

Javascript

#01

Matthieu Nicolas
Licence Pro CIASIE
Slides de Christophe Bouthier

Se présenter

- matthieu.nicolas@univ-lorraine.fr
- Doctorant, équipe COAST, LORIA
- Algorithmes pour l'édition collaborative pair-à-pair

Quelques règles

- Présence obligatoire
- 5min de retard tolérées
 - mais le cours commence à l'heure
- Si possible, chacun sur sa machine
 - encouragés à réfléchir en binôme

Organisation

- Volume horaire : 24h
- Enseignement intégré : CM et TD fusionnés
- 2 notes pour l'évaluation
 - Note de contrôle continu
 - Note de projet

Objectifs

- Hypothèse
 - Vous possédez tous déjà un background avec JavaScript
- But
 - Revoir et consolider les bases...
 - ... avant d'aller plus loin

Syllabus

- Bases du langage
- Programmation orientée objet
- Manipulation du DOM
- Requêtes AJAX et code asynchrone
- Bundler
- ...

Plan

- Historique
- Contexte
- Aspects simples du langage

Historique

Javascript
#01



Brendan Eich

Naissance

- Mai 1995
- Netscape
- 10 jours !!!
- Nom successifs: Mocha -> LiveScript
-> JavaScript
 - raison uniquement marketing !

À l'origine...

- Communication entre plugins et applets Java
- Modifications cosmétiques et dynamiques de la page

... à aujourd'hui

- Utilisé par tous les services
 - Google, Facebook, Netflix, Microsoft Office
- Utilisé par tous les appareils
 - PCs, Smartphones, Serveurs
- La frontière entre applications web et applications desktop disparaît
 - Skype, Discord, Slack, Visual Studio Code

«Standardisation»

- ECMA
 - ECMAScript - ECMA-262
 - ActionScript (Flash), JScript (Microsoft)
- ECMAScript 3 (1999)
 - Javascript «minimal»
- ECMAScript 5 (2009)
 - «Modern» browsers

