



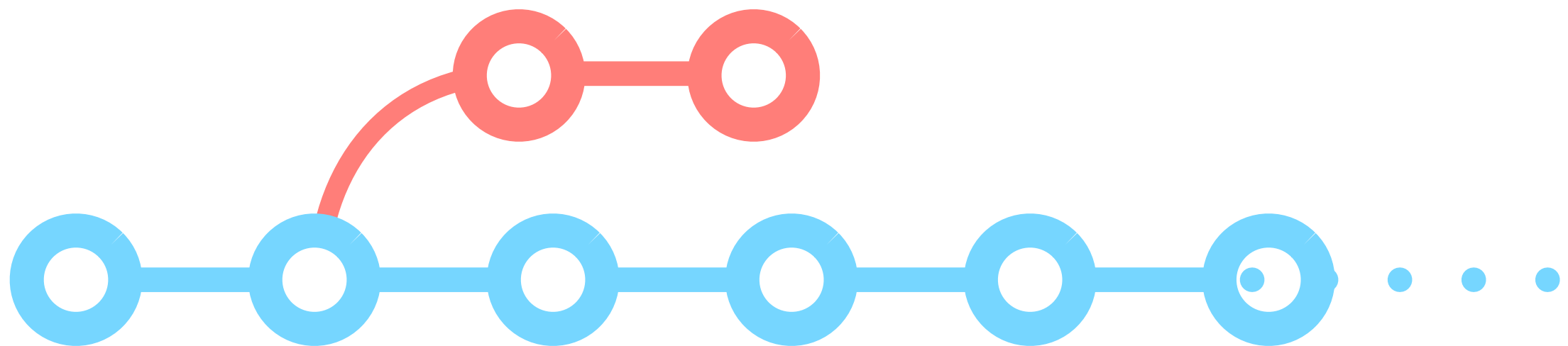
git & github

introduction par @RayFranco

Qu'est-ce que le versioning ?

(en deux slides...)

- **Gestion des versions de fichiers/projets**
- **Enregistrement d'états (snapshots) d'un projet à un instant donné**
- **Gestion de branches pour des évolutions parallèles**



Quelques exemples

- **Git**
- **CVS**
- **Subversion (SVN)**
- **Mercurial**
- **Même... Time Machine ?**

Qu'est-ce que 'git' ?



(en une slide...)

- **Systeme de Gestion de Version Distribué (DRCS)**
- **Créé par Linus Torvalds pour Linux**
- **Décentralisé, contrairement à CVS/SVN**
- **Voilà un sujet parfait pour troller**



Les avantages de 'git' ? et du “décentralisé”



(en une — ou deux — slides...)

