GROS

CONNARDS !

CRIE!

**Ton contenu est un tas de glaise.**
**Ton prototype est une vision.**
**La vision d'un vase,**
**dont la forme suit la fonction.**

**Aujourd'hui, c'est poterie.**

DVM

La vie est parfois compliquée, mammy.

DVM

**Qui a, comme moi, ce tic du ⌘+S ?**

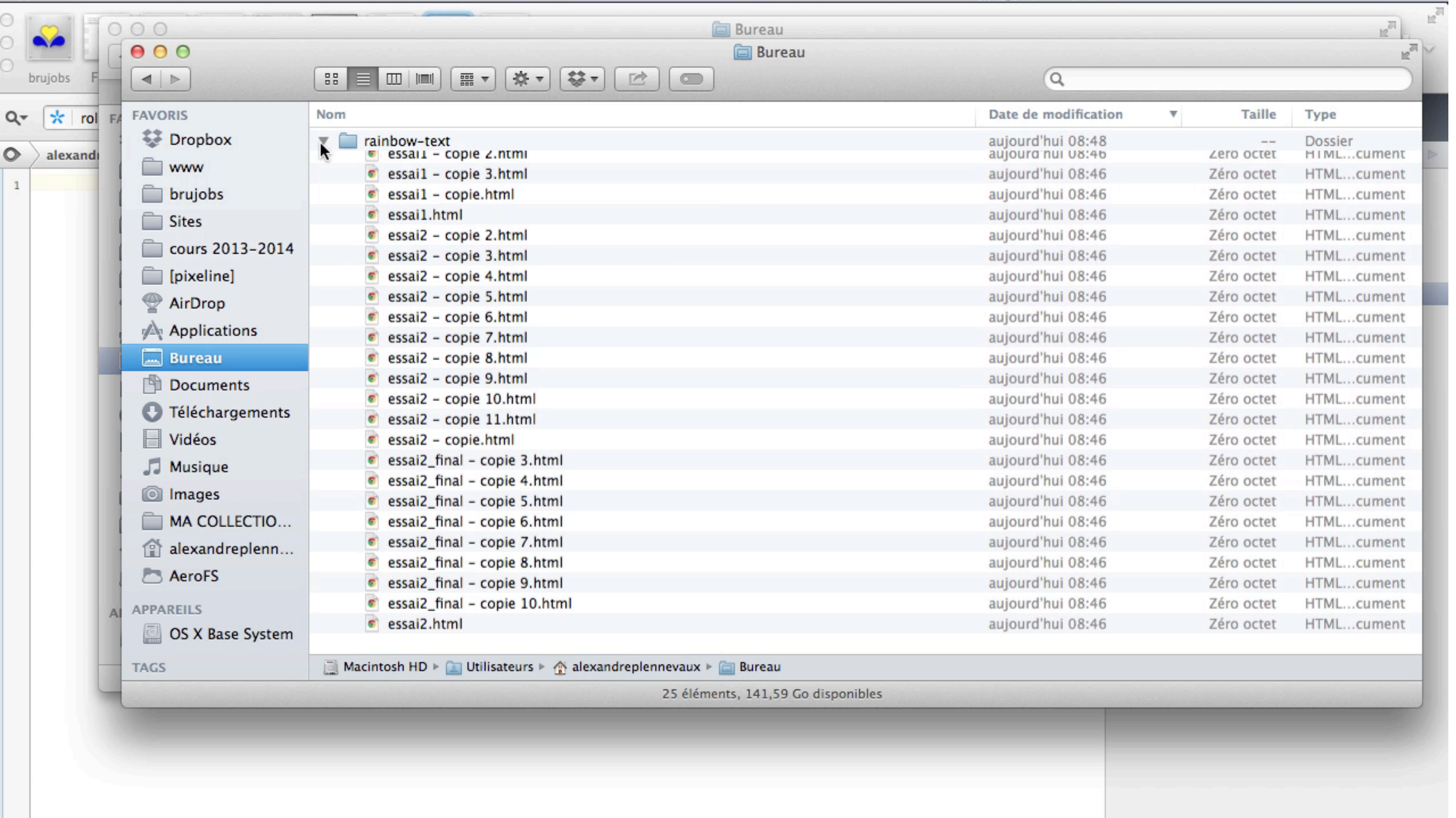
(CTRL+ S sur Windows)

Tu codes un chouette truc: du texte dont la couleur s'anime **de toutes les couleurs de l'arc-en-ciel**. Tu sauvegardes régulièrement, (crashdisks, vols, on n'est jamais à l'abri, hein mammy?) ⌘-ESSE, ⌘-ESSE.

Tu arrives à un bon premier résultat. Tu te dis: "tiens, je vais animer les lettres en fonction de l'orientation des planètes.

...

