2014

GMIN 309 - TAWeb

Divers et rappels

Bérenger ARNAUD
berenger.arnaud@itkweb.com

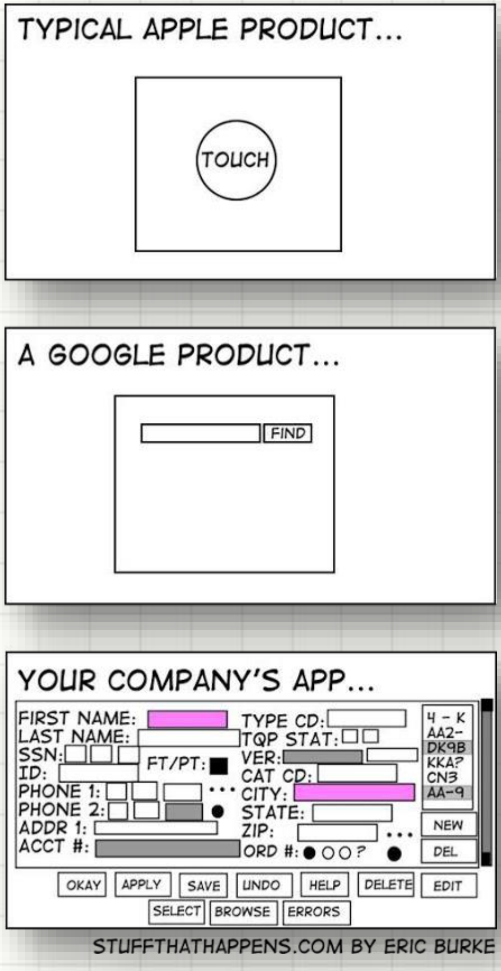
iTK
Créateur de logiciels experts pour l'agriculture

Sommaire (cours 7)

- UI /UX
  - scénarisation
  - internationalisation
  - éléments HTML 5
  - formulaires HTML 5
- développement
  - GIT
  - PAC
  - TDD
- Programmation
  - astuces JS
  - worker
  - promise
  - factory
- prototypes
  - prototype
  - proxy
  - RMI
- Sécurité
  - mots de passe
  - attaques
  - SQL injection
  - code injection

UI / UX

Scénarisation

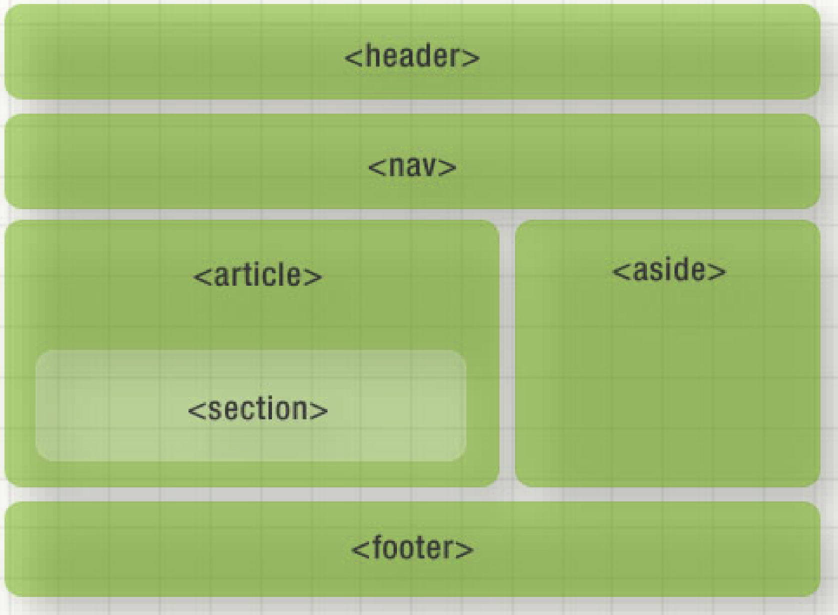


- scénariser de la navigation
  - guider, orienter
  - ne pas perdre
- règles
  - trop vs. assez
    - 7 éléments et variations
  - pas d'interruption

