

## Projet

### Git et GitHub

Il existe différents outils de gestion de version de code (par exemple SVN ou CVS). Pour votre projet, vous allez utiliser Git. Votre code sera stocké sur le serveur GitHub (GitHub est le principal serveur utilisé par la communauté des logiciels open source, mais il en existe d'autres, comme BitBucket). Voici les avantages d'une telle structure :

- Vous pourrez accéder à votre code depuis n'importe quel ordinateur,
- Vous pourrez travailler en simultané depuis différentes machines sur les mêmes fichiers,
- Vous aurez une sauvegarde en ligne de votre projet,
- Vous pourrez si besoin accéder aux anciennes versions de votre code.

Avant de commencer, chaque membre du groupe doit aller sur la page de GitHub <https://github.com/> et créer un compte.

#### Exercice 1 : Paramétrer Git

Chaque membre du groupe doit, sur sa machine :

1. Installer GIT : <https://git-scm.com/downloads>.
2. Ouvrir un terminal si la machine est sous MacOSX ou Linux, Git Bash si la machine est sous windows (Git Bash sera appelé terminal dans la suite du TP).
3. Configurer git de la façon suivante :
  - Ajouter un nom qui sera utilisé pour connaître qui a envoyé quoi sur le serveur : `git config --global user.name "VotreNom"`
  - Ajouter votre adresse e-mail : `git config --global user.email "votre_email@votre_email.com"`

#### Exercice 2 : Créez un dépôt en ligne

**Un membre** du groupe doit créer un répertoire sur le serveur en ligne. Pour cela, il doit se connecter à GitHub et cliquer sur « New repository », puis spécifier le nom du répertoire.

**Attention :** il vous est proposé d'ajouter un fichier README.md<sup>1,2</sup>. Ne le faites pas pour l'instant. Cependant, il sera utile d'en ajouter un quand le code que vous créerez sera destiné à être partagé. Vous pourrez y spécifier toutes les informations utiles pour utiliser votre programme.

Vous disposez maintenant d'un répertoire en ligne.

---

<sup>1</sup> Qu'est-ce qu'un fichier README ? <https://fr.wikipedia.org/wiki/Readme>

<sup>2</sup> Qu'est-ce que le format .md ? <https://fr.wikipedia.org/wiki/Markdown>

### Exercice 3 : Créez votre dépôt local

A partir d'un terminal, vous pouvez naviguer à travers la hiérarchie des dossiers de votre ordinateur grâce aux commandes `cd` et `ls`. `ls` permet d'afficher le contenu du répertoire courant (essayez en tapant `ls` dans le terminal puis en appuyant sur entrée). `cd` permet de se déplacer dans un répertoire. Par exemple, si après avoir tapé `ls`, vous avez vu que le répertoire courant contient un répertoire « toto », vous devez taper `cd toto` pour rentrer dans ce répertoire. `cd ../` permet de se déplacer vers le répertoire parent du répertoire courant.

En utilisant les commandes `cd` et `ls`, **chaque membre du groupe** doit aller dans le répertoire « Documents » de son ordinateur, puis créer un nouveau répertoire grâce à la commande `mkdir nomRepertoire`. Il doit ensuite entrer dans ce répertoire grâce à la commande `cd` et taper la commande `git init` pour spécifier à son ordinateur que ce répertoire sera dédié à des fichiers Git.

### Exercice 4 : Créer un fichier GIT

**Sur l'ordinateur de la personne qui a créé le répertoire** en ligne, ajoutez un fichier quelconque. Ensuite, tapez la commande suivante pour que git prenne en compte le fichier que vous venez d'ajouter au répertoire :

```
git add votreFichier.
```

Enfin, « consignez » l'état actuel de votre projet grâce à la commande :

```
git commit -m "commentaire (eg. Ajout du fichier votreFichier)".
```

### Exercice 5 : Connecter le dépôt local et le dépôt en ligne

**Sur l'ordinateur de la personne qui a créé le répertoire en ligne**, dans le terminal, tapez la ligne suivante pour connecter les deux répertoires :

```
git remote add origin https://github.com/nomutilisateur/MonProjet.git
```

Tapez ensuite la commande `git push -u origin master` pour envoyer votre fichier git sur le serveur. Allez sur la page GitHub de votre projet et observez que votre fichier a été envoyé.

Dans le répertoire courant, ajoutez un nouveau fichier et envoyez-le sur le serveur :

```
git add nouveauFichier
git commit -m "commentaire"
git push
```

Allez sur la page GitHub de votre projet et observez que votre fichier a été envoyé.

Modifiez le fichier sur votre machine et envoyez le de nouveau (commandes `add` pour prendre en compte le fichier, `commit` pour consigner l'état et `push` pour envoyer).

**Remarque :** Si vous avez modifié plusieurs fichiers, vous devez taper la commande `add` pour tous ces fichiers, puis taper une seule fois les commandes `commit` et `push` :

```
git add fichier1
git add fichier2
git add fichier3
git add fichier4
git commit -m "commentaire précisant les modifications apportées"
git push
```

Trouvez la ligne de commande permettant d'appliquer `add` à tous les fichiers d'un répertoire. Testez.

Vous savez maintenant utiliser git pour stocker des fichiers et les modifier. Nous allons maintenant voir comment le second membre de votre groupe peut accéder et modifier ces fichiers.

### Exercice 6 : Permettre au second membre d'accéder et de modifier le répertoire GitHub

Tout d'abord, **la personne qui a créé le répertoire** sur GitHub doit se connecter au serveur et définir l'autre membre du groupe comme un collaborateur (Settings -> Collaborators).

Ensuite, **l'autre membre du groupe** doit se placer dans le répertoire qu'il a créé à l'exercice 3 et taper la commande :

```
git clone https://github.com/nomutilisateur/MonProjet.git.
```

Les fichiers se trouvent maintenant dans son ordinateur. Il peut en ajouter un nouveau :

```
git add nouveauFichier
git commit -m "commentaire"
git push
```

Pour récupérer ce nouveau fichier, **la personne qui a créé le répertoire** doit taper la commande `git pull`.

Vous connaissez maintenant les principales commandes de git.

Pendant votre projet, la bonne pratique à adopter quand vous commencez à travailler sur les codes est de récupérer tous les fichiers du serveur avec la commande `git pull`, puis d'envoyer régulièrement les fichiers ajoutés ou modifiés avec les commandes `add`, `commit` et `push`.

### Exercice 7 : Supprimer un fichier

Pour supprimer un fichier du serveur, il faut taper les lignes suivantes depuis l'un des répertoires locaux :

```
git rm leFichier  
git commit -m "commentaire"  
git push
```

Supprimez les fichiers que vous avez créés précédemment (n'oubliez pas de mettre à jour l'ordinateur de la personne qui ne supprime pas avec un `pull`).

### Exercice 8 : Ajoutez les fichiers de votre projet

Maintenant que votre répertoire est propre, vous pouvez commencer à travailler sur votre projet. Ajoutez tous vos fichiers code ou autres sur le serveur et avancez sur votre projet.