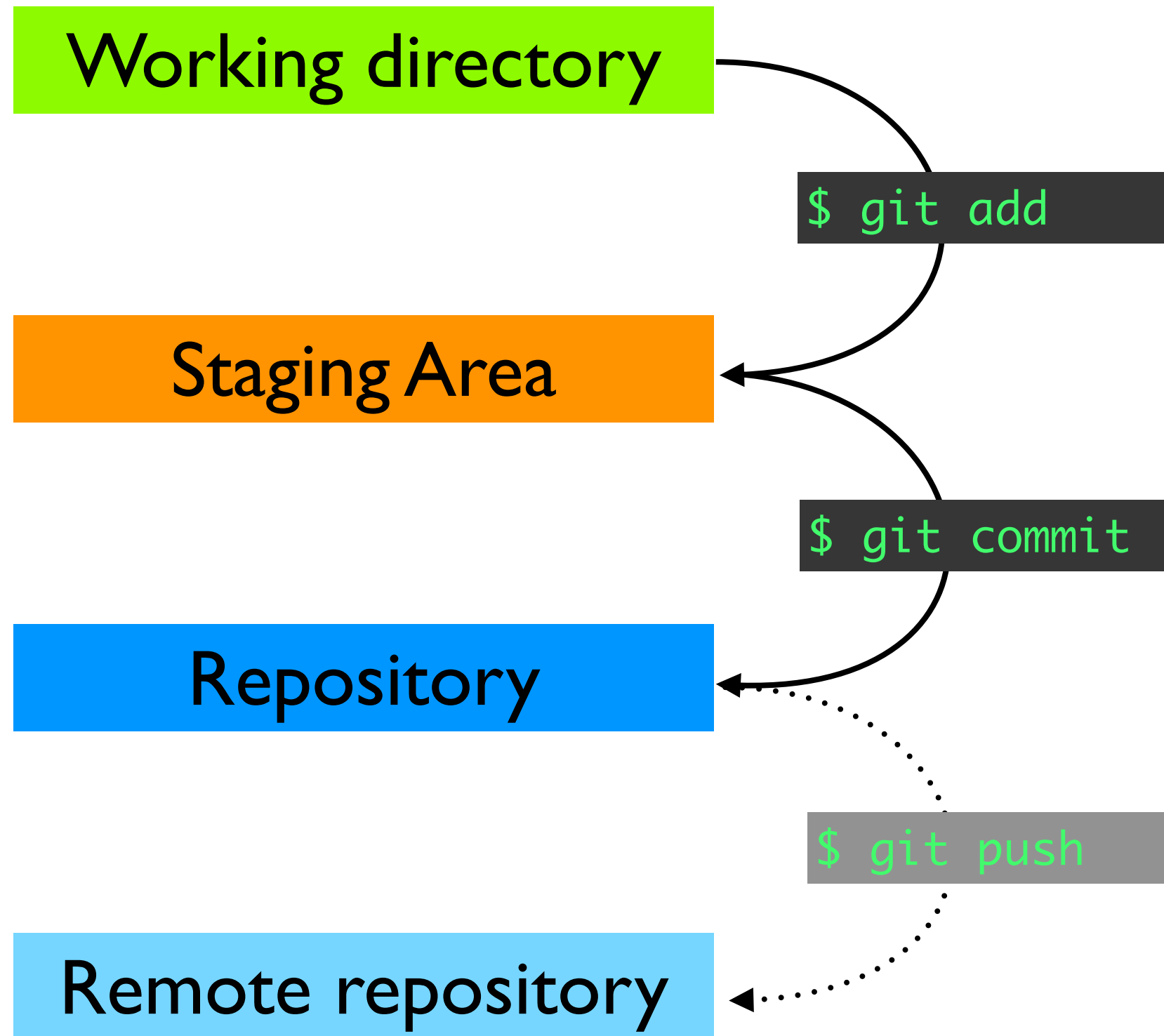


- **Ne pas être dépendant d'une seule machine**
- **Travailler sans être connecté**
- **Rapide car en local**
- **Permet le travail en privé (et sans droits d'écriture)**
- **Permet quand même de garder un (ou plusieurs) dépôt(s) de référence(s)**



“C’est surtout tellement plus **SEXY que **SVN**”**





Let's do it !

Configuration de git

- Renseigner son nom et son email

```
$ git config --global user.name "John Doe"  
$ git config --global user.email "john@doe.com"
```

Création d'un projet

- Créer un dossier “hello”

```
$ mkdir intro-git  
$ cd intro-git
```

- Ajouter un fichier “index.html”

```
$ touch index.html
```

- Editer le fichier et créer un Hello World en HTML

Création d'un projet

```
<!DOCTYPE HTML>
<html lang="fr-FR">
<head>
  <meta charset="UTF-8">
  <title>Hello World</title>
</head>
<body>
  <h1>Hello World</h1>
</body>
</html>
```

Création d'un projet

- Initialiser le dépôt

```
$ git init  
Initialized empty Git repository in ../.git/
```

- Ajout du fichier au dépôt

```
$ git add index.html  
$ git commit -m "First Commit"
```

Vérifier l'état du dépôt

- Utiliser la commande git status

```
$ git status  
# On branch master  
nothing to commit (working directory clean)
```

Rien à commiter, puisqu'on vient de le faire !

Modification du projet

- **Ajouter du contenu au fichier index.html**

```
<h1>Hello World<h1>  
<p>Ceci est un hello world très puissant !<p>
```

- **Verifier le statut du dépôt**

```
$ git status
```

Modification du projet

```
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will
#   be committed)
#   (use "git checkout -- <file>..." to discard
#   changes in working directory)
#
#   modified:   index.html
#
no changes added to commit (use "git add" and/or
"git commit -a")
```

Nos fichier modifiés ne sont pas “staged”

Modification du projet

- **Stager le fichier avec la commande git add**

```
$ git add index.html
```

- **Vérifions de nouveau l'état du dépôt**

```
$ git status
```

Modification du projet

```
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   modified:   index.html
#
```

Nous sommes prêts à commiter !