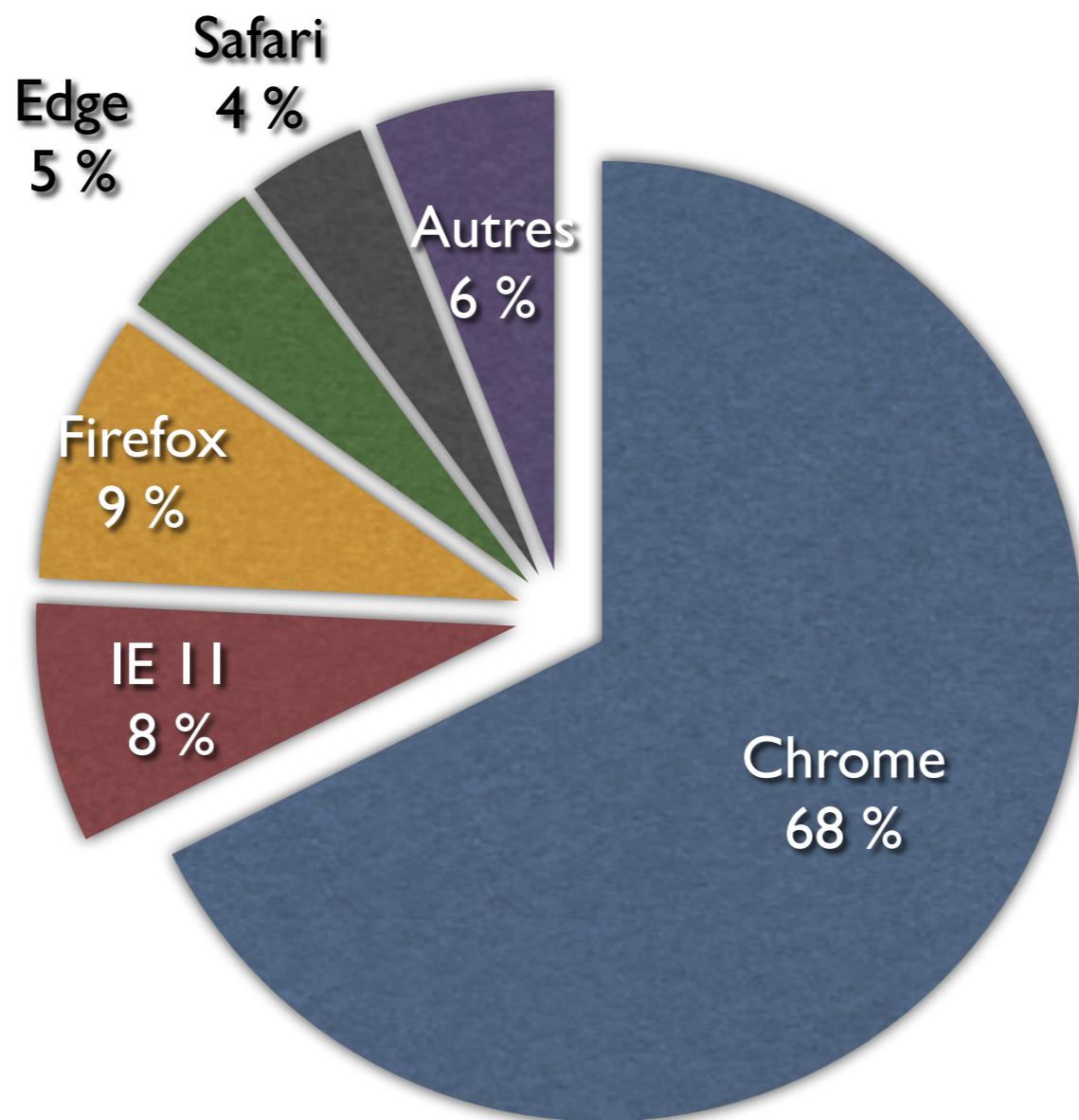
ECMAScript 4

- Macromedia/Adobe Flash
 - ActionScript 3 = ECMAScript 4
- Incompatibilité
 - Microsoft : «break the web»
- Accords d'Oslo
 - ECMAScript 3.1 = ECMAScript 5
 - Harmony pour le reste

Compatibilité ECMAScript 5

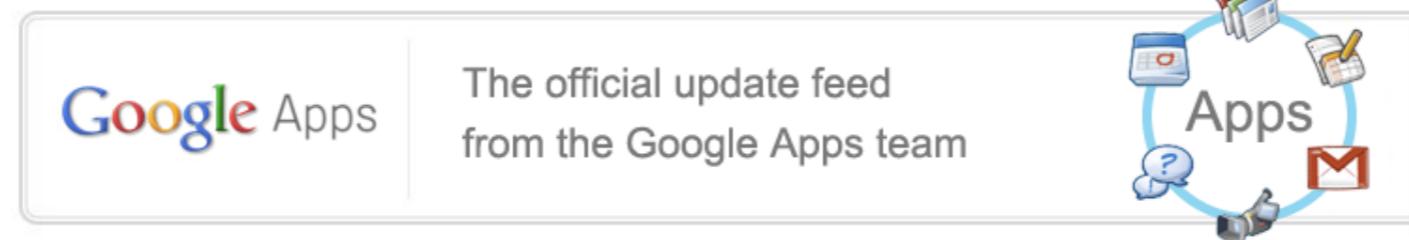
<http://kangax.github.io/es5-compat-table>

Parts de marchés des navigateurs webs



<http://www.netmarketshare.com/browser-market-share.aspx?qprid=2&qpcustomd=0>

Google et IE9



[Get update alerts by email](#)

[Subscribe to the update alerts RSS feed](#)

[Follow these updates on Twitter](#)

65385 readers
BY FEEDBURNER

Update topics

- [Google Apps \(236\)](#)
- [End-user \(235\)](#)
- [Google Apps for Government \(234\)](#)
- [Google Apps for Business \(208\)](#)
- [Google Apps for Education \(204\)](#)
- [Google Docs \(168\)](#)
- [Gmail \(129\)](#)
- [Premier Edition \(115\)](#)
- [Education Edition \(95\)](#)
- [Scheduled Release \(61\)](#)
- [Standard Edition \(56\)](#)
- [Google Calendar \(48\)](#)
- [Partner Edition \(47\)](#)
- [Mobile \(43\)](#)
- [Google Sites \(42\)](#)

End of support for Internet Explorer 9

Released on 11/05/2013

We would like to remind you of the Google Apps browser support policy, the set of guidelines for Google Apps services interoperability support. We support the latest version of Google Chrome (which automatically updates whenever it detects that a new version of the browser is available) as well as the current and prior major release of Firefox, Internet Explorer and Safari on a rolling basis. Each time a new version of one of these browsers is released, we begin supporting the update and stop supporting the third-oldest version.

<http://googleappsupdates.blogspot.fr/2013/11/end-of-support-for-internet-explorer-9.html>

Microsoft et IE 11

Support for older versions of
Internet Explorer ended

Beginning January 12, 2016, only the most current version of Internet Explorer available for a supported operating system will receive technical supports and security updates. Internet Explorer 11 is the last version of Internet Explorer, and will continue to receive security updates, compatibility fixes, and technical support on Windows 7, Windows 8.1, and Windows 10.

[UPGRADE NOW >](#)



<https://www.microsoft.com/en-us/windowsforbusiness/end-of-ie-support>

ECMAScript 6 (2015)

- Améliore le langage
- Simplifie son utilisation
- Beaucoup d'idées piochées depuis les libraires de la communauté
- Pendant longtemps peu supporté par les navigateurs

Compatibilité ECMAScript 6

<http://kangax.github.io/compat-table/es6/>

Compatibilité ECMAScript 6

<http://kangax.github.io/compat-table/es6/>

Nouvelles versions

- Depuis, une nouvelle version déployée chaque année
 - ES6 -> ES 2015
 - ES 2016, ES 2017, ..., ES 2019
- ES.Next correspond à la prochaine

Résumé

- JavaScript n'est pas utilisé aujourd'hui ce pour quoi il a été conçu
- Le langage possède des lacunes qu'on doit combler en tant que devs
- Mais ça s'améliore !
- De moins en moins besoin de supporter les navigateurs obsolètes

Intérêts contraires des acteurs du web

INTERNET SERVICES / LEER EN ESPAÑOL

Mozilla engineer says Google slowed YouTube down on non-Chrome browsers

A redesign allegedly made the video site more dependent on Chrome's architecture.

