Git : mise en pratique

Ces exercices vont vous guider pour créer votre (potentiellement) premier dépôt git et vous donner les bases des fonctionnalités pour travailler sur git. On illustrera le comportement de commandes de bases pour récupérer des modifications depuis un dépôt distant ou envoyer des modifications vers un dépôt distant.

Cette fiche d'exercice donnera également le schéma de rendu pour les séances suivantes avec l'utilisation des étiquettes (*tags*). Une étiquette permet de marquer un état spécifique de votre projet

Exercice 1 Créer un dépôt Git avec GitHub

Exercice 1.a Créer un compte GitHub

1. Allez sur `https://github.com`.
2. En haut à droite ou à gauche (selon la largeur de l'écran), cliquez sur le bouton *sign up*.
3. Remplissez avec les informations de votre choix (de préférence une adresse mail pérenne).

À partir d'Octobre 2024, Github risque de demander une autorisation à deux facteurs sous peine de réduire drastiquement les fonctionnalités du site pour une personne connectée. Vous pouvez sans doute passer cette étape pour le moment, mais il faudra sans doute le faire à un moment.

Exercice 1.b Créer le dépôt depuis GitHub

Une fois connecté·e sur `https://github.com`, en haut à droite, cliquez sur "+" puis "*new repository*" comme indiqué sur la figure 1.

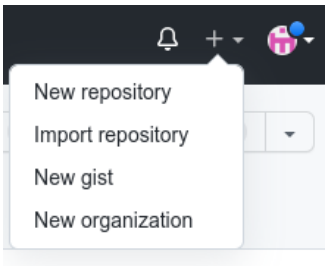

FIGURE 1 – La popup pour créer un nouveau dépôt en ligne.

Cela vous mènera vers une nouvelle page afin de configurer l'initialisation de votre dépôt. La figure 2 vous donne un aperçu de cette page.

Il faudra rentrer les paramètres suivants :
- **Repository Name** : PPE1-2023
- **Description** : Programmation et Projet Encadré 1
- **Public** : cocher
- **Add a README file** : cocher
- **Add a .gitignore** : laisser à *None*
- **Licence** : comme vous voulez
- cliquez sur "create repository"

Exercice 1.c Récupérer le dépôt depuis GitHub

Allez sur votre dépôt nouvellement créé et cliquez sur le bouton vert indiquant "Code" puis allez sur l'onglet "SSH" comme indiqué sur la figure 3. Copiez alors l'URL qui s'affiche. Utiliser le protocole SSH vous permettra de publier vos modifications. En effet, GitHub ne supporte plus de publier les modifications de dépôt via le protocole HTTPS pour des raisons de sécurité [1].

1. Voir : `https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls`

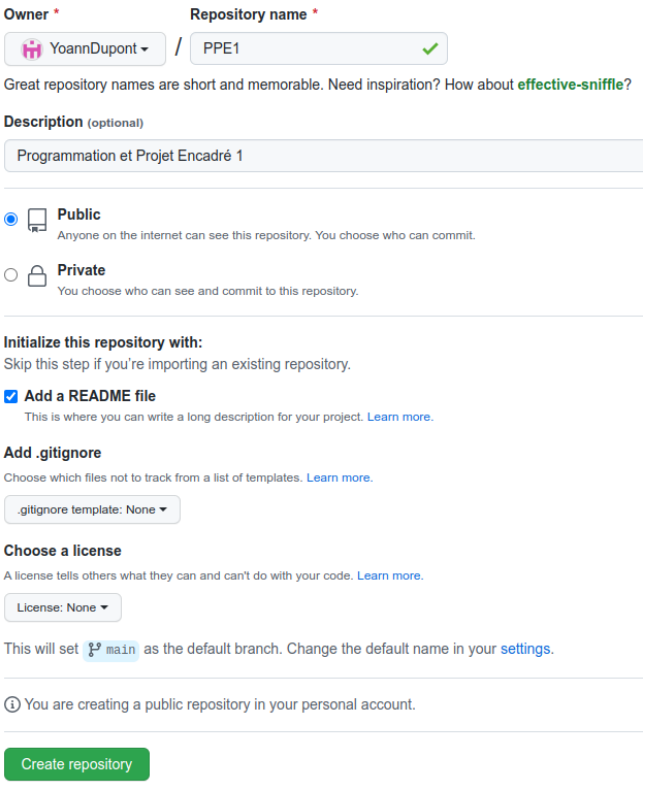

FIGURE 2 – La popup pour créer un nouveau dépôt en ligne. Non, ce dépôt n'existe pas, donc ne vous fatiguez pas à fouiller mes dépôts.

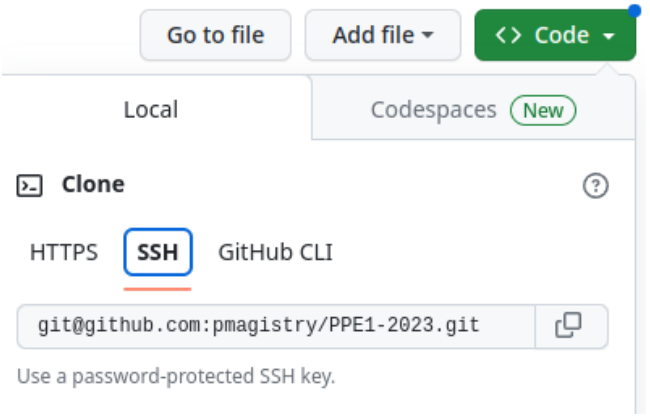

FIGURE 3 – Exemple de popup pour récupérer le lien vers le dépôt. L'URL affichée dépend du dépôt, vous devriez voir autre chose si vous suivez les consignes.

