

Git & GitHub

- [Налаштування Git](#)
- [Ініціалізація репозиторію](#)
- [Перевірка стану репозиторію](#)
- [Додавання файлів до індексації](#)
- [Створення коміту](#)
- [Перегляд історії комітів](#)
- [Робота з віддаленими репозиторіями на GitHub](#)
 - [Клонування репозиторію](#)
 - [Додавання віддаленого репозиторію](#)
- [Відправка змін на GitHub](#)
- [Отримання змін з GitHub](#)
- [Робота з гілками](#)
 - [Перегляд гілок](#)
 - [Створення нової гілки та перемикання на неї](#)
 - [Злиття гілок](#)
- [Отримання довідки](#)

Git — це розподілена система контролю версій, яка дозволяє відслідковувати зміни в коді та ефективно координувати роботу над проектами. GitHub — це веб-сервіс для хостингу Git-репозиторіїв, що надає додаткові інструменти для співпраці, рецензування коду та управління проектами.

Налаштування Git

Перед початком роботи з Git важливо налаштувати ваше ім'я користувача та електронну адресу. Це дозволить відстежувати, хто вносить зміни до коду.

Виконайте наступні команди в терміналі:

```
git config --global user.name "GitHub username"
git config --global user.email <GitHub email>
```

Ці налаштування застосовуються глобально (до всіх ваших репозиторіїв) та дозволяють Git додавати правильну інформацію до кожного коміту.

Ініціалізація репозиторію

Щоб почати працювати з Git у новій папці, необхідно створити локальний репозиторій. Для цього використовується команда:

```
git init
```

Ця команда створює підпапку `.git`, яка містить всю необхідну інформацію про версії вашого проекту.

Перевірка стану репозиторію

Команда `git status` дозволяє переглянути поточний стан робочої директорії та індексації:

```
git status
```

Вона показує, які файли змінені, додані або знаходяться в стані очікування на коміт.

Додавання файлів до індексації

Щоб підготувати файли до коміту, їх потрібно додати до індексації за допомогою команди `git add`:

```
git add filename.txt
```

Або, щоб додати всі змінені файли:

```
git add .
```

Створення коміту

Коміт фіксує поточний стан проекту. Для створення коміту виконайте:

```
git commit -m "Опис внесених змін"
```

Параметр `-m` дозволяє додати короткий опис змін.

Перегляд історії комітів

Для перегляду історії змін використовується команда:

```
git log
```

Вона виводить список комітів з їхніми ідентифікаторами, датами та описами.

Робота з віддаленими репозиторіями на GitHub

Клонування репозиторію

Щоб завантажити існуючий репозиторій з GitHub на свій комп'ютер, застосовується команда:

```
git clone https://github.com/username/repository.git
```

Це створить копію проекту у вказаній папці.

Додавання віддаленого репозиторію

Якщо ви створили локальний репозиторій і бажаєте пов'язати його з віддаленим на GitHub, використайте команду:

```
git remote add origin https://github.com/username/repository.git
```

Команда «origin» — це стандартна назва для віддаленого репозиторію.

Щоб переглянути список доданих віддалених репозиторіїв, використовуйте:

```
git remote -v
```

Це виведе URL-адреси, асоційовані з кожною назвою віддаленого репозиторію.

Відправка змін на GitHub

Щоб надіслати ваші коміти на GitHub, використовується команда:

```
git push -u origin master
```

(або `main`, якщо ваша головна гілка має таку назву). Прапорець `-u` встановлює віддалену гілку як upstream для локальної.

Отримання змін з GitHub

Для синхронізації з віддаленим репозиторієм виконайте:

```
git pull
```

Ця команда завантажує та інтегрує зміни з віддаленого репозиторію у вашу поточну гілку.

Робота з гілками

Перегляд гілок

Щоб переглянути наявні гілки у локальному репозиторії, застосовується команда:

```
git branch
```

Створення нової гілки та перемикання на неї

Для створення нової гілки та одночасного переходу на неї використовуйте:

```
git checkout -b new-feature
```

Злиття гілок

Після завершення роботи над новою функцією її можна злити з основною гілкою. Наприклад, перебуваючи в гілці master, виконайте:

```
git merge new-feature
```

Якщо виникнуть конфлікти, Git повідомить про це, і вам доведеться вирішити їх вручну.

Отримання довідки

Git має вбудовану довідку, яка допомагає ознайомитися з командами та їх параметрами.

- Щоб переглянути загальний список команд, використовуйте:

```
git help -a
```

- Щоб отримати довідку для конкретної команди, наприклад, `commit`, виконайте:

```
git help commit
```

Це відкриє документацію з описом використання, параметрів та прикладів для вибраної команди.