BY SEAN KEANE / JULY 26, 2018 1:43 AM PDT

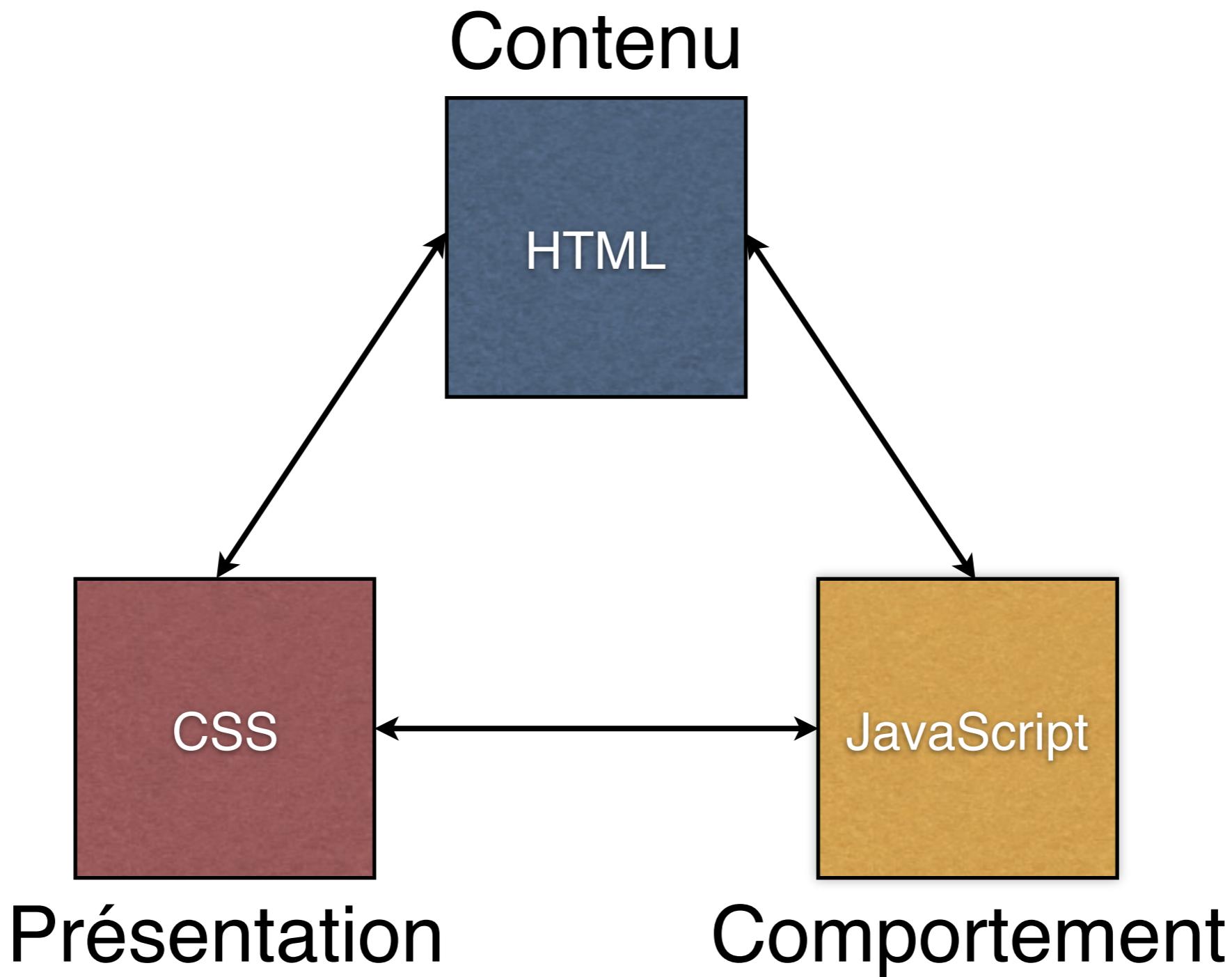


<https://www.cnet.com/news/mozilla-exec-says-google-slowed-youtube-down-on-non-chrome-browsers/>

