**GIT étape par étape**

1. Sur mon poste de travail, je crée un dossier dans lequel sera sauvegardé mon travail.
2. Image
3. Je copie l’adresse du fichier ou j’ouvre PowerShell depuis ce point. (Dans l’invit de commande faire cd puis le nom du dossier)
4. Image
5. Maintenant utilisez la commande “git init” pour dire que ce répertoire est un dépôt git. Si vous allez voir dans le répertoire vous verrez un dossier nommé “.git”.
6. Image
7. Maintenant nous pouvons ajouter des fichiers, pour l’exemple j’ai choisi de créer un fichier README.md avec Visual Studio Code.
8. Image
9. Nous pouvons préparer notre premier commit.
10. Faire un git status (déjà pour voir 😉)
11. Image
12. Nous voyons que dans notre dossier Exercice\_Algo\_Serie\_1\_Exemple, sur la branch master, il nous indique la mention “No commits yet” “pas encore commit” ainsi que la mention “Untracked files” “fichiers non tracés” avec le fichier README.md en rouge.
13. Il nous indique la situation actuelle, “nothing added to commit but untracked files present” “rien n'a été ajouté au commit mais des fichiers non suivis sont présents”
14. Mais pas que ! au-dessus de README.md, “use "git add <file>" to include in what will be committed”, “utiliser "git add <nom du fichier>" pour inclure dans ce qui sera livré”.
15. Donc maintenant nous allons PREPARER notre commit avec la commande “git add <nom du fichier>” ou “git add \*” (add[espace]\*(l’étoile est le symbole pour dire ajouter tout))”
16. Image
17. On constate que le fichier est passé au vert et les modifications sont donc prêtes à être commit !
18. Il est temps à présent de faire notre commit avec la commande “git commit” et un commit se fait avec un petit message (de préférence), cette bonne habitude permet de voire au premier coup d’œil à quoi correspond ce commit (exemple : premier commit, création du fichier README.md ou modification du ficher etc.…).
19. Allons’y : git commit –m “Premier commit et ajout du fichier README”.
20. Image
21. Voilà, LOCALEMENT notre travail est prêt à l’emploi 😉.
22. PREPARATION DU DEPOT DISTANT
23. Maintenant sur GitHub.com nous allons créer un nouveau dépôt DISTANT.
24. Pour le faire connectez-vous à votre compte GitHub (Pour la démo je concidère que vous disposez déjà d’un compte 😊).
25. Cliquez sur la touche new comme sur l’image qui suit.
26. Image
27. Puis remplissez la fiche qui apparaît comme sur l’image 👇
28. Image
29. Owner : est votre pseudo de compte donc sauf si vous avez d’autres compte TPACPC.
30. Repository name : est le nom que vous donnez à votre repository, j’ai choisi délibérément le même nom que mon dossier dans lequel il y a mon dépôt local.
31. Description : ici vous pouvez indiquer à quoi correspond ce repository (optinnel).
32. Public : Si vous choisissez cette option, tout le monde peut voir votre ce référentiel mais vous choisissez qui peut le commit.
33. Private : Cette fois vous pouvez choisir qui peut voir votre référentiel ET qui peut le commit.
34. Add a README file (Ajouter un fichier README) : C'est ici que vous pouvez rédiger une longue description de votre projet. Je l’ai déjà fait donc je ne coche pas cette case.
35. Maintenant vous pouvez cliquer sur “Create repository” en bas à gauche du formulaire.
36. Image
37. Voilà, votre repository est prêt, n’oubliez pas de copier l’URL souligné en jaune vous allez en avoir besoin.
38. Création du lien entre le dépôt local et le dépôt distant.
39. En ligne de commande, git remote add orin <lien URL du dépôt>.
40. AND VOILÀ !! Vous pouvez travailler 😉.
41. À savoir, le soir après votre journée de travail ou si vous changez de projet vos seules commandes seront :
42. Git add \*
43. Git commit –m “petit message”
44. Git push (pour envoyer votre travail sur le serveur distant 😉.

SUR LE PC PERSO

1. Ouvrir PowerShell depuis le dossier de travail local, faire git clone <URL du dépôt distant>
2. Si vous avez déjà exécuté des taches avant, faire un git add \* pour préparer le prochain commit.
3. Faire le commit avec les éléments ajoutés avec git commit -m "votre message"
4. Test avec git push –set-upstream.
5. Ok ça marche je test le git push simple.