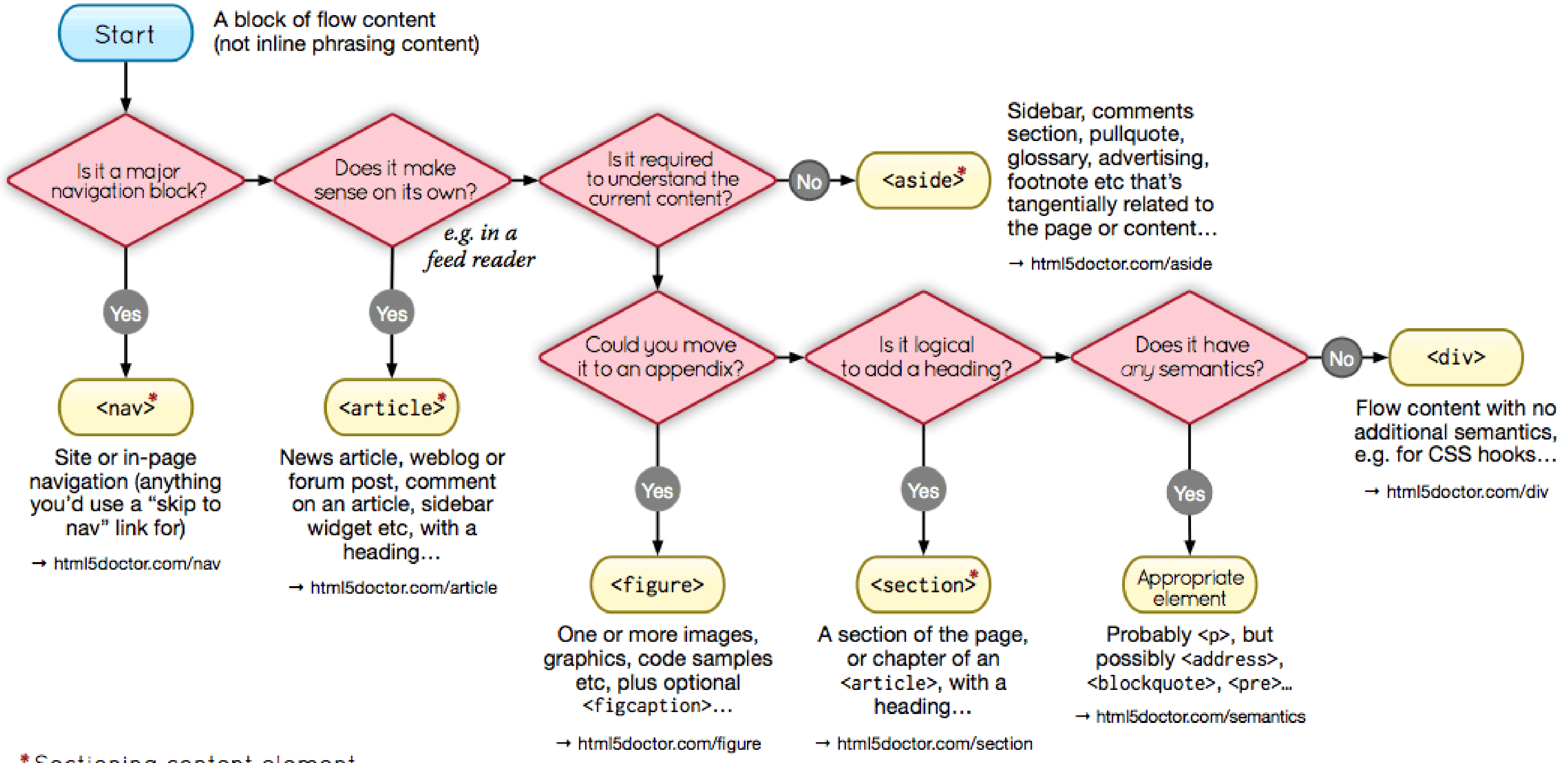
Internationalisation & localisation

- aka i18n & l10n
- non natif, libs JS
- fichiers ressources
  - textes
  - images
  - …

Éléments HTML 5



- article
- aside
- details
- figure
- footer
- header
- main
- nav
- progress
- section
- time
- [...]


Start
Is it a major navigation block?
Yes
<nav>*
Does it make sense on its own?
e.g. in a feed reader
Yes
<article>*
Is it required to understand the current content?
No
<aside>*
Could you move it to an appendix?
Yes
<figure>
Is it logical to add a heading?
Yes
<section>*
Does it have any semantics?
Yes
Appropriate element
No
<div>

HTML5 Element Flowchart

Sectioning content elements and friends

By @riddle & @boblet
www.html5doctor.com

A block of flow content (not inline phrasing content)

Site or in-page navigation (anything you'd use a "skip to nav" link for)
→ html5doctor.com/nav

News article, weblog or forum post, comment on an article, sidebar widget etc, with a heading…
→ html5doctor.com/article

Sidebar, comments section, pullquote, glossary, advertising, footnote etc that's tangentially related to the page or content…
→ html5doctor.com/aside

One or more images, graphics, code samples etc, plus optional `<figcaption>`…
→ html5doctor.com/figure

A section of the page, or chapter of an `<article>`, with a heading…
→ html5doctor.com/section

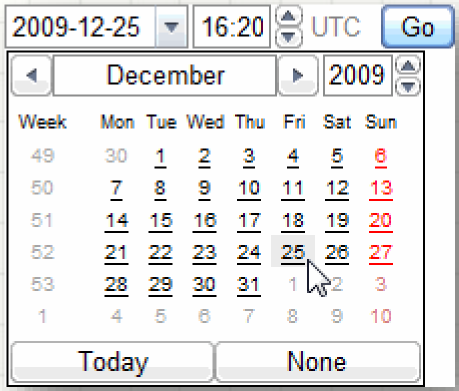
Probably `<p>`, but possibly `<address>`, `<blockquote>`, `<pre>`…
→ html5doctor.com/semantics