Contexte

Javascript
#01

Architecture Web



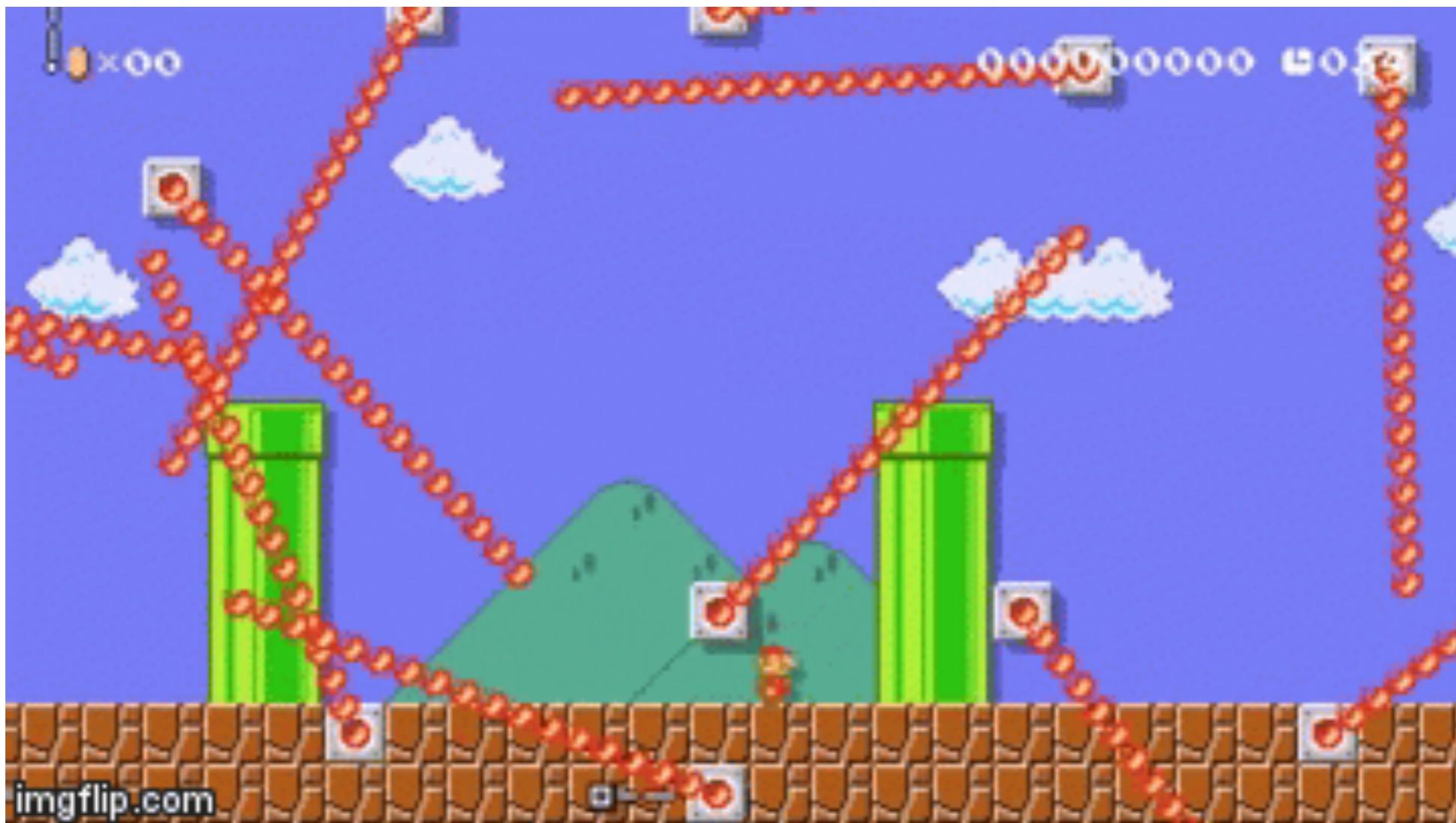
Good parts

- Paradigme fonctionnel
- Programmation événementielle
- Flexibilité
- Universalité

Bad parts

- Typage dynamique
 - Variables non typées, valeurs typées
- Conversions automatiques de types
- Mono-threadé
- Lourdeur de la toolchain actuelle

Coder sans types



Réponse de la communauté

- Librairies
 - JQuery, Dojo, Prototype, YUI...
 - Apporter à JS ce qui lui manque

Node.js

- Un seul langage frontend et backend
- JavaScript «obligatoire» pour frontend
- Donc JavaScript aussi pour backend

Transpiler

- Les nouvelles spécifications améliorent le langage...
- ... mais mettent du temps à être supportées par les navigateurs
- “Compile” votre nouveau code JS en ancien code

TypeScript

- Un nouveau langage basé sur JS
- Ajoute un typage statique
- A besoin d'être transpilé en JS pour pouvoir être exécuté

WebAssembly

- Permet d'exécuter du code d'un langage tel que C, C++ ou Rust dans le navigateur
- Offre des performances proches du code natif...
- ... tout en interagissant avec le code JS