Modification du projet

- **Stager le fichier avec la commande git add**

```
$ git commit -m "Awesome paragraph added"
```

```
[master 2c4433b] Awesome paragraph added  
1 files changed, 11 insertions(+), 0 deletions(-)
```

- **Vérifions de nouveau l'état du dépôt**

```
# On branch master  
nothing to commit (working directory clean)
```

Modification du projet

Vous avez fait votre premier commit.
Le projet est désormais sous contrôle de versions.

Vérifions ça !

Visionner les commits

- Voir les commits avec la commande git log

```
$ git log
```

```
$ git log --pretty=oneline
```

```
$ git log --pretty=oneline --max-count=1
```

```
$ git log --pretty=oneline --since='5 minutes'
```

```
$ git log --pretty=oneline --author='John Doe'
```

```
$ git log --pretty=oneline --all
```

Visionner les commits

```
2c4433b6b5cc859b6d4634e72b08390dc380d6c9 Awesome paragraph  
945c7aa2711c5a7dbcf86e7364153954c376065d First Commit
```

Nos commits ont bien été pris en compte !

Visionner les commits

- Ajoutons un alias dans notre fichier de configuration

```
$ nano ~/.gitconfig
```

```
[alias]
```

```
hist = log --pretty=format:@"%h %ad | %s%d [%an]" --graph --date=short
```

- Utilisons notre alias git hist

```
$ git hist
```

```
* 2c4433b 2012-04-29 | Awesome paragraph added (HEAD, master) [Franco Bouly]  
* 945c7aa 2012-04-29 | First Commit [Franco Bouly]
```

Beaucoup mieux non ?

Reverts

- **Annuler les changements d'un fichier pas encore staged**
- **Annuler les changements d'un fichier staged**
- **Annuler les changement d'un fichier commité**
- **Annuler un commit**

Modifier le fichier

- Modifions notre fichier index.html

```
<title>Voici un mauvais titre</title>
```

- Vérifions l'état du dépôt

```
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#    modified:   index.html
#
no changes added to commit (use "git add" and/or "git commit -a")
```

Revert avant le stage

- Checkout au dernier commit de index.html

```
$ git checkout index.html
```

- Vérifions l'état du dépôt

```
$ git status  
# On branch master  
nothing to commit (working directory clean)
```

- Vérifions notre index.html : le titre a bien disparu.

Modifier le fichier + stage

- Modifions notre fichier index.html

```
<title>Voici un mauvais titre</title>
```

- Stager le fichier

```
$ git add index.html
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   index.html
#
```

Revert après le stage

- **Reset le stage area**

```
$ git reset HEAD index.html  
Unstaged changes after reset:  
M   index.html
```

- **Checkout au dernier commit de index.html**

```
$ git checkout index.html
```

- **Vérifions notre index.html : le titre a bien disparu.**

Modifier le fichier + commit

- Modifions notre fichier index.html

```
<title>Voici un mauvais titre</title>
```

- Stager et commiter le fichier

```
$ git add index.html  
$ git commit -m "Mauvais commit d'un mauvais titre"  
[master 0ed3d6a] Mauvais commit d'un mauvais titre  
1 files changed, 1 insertions(+), 1 deletions(-)
```

Revert après le commit

- **Revert le dernier commit (HEAD)**

```
$ git revert HEAD --no-edit  
[master 2d663cd] Revert "Un mauvais commit d'un mauvais titre"  
1 files changed, 1 insertions(+), 1 deletions(-)
```

- **Vérifier l'historique**

```
$ git hist  
* 2d663cd 2012-05-13 | Revert "Un mauvais commit d'un mauvais titre" (HEAD, mast  
* 0ed3d6a 2012-05-13 | Un mauvais commit d'un mauvais titre [Franco Bouly]  
* 2c4433b 2012-04-29 | Awesome paragraph added [Franco Bouly]
```

- **Le revert crée donc un nouveau commit**
- **Vérifions le index.html : le revert a bien eu lieu.**

Le reset d'un peu plus près

- **Reset au commit “Awesome paragraph...”**

```
$ git reset --hard 2c4433b  
HEAD is now at 2c4433b Awesome paragraph added
```

- **Vérifions l'historique**

```
$ git hist  
* 2c4433b 2012-04-29 | Awesome paragraph added (HEAD, master)  
[Franco Bouly]  
* 945c7aa 2012-04-29 | First Commit [Franco Bouly]
```

Attention aux resets sur les dépôts partagés !