Flow content with no additional semantics, e.g. for CSS hooks…
→ html5doctor.com/div

* Sectioning content element
These four elements (and their headings) are used by HTML5's outlining algorithm to make the document's outline
→ html5doctor.com/outline

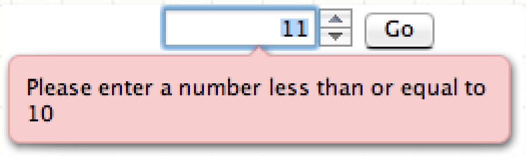
2011-07-22 v1.5
For more information:
www.html5doctor.com/semantics

Formulaire HTML 5 : types



- button
- checkbox
- color
- date
- datetime
- datetime-local
- email
- month
- number
- password
- radio
- range
- search
- submit
- tel
- text
- time
- url
- week

Formulaire HTML 5 : attributs

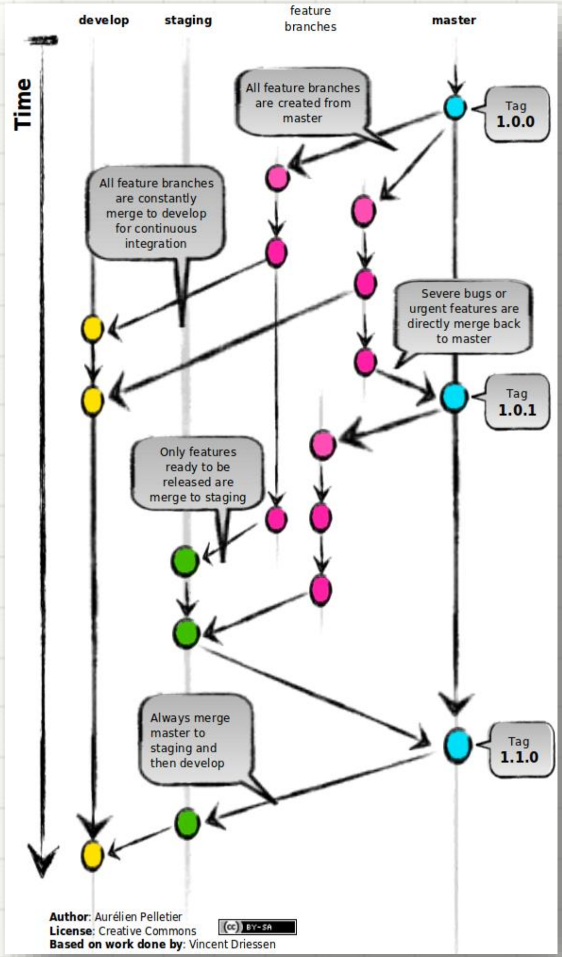


Please enter a number less than or equal to 10

- auto-complete
- autofocus
- disabled
- form*
- list
- min and max
- maxlength
- multiple
- pattern
- place-holder
- readonly
- required
- size
- step
- value

DÉVELOPPEMENT

GIT


develop
staging
feature branches
master
Time
All feature branches are created from master
Tag 1.0.0
All feature branches are constantly merge to develop for continuous integration
Severe bugs or urgent features are directly merge back to master
Tag 1.0.1
Only features ready to be released are merge to staging
Always merge master to staging and then develop
Tag 1.1.0
Author: Aurélien Pelletier
License: Creative Commons
Based on work done by: Vincent Driessen

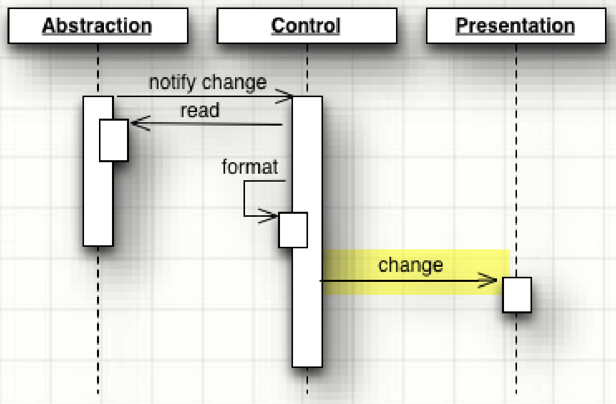
- git clone

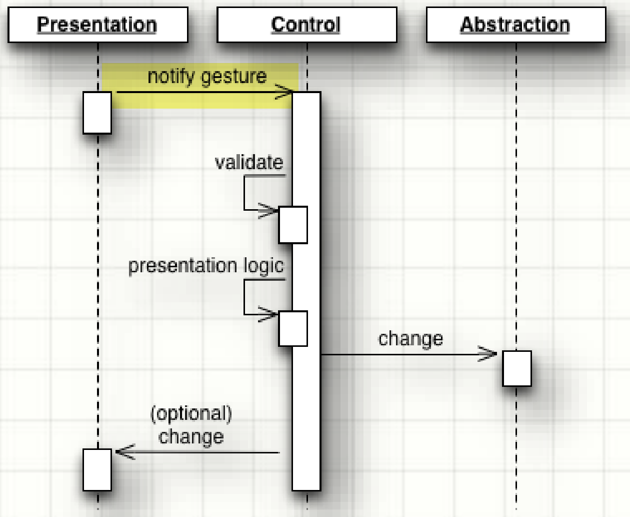
- git pull --rebase
- git branch "#001 fix things"