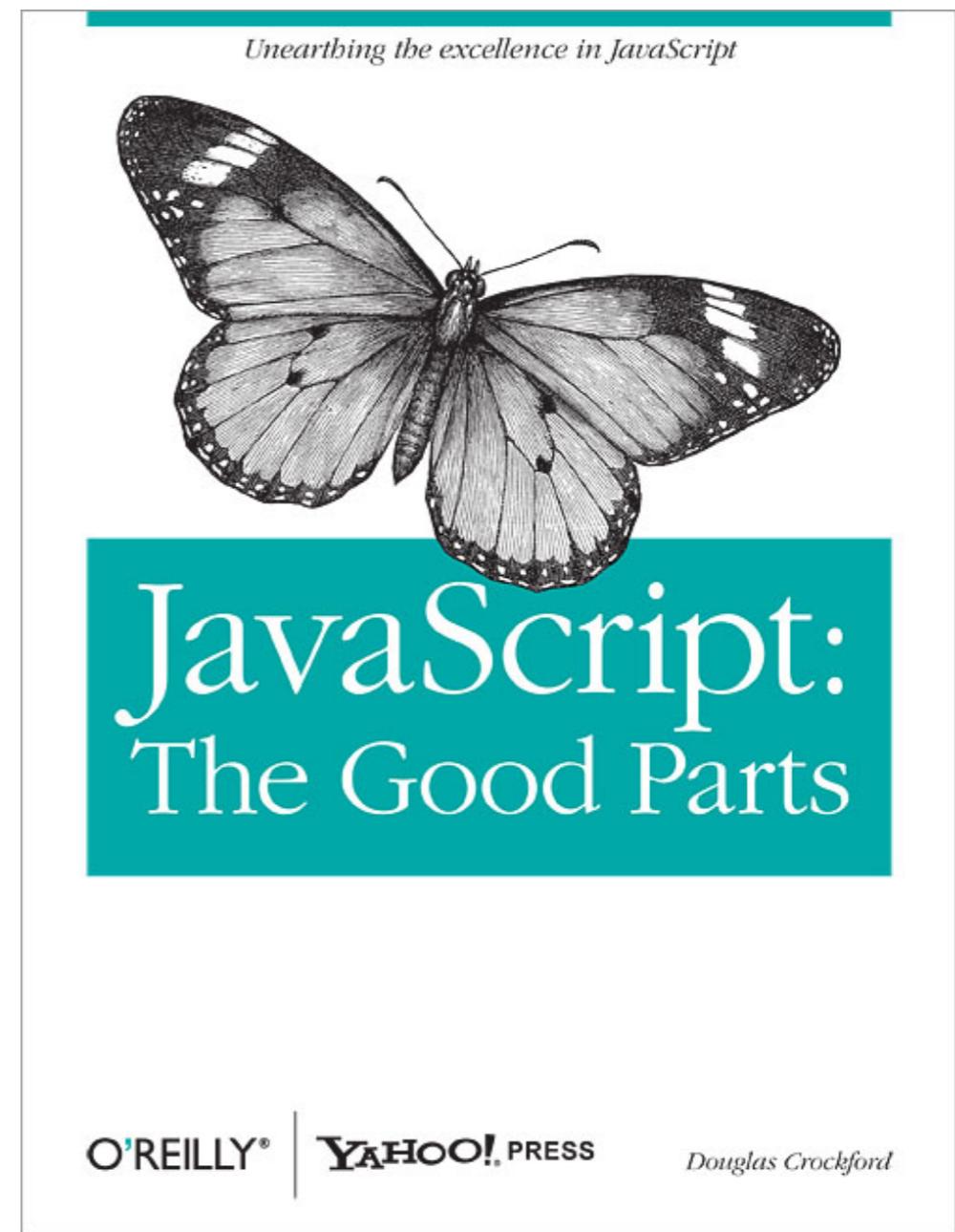
Langage - 1

Javascript
#01

Références

JavaScript:
The Good Parts

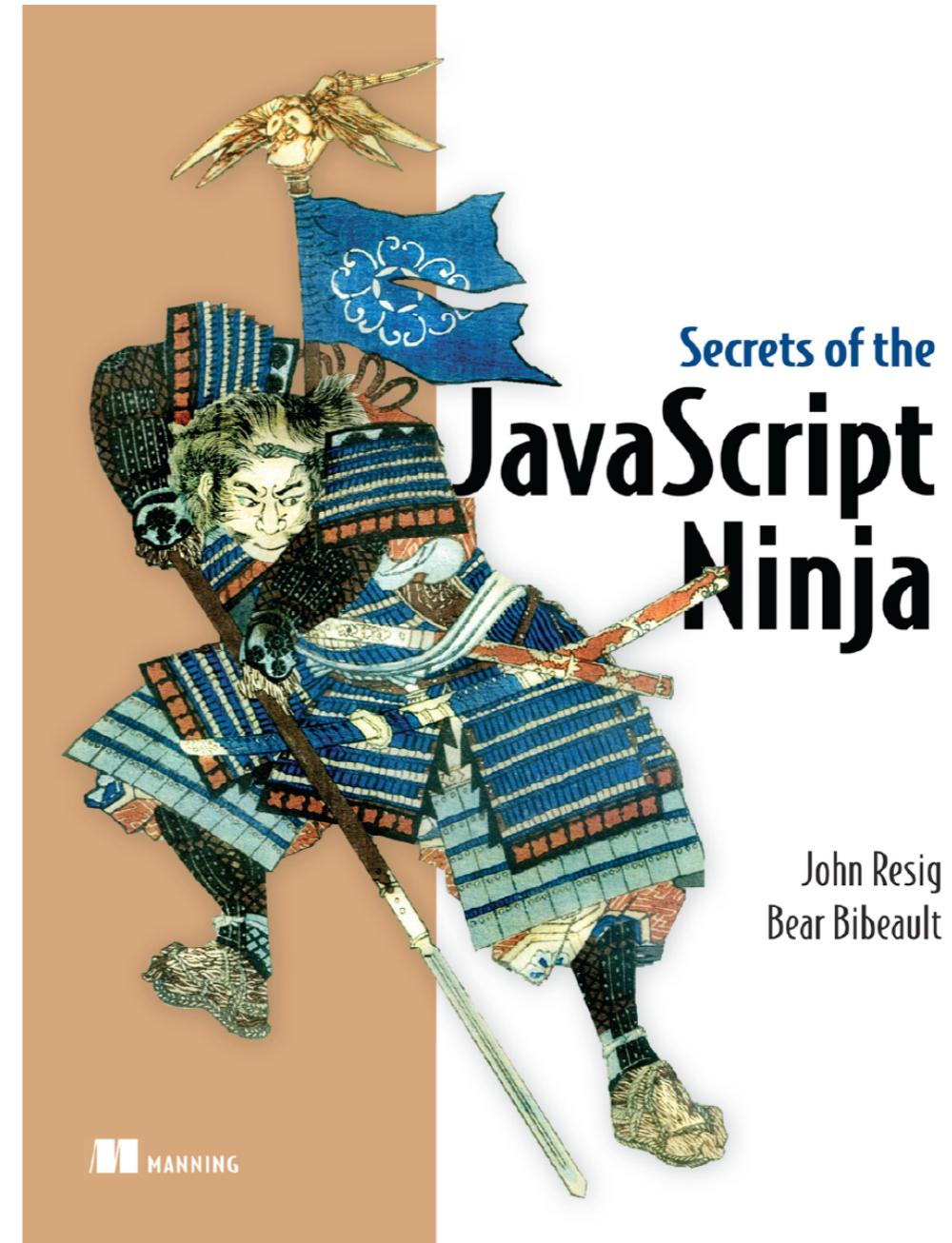
Douglas Crockford - Yahoo!



Références

Secrets of the
JavaScript Ninja

John Resig - JQuery
Bear Bibeault



Références

Référence JavaScript - Mozilla Developer Network

<https://developer.mozilla.org/fr/docs/Web/JavaScript>

Variables

- Plusieurs mots clés pour déclarer une variable
 - let a = 1
 - let b = “toto”

Primitives

- Number
- String
- Array
- Boolean
- Object
- Function

Number

- 2, -4, 8.5
- pas de ≠ entre int et float
- Valeurs particulières:
 - Infinity, NaN (Not a Number)

Number

- Opérateurs disponibles
 - +, -, *, /, %
 - +=, ++, -=, --
 - <, >, <=, >=, ===, !==
- Ensemble de fonctions disponibles
 - Math.xxx

String

- “test” ou ‘test’
- propriétés :
 - length
- méthodes :
 - toUpperCase, ...
- immutable
- concaténation par ‘+’

Boolean

- 2 valeurs: true, false
- false :
 - false, 0, null, undefined, NaN, “” (empty string)
- true :
 - tout le reste !

Boolean

- Opérateurs disponibles
 - et logique: &&
 - ou logique: ||
 - négation: !

Array

- [1, “a”, -4, false]
