two commits](#)
- v. [Commit signature verification](#)

The Repo Commits API supports listing, viewing, and comparing commits in a repository.

List commits on a repository ⓘ

```
GET /repos/:owner/:repo/commits
```

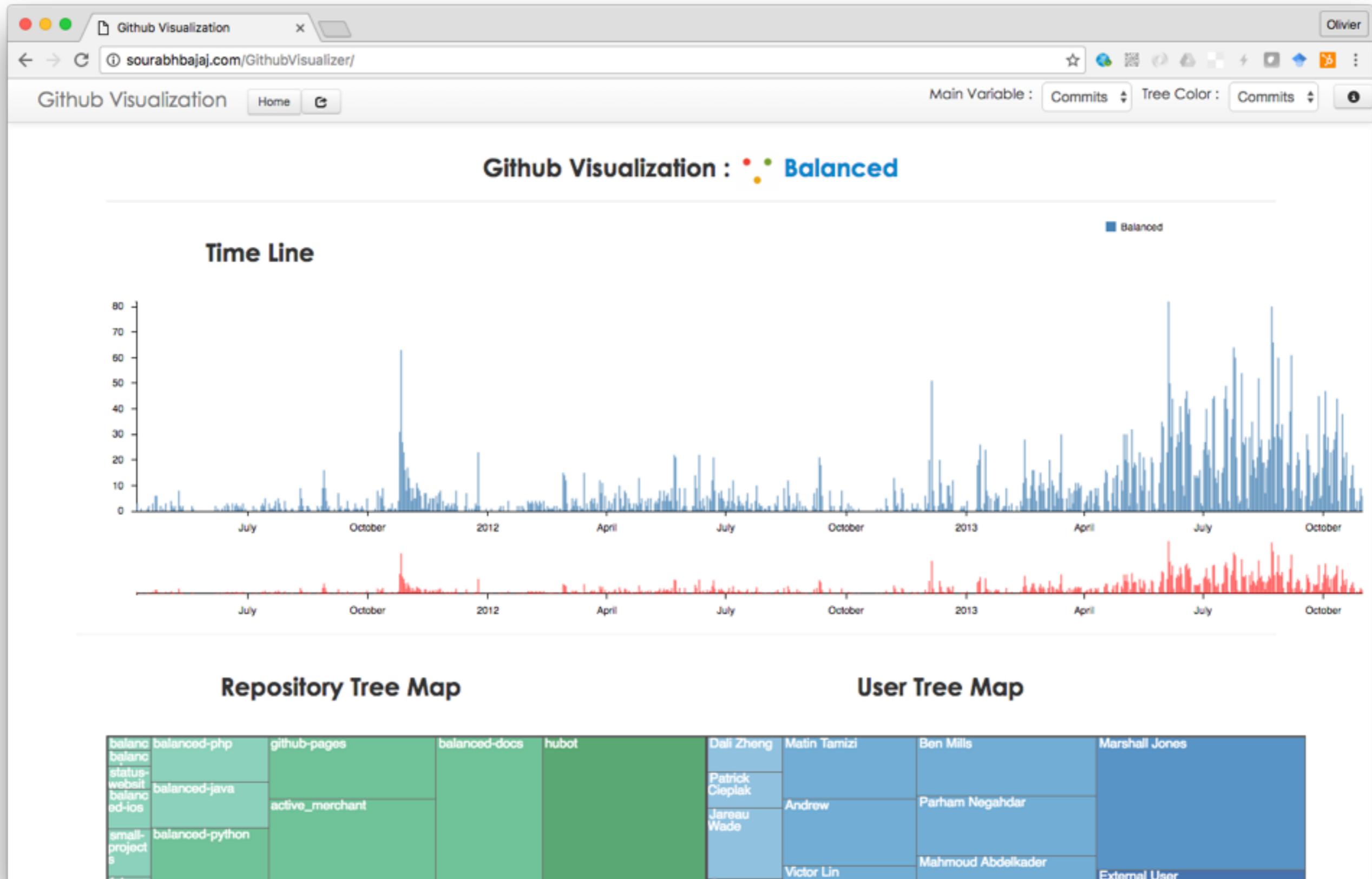
Parameters

Name	Type	Description
sha	string	SHA or branch to start listing commits from. Default: the repository's default branch (usually <code>master</code>).
path	string	Only commits containing this file path will be returned.
author	string	GitHub login or email address by which to filter by commit author.
since	string	Only commits after this date will be returned. This is a timestamp in

- ▶ Overview
- ▶ Activity
- ▶ Gists
- ▶ Git Data
- ▶ Issues
- ▶ Migration
- ▶ Miscellaneous
- ▶ Organizations
- ▶ Pull Requests
- ▶ Reactions
- ▼ Repositories
 - Branches
 - Collaborators
 - Comments
 - Commits (selected)
 - Contents
 - Deploy Keys

Project 1: Mining GitHub data

- You will write a multi-tiered application that:
 - Makes API calls to GitHub to extract data
 - Does some processing and store information in a NoSQL database
 - Presents a UI to the users, so that they can explore/query the information
 - Creates a visualization of the information
- You should start with a simple question and a simple visualization (to validate the e2e process). That will be the focus of the 5-weeks “bootcamp”.
- When that works, you will think of more interesting and challenging questions and visualizations. That will be the focus of the 2-weeks “excursion”.