Qu'arrive t-il si j'ai oublié quelque chose dans mon dernier commit ?

- **Méthode noob** : Je crée un nouveau commit avec la modif'
- **Méthode like a boss** : J'amend mon commit.

Modifier le fichier

- **Modifions notre fichier index.html**

```
<head>  
  <meta charset="UTF-8">  
  <title></title>  
  <link rel="stylesheet" href="style.css">  
</head>
```

- **Créer une feuille de style “style.css”**

```
$ touch style.css
```

Faire le commit

- **Stager le fichier index.html**

```
$ git add index.html
```

- **Committer**

```
$ git commit -m "Ajout de la feuille de style"  
[master 9554345] Ajout de la feuille de style  
1 files changed, 1 insertions(+), 0 deletions(-)
```

Vérifier le dépôt

- **Juste une vérification de routine**

```
$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be
#   committed)
#
#   style.css
nothing added to commit but untracked files present (use "git
add" to track)
```

- **Ooops, nous avons oublié de stager style.css**
Il n'est donc pas dans le dernier commit.



No problem !

- **Stager le fichier style.css**

```
$ git add style.css
```

- **Amend le dernier commit**

```
$ git commit --amend -m "Ajout de la feuille de style like a boss"
[master e8ce65c] Ajout de la feuille de style like a boss
1 files changed, 1 insertions(+), 0 deletions(-)
create mode 100644 style.css
```

- **Vérifier l'historique**

```
$ git hist
* b6b2d04 2012-05-13 | Ajout de la feuille de style like a boss
(HEAD, master) [Franco Bouly]
* 2c4433b 2012-04-29 | Awesome paragraph added [Franco Bouly]
```



Créer un dépôt

- Créer un dossier “repos”

```
$ cd ../  
$ mkdir repos  
$ cd repos
```

- Cloner notre dépôt en mode --bare

```
$ git clone --bare ../intro-git  
Cloning into bare repository intro-git.git...  
done.
```

- Vérifier la présence de notre “bare repository”

```
$ ls -l  
total 0  
drwxr-xr-x  11 franco  staff  374 May 13 21:33 intro-git.git
```

Utilisons notre dépôt

- Revenons à nos moutons

```
$ cd ../intro-git
```

- Apportons des modifications à style.css

```
body {  
    background: rgb(66,175,255);  
    color: rgb(255,255,255);  
}
```

- Commitons !

```
$ git add .  
$ git commit -m "Ajout du background/colors"  
[master 4edda09] Ajout du background/colors  
1 files changed, 4 insertions(+), 0 deletions(-)
```


Utilisons notre dépôt

- Ajoutons notre remote repository

```
$ git remote add origin ../repos/intro-git.git
```

- Vérifions l'existence de notre remote nommé "origin"

```
$ git remote  
origin
```

- Nous allons pusher le dernier commit vers "origin"

```
$ git push origin  
Counting objects: 5, done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 350 bytes, done.  
Total 3 (delta 0), reused 0 (delta 0)  
Unpacking objects: 100% (3/3), done.  
To ../repos/intro-git.git  
    b6b2d04..4edda09  master -> master
```

Clônons notre dépôt

- Nous allons créer un dossier un peu plus loin

```
$ mkdir ../clones  
$ cd ../clones
```

- Et cloner notre “bare repository”

```
$ git clone ../repos/intro-git.git  
Cloning into intro-git...  
done.
```

- Nous pouvons désormais travailler dans notre nouveau dossier “clone”.

Travaillons sur le clone

- Ajoutons quelques modifications à index.html

```
<p>Ceci est un hello world très puissant, encore plus maintenant !</p>
```

- Commitons ces modifications

```
$ git commit -m "Ajout de texte"  
[master e6cfd12] Ajout de texte  
1 files changed, 1 insertions(+), 1 deletions(-)
```

- Et renvoyons nos modifications au dépôt

```
$ git push origin  
Counting objects: 5, done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 357 bytes, done.  
Total 3 (delta 1), reused 0 (delta 0)  
Unpacking objects: 100% (3/3), done.  
To /Users/franco/Sites/tests/clones/../../repos/intro-git.git  
4edda09..e6cfd12 master -> master
```