- Pas forcément le même type dans chaque case
- taille qui augmente automatiquement
- possible d'avoir des trous
 - assignage direct
 - undefined pour les trous

Object

- { prop: value, prop2: value2 }
- Peut posséder des fonctions comme propriétés
 - S'agit de méthodes
- Hérite automatiquement des méthodes de Object
 - `toString()`

Accès Propriétés

- 2 possibilités
 - objet.prop
 - objet["prop"]
- 2ème obligatoire si :
 - “prop” est un mot réservé
 - https://www.w3schools.com/js/js_reserved.asp
 - Accès dynamique

Propriétés

```
1 const obj = {  
2   prop1: "hello"  
3 }  
4  
5 const a = obj.prop1 // Assigne "hello" à a  
6 const b = obj.prop2 // Assigne undefined à b  
7  
8 obj.prop1 = "hello" // Assigne "hello" à obj.prop1  
9  
10 obj.prop2 = "world" // Ajout propriété prop2  
11 |
```

Supprimer propriété

- À l'aide du mot clé delete

```
1 const obj = {  
2   prop1: "hello",  
3   prop2: "world"  
4 }  
5  
6 obj.prop1 = null  
7 obj // {prop1: null, prop2: "world"}  
8  
9 obj.prop1 = undefined  
10 obj // {prop1: undefined, prop2: "world"}  
11  
12 delete obj.prop1  
13 obj // {prop2: "world"}  
14
```