Quelques heures plus tard, tu n'arrives à rien, c'est la cata.
Ton dossier de travail ressemble à ceci:


Bureau
Bureau
FAVORIS
Dropbox
www
brujobs
Sites
cours 2013-2014
[pixeline]
AirDrop
Applications
Bureau
Documents
Téléchargements
Vidéos
Musique
Images
MA COLLECTIO…
alexandreplenn…
AeroFS
APPAREILS
OS X Base System
TAGS
Nom
Date de modification
Taille
Type
rainbow-text
aujourd'hui 08:48
--
Dossier
essai1 - copie 2.html
essai1 - copie 3.html
essai1 - copie.html
essai1.html
essai2 - copie 2.html
essai2 - copie 3.html
essai2 - copie 4.html
essai2 - copie 5.html
essai2 - copie 6.html
essai2 - copie 7.html
essai2 - copie 8.html
essai2 - copie 9.html
essai2 - copie 10.html
essai2 - copie 11.html
essai2 - copie.html
essai2_final - copie 3.html
essai2_final - copie 4.html
essai2_final - copie 5.html
essai2_final - copie 6.html
essai2_final - copie 7.html
essai2_final - copie 8.html
essai2_final - copie 9.html
essai2_final - copie 10.html
essai2.html
aujourd'hui 08:46
Zéro octet
HTML…cument
Macintosh HD ▸ Utilisateurs ▸ alexandreplennevaux ▸ Bureau
25 éléments, 141,59 Go disponibles

DVM

Cela te parle?

DVM

No more.

Get Get Get Get Get
GIT HUP!

# DÉVELOPPER LA GITHUB ATTITUDE

*. . . ou comment le développement web est un conte de Perrault.*

DVM

se prononce: "guide heupe".

(vient probablement d'une chanson de James Brown.)

GITHUB
est logotypé par un octocat
= un chat-pieuvre
= une pieuvre chatte
= un octopussy ?

autrement dit:

Un chat qui n'a pas peur de se mouiller.

DVA

Et toi, as-tu peur parfois?

C'est bien.

DVM

Poursuivons...

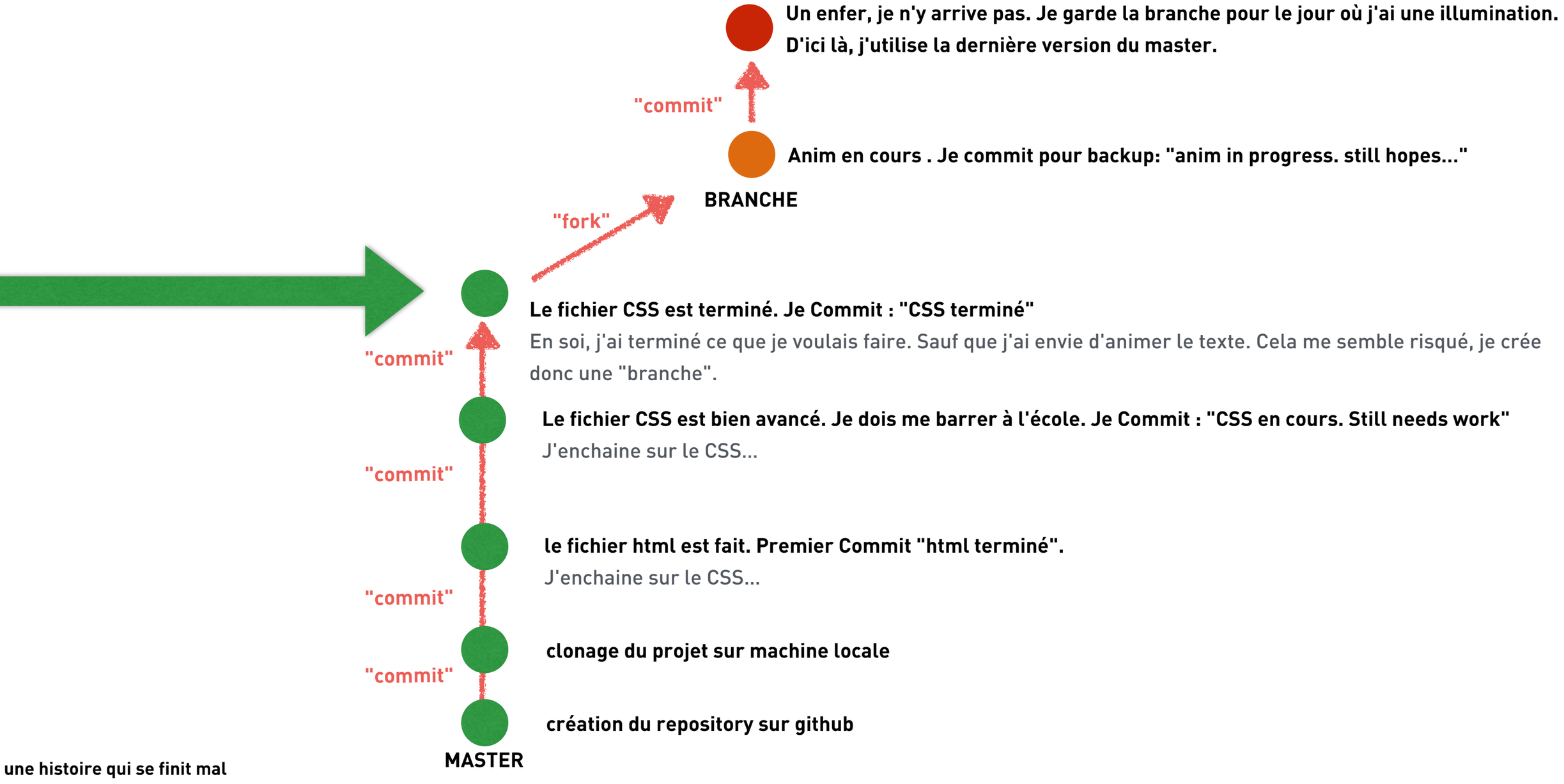
# POURQUOI C'EST COOL ?