Project 2: Interactive polls

- As a professor, I would like to have an app that I can use to ask questions (polls) to my audience.
- Students should have an easy way to answer the questions.
- The audience should see the results in realtime.
- Beyond the basics (what happens “live”), what could be offered to professors and students, both before and after the lecture?
- You will be asked to:
 - specify the product roadmap (what you would like to offer to users in the long term)
 - design the UX (mockups that documents the interaction between the users and the app)
 - implement a first set of features (focus on quality and not on quantity).

SpeakUp

speakup.info Olivier

SpeakUp

Let your audience anonymously share and rate each other's questions to answer the best ones.

Available on the iPhone App Store

Available on the Web

Facebook Twitter Google+

Today's agenda

15:45 - 16:05	20'	Intro
16:05 - 16:25	20'	JavaScript 101 (part 1)
16:25 - 16:45	20'	Guidelines for phase 1
16:45 - 17:55	70'	Group in pairs
17:55 - 18:05	10'	Wrap-up

JS

JavaScript 101 (Part 1)

ECMAScript 6? TypeScript?

Initially, we will stick to ES 5.



<http://es6-features.org/>

JavaScript 101, Part 1

- Types
- Scopes
- Objects
- Prototypal inheritance
- Functions
- Constructors
- Arrays

Rule #1

JavaScript defines 6 types

```
var aNumber = 3.12;
var aBoolean = true;
var aString = "HEIG-VD";
var anObject = {
  aProperty: null
};

// t is true for all of these:
var t;
t = typeof aNumber === "number";
t = typeof aBoolean === "boolean";
t = typeof aString === "string";
t = typeof anObject === "object";
t = typeof anObject.aProperty ===
  "object";
t = typeof anObject.foobar ===
  "undefined";
```

- The 6 types are:
 - number
 - boolean
 - string
 - object
 - undefined
 - null
- null is a type, but `typeof null === object`.
- JavaScript is a dynamic language: when you declare a variable, you don't specify a type (and the type can change over time).

Rule #2

There are 2 scopes for variables:

the (evil) global scope and the function scope

```
var aVariableInGlobalScope;

function myFunction() {
    var aVariableInFunctionScope;
    anotherVariableInGlobalScope;
}

function myFunction2() {
    for (i=0; i<10; i++) {
        //i is in global scope!
    }
    for (var j=0; j<10; j++) {
        //j is in function scope!
    }
}
```

- A variable declared within a function is **not accessible** outside this function.
- Unless using **strict mode**, it is not mandatory to declare variables (beware of typos...)
- Two scripts loaded from the same HTML page share the same global scope (beware of **conflicts**...).
- There is **no block scope** (well, that's not really true anymore since ES6)

Rule #3

Objects are dynamic bags of properties

```
// let's create an object
var person = {
  firstName: 'olivier',
  lastName: 'liechti'
};

// we can dynamically add properties
person.gender = 'male';
person['zip'] = 1446;

// and delete them
delete person.zip;

for (var key in person) {
  console.log(key + " : " +
  person[key]);
};
```

- There are different ways to **access properties** of an object.
- JavaScript is **dynamic**: it is possible to **add** and **remove** properties to an object at any time.
- Every object has a different list of properties (**no class**).

Rule #4

The language has no support for classes

There are 3 ways to create objects

```
// create an object with a literal
var person = {
  firstName: 'olivier',
  lastName: 'liechti'
};

// create an object with a prototype
var child = Object.create(person);

// create an object with a constructor
var child = new Person('olivier',
  'liechti');
```

- **class** is a reserved word in JavaScript, but it is not used in the current version of the language (reserved for the future).
- A **constructor** is function like any other (uppercase is a coding convention).
- It is the use of the **new** keyword that triggers the object creation process.