Function

- Objet comme les autres
 - Possède des propriétés et méthodes
 - Peut être assigné à une variable
 - Peut être passé en argument : callbacks
 - Peut être retourné comme valeur de retour
- En plus : peut être invoqué « () »

Déclaration

```
1  function nomFonction (param1, param2) {  
2      ... code  
3  }  
4  
```

- Peut être définie n'importe où
- nomFonction : pas obligatoire
 - Peut être utilisé dans le scope pour appeler la méthode
 - Sert DANS la méthode pour récursion
 - Propriété name

Hoisted (hissé)

- Définie dans TOUT le scope, même AVANT la définition :

```
1   foo() // 1
2
3   function foo() {
4       return 1
5   }
6
```

Arguments

- Pas de vérification de type
- Pas de vérification du NOMBRE !!!
 - Passés en plus : ignorés !
 - Manquants : mis à undefined

arguments

- Paramètre toujours passé à une fonction
- Contient TOUS les arguments passés
 - Indépendant de la déclaration
- “Pseudo” Array
 - Contient length, mais PAS les méthodes !
 - arguments.length ≠ nomFonction.length

arguments

```
1  foo(1, 2, 3)
2
3  function foo(arg1, arg2) {
4      arguments // [1, 2, 3]
5      arguments.length // 3
6      arguments.slice // undefined
7  }
8
```

Valeurs Spéciales

- **undefined**
 - variable qui n'a pas encore de valeur
 - dans un array...
- **null**
 - valeur spécifique signifiant aucune valeur

typeof

- opérateur
- “number”, “string”, “boolean”, “object”,
“function”, “undefined”
- Array -> “object”
- typeof null ?
- typeof NaN ?

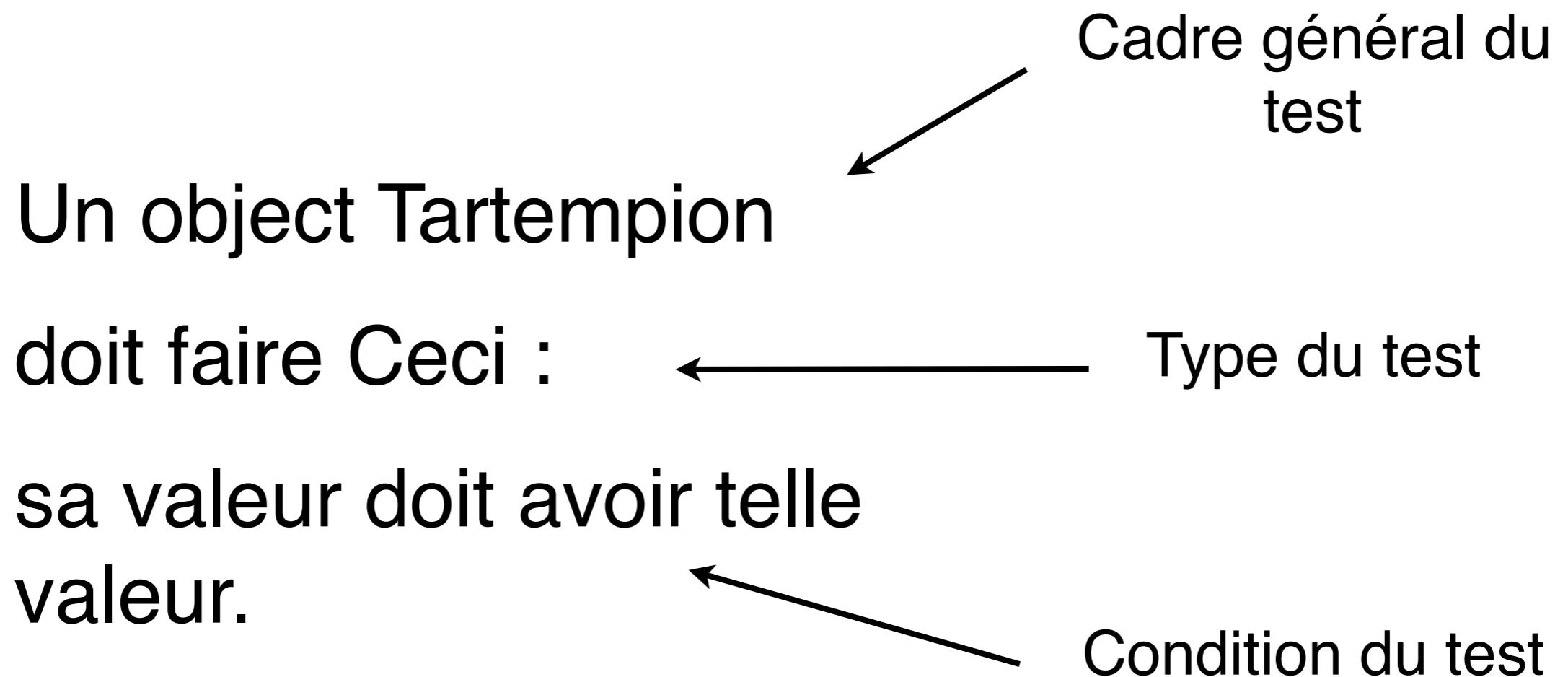
Opérateurs

- if else
- switch case default
- while
- for (; ;)
- return

Tests unitaires

- But :
 - s'assurer du fonctionnement d'un morceau de code
 - de manière automatisée
- Avantages :
 - Vérification
 - Documentation
 - Pas de régression