- grace à Git vous pouvez travailler sur plusieurs parties du projet de façon complètement parallèle et isolée les unes des autres et sans risque de "casser" ce qu'ont fait les autres.
- **backup automatique**, permettant de revenir à toutes les versions précédentes du projet ("versioning").
- fonctionne **pour tout**, pas que pour le code: html, css, js, actionscript, bash, c++,... Mais aussi slides, contrats, documents word...
- C'est aussi un réseau social autour du partage de code open source, **un bon endroit pour découvrir et apprendre**, comme stackoverflow.com
- Simple comme bonjour, une fois compris le principe.

DVVA

rainbow text qui se finit mal, avec github

rainbow text qui se finit bien, avec github

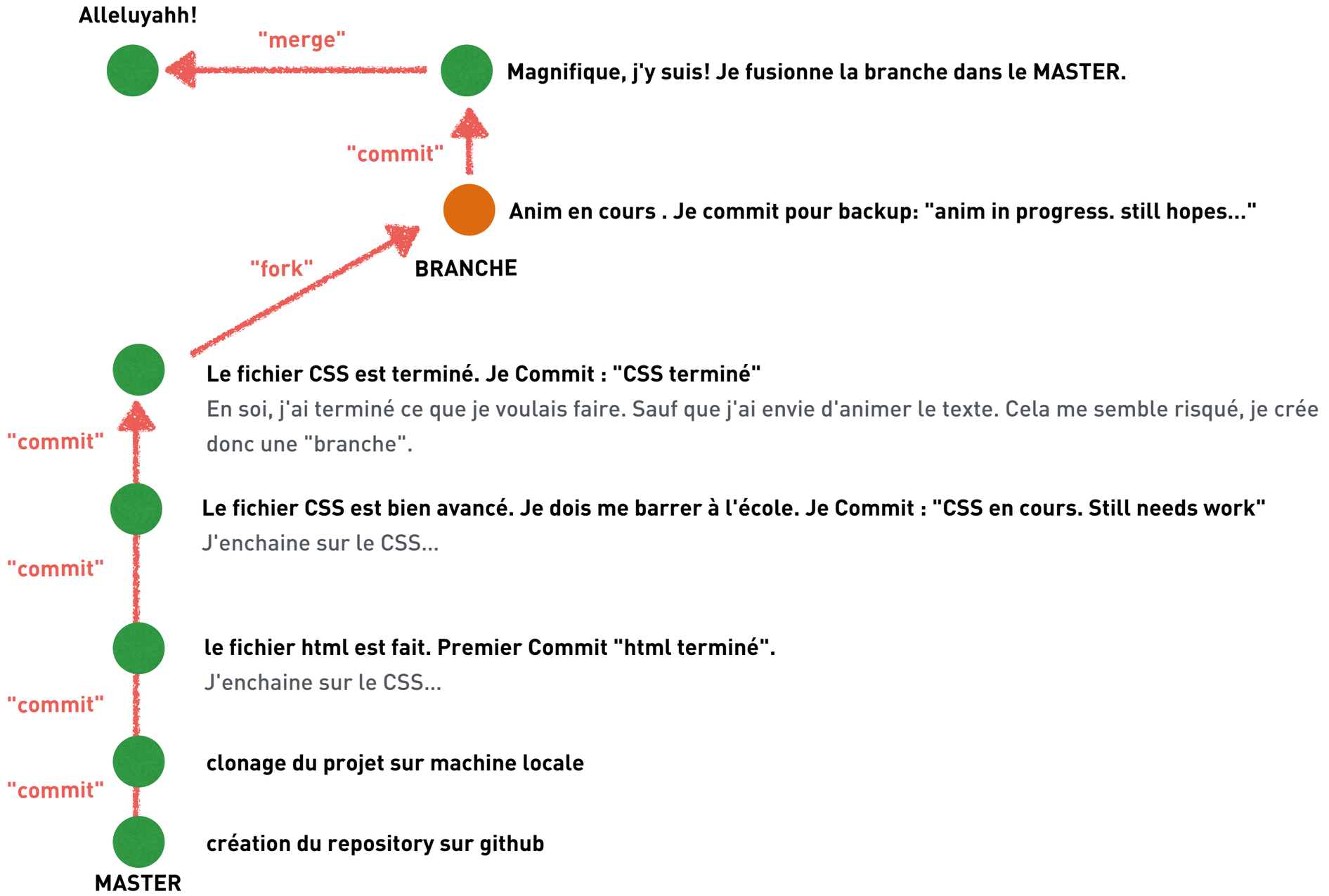



une histoire qui se finit mal

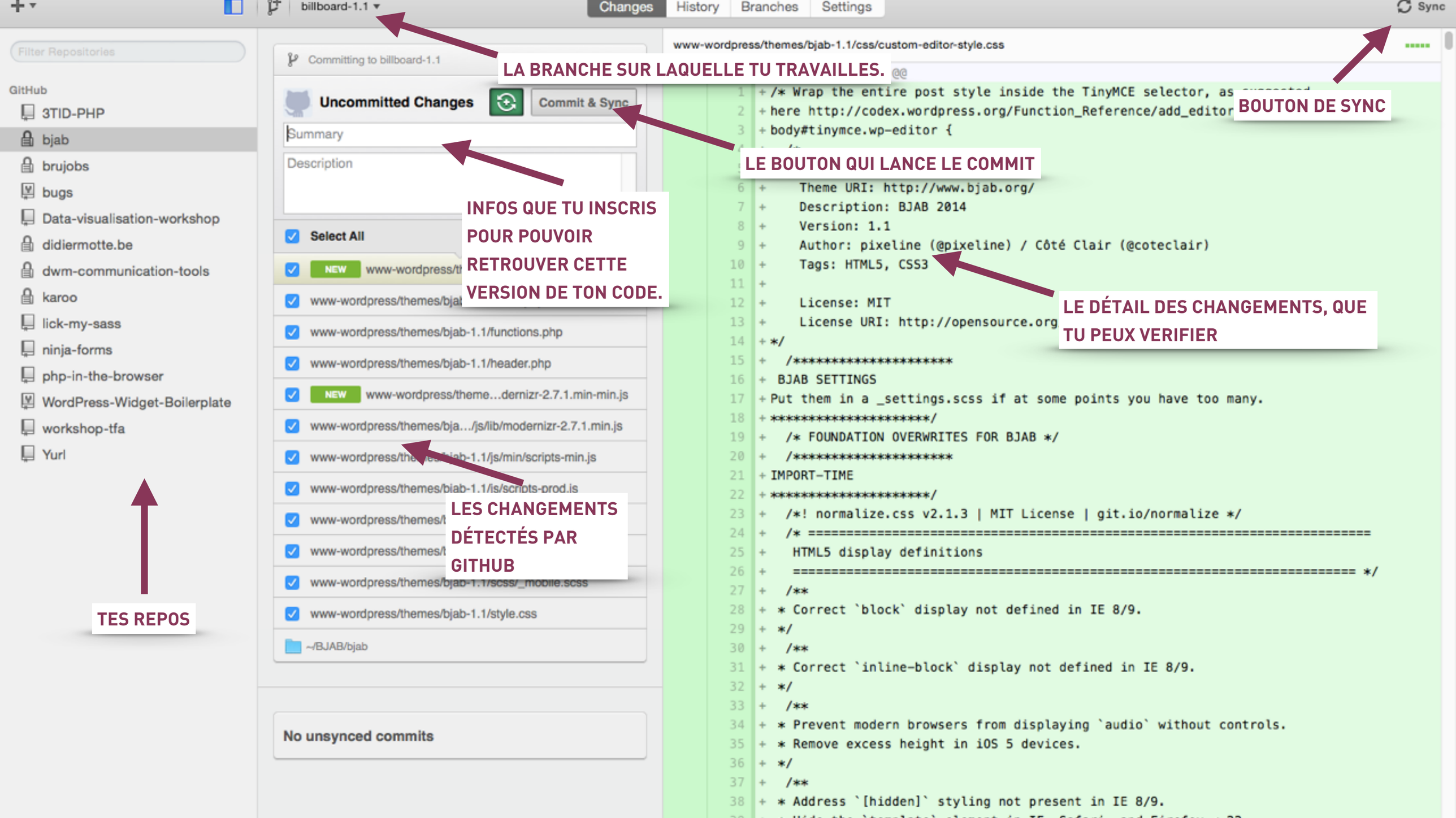
En fait,
c'est comme le petit poucet.
Il ne sait pas où il va,
alors il laisse un petit "commit" pour
pouvoir retrouver son chemin.