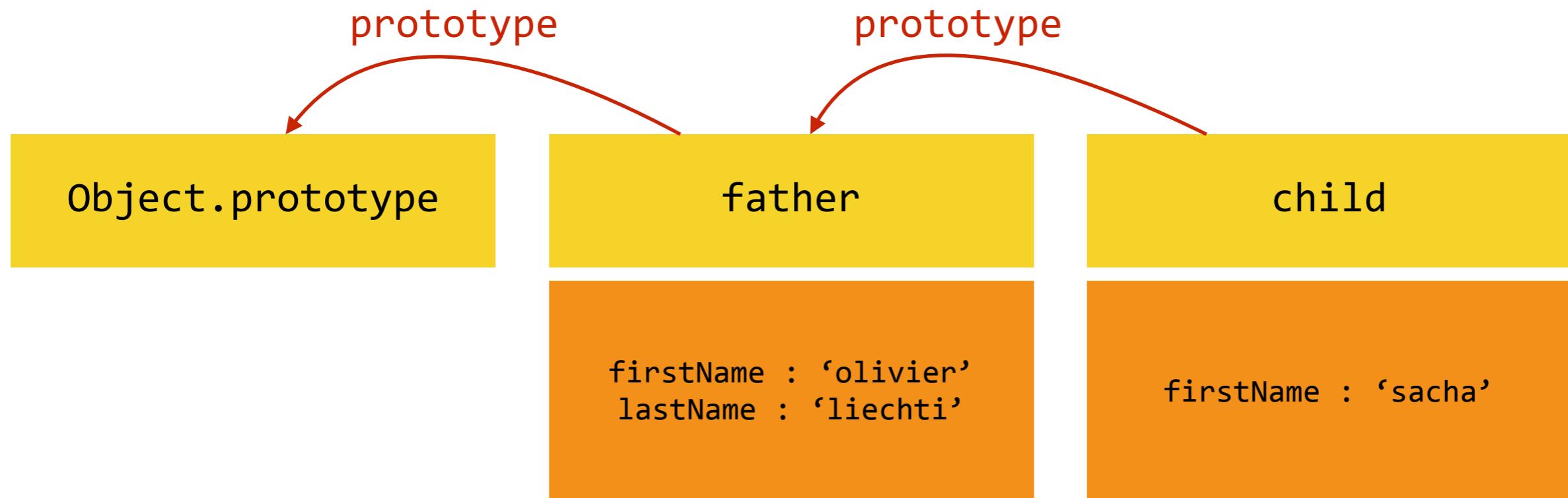
Rule #5

Every object inherits from a prototype object

```
var person = {  
    firstName: "olivier",  
    lastName: "liechti"  
};  
// person's prototype is Object.prototype  
  
var father = {};  
var child = Object.create(father);  
// child's prototype is father  
  
function Person(fn, ln) {  
    this.firstName = fn;  
    this.lastName = ln;  
}  
var john = new Person("John", "Doe");  
// john's prototype is Person.prototype
```

Rule #5

Every object inherits from a prototype object



```
console.log(child.lastName);
// prints 'liechti' on the
console
```

- Every object inherits from a prototype object. **It inherits and can override its properties**, including its methods.
- Objects created with object literals inherit from **Object.prototype**.
- When you access the property of an object, JavaScript **looks up the prototype chain** until it finds an ancestor that has a value for this property.

Rule #6

With patterns, it is possible to implement class-like data structures

```
function Person(fn, ln) {  
    var privateVar;  
    this.firstName = fn;  
    this.lastName = ln;  
    this.badGreet = function() {  
        console.log("Hi " + this.firstName);  
    };  
};  
  
Person.prototype.greet = function() {  
    console.log("Hey " + this.firstName);  
};  
  
var p1 = new Person("olivier", "liechti");  
  
p1.badGreet();  
p1.greet();
```

- **badGreet** is a property that will be replicated for every object created with the Person constructor:
 - poor memory management
 - not possible to alter behavior of all instances at once
- **greet** is a property that will be shared by all instances (because it will be looked up along the object inheritance chain).
- **privateVar** is not accessible outside of the constructor.
- **firstName** is publicly accessible (no encapsulation).

