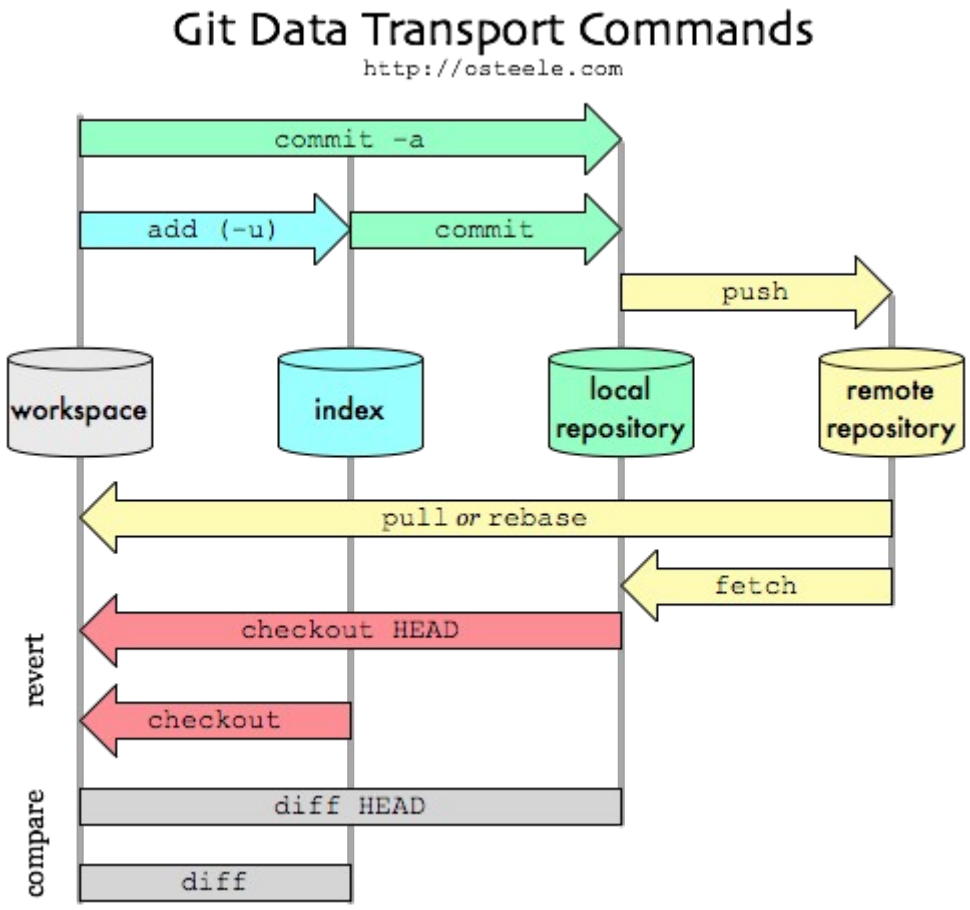
Git permet de versionner votre code source. Il permet, à mesure que vous codez, de placer des « sauvegardes » dans votre code, appelées **commit**, qui vous permettront de revenir dans le passé (en cas de bug) et de collaborer avec vos collègues. Aussi bien seul qu'à plusieurs, **git** deviendra vite votre meilleurs ami. Soyons honnêtes : cet outil est complexe et passera d'abord par l'étape pire ennemi.

Github est un service gratuit pour héberger le code des projets open source (option payante pour les projets privés). C'est très pratique pour rappatrier son code d'un ordinateur à l'autre, pour le présenter de façon élégante aux autres. Cela sert de CV / Book pour un développeur qui postule en entreprise.

Faire le Try Git de Github.

Quand vous publiez du code (autre que des petits exos) sur Internet, pensez à choisir une licence.



Commandes de bases

```
### Installation

sudo apt-get install git-core

### Configurer git

git config --global core.editor "vim"
git config --global user.name "Billy Everyteen"
git config --global user.email "your_email@example.com"

### Créer une paire de clés pour les mettre dans son profil Github

ssh-keygen -t rsa
cat ~/.ssh/id_rsa.pub

### Récupérer un projet

git clone git@github.com:...

### Ou initialiser un nouveau projet

git init
git remote add origin git@github.com:...

### Faire une modification

# 1. Dans le doute, on revient sur master
git checkout master

# 2. On fait notre modification
vim index.html

# 3. Avant d'envoyer notre modification, on récupère ce qu'ont éventuellement
fait nos collaborateurs
git stash
git pull origin master
git stash pop

# 4. Préparons notre commit avec l'une des deux lignes suivantes
git add index.html
git add --all

# 5. On valide le commit (la sauvegarde)
git commit -m"Ajout header"

# 6. On la partage à nos autres collaborateur sur Github
git push [-u] origin master

### Publier sur les Github Pages

# 1. aller sur gh-pages
git checkout [-b] gh-pages

# 2. mettre à jour notre branche
git pull origin gh-pages

# 3. mettre les données de master dans gh-pages
git merge master

# 4. publier
git push origin gh-pages
```

```
# 5. revenir sur master
git checkout master

### Analyser le passé

# 1. trouver le commit que l'on souhaite analyser
git log [-p]

# 2. mettre de coté nos modifications en cours
git stash

# 3. remonter le temps jusqu'à ce commit
git checkout numéro_de_commit

# 4. après l'analyse, on revient dans le présent
git checkout master

# 5. éventuellement, on remet nos modifications
git stash pop

### Annuler un commit

git log [-p]
git revert numéro_de_commit
git push [-u] origin master

### Annuler les add en cours

git reset

### Annuler les modifications en cours

git checkout -- nom_du_fichier
```