On référera à l'URL que vous avez copiée par <URL> dans la suite. Dans un endroit **rangé** (donc ni le bureau ni votre dossier utilisateur), par exemple `~/cours/plurital/`, ouvrez un terminal et lancez la commande :

```
git clone <URL>
```

Cela créera un nouveau dossier appelé "PPE1-2023" qui sera votre copie locale du dépôt git en ligne. Lancez les commandes suivantes pour voir l'état de votre dossier :

```
cd PPE1-2023/
git status
```

Cela devrait vous indiquer que vous êtes à jour par rapport à la branche main.

Exercice 2 Gérer des modifications

Dans cet exercice, on commencera par laisser tranquille le dépôt local une fois cloné.

Pour la suite, il faut que la commande `git status` indique un message comme celui-là :

```
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Exercice 2.a Faire des modifications

Exercice 2.a.1 Si vous n'avez pas déjà créé votre journal de bord

— Allez sur votre dépôt et cliquez sur "*Add File*" (à côté du bouton "Code") puis "*Create new file*".
— Nommez le fichier "journal.md".
— mettez comme contenu "# Journal de bord du projet encadré" (sans les guillemets).
— Dans "*Commit new file*" plus bas, dans la zone "*Add an optional extended description*", écrivez "Ajout du journal de bord." (sans les guillemets).
— Appuyez sur le bouton "*Commit new file*".

Exercice 2.a.2 Si vous avez déjà créé votre journal de bord

On veut obtenir de `git status` la réponse donnée dans Exercice 2. Faites le nécessaire pour que cela soit le cas en gardant dans un autre dossier tous les changements que vous souhaiteriez garder.

Refaites ensuite les étapes de Exercice 2.a.1, mais en indiquant plutôt que vous aviez déjà un journal dans votre dépôt git.

Exercice 2.a.3 Pourquoi ce journal

Le journal de bord est un fichier `markdown`, un fichier texte suivant certaines conventions que GitHub peut interpréter pour fournir du texte formatté. Vous pouvez trouver un résumé des fonctionnalités principales qu'on vous demandera d'utiliser à l'adresse suivante : `https://www.markdownguide.org/cheat-sheet`.

Le journal de bord vous servira tout au long de cette unité d'enseignement, vous **devrez** y écrire régulièrement pour faire part de votre avancement. Vous devrez y indiquer notamment les problèmes que vous avez rencontrés et les solutions que vous avez trouvées.

Exercice 2.b Synchroniser depuis le dépôt

En parlant de problème, vous en avez un à présent ! Le répertoire que vous avez sur votre machine n'est plus à jour par rapport au dépôt en ligne. Comment pouvez-vous :

1. vérifier que vous êtes en retard par rapport à la version en ligne ?
2. répercuter les changements du dépôt sur votre machine ?

La situation dans laquelle vous étiez est la cause d'un problème assez récurrent qu'on appelle des *conflits*. Lorsque vous travaillez sur votre machine sur une version qui n'est pas celle du dépôt synchronisée, vous pouvez effectuer des changements incompatibles avec la version en ligne. Il convient de synchroniser régulièrement son répertoire afin de minimiser les *conflits*.

Lancez la commande "`git log`" pour voir votre dernier *commit*. Il faut appuyer sur la touche "q" pour sortir du log.

Exercice 2.c Synchroniser vers le dépôt

Modifiez le fichier "journal.md" **sur votre machine** :

— Ajoutez une sous-section (une ligne qui commence par "## ") pour le travail d'ajourd'hui sur git et la manipulation de fichiers.
— Remplissez la sous-section en indiquant le travail que vous avez fait.

Lancez la commande "`git status`" pour vous assurer que git détecte bien les modifications. À présent, quelles commandes devez-vous utiliser pour ajouter vos modifications à votre dépôt ?

Exercice 2.d Ignorer les fichiers macOS inutiles

MacOS a la fâcheuse tendance à créer des fichiers ayant pour nom `.DS_STORE`, qui sont utiles pour macOS mais qui polluent le git et qu'on ne souhaite pas voir. Pour éviter de les voir, créez avec un éditeur de texte un fichier qui s'appelle ".gitignore" et lui donner comme contenu :

```
.DS_STORE
```

Les fichiers dont le nom commence par un "." sont des fichiers cachés sous Linux, il est possible que vous ne les voyiez pas sur votre navigateur de fichiers ou avec un simple `ls` (regardez le man de `ls` pour voir quelle option permet d'afficher ces fichiers qui commencent par un "."). Pour cette raison, il vaut mieux créer ce fichier via un éditeur de texte.

Une fois ce fichier créé et rempli, ajoutez-le à votre git distant avec les commandes déjà vues précédemment.

Exercice 3 Créer un tag sur votre dernier commit

Une fois l'ensemble du travail de la semaine fini (d'autres exercices peuvent être donnés en plus de cette feuille, du *bash* entre autre), créez un tag appelé "seance2" (ou la date de la dernière séance si on a oublié changé) avec le message "version finie séance 2".

Une fois ce tag créé, poussez-le et vérifiez que vous y accédez bien sur GitHub.