Rule #7

Arrays are objects

```
var fruits = ["apple", "pear"];

fruits.push("banana");
console.log(Object.getPrototypeOf(fruits));

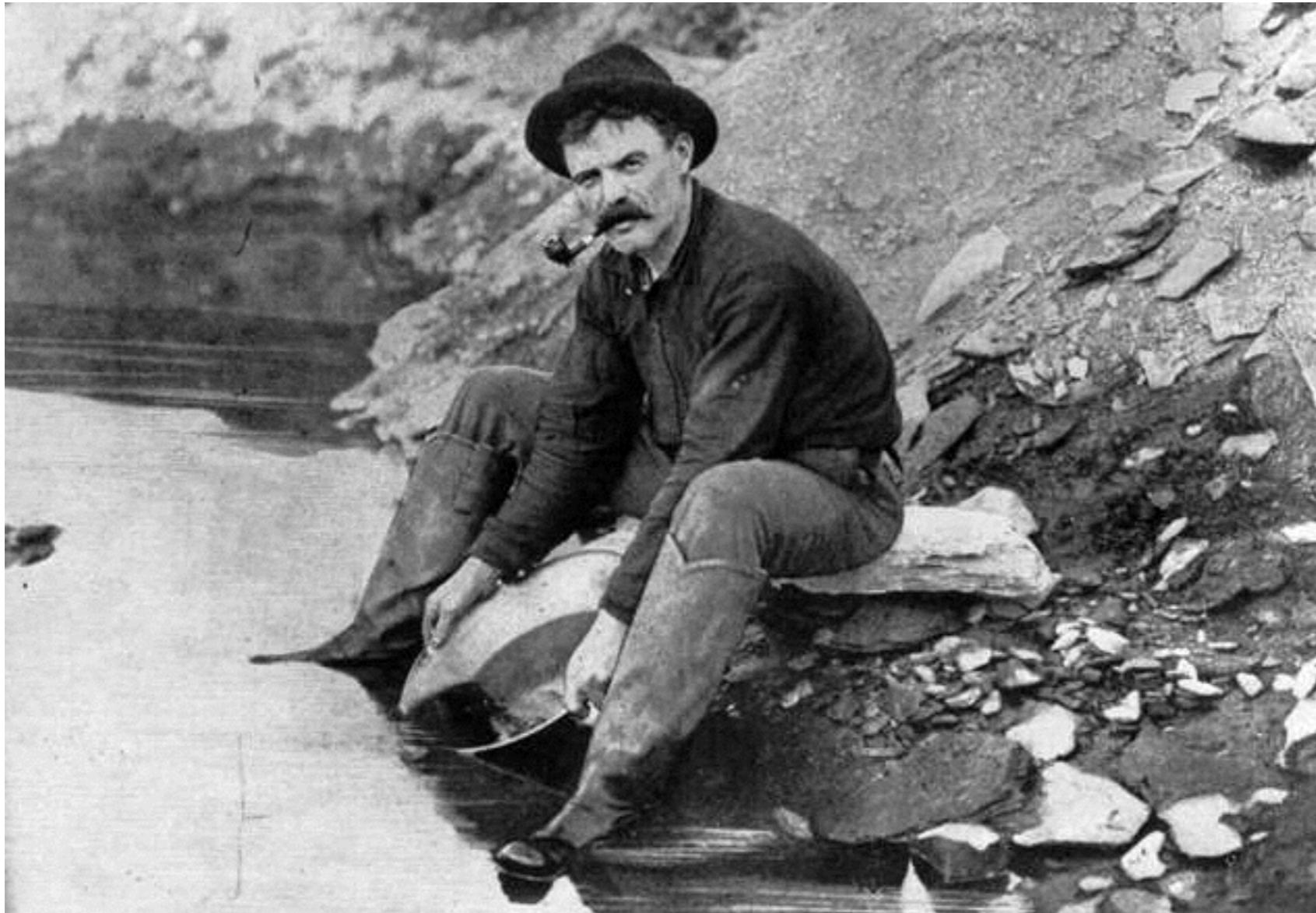
for (var i=0; i<fruits.length; i++) {
  console.log("fruits[" + i + "] = " + fruits[i]);
}

var transformedFruits = fruits.map( function(fruit) {
  return fruit.toUpperCase();
});
transformedFruits.forEach( function(fruit) {
  console.log(fruit);
});

var count = fruits.reduce( function(val, fruit) {
  console.log("reducer invoked with " + val);
  return val+1;
}, 0);
console.log("There are " + count + " fruits in the array");
```

Today's agenda

15:45 - 16:05	20'	Intro
16:05 - 16:25	20'	JavaScript 101 (part 1)
16:25 - 16:45	20'	Guidelines for phase 1
16:45 - 17:55	70'	Group in pairs
17:55 - 18:05	10'	Wrap-up



GitHub Explorer - Step 1

The GitHub Explorer



Carl is a end-user



Bob is the developer



Alice is the product owner

The GitHub Explorer



Carl is a end-user

"I build and deploy a web app"