- git add […]
- git commit "Thing 1 fixed"

- git checkout master
- git merge "#001 fix things«
- git push

Présentation, Abstraction, Contrôle





Présentation

- visualisation
- interactions utilisateurs

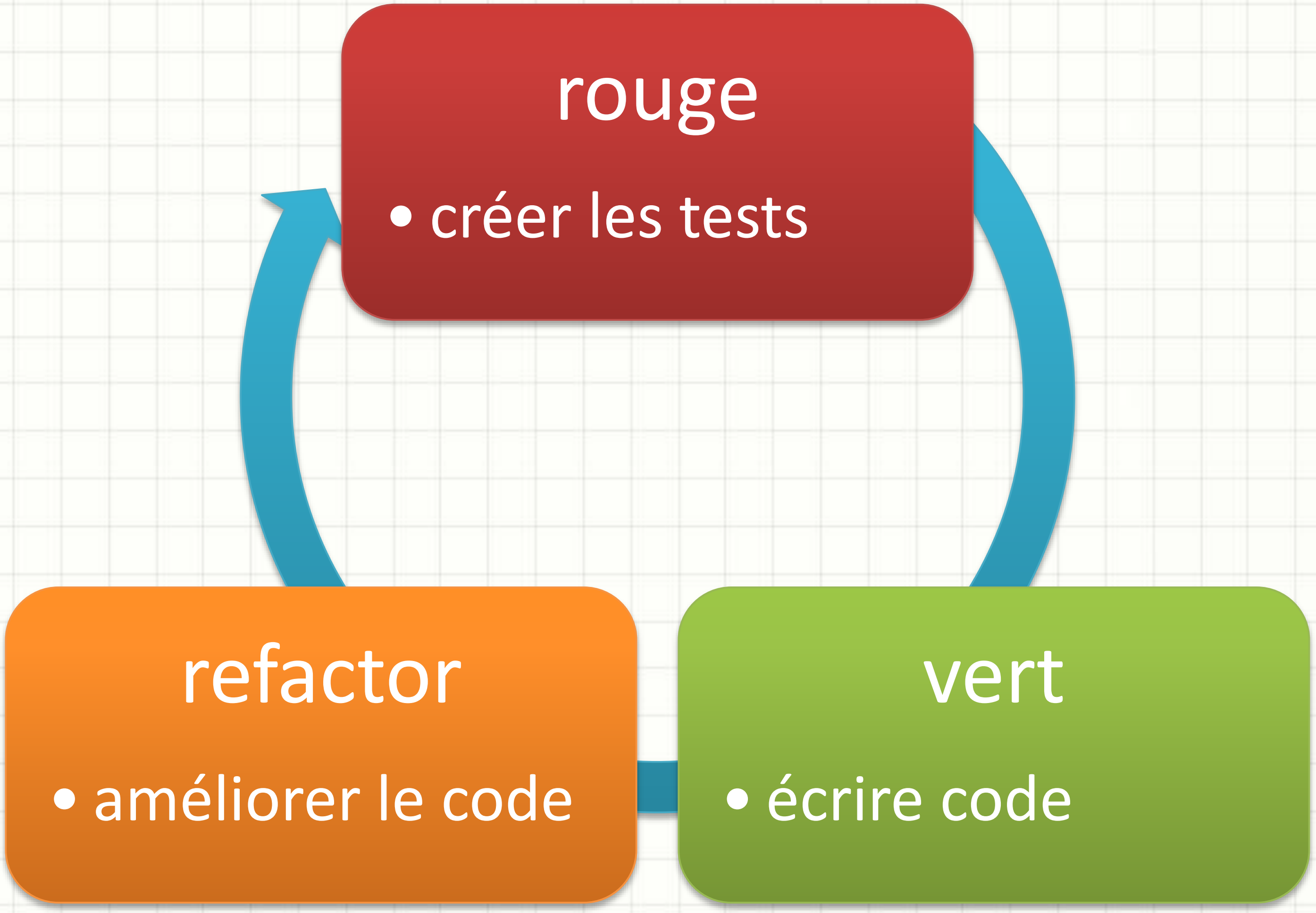
Abstraction

- données
- méthodes métiers

Contrôle

- inter-actions

Test Driven Development