Synchronisons notre 1er dépôt

- Retournons d'où nous venons

```
$ cd ../../intro-git
```

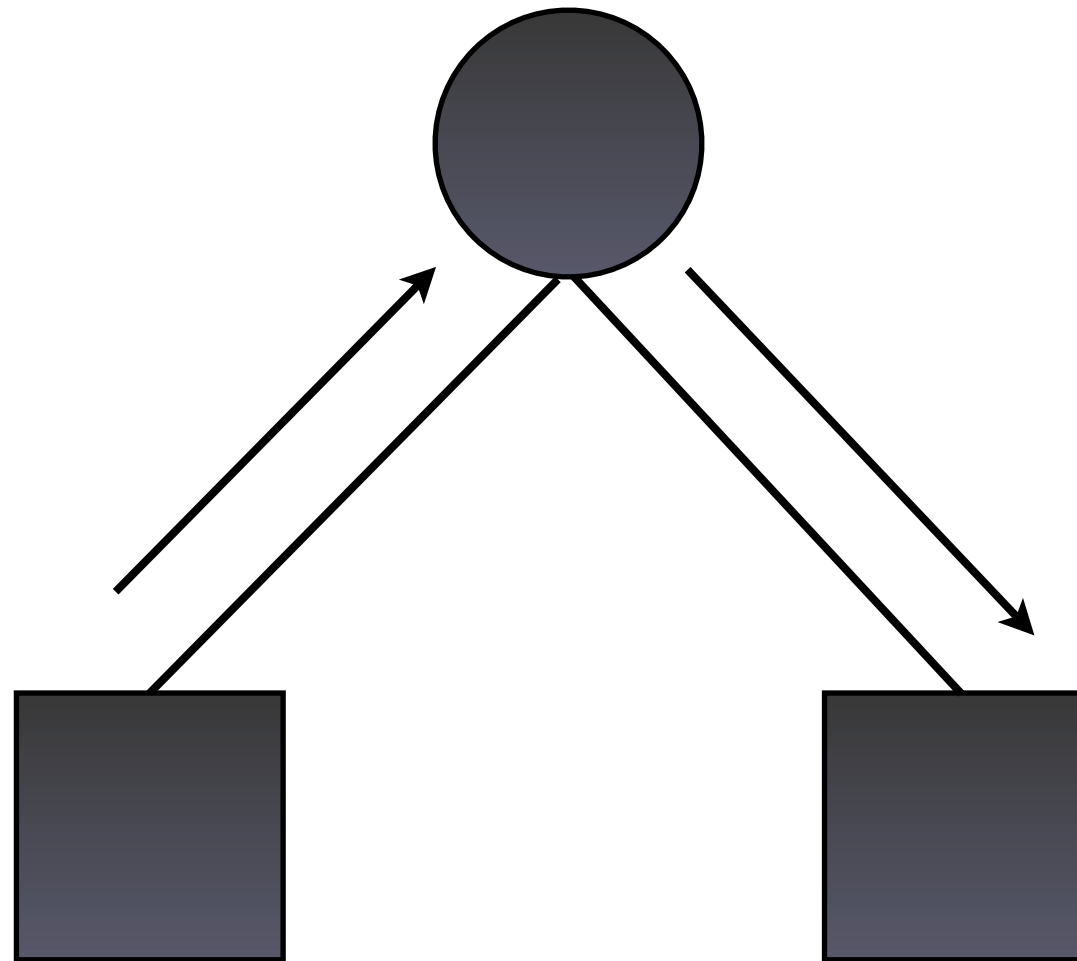
- Vérifions l'historique

```
$ git hist
* 4edda09 2012-05-13 | Ajout du background/colors (HEAD, origin/master, master)
* b6b2d04 2012-05-13 | Ajout de la feuille de style like a boss [Franco Bouly]
```

- Nous ne sommes pas à jour : Nous allons “pull”

```
$ git pull origin master
From ../repos/intro-git
* branch          master      -> FETCH_HEAD
Updating 4edda09..e6cfd12
Fast-forward
 index.html |    2 +-
1 files changed, 1 insertions(+), 1 deletions(-)
```

Bare Repository



Working Repositories

Créer une nouvelle branche

- Vérifiez que vous êtes bien dans le dépôt initial
- Créons une branche 'dev'

```
$ git checkout -b dev  
$ git status
```