Bob is the developer

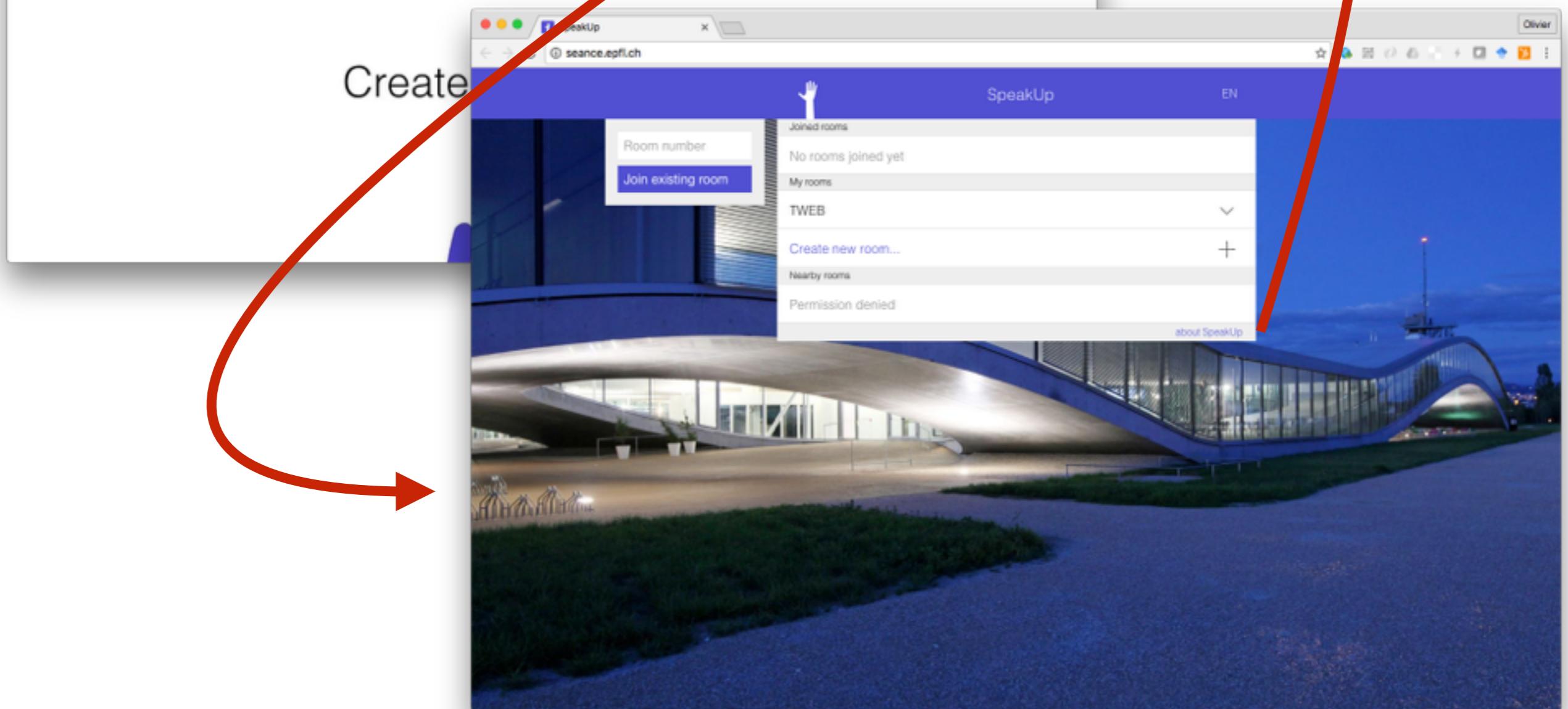
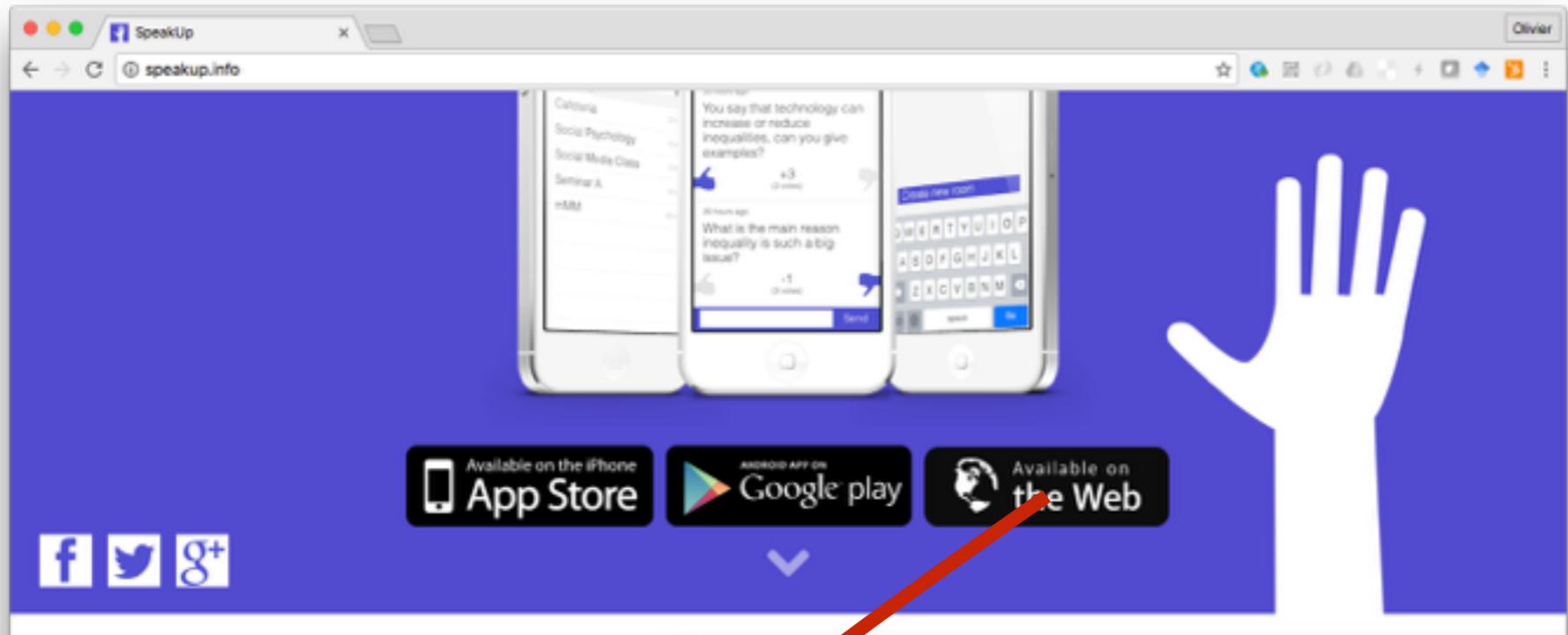