rainbow text qui se finit mal, avec github

rainbow text qui se finit bien, avec github




chapitre en cours


billboard-1.1
Changes
History
Branches
Settings
Sync
Filter Repositories
GitHub
3TID-PHP
bjab
brujobs
bugs
Data-visualisation-workshop
didiermotte.be
dwm-communication-tools
karoo
lick-my-sass
ninja-forms
php-in-the-browser
WordPress-Widget-Boilerplate
workshop-tfa
Yurl
Committing to billboard-1.1
Uncommitted Changes
Commit & Sync
Summary
Description
Select All
NEW
www-wordpress/th
www-wordpress/themes/bja
www-wordpress/themes/bjab-1.1/functions.php
www-wordpress/themes/bjab-1.1/header.php
NEW
www-wordpress/theme...dernizr-2.7.1.min-min.js
www-wordpress/themes/bja.../js/lib/modernizr-2.7.1.min.js
www-wordpress/th
jab-1.1/js/min/scripts-min.js
www-wordpress/themes/bjab-1.1/js/scripts-prod.js
www-wordpress/themes/t
www-wordpress/themes/t
www-wordpress/themes/bjab-1.1/scss/_mobile.scss
www-wordpress/themes/bjab-1.1/style.css
~/BJAB/bjab
No unsynced commits
www-wordpress/themes/bjab-1.1/css/custom-editor-style.css
@@
1 + /* Wrap the entire post style inside the TinyMCE selector, as
2 + here http://codex.wordpress.org/Function_Reference/add_editor
3 + body#tinymce.wp-editor {
6 + Theme URI: http://www.bjab.org/
7 + Description: BJAB 2014
8 + Version: 1.1
9 + Author: pixeline (@pixeline) / Côté Clair (@coteclair)
10 + Tags: HTML5, CSS3
11 +
12 + License: MIT
13 + License URI: http://opensource.org
14 + */
15 + /**********************
16 + BJAB SETTINGS
17 + Put them in a _settings.scss if at some points you have too many.
18 + **********************/
19 + /* FOUNDATION OVERWRITES FOR BJAB */
20 + /**********************
21 + IMPORT-TIME
22 + **********************/
23 + /*! normalize.css v2.1.3 | MIT License | git.io/normalize */
24 + /* ==========================================================================
25 + HTML5 display definitions
26 + ========================================================================== */
27 + /**
28 + * Correct `block` display not defined in IE 8/9.
29 + */
30 + /**
31 + * Correct `inline-block` display not defined in IE 8/9.
32 + */
33 + /**
34 + * Prevent modern browsers from displaying `audio` without controls.
35 + * Remove excess height in iOS 5 devices.
36 + */
37 + /**
38 + * Address `[hidden]` styling not present in IE 8/9.
LA BRANCHE SUR LAQUELLE TU TRAVAILLES.
BOUTON DE SYNC
LE BOUTON QUI LANCE LE COMMIT
INFOS QUE TU INSCRIS POUR POUVOIR RETROUVER CETTE VERSION DE TON CODE.
LE DÉTAIL DES CHANGEMENTS, QUE TU PEUX VERIFIER
LES CHANGEMENTS DÉTECTÉS PAR GITHUB
TES REPOS

DVM

# DEMONSTRATION ?

# ÇA VA?

TOUT D'ABORD, VOUS DEVEZ DÉCIDER ENSEMBLE
**COMMENT** VOUS ALLEZ TRAVAILLER AVEC GITHUB.