- Faisons une modifications sur la nouvelle branche

```
<h1>Hello World</h1>  
  <p>Ceci est un hello world très puissant, encore plus  
maintenant !</p>  
  <p>Nous ajoutons une superbe nouvelle fonctionnalité dans  
la branche dev</p>
```

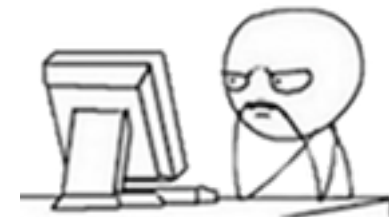
Commitons nos modifications

```
$ git add .  
$ git commit -m "Nouvelle fonctionnalité de ouf"
```

- Nous avons bien commité sur la branche dev

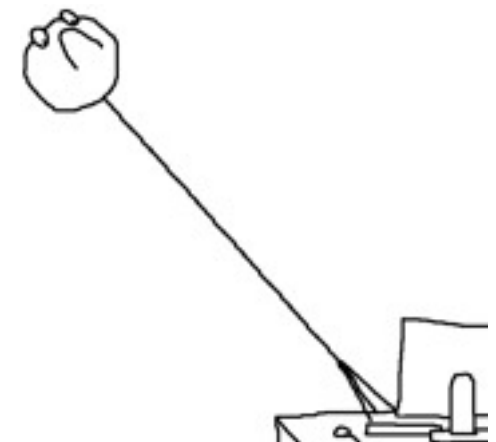
```
[dev 5b48898] Nouvelle fonctionnalité de ouf  
1 files changed, 1 insertions(+), 0 deletions(-)
```

- Mais que s'est-il passé exactement ?



Ce qui s'est passé :

```
$ git hist
* 5b48898 2012-05-25 | Nouvelle fonctionnalité de ouf (HEAD,
dev) [Franco Bouly]
* e6cfd12 2012-05-13 | Ajout de texte (origin/master, master)
[Franco Bouly]
* 4edda09 2012-05-13 | Ajout du background/colors [Franco
Bouly]
* b6b2d04 2012-05-13 | Ajout de la feuille de style like a boss
[Franco Bouly]
* 2c4433b 2012-04-29 | Awesome paragraph added [Franco Bouly]
* 945c7aa 2012-04-29 | First Commit [Franco Bouly]
```



Ooops

**Nous avons oublié quelque chose dans notre
version de production (master)**



Faire un commit sur une autre branche

- Revenons sur la branche master

```
$ git checkout master  
$ git hist
```

- Faisons la modif demandée

```
<h1>Cheers World !</h1>
```

- Commiter les mods

```
$ git add .  
$ git commit -m "Modification du titre"  
$ git hist --all
```

```
* 3cbb8f1 2012-05-25 | Modification du titre (HEAD, master)
[Franco Bouly]
| * 5b48898 2012-05-25 | Nouvelle fonctionnalité de ouf (dev)
|/
* e6cfd12 2012-05-13 | Ajout de texte (origin/master) [Franco
Bouly]
* 4edda09 2012-05-13 | Ajout du background/colors [Franco
Bouly]
* b6b2d04 2012-05-13 | Ajout de la feuille de style like a boss
[Franco Bouly]
* 2c4433b 2012-04-29 | Awesome paragraph added [Franco Bouly]
* 945c7aa 2012-04-29 | First Commit [Franco Bouly]
```

Merger les modifications

- Checkout sur la branche dev (retour a la branche dev)

```
$ git checkout dev  
$ git status
```

- Merger la branche master —> dev

```
$ git merge master  
$ git hist --all
```

```
*      870206d 2012-05-25 | Merge branch 'master' into dev (HEAD, dev)
|\
| * 3cbb8f1 2012-05-25 | Modification du titre (master)
* | 5b48898 2012-05-25 | Nouvelle fonctionnalité de ouf
|/
* e6cfd12 2012-05-13 | Ajout de texte (origin/master)
* 4edda09 2012-05-13 | Ajout du background/colors
* b6b2d04 2012-05-13 | Ajout de la feuille de style like a boss
* 2c4433b 2012-04-29 | Awesome paragraph added
* 945c7aa 2012-04-29 | First Commit [Franco Bouly]
```

Créer un conflit

- Checkout sur la branche master

```
$ git checkout master
```

- Modifier le fichier HTML

```
<span>Le span qui va foutre la merde... Ready ?</span>
```