"I describe the app and
publish a landing page"

Alice is the product owner

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



The GitHub Explorer



Carl is a end-user

"What do I use to write,
store and run my code?"

"I build an deploy a web app"



Bob is the developer



Alice is the product owner

"What do I use to write
and host my content?"

"I describe the app and
publish a landing page"

"I have a seamless and unified
experience across the landing
page and the app"

"I start my journey at
<https://github-explore.io>"

The GitHub Explorer



Carl is a end-user



"What do I use to write,
store and run my code?"

"I build an deploy a web app"



Bob is the developer



Alice is the product owner



"What do I use to write
and host my content?"

"I describe the app and
publish a landing page"



"I have a seamless and unified
experience across the landing
page and the app"

"I start my journey at
<https://github-explore.io>"

Cloud Application Platform | Heroku

Heroku, Inc. [US] https://www.heroku.com/home Olivier

Products Elements Pricing Documentation Support More

SEARCH

≡

Oliver

HEROKU



NEW

Heroku Teams

Heroku Teams provides powerful collaboration and controls so you can bring your people, processes and tools together to build great software, with the ease of the Heroku developer experience.

EXPLORE HEROKU TEAMS

• • • •



dreamforce® Visit Heroku at Dreamforce
OCTOBER 4-7, 2016 SAN FRANCISCO, CA

LEARN MORE

calm-bayou-42065 | Heroku

Heroku, Inc. [US] https://dashboard.heroku.com/apps/calm-bayou-42065

HEROKU Jump to Favorites, Apps, Pipelines, Spaces... Olivier

Personal apps > calm-bayou-42065 Open app More

GITHUB wasadigi/tempTest master

Overview Resources Deploy Metrics Activity Access Settings

Installed add-ons \$0.00/month Configure Add-ons

There are no add-ons for this app

You can add add-ons to this app and they will show here. [Learn more](#)

Dyno formation \$0.00/month Configure Dynos

calm-bayou-42065 is using free dynos

web node index.js ON

Collaborator activity Manage Access

olivier.liechti@wasabi-tech.com 4 deploys

Latest activity All Activity

- olivier.liechti@wasabi-tech.com: Deployed d544b4e about 21 hours ago • v6 • [Compare diff](#)
- olivier.liechti@wasabi-tech.com: Build succeeded about 21 hours ago • [View build log](#)
- olivier.liechti@wasabi-tech.com: Deployed 2964d81 about 21 hours ago • v5 • [Compare diff](#)
- olivier.liechti@wasabi-tech.com: Build succeeded about 21 hours ago • [View build log](#)
- olivier.liechti@wasabi-tech.com: Deployed 33d6cbe about 21 hours ago • v4 • [Compare diff](#)
- olivier.liechti@wasabi-tech.com: Build succeeded about 21 hours ago • [View build log](#)
- olivier.liechti@wasabi-tech.com: Deployed 2e1ba6f about 22 hours ago • v3
- olivier.liechti@wasabi-tech.com: Build succeeded about 22 hours ago • [View build log](#)
- olivier.liechti@wasabi-tech.com: Enable Logplex

GitHub Pages - Websites for you and your projects.

https://pages.github.com Olivier

GitHub Pages

Websites for you and your projects.

Hosted directly from your [GitHub repository](#). Just edit, push, and your changes are live.

GitHub Training

Education

We love helping people just like you use GitHub and Git to collaborate and be more productive. No matter your level of experience, we have something for you.

Education from the experts

We love helping people just like you use GitHub and Git to collaborate and be more productive. No matter your level of experience, we have something for you.

What is GitHub Pages?