«Shoulda»



«Shoulda»

describe ‘tartempion’

it should ‘do something’

expect ‘value’ to be equal
to ‘val’

Cadre général du
test

Type du test

Condition du test

Tests Jasmine

<https://jasmine.github.io/>

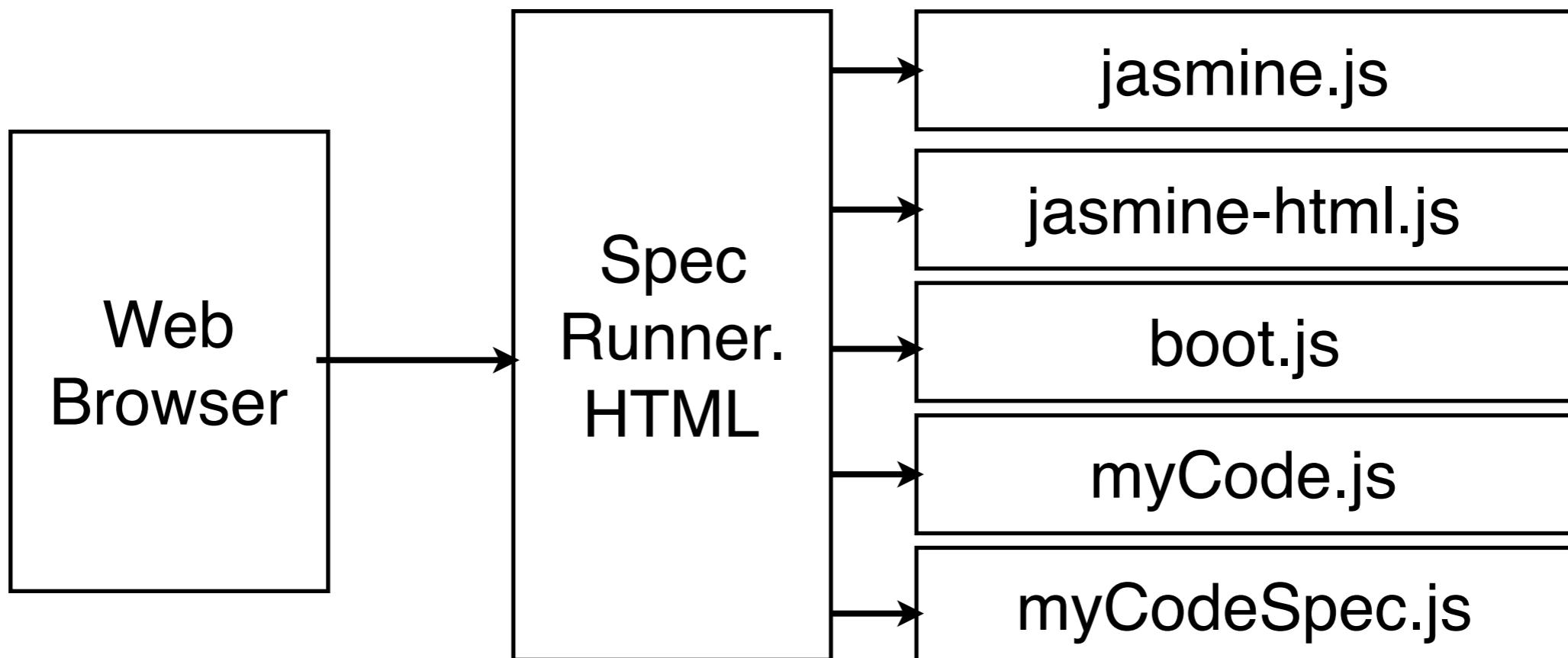
```
describe("A suite", function() {  
  it("contains spec with an expectation", function() {  
    expect(true).toBe(true);  
  });  
});
```

Matchers

- `expect(true).toBe(true);`
- `expect(false).not.toBe(true);`
- `toEqual`
- `toMatch(/bar/)`
- `toBeDefined / toBeUndefined`
- `toBeNull`
- `toBeTruthy / toBeFalsy`

Installation

- Tests : 1 fichier js («spec»)
- Framework : 1 fichier js
- Runner : 1 fichier HTML + 2 fichier js



Arche

- Infos du Module
 - 2019 - LP CIASIE LP2 : Bases de la programmation web / client
 - Identifiant : 29227

TD

https://classroom.github.com/a/2pyJ_5v8

(lien dispo sur Arche)

- Cloner le projet
- Suivre l'énoncé se trouvant dans le répertoire sujets/