- Commit !

```
$ git add .  
$ git commit -m "Ajout du span qui va foutre la merde"
```

Faire un merge de merde

- Revenons sur la branche dev

```
$ git checkout dev
```

- Merger les modifs de master dans dev

```
git merge master
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the
result.
```

Résoudre le conflit

- Ouvrir index.html
- Merger les modifs de master dans dev

```
<<<<<<< HEAD
    <p>Nous ajoutons une superbe nouvelle fonctionnalité dans
la branche dev</p>
=====
    <span>Le span qui va foutre la merde... Ready ?</span>
>>>>>>> master
```


Résoudre le conflit

- **Nous allons modifier le fichier comme nous le souhaitons**

```
<p>Nous ajoutons une superbe nouvelle fonctionnalité dans  
la branche dev</p>  
<span>Le span qui va foutre la merde... Ready ?</span>
```

- **Puis commiter le merge manuel de r.d.c**

```
$ git add index.html  
$ git commit -m "Merge master fixed conflict"
```

```
*      870206d 2012-05-25 | Merge branch 'master' into dev (HEAD, dev)
|\
| * 3cbb8f1 2012-05-25 | Modification du titre (master)
* | 5b48898 2012-05-25 | Nouvelle fonctionnalité de ouf
|/
* e6cfd12 2012-05-13 | Ajout de texte (origin/master)
* 4edda09 2012-05-13 | Ajout du background/colors
* b6b2d04 2012-05-13 | Ajout de la feuille de style like a boss
* 2c4433b 2012-04-29 | Awesome paragraph added
* 945c7aa 2012-04-29 | First Commit [Franco Bouly]
```

Faire un commit sur une autre branche

- Revenons sur la branche master

```
$ git checkout master  
$ git hist
```

- Faisons la modif demandée

```
<h1>Cheers World !</h1>
```

- Commiter les mods

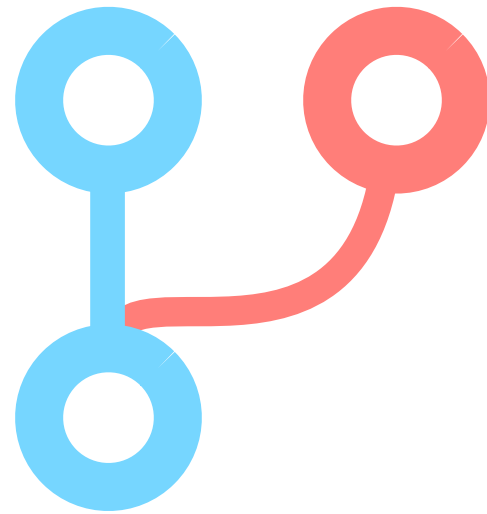
```
$ git add .  
$ git commit -m "Modification du titre"  
$ git hist --all
```

Qu'est-ce que github ?

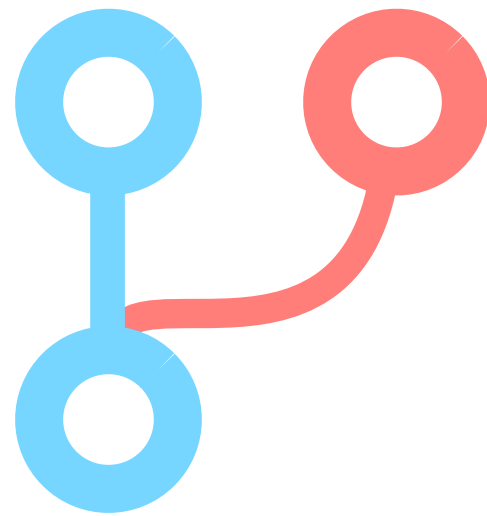


- **Hébergement de dépôts (publics/privés)**
- **Le “Coding Social” revisité**
- **Collaboration sur des projets open-source**
- **Interface graphique et interactive de dépôts**