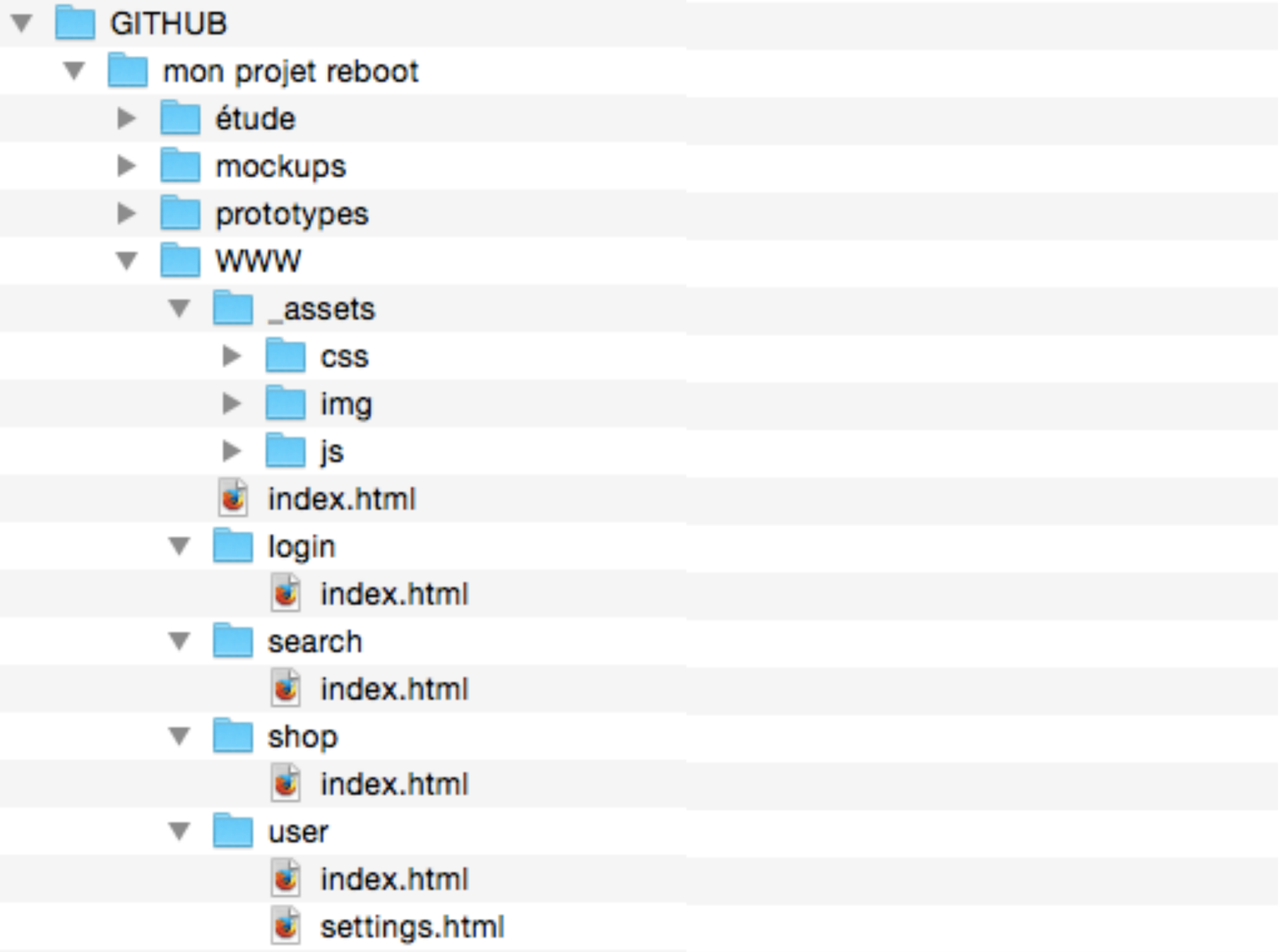
DÉSIGNEZ 1 "REPO MASTER" (LE "LIEUTENANT").

CHOISISSEZ ENTRE 2 STRATÉGIES PRINCIPALES.

# STRATÉGIE 1 : "LES 3 MOUSQUETAIRES"

## VOUS TRAVAILLEZ À PLUSIEURS
## **SUR LES MÊMES FICHIERS**

( c'est le plus "pro" à mon avis)




**exemple organisation de fichiers "Les 3 Mousquetaires"**

serveur de production

serveur de github

ton ordinateur

LE "PROD"

