

02-gitbasics

August 21, 2020

1 Git

Git est un logiciel de gestion de versions.

Site officiel <https://git-scm.com>

1.1 GitHub

- Service web utilisant Git comme gestionnaire de version.
- Github permet d'enregistrer et partager toutes les étapes de développement d'un logiciel ou de la rédaction d'un document au format texte au cours du temps. A tout moment vous pouvez revenir à une précédente version, l'ensemble de l'historique est toujours préservé.
- Github permet d'organiser un travail collaboratif avec différents niveaux d'accès, la gestion des bugs et hébergement de la documentation.
- Github est la plus grande bibliothèque de codes du monde.

1.2 SCM (Source Code Management)

- Enregistrement des changements dans un ou plusieurs fichiers au cours du temps.
- Vous pouvez revenir a tout moment vers une version précédente.
- Fonctionne avec tout type de document.
- Excellent moyen pour collaborer sur un même document.
- Chaque modification enregistrée est datée avec le nom de son auteur.
- Les figures proviennent du [Pro Git book](#).
- Ce support est fortement inspiré de "Become a git guru" (<https://www.atlassian.com/git/tutorials>).

1.3 Gestion distribuée des versions

- Chaque membre du projet possède ne copie totale du projet
- On peut contribuer seul ou a plusieurs.
- Pas besoin de connexion réseau
- Les sauvegardes sont très nombreuses grâce à ce système.

1.4 Configuration

Il faut renseigner votre identité avec votre adresse de messagerie

```
git config --global user.name "Prenom Nom"
git config --global user.email "prenom.nom@domaine.fr"
```

1.5 Clonage du dépôt

```
git clone https://github.com/pnavaro/big-data.git
```

1.6 Branche

```
git checkout -b my_branch
```

1.7 Four File status in the repository

1.8 Git Workflow

1.9 Verifier les fichiers modifiés

```
[1]: %%bash  
git status
```

On branch master

Your branch is up to date with 'origin/master'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: 01-introduction.ipynb

modified: 02-gitbasics.ipynb

no changes added to commit (use "git add" and/or "git commit -a")

1.10 Ajouter le fichier dans l'index

```
git add your_notebook_copy.ipynb
```

1.11 Verification avant validation

```
git status
```

1.12 Validation (commit)

```
git commit -m 'Some modifications'
```

1.13 Synchroniser avec la version distante sur le dépôt

```
git fetch origin
```

1.14 Résoudre les conflits

- En cas de problèmes sur un fichier, vous pouvez annuler toutes les modifications avec :

```
git checkout notebook.ipynb
```

- Si vraiment il y a de gros soucis

```
git reset --hard
```

Attention cette dernière commande, annule tous les changements qui n'ont pas été validés.

1.15 Git en version graphique

- [GitHub Desktop](#)
- [Sourcetree](#)
- [GitKraken](#)
- [Fork](#)
- Plugins
 - [RStudio](#)
 - [Eclipse](#)
 - [JetBrains](#)