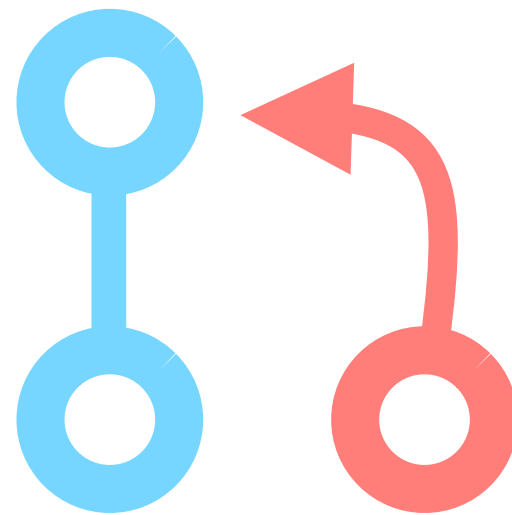
Le vocabulaire



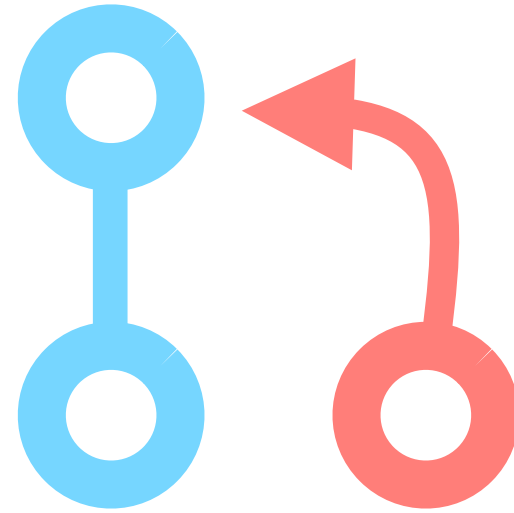
fork



- **Cloner un dépôt public sur son compte github**
- **On va ensuite apporter des modifications à son propre “fork”**
- **Une fois les modifications terminées (et testées) on fait un ...**



pull request
a.k.a “PR”



- **Demande au propriétaire du dépôt d'origine d'y “merger” les “commits” de son “fork”**

(let's say you speak github now !)

Issues

Set up

Création de clés ssh

- <http://help.github.com/mac-set-up-git/>
- <http://help.github.com/win-set-up-git/>

Ajout des clés sur github







- <https://github.com/settings/ssh>


Test des clés

```
$ ssh -T git@github.com  
Hi RayFranco! You've successfully authenticated, but GitHub  
does not provide shell access.
```

Cas d'utilisation : lab215

Organization

 [Explore](#) [Gist](#) [Blog](#) [Help](#)  RayFranco    

 **lab215 (LAB 215)** You are an owner of this Organization! [Edit lab215's Profile](#)

Email

odnl2012@gobelins.fr

URL

<http://odnl2012.gobelins.fr/>

Location

Paris, France

Member Since

Apr 12, 2012

1

Public Repo

8

Private Repos

12

Members

Repositories (9)

All Repositories

Public Private Sources Forks Mirrors

 **trackline-project-ios** Objective-C 3 1

Last updated 2 days ago



 **natum-ios** Objective-C 1 1

Last updated 4 days ago



 **natum-wah** 1 1

Organization Members (12)

 **arkader**
0 public repos, 0 followers [Conceal membership](#)

 **B-Rass**
0 public repos, 0 followers [Conceal membership](#)

 **damienromito** (Damien)
1 public repos, 0 followers [Conceal membership](#)

 **Flo-Schield-Bobby** (Florent SCHILDE)
1 public repos, 0 followers [Conceal membership](#)

 **jlima**
0 public repos, 1 followers [Conceal membership](#)

 **ketchupmoutarde**
0 public repos, 1 followers [Conceal membership](#)

 **nicolas-brousse** (Nicolas Brousse)
4 public repos, 6 followers [Conceal membership](#)

 **RayFranco** (Franco Bouly)
5 public repos, 5 followers [Conceal membership](#)

Open Source

- **Projets d'intérêt public :**
 - **Librairies, SDKs**
 - **Plugins, Bundles**
 - **Projets de démonstration / R&D / Labs**

Exemples

- **Bundle Portfolio**
- **Applications Arles**
- **TweetCanon**
- **SDK et Librairies iOS/Symfony2**
- **Autres projets perso au profit de la classe...**

Projets de fin d'année

Offert par github

- **10 dépôts privés pour toute la classe soit :**
 - **2 dépôts par groupe :**
 - **1 dépôt iOS (nomdeprojet-ios)**
 - **1 dépôt Web (nomdeprojet-web)**

Now clone, commit and push !

Merci !

<http://github.com/rayfranco>