A screenshot of a web browser window displaying the GitHub Pages landing page at <https://pages.github.com>. The page features a large, bold headline: "Ready to get started? Roll vanilla, or generate a site for your project." Below the headline, a subtext reads: "You get one site per GitHub account and organization, and unlimited project sites. Ready? Let's get started." At the bottom of the main section are two buttons: "User or organization site" (unselected) and "Project site" (selected). A blue circular button with a question mark icon is located on the left side of the page. A callout box on the right contains the text: "Generate a site, or start from scratch? For Project sites, you have the option to generate a single page website with one of the amazing pre-built themes, or to create a site from scratch." It also includes two buttons: "Generate a site" (unselected) and "Start from scratch" (selected).

GitHub Pages - Websites for your projects

https://pages.github.com

Ready to get started? Roll vanilla, or generate a site for your project.

You get one site per GitHub account and organization, and unlimited project sites. Ready? Let's get started.

User or organization site Project site

?

Generate a site, or start from scratch?

For Project sites, you have the option to generate a single page website with one of the amazing pre-built themes, or to create a site from scratch.

Generate a site Start from scratch

Using Jekyll as a static site generator with GitHub Pages

GitHub Help Version Contact Support Return to GitHub

Customizing GitHub Pages / Using Jekyll as a static site generator with GitHub Pages How can we help?

Using Jekyll as a static site generator with GitHub Pages

If you use Jekyll as a static site generator with GitHub Pages, you benefit from more support with setting up, updating, and troubleshooting your site.

About GitHub Pages and Jekyll

In addition to supporting regular HTML content, GitHub Pages supports [Jekyll](#), a popular static site generator.

Setting up your GitHub Pages site locally with Jekyll

You can set up a local version of your Jekyll GitHub Pages site to test changes to your site locally. We highly recommend installing Jekyll to preview your site and help troubleshoot failed Jekyll builds.

Adding a Jekyll theme to your GitHub Pages site

You can add a Jekyll theme to your GitHub Pages site by editing your site's `_config.yml` file.

Configuring Jekyll

You can configure most Jekyll settings by editing your `_config.yml` file.



Project - Feature 1 acceptance criteria

As a user, I can visit the product landing page, transition to the web app and back to the landing page.

- You can work in **pairs of 2 students** (one can focus on the webapp to be deployed on Heroku, the other on the landing page to be deployed in GitHub pages)
- Every team must create **a new GitHub repo**
- For the feature to be “**done**”, the customer must be able to:
 - Visit the GitHub repo of the project and find one branch with the app code (master) and one branch with the landing page (gh-pages).
 - Have a link in the README.md that points to the landing page.
 - See your welcome message in a nice HTML template.
 - Be able to move back and forth between the landing page and the web app.
 - See a consistent look&feel across the landing page and the web app.

Feature 1 tasks

Do the Heroku tutorial (in a temporary git repo... it's about discovering the platform)

Do the GitHub Pages tutorial (in a temporary git repo... it's training time)

Pick an HTML template

Implement a simple express.js web app that serves a welcome page (and has a link back to the landing page)

Deploy the app on heroku

Do the GitHub Pages tutorial (in a temporary git repo... it's training time)

Prepare basic content and a link to the heroku app. Deploy.

Validate the acceptance criteria.

References

JavaScript | MDN

Mozilla Foundation [US] https://developer.mozilla.org/en/docs/Web/JavaScript

Sign in with mozilla

MDN MOZILLA DEVELOPER NETWORK

ZONES WEB PLATFORM TOOLS DEMOS CONNECT

MDN > Web technology for developers > JavaScript

LANGUAGES EDIT ⚙️

JavaScript

▲ HIDE SIDEBAR

SEE ALSO

JavaScript Tutorials:

- ▶ JavaScript Guide
- ▼ Introductory
 - Getting started
 - JavaScript technologies overview
 - Introduction to Object Oriented JavaScript
- ▼ Intermediate
 - A re-introduction to JavaScript
 - JavaScript data structures
 - Equality comparisons and when to use them
- ▶ Advanced

JavaScript® (often shortened to JS) is a lightweight, interpreted, object-oriented language with [first-class functions](#), most known as the scripting language for Web pages, but [used in many non-browser environments](#) as well such as [node.js](#) or [Apache CouchDB](#). It is a [prototype-based](#), multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles. Read more [about JavaScript](#).

The JavaScript standard is [ECMAScript](#). As of 2012, all modern browsers fully support ECMAScript 5.1. Older browsers support at least ECMAScript 3. A 6th major revision of the standard is in the works.

This section of the site is dedicated to the JavaScript language itself, the parts that are not specific to Web pages, or other host environments. For information about APIs specific to Web pages, please see [Web APIs](#) and [DOM](#).

JavaScript is not to be confused with the [Java programming language](#). Java is a trademark or registered trademark of Oracle in the U.S. and other countries.

Tutorials

Learn how to program JavaScript.