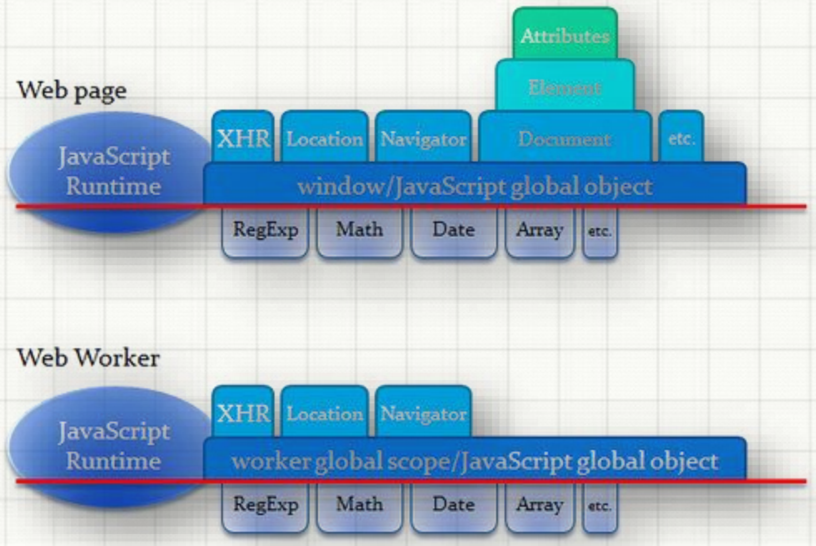


Programmation

Astuces

`var o = {};`	`var o = new Object();`
`var a = [];`	`var a = new Array();`
`function() {  }`	*fonction anonyme*
`foo = function foo() {  }`	*fct nommée & référencée*

Worker

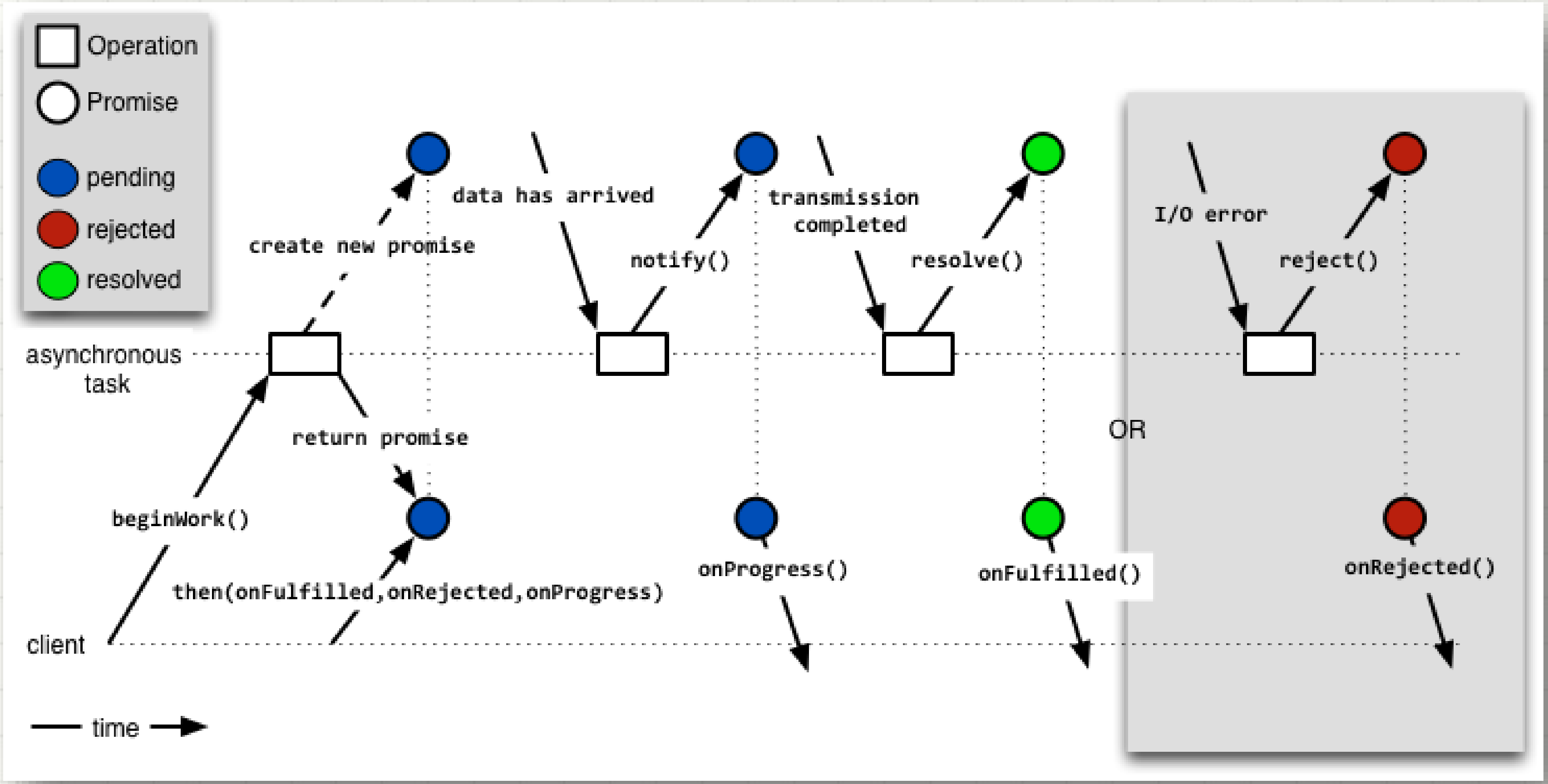


- w = new Worker(…)
- w.postMessage(…)

