вступ

https://git-scm.com/downloads

https://github.com/

git --version

git config --global user.name "Ваше ім'я"

git config --global user.email "Ваш e-mail"

git config --list

.gitignore (logs/)

git init

git status

git add

git commit -m "Початковий коміт: Додано README проєкту"

git log

git reset hash

git branch

git checkout new-feature

git checkout -b new-feature

git merge new-feature

git branch -d new-feature

Використання SSH-ключів для підключення до GitHub є більш безпечним та зручним, ніж використання паролів (HTTPS). Це галузевий стандарт для взаємодії з такими сервісами. SSH-ключ — це пара криптографічних ключів: публічний та приватний. Публічний ключ ви завантажуєте на GitHub, а приватний залишається на вашому комп'ютері. Коли ви підключаєтесь, GitHub використовує ваш публічний ключ, щоб перевірити, чи маєте ви відповідний приватний ключ, підтверджуючи вашу особу без необхідності вводити пароль.

ssh-keygen -t rsa -C "ваш\_email@example.com"

ssh -T git@github.com

git remote add origin <url\_репозиторію>

git push -u origin main (Прапорець -u (set upstream) встановлює зв'язок між вашою локальною гілкою та її віддаленим аналогом, що дозволяє в майбутньому використовувати просту команду git push.)

git clone

git push

git pull (Завжди виконуйте git pull перед початком нової роботи)

Pull Request — це формальний запит на злиття вашої гілки в основну гілку проєкту. Він дозволяє команді переглянути ваш код (code review), обговорити зміни та переконатися, що все працює правильно, перш ніж зміни будуть інтегровані.

git stash Ця команда бере всі змінені відстежувані файли (як з робочої директорії, так і з індексу), зберігає їх у спеціальному сховищі (стеку схованок) і повертає вашу робочу директорію до стану останнього коміту

git stash push -m "Описове повідомлення": Дуже рекомендована практика — створювати схованку з коротким описом. Це допоможе вам через деякий час згадати, які саме зміни були збережені, особливо якщо у вас кілька схованок.

git stash pop