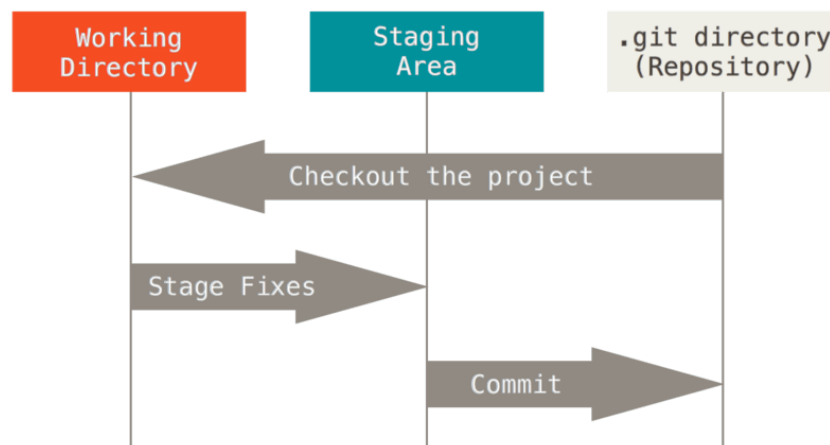


Initiation GIT

Pour obtenir un dépôt Git, on peut soit créer un dépôt vide (typiquement utilisé pour commencer un nouveau projet de développement); soit copier un dépôt existant pour travailler sur cette copie de travail (typiquement utilisé pour collaborer avec les développeurs d'un projet en cours). On parle de `clone` de dépôt.

Pour créer un nouveau dépôt, on utilise la commande `git init monrepo`. Cette commande initialise un dépôt Git dans le répertoire `monrepo` (celui-ci est créé s'il n'existe pas). Ce répertoire contiendra alors ce qu'on appelle l'arborescence de travail (dans `monrepo`) et les données du dépôt Git (dans `monrepo/.git`). L'arborescence de travail permet d'éditer différentes versions des fichiers, tandis que le répertoire `.git` contient tout l'historique du dépôt.



Source de l'illustration : <https://git-scm.com/book>

I) En autonomie, en local

Cette première partie est à réaliser individuellement et ne nécessite pas de connexion internet.

1. Initialiser un nouveau dépôt Git intitulé `recettes`.
2. Créer un fichier texte `mongateau.txt` contenant la recette de votre gâteau préféré.
3. Vérifiez avec `git status` l'état dans lequel se trouve votre dépôt. Vos modifications (l'ajout du fichier `mongateau.txt`) devraient être présentes seulement dans la copie de travail.
4. Préparer `mongateau.txt` pour le prochain commit avec la commande `git add mongateau.txt`, observer à nouveau l'état de votre dépôt à l'aide de la commande `git status`. Votre fichier a dû être ajouté à l'index. Observer les résultats de commandes `git diff` et `git diff -cached`.
5. Commitez votre modification avec `git commit -m "<votre message de commit>"`.
6. Observer l'état de votre dépôt à l'aide de `git status` et `git log`.

7. Créer quelques autres fichiers de recettes et modifier les recettes existantes de manière à avoir au moins 5 commits. Utiliser la commande `git mv` pour renommer un fichier qui était déjà dans le dépôt.
8. Expérimentez la modification du dernier message de commit avec la commande `git commit -amend -m "<nouveau message de commit>"`
9. Utiliser `git log -p` pour observer votre historique de commits avec le détail de chaque changement.
10. Simulons une erreur : modifier une de vos recettes de manière erronée. Ajouter le fichier à l'index avec la commande `git add`. Annulez l'ajout à l'index du fichier avec la commande `git reset`. Vérifiez l'état du dépôt avec `git status`. Utiliser `git checkout monfichier.txt` pour récupérer la dernière version commitée du fichier.
11. Regarder l'historique de votre dépôt avec `git log`; choisir dans la liste un commit (autre que le dernier). Exécuter `git checkout COMMITID` où `COMMITID` est le numéro de commit que vous avez choisi. Observer le résultat de `git log` et `git log -all`.
12. Revenir à la version actuelle de votre dépôt grâce à la commande `git checkout main`.
13. On décide que les données ne sont pas assez structurées, créer une branche `structure` dans laquelle travailler avec la commande `git branch structure`.
14. Observer la liste de vos branches avec la commande `git branch`
15. Basculer sur la branche `structure` avec la commande `git checkout structure`
16. Modifier la description de vos recettes pour qu'elles contiennent chacune trois sections : Titre : Ingrédients : Préparation :, intégrez vos changements à la branche `structure` en plusieurs commits.
17. Basculer sur la branche `main` et modifier le texte d'une recette et commiter le changement.
18. Fusionner la branche `structure` avec la commande `git merge structure`. Traiter les conflits éventuels.
19. Effacer la branche `structure` avec la commande `git branch -d structure`.
20. Lire la doc sur la différence entre `git merge` et `git rebase`. Créer une branche et quelques commits pour expérimenter et procéder à un `rebase`.
21. Quand utiliser `merge/rebase` faire une recherche et synthétiser la réponse.

II) En groupe avec GitLab ou GitHub

- Créer un dépôt privé sur le GitLab <https://moule.informatique.u-paris.fr/>.
- Ajouter vos camarades au dépôt.
- Ajouter chacun vos recettes au dépôt à l'aide des commandes `git pull` et `git push`. Modifier les recettes des autres, organiser les fichiers et corriger quelques fautes de frappes. Expérimenter la résolution de conflits.
- Créer une branche `dev` et des branches de fonctionnalité à partir de `dev`. Procéder à quelques ajouts/modifications de recettes et commits dans chacune des branches. Expérimenter leur fusion.
- Transformer cette collection de recettes en un livre ou un site web de recettes. Choisir une technologie (Latex/Markdown/HTML...), discuter des tâches à réaliser, créer des tickets, discuter les tickets et les résoudre en proposant des modifications.

- Certains fichiers temporaires, exécutable ou de compilation ne devraient pas être suivis par git. Configurez un fichier `.gitignore` pour que certains fichiers ne soient pas partagés entre vos dépôts locaux.
- Faire une 'release' de votre livre/site.

III) Quelques liens :

Cheatsheet : <https://ndpsoftware.com/git-cheatsheet.html#loc=workspace>;

Un super tutoriel pour apprendre les branches : https://learngitbranching.js.org/?locale=fr_FR&demo=