- addEventListener(‘message’, …)

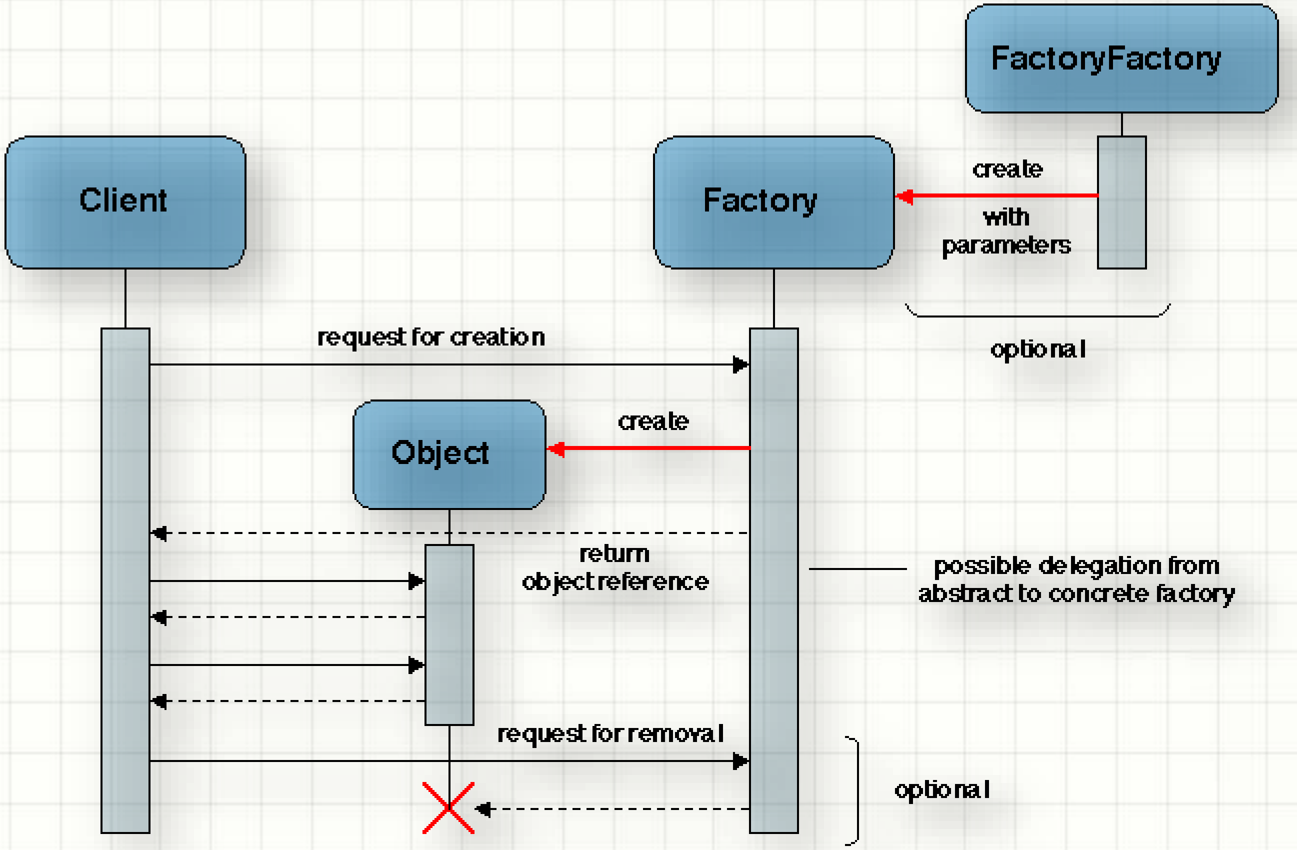
- w.terminate()
- close()