LE REPO

CLONES


## Master (= le stable)

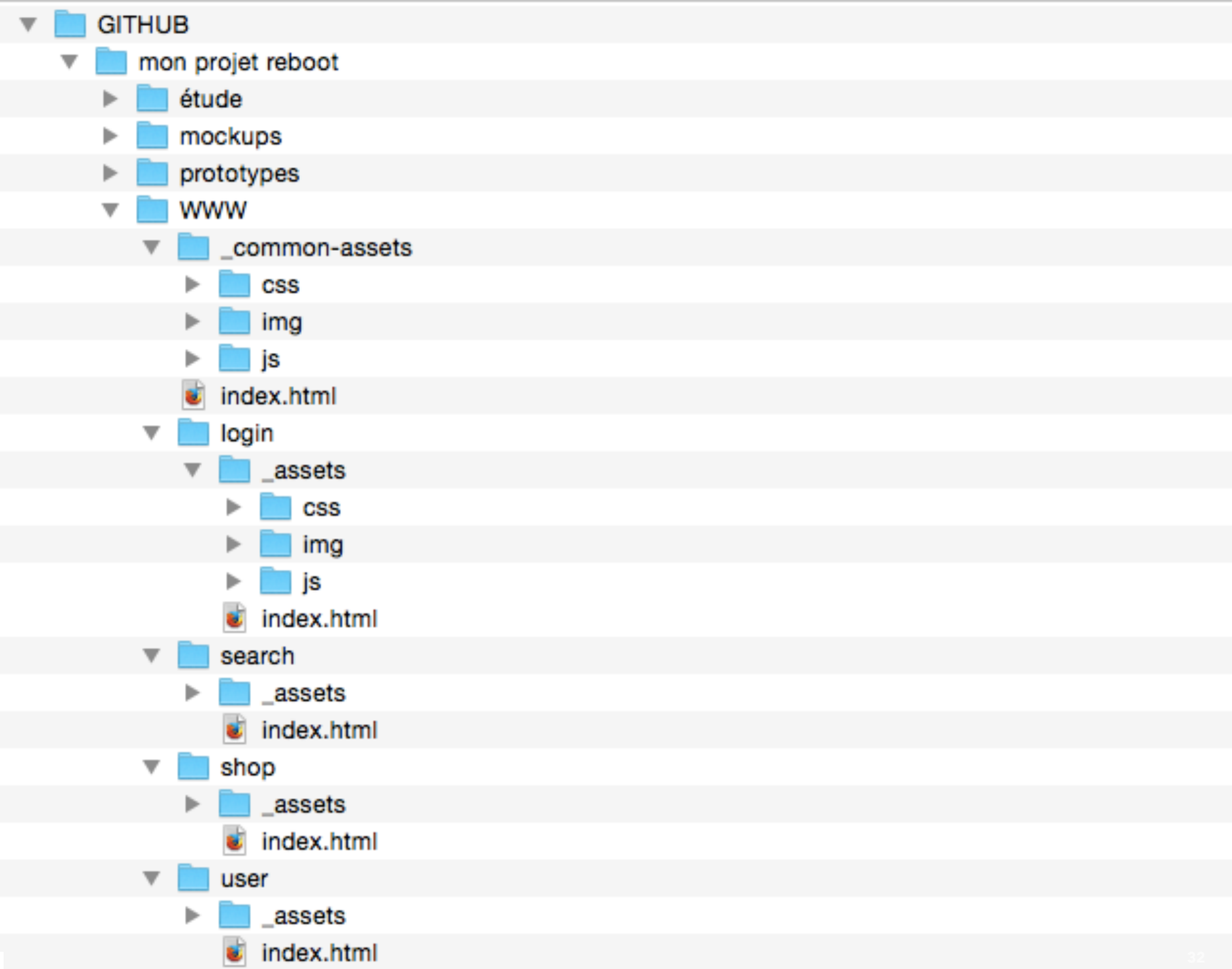
## Branches (= le chantier)

copies locales du master. ("clone")
Lorsque stable > fusion dans master
( = "merge")

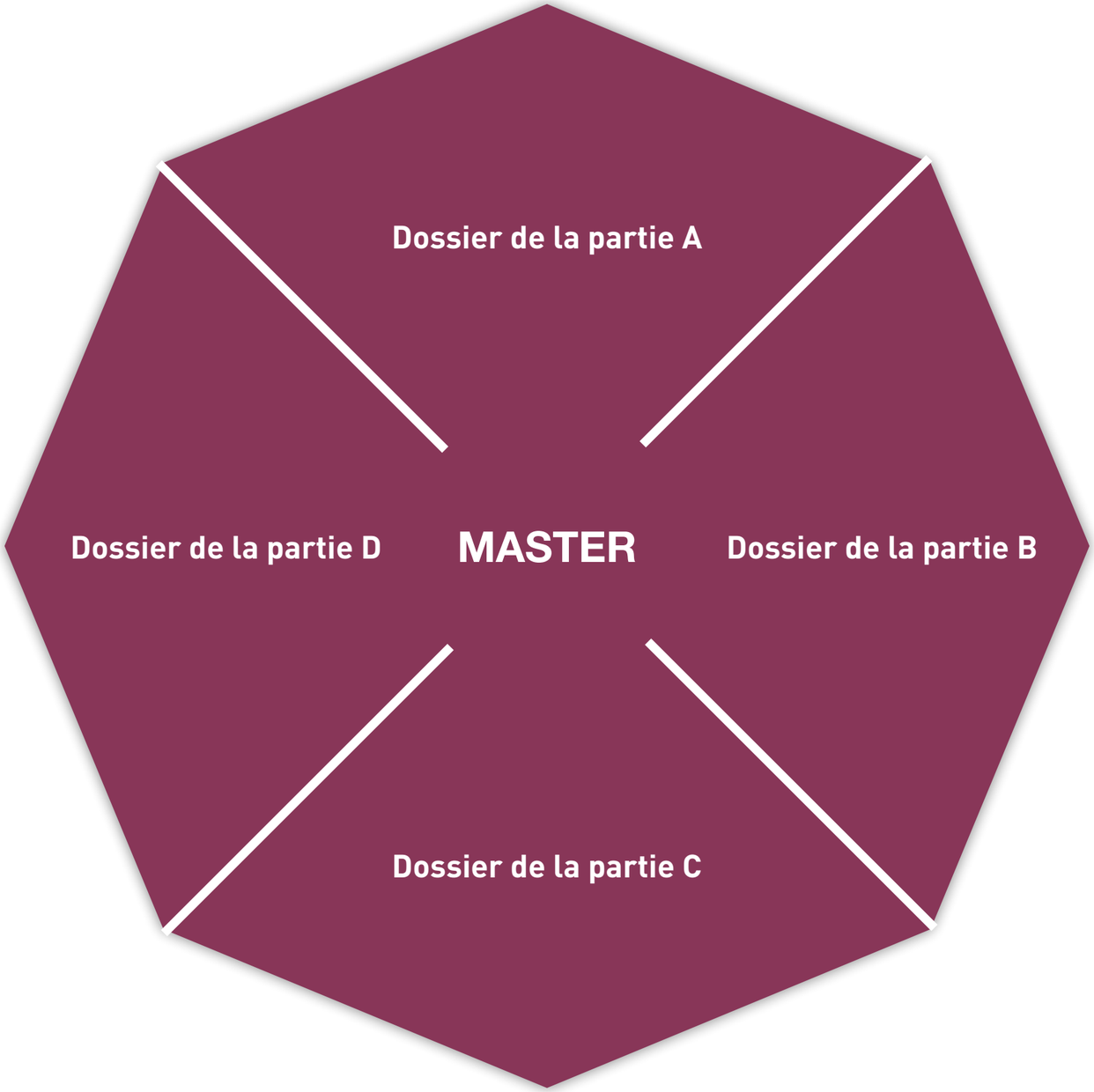
STRATÉGIE 2:

