

Projet Foot 2I013

Tuto git

inspiré de <http://www.atlassian.com/git>

<http://www.gitguys.com/topics/>

Nicolas Baskiotis

`nicolas.baskiotis@lip6.fr`

Université Pierre et Marie Curie (UPMC)
Laboratoire d'Informatique de Paris 6 (LIP6)

S2 (2014-2015)

Commandes

- Initialisation : `git init <directory>`
- Copier un projet : `git clone <repository>`
- Configuration : `git config` (voir fichier `gitconfig`)
 - `git config --global user.name <name>`
 - `git config --global user.email <email>`
 - `git config --global core.editor <editor>`
- Changer un fichier : `git add <file>|<directory>`
- Commiter les changements : `git commit`
- Observer les changements : `git status`
- Ignorer des fichiers : fichier `.gitignore` (par exemple `*.pyc`)
- Qui a fait quoi : `git log <option>`

Collaborer

- **Bookmark :** `git remote`
- **Merge :** `git pull <remote>`
- **Transferer :** `git push <remote>`
- **Branche :** `git branch <name>`
 - **Selectionner :** `git checkout <name>`
 - **Supprimer :** `git branch -D name`
 - **renommer :** `git branch -m <branch>`
 - **Transferer :** `git push origin <branch>`
 - **Merger :** `git merge <branche a merger>`

Resume

Entre depots :

- initialiser : `git clone https://github.com/..`
- telecharger les modifications : `git pull`
- uploader les modifications : `git push`

Pour les branches :

- on cree une branche (a partir du master par exemple) :
`git branch mabranche`
- on se met sur cette branche : `git checkout mabranche`
- on modifie, on debug, on sauve dans mabranche sans toucher au master : `git commit`
- je veux revenir au master : `git checkout master`
- je veux definir une nouvelle branche a partir du master :
`git branch autrebranche`
-
- je reviens a ma branche : `git checkout mabranche`
- je veux apporter les modifs de mabranche au master :
`git checkout master; git merge mabranche`