Promise



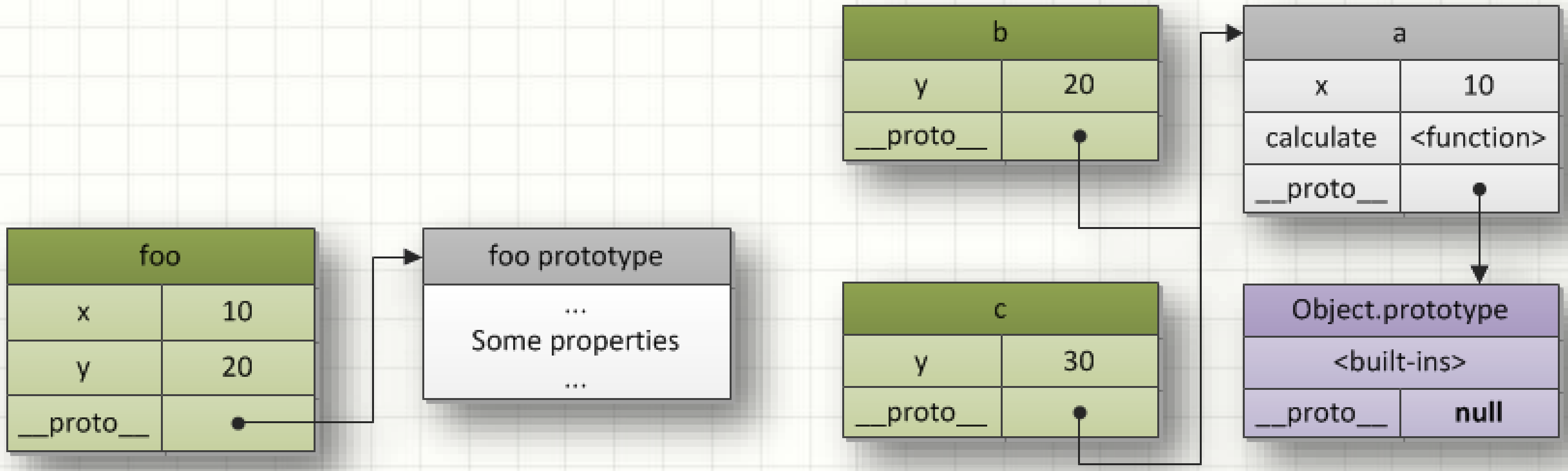
Operation
Promise
pending
rejected
resolved
asynchronous task
client
time
create new promise
return promise
beginWork()
then(onFulfilled,onRejected,onProgress)
data has arrived
notify()
onProgress()
transmission completed
resolve()
onFulfilled()
OR
I/O error
reject()
onRejected()

Factory


FactoryFactory
Client
Factory
create
with parameters
optional
request for creation
Object
create
return object reference
possible delegation from abstract to concrete factory
request for removal
optional

PROTOTYPE

Prototype



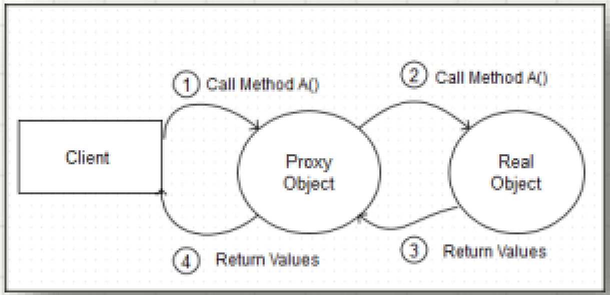
```
var b = Object.create(a,
 {y: {value: 20}});

var c = Object.create(a,
 {y: {value: 30}});
```

http://yehudakatz.com/2011/08/12/understanding-prototypes-in-javascript/

Prototype

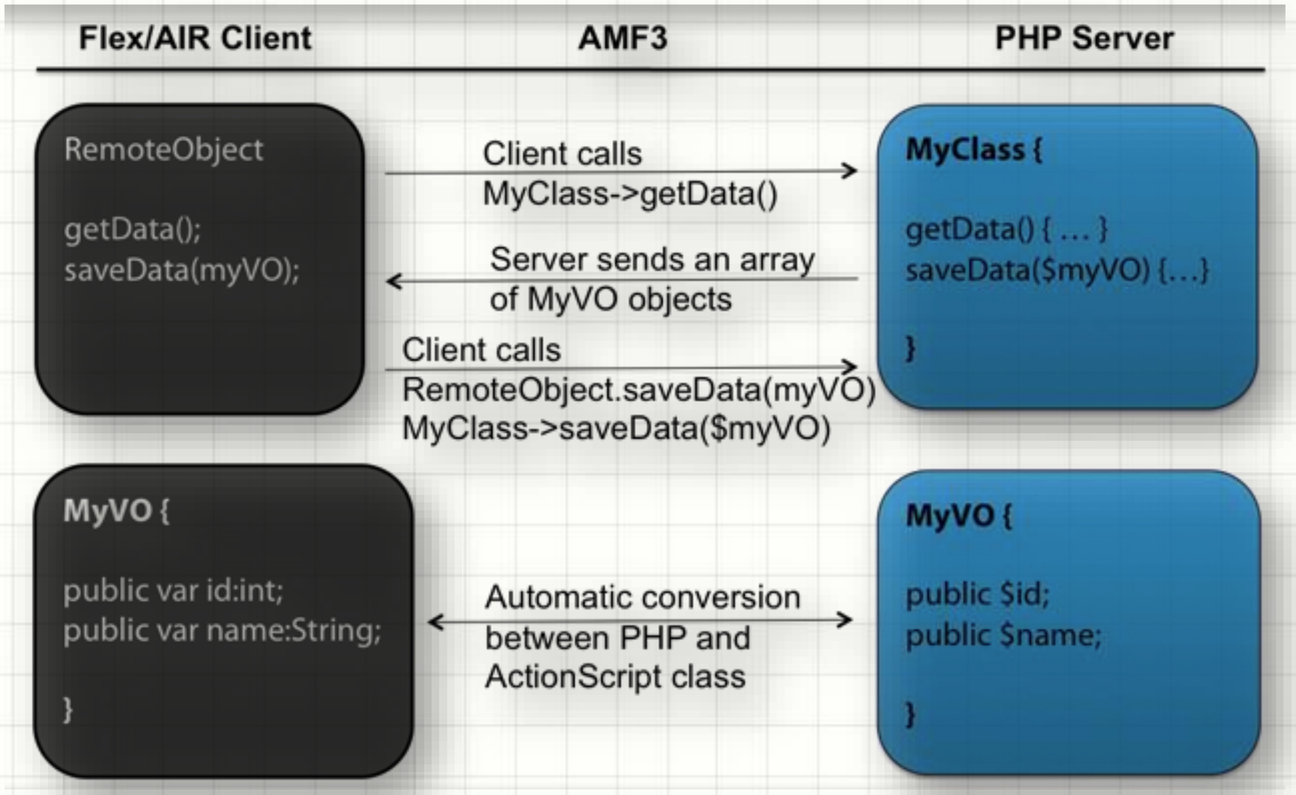
```
var A = {};
var B = (
    function() {
        var F = function () {};
        F.prototype = A;
        return new F();
    } ());
var B = Object.create(A);

A.compte = 0;
A.separ = ": ";
B.separ = ", ";

for (A.compte = 0; A.compte < 10; A.compte++) {
    document.write(B.compte + B.separ)
}

// 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
```

