

	GIT	
	TP BINOME	

GIT : TP à faire en binôme

Objectif de l'exercice Introduire les participants aux concepts fondamentaux de Git, développer des compétences pratiques pour travailler avec des dépôts locaux et distants, et encourager la collaboration à l'aide de Git.

Pré-requis : Chaque participant doit avoir :

1. Un ordinateur avec Git installé.
2. Un compte sur GitHub.com.

Pré-requis :

Pour chaque étape, donnez la commande et répondez aux questions

Objectif de l'exercice Introduire les participants aux concepts avancés de Git

Pré-requis : Chaque participant doit avoir :

1. Un ordinateur avec Git installé.
2. Un compte sur GitHub.com.
3. Une clef *token* (c.f. <https://www.geeksforgeeks.org/git-how-to-solve-remote-invalid-username-or-password-fatal-authentication-failed/>)

Travail à rendre

Pour chaque étape, donnez la commande et répondez aux questions

I. 1. Étape 1: Configuration initiale

Étape N°1 : Configurez votre nom

Étape N°2 : Configurez votre email

Étape N°3 : Vérifiez les précédentes étapes :

II. Étape 2: Création d'un dépôt local

a) Questions préliminaires

Étape N°4 : Dans quel cas peut-on utiliser `git init` à la place de `git init NomDuDépôt` ?

b) Etapes à réaliser

	GIT	
	TP BINOME	

Étape N°5 : Initialisez un dépôt Git

Étape N°6 : Naviguez dedans

Étape N°7 : Vérifiez qu'une branche *master* ou *main* a bien été créée si ce n'est pas le cas, créez-la.

Étape N°8 : Créez un fichier README.md et ajoutez-y du contenu avec *echo*. Vérifiez avec *cat*.

Étape N°9 : Vérifiez les précédentes étapes. Que dit *git* à propos de README.md ?

III. Étape 3: Premier commit

c) Question préliminaire

Étape N°10 : Donnez la signification de ces deux commandes :

- git log
- git status

d) Étapes

Étape N°11 : Ajoutez le fichier README.md au staging area (zone d'attente)

Étape N°12 : Committez avec un message « Ajout de README.md »

Étape N°13 : Vérifiez les précédentes étapes

IV. Étape 4: Création d'un dépôt sur GitHub et push

e) Questions préliminaires

Voici les nouvelles commandes utilisées dans cette partie :

- git remote add origin [URL_du_dépôt_GitHub]
- git push -u origin master
- git push --set-upstream origin master

Étape N°14 : Expliquez ces trois commandes

Étape N°15 : Quelles sont les différences entre git clone URL et git remote add origin URL ?

f) Étapes à réaliser

	GIT	
	TP BINOME	

- Étape N°16 : L'étudiant 1 crée un nouveau dépôt sur GitHub.
- Étape N°17 : L'étudiant ajoute son binôme comme collaborateur sur ce dépôt via les paramètres GitHub.
- Étape N°18 : Liez votre dépôt local à ce dépôt distant.
- Étape N°19 : Poussez (push) vos changements sur la branche principale
- Étape N°20 : Vérifiez les précédentes étapes sur github

CORRECTION

V. Étape 5: Collaboration - Clonage et modifications

1. Première modification

- Étape N°21 : L'étudiant 2 crée un nouveau répertoire puis clone le dépôt
- Étape N°22 : L'étudiant 2 modifie le fichier README.md, ajoute et committe le changement.
- Étape N°23 : L'étudiant 2 pousse ses modifications.
- Étape N°24 : Vérifiez les précédentes étapes : L'étudiant 1 récupère les modifications

2. Seconde modification

- Étape N°25 : Refaites ces manipulations avec uniquement des *git push* et *git pull* sans l'option *origin main*

VI. Étape 6: Gestion des conflits

- Étape N°26 : Les deux étudiants modifient la même ligne de README.md localement.
- Étape N°27 : L'étudiant 1 ajoute, committe et pousse sa modification.
- Étape N°28 : L'étudiant 2 tente également de pousser après avoir commité.

VII. Comprendre la différence entre git pull et git fetch

Dans cette partie, il faut deux étudiants, notés ÉTUDIANT A et ÉTUDIANT B.

- Étape N°29 : ÉTUDIANT A crée un nouveau dépôt sur GitHub et y ajoute un fichier notes.txt avec le contenu "Première note". Committez "EA : premier commit »
- Étape N°30 : Il partage l'URL du dépôt avec ÉTUDIANT B

	GIT	
	TP BINOME	

Étape N°31 : ÉTUDIANT B clone le dépôt sur son ordinateur avec git clone

Modification par Étudiant A :

Étape N°32 : Étudiant A ajoute à notes.txt le texte "Deuxième note" sur une nouvelle ligne, puis committe et pousse cette modification.

Fetch par Étudiant B :

Étape N°33 : Sans modifier le fichier localement, Étudiant B utilise la commande git fetch.

Étape N°34 : Après le fetch, Étudiant B vérifie le contenu de notes.txt. Que remarque-t-il ?

Comprendre la différence :

Étape N°35 : Étudiant B utilise la commande git log pour voir l'historique des commits. Que remarque-t-il par rapport au dépôt distant ?

Étape N°36 : Étudiant B bascule vers la branche distante pour voir les changements avec : git checkout origin/master et ouvre notes.txt.

Étape N°37 : Pour voir les différences entre votre branche et la branche distante, utilisez la commande :
git diff HEAD..origin/master

Utilisation de git pull :

Étape N°38 : Étudiant B retourne sur sa branche master locale avec : git checkout master.

Étape N°39 : Maintenant, Étudiant B utilise git pull pour récupérer les modifications.

Étape N°40 : Après le pull, il vérifie à nouveau le contenu de notes.txt.

Étape N°41 : En conclure sur git fetch et git pull