"CHACUN CHEZ SOI"

**CHACUN DANS UN DOSSIER SÉPARÉ**,
AINSI MOINS DE RISQUES DE "CONFLITS".




exemple organisation de fichiers "chacun chez soi"


Dossier de la partie A
Dossier de la partie D
MASTER
Dossier de la partie B
Dossier de la partie C

# MISE EN PLACE DE GITHUB POUR TON PROJET

1. Chaque étudiant crée son compte sur github.com
2. Chaque étudiant installe l'application gratuite de github.
3. Un étudiant crée un "repository" via l'interface en ligne. Il inclut les autres membres comme "collaborateurs". Ainsi, ils pourront aussi travailler sur leur propre machine et faire des "commit".
4. Il clone ce repo dans un dossier sur son disque dur.
5. Dans ce dossier, il y jette ses fichiers de départ. Tout ce qui s'y trouve sera surveillé  par l'application Github.
6. A chaque fois qu'il arrive à un stade "stable", ⌘+S  puis "commit" (= sauvegarde) via l'application. Cela va mettre à jour le repository sur le serveur de Github.
7. Chaque étudiant crée une branche avec ce sur quoi il va travailler. Il commit alors dans sa propre branche.
8. Lorsque son dev est terminé, il fait un commit, puis "merge" sa branche dans le master, pour le mettre à jour.

# ATTENTION À ...

- Toujours commencer sa séance de travail par un "sync", histoire de récupérer les mises à jour faites par les autres.

- Terminer sa séance de travail par un "commit", pour profiter du backup fournit par github. Tant que ce n'est pas stable, ne pas "commiter" dans le Master, uniquement dans sa branche.

- L'application github travaille en arrière plan. Tu peux aussi la fermer après le "sync" initial et la réouvrir pour faire un "commit"; elle détectera instantanément les fichiers qui ont été modifiés.

# S'EXERCER

- crée un repo sur le site de github, intitulé "tisse la toile"
- clone ce repo sur ton disque dur.
- jette dance le clone tes fichiers composant ta création TLT
- Commit dans le "Master".
- Sync. Voilà!

- Le jour où tu veux faire des correctifs, fais-les, puis "commit".
- Le jour où tu estimes que le projet est terminé pour toi à tout jamais, efface ton clone local.
  N'aie crainte; ton projet reste bien au chaud sur Github, si jamais un jour tu voulais faire une nouvelle modif, que devras-tu simplement faire?

# RÉCAPITULATIF

- Qu'est-ce qu'un "repository"?
- Qu'est-ce qu'un "commit"?
- à ton avis, qu'est-ce que veut dire "forker" ?
- qu'est ce qu'une "branche" ?
- qu'est-ce que le "Master" ?
- Qu'est-ce qu'un "merge" ?
- Quelle sont les 2 stratégies suggérées ?

# DWM SUR GITHUB

- github.com/dwmaj = compte "education"
- tu peux y trouver
  - la dernière version du briefing
  - des maquettes de code & demos mis à ta disposition par tes ~~bienfaiteurs~~ professeurs. Clone-les.

# APPROFONDIR

- Tu en sais suffisamment, pour prendre l'outil en main. Poursuis ton exploration, va voir ce que signifie "pull request", explore les "issues", les "github pages", etc.

- https://education.github.com

- Github fundamentals (excellente explication de github)
  http://www.teehanlax.com/blog/github-fundamentals/

- Understanding the github workflow
  http://guides.github.com/overviews/flow/

- Github pour les étudiants
  https://education.github.com/

# FORK

# YOU, DWM!

CRIE!