[class,funtion,attribute]Proxy



- property/function
  - has…()
  - set…()
  - get…()
  - delete…()
  - call…()
- external
  - read…()
  - write…()

Remote Method Invocation



SÉCURITÉ

Mots de passe

1. 123456
2. password
3. 12345678
4. qwerty
5. abc123
6. 123456789
7. 111111
8. 1234567
9. iloveyou
10. adobe123
11. 123123
12. Admin
13. 1234567890
14. letmein
15. photoshop
16. 1234
17. monkey
18. shadow
19. sunshine
20. 12345
21. password1
22. princess
23. azerty
24. trustno1
25. 000000

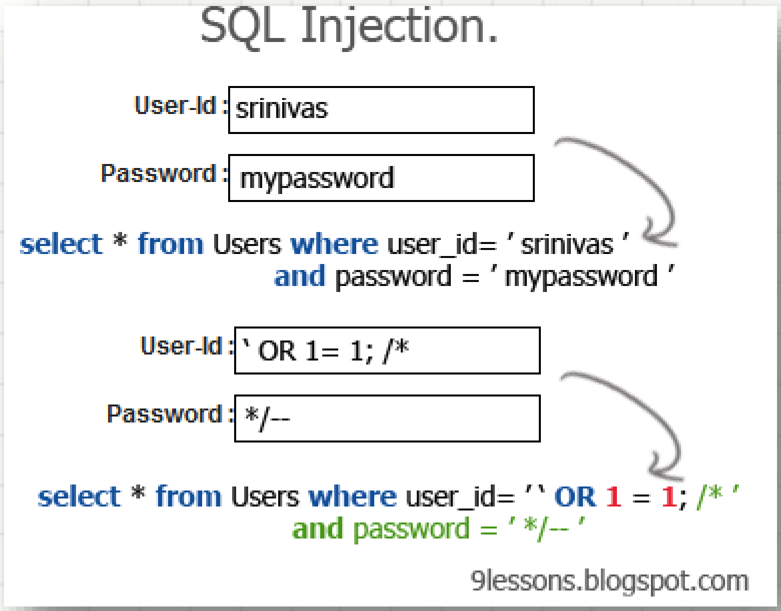
(2013)

Attaques

1. Injection
2. Broken Authentication and Session Management
3. Cross-Site Scripting (XSS)
4. Insecure Direct Object References
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross-Site Request Forgery (CSRF)
9. Using Components with Known Vulnerabilities
10. Unvalidated Redirects and Forwards

https://www.owasp.org/index.php/Top10

SQL injection


SQL Injection.

User-Id : srinivas

Password : mypassword

```
select * from Users where user_id= ' srinivas '
        and password = ' mypassword '
```

User-Id : ' OR 1= 1; /*

Password : */--

```
select * from Users where user_id= '' OR 1 = 1; /* '
        and password = ' */-- '
```

9lessons.blogspot.com

- ' or '1'='1
- ' or '1'='1' --
- ' or '1'='1' ({
- ' or '1'='1' /*

- a'; DROP TABLE users; SELECT * FROM userinfo WHERE 't' = 't

Code injection

- PHP
  - caractères spéciaux
  - interprétation code / fonctions
  - appel de code
- HTML
  - interprétation code
  - injection code
- JS
  - interprétation code

Social engineering

Fini !