Reference

Browse the complete [JS reference](#) documentation.

Standard objects

MUST READ for the Tests

- **A re-introduction to JavaScript**
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript
- **Inheritance and the prototype chain**
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Inheritance_and_the_prototype_chain
- **Introduction to Object-Oriented JavaScript**
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript

JS JavaScript Objects in Detail X

javascriptissexy.com/javascript-objects-in-detail/

JavaScript Objects in Detail

January 27 Last Year

Show Modern Dev Ad

JavaScript.is (Sexy)

Learn everything about modern web application development with JavaScript and HTML5.

Sponsored By:

Grammar & Writing for Creators

Improve Quickly and Significantly
Write Powerful & Eloquent Prose

Bloggers Writers Programmers Designers Other Professionals

RICHARD OH STANLEY

Follow on Twitter

Recent Posts

▶ Beautiful JavaScript: Easily Create

JavaScript's core—most often used and most fundamental—data type is the Object data type. JavaScript has one complex data type, the Object data type, and it has five simple data types: Number, String, Boolean, Undefined, and Null. Note that these simple (primitive) data types are immutable, they cannot be changed, while objects are mutable.

What is an Object

An object is an unordered list of primitive data (and sometimes reference data types) types that are stored as name-value pairs. Each item in the list is called a property (functions are called methods) and each property name has to be unique and can be a string or a number.

Here is a simple object:

```
var myFirstObject = {firstName: "Richard", favoriteAuthor: "Conrad"};
```

To reiterate: Think of an object as a list that contains items and each item (a property) in the list is stored by a name-value pair. The property names in the example above are firstName and favoriteAuthor. And the values for each are "Richard" and "Conrad."

ECMA-262 » JavaScript. Th x

dmitrysoshnikov.com/ecmascript/javascript-the-core/

Home About 

Menu

ECMA-262

by Dmitry Soshnikov

JavaScript. The core.

 Tweet 285  Like 168  +1 117  Share 79  Share

Read this article in: Japanese, German (version 2), Arabic, Russian, French, Chinese.

1. An object
2. A prototype chain
3. Constructor
4. Execution context stack
5. Execution context
6. Variable object
7. Activation object
8. Scope chain
9. Closures
10. This value
11. Conclusion

This note is an overview and summary of the "ECMA-262-3 in detail" series. Every section contains references to the appropriate matching chapters so you can read them to get a deeper understanding.

Intended audience: experienced programmers, professionals.

We start out by considering the concept of an *object*, which is fundamental to ECMAScript.

An object

ECMAScript, being a highly-abstracted object-oriented language, deals with *objects*. There are also *primitives*, but they, when needed, are also converted to objects.

An object is a *collection of properties* and has a *single prototype object*. The prototype may be either an object or the *null* value.

Search...

Articles

- ES6 Notes: Default values of parameters
- Pattern Matching
- Essentials of Interpretation. Checkpoint: part 1
- Essentials of Interpretation. Intro.
- ECMA-262-5 in detail. Chapter 3.2. Lexical environments: ECMAScript implementation.

Comments

-  micha-s on Note 2. ECMAScript. Equality operators. Another way to check for NaN is [js]NaN != NaN[/js]
-  micha-s on Note 1. ECMAScript. Bound functions. I am completely misunderstanding the "Constructor with various number of arguments" section.
-  Dmitry Soshnikov on ES6

javascript the good parts - X

https://www.youtube.com/results?search_query=javascript+the+good+parts

YouTube CH

javascript the good parts

Upload Sign In

Filters ▾

About 27,000 results

JavaScript: The Good Parts
by GoogleTechTalks • 5 years ago • 409,631 views
Google Tech Talks Web Exponents presented by Doug Crockford February 27, 2009
blog post: ...
OFFICIAL CC
1:03:48

Douglas Crockford: JavaScript: The Good Parts
by YUI Library • 3 years ago • 13,180 views
In this talk from 2007, Douglas Crockford takes us on a journey through the lens of his own personal experience with JavaScript ...
39:38

Flexible Navigation | JavaScript Good Parts | CSS Animation Tricks | The Treehouse Show Episode 75
by Treehouse • 7 months ago • 9,750 views
In this episode of The Treehouse Show, Nick Pettit (@nickrp) and Jason Seifer (@jseifer) talk about the latest in web design, web ...
14:41 HD

Houston Developers Book Club: JavaScript the Good Parts Ch 4
by Jonathan Birkholz • 8 months ago
This meeting will be about Chapter 4: Functions from JavaScript the Good Parts. For this time: - Less confusion with Google ...
1:10:21 HD

Test Your Knowledge of Function Scope with Douglas Crockford
by MJG International (Frontend Masters) • 1 year ago • 9,342 views
In this lesson Douglas Crockford tests your knowledge of JavaScript function scope.
10:31 HD