



---

## ЛЕКЦІЯ 3. *Git та GitHub*

# ПЛАН

1. Встановлення та налаштування Git
2. Реєстрація на GitHub
3. Стан файлу у репозиторії.
4. Команди роботи з репозиторієм у терміналі.
5. Завантаження проєкту

# ЛІТЕРАТУРНІ ДЖЕРЕЛА

1. **Git Book** <https://git-scm.com/book/en/v2>
2. **GitHowTo** <https://githowto.com/>
3. **Основи роботи з Git. Базові команди** [https://hyperhost.ua/info/uk/osnovi-roboti-z-git-bazovi-komandi?gad\\_source=1&gclid=CjwKCAjwufq2BhAmEiwAnZqw8tYQJKMQT0BP\\_MPScCsc10IBE1Mg7cM9Duv\\_4mapN1SWfFfBEnHilhoC\\_XMQAvD\\_BwE](https://hyperhost.ua/info/uk/osnovi-roboti-z-git-bazovi-komandi?gad_source=1&gclid=CjwKCAjwufq2BhAmEiwAnZqw8tYQJKMQT0BP_MPScCsc10IBE1Mg7cM9Duv_4mapN1SWfFfBEnHilhoC_XMQAvD_BwE)
4. **Основи роботи з GitHub для Unity** [https://youtu.be/z6GM\\_qb-sUY?si=gne1JKl9cuwDngn6](https://youtu.be/z6GM_qb-sUY?si=gne1JKl9cuwDngn6)



# Встановлення та налаштування Git

## *Інсталяція на Windows*

Є декілька шляхів встановити Git під Windows:

- Офіційна збірка доступна для завантаження з сайту Git. <http://git-scm.com/download/win>
- Ще один простий спосіб встановити Git це встановити GitHub для Windows. Установка включає версію командного рядка Git та графічну теж. Ви можете завантажити GitHub для Windows за адресою <http://windows.github.com>.



# Встановлення та налаштування Git

## Download for Windows

[Click here to download](#) the latest (**2.46.0**) **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **about 1 month ago**, on 2024-07-29.

### Other Git for Windows downloads

#### Standalone Installer

[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#)

#### Portable ("thumbdrive edition")

[32-bit Git for Windows Portable.](#)

[64-bit Git for Windows Portable.](#)

#### Using winget tool

Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version **2.46.0**. If you want the newer version, you can build it from [the source code](#).

# Оновлення Git

**git version** – версія Git встановленої на вашому комп'ютері

```
PS D:\Learn_BS> git version  
git version 2.44.0.windows.1
```

Якщо ваша версія Git 2.14.1 або молодша:

Видаліть Git, завантажте останню версію Git і інсталюйте її знову.

Версія між 2.14.2 і 2.16.1:

Використовуйте команду **git update**

Якщо версія дорівнює або перевищує Git 2.16.1(2):

Використовуйте команду **git update-git-for-windows**





# Реєстрація на GitHub

Для реєстрації на GitHub зайдіть до <https://github.com> :

- оберіть ім'я користувача
- надайте адресу електронної пошти
- вкажіть пароль,
- натисніть зелену кнопку "Sign up for Github"

A screenshot of the GitHub sign-up form. It features three input fields: "Username" with the placeholder "Pick a username", "Email" with the placeholder "you@example.com", and "Password" with the placeholder "Create a password". Below the password field is a note: "Make sure it's more than 15 characters, or at least 8 characters, including a number, and a lowercase letter." A large green button labeled "Sign up for GitHub" is positioned below the form. At the bottom, a disclaimer states: "By clicking 'Sign up for GitHub', you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails."

Username

Pick a username

Email

you@example.com

Password

Create a password

Make sure it's more than 15 characters, or at least 8 characters, including a number, and a lowercase letter.

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

# Обліковий запис на GitHub



Dashboard

Search

Top repositories

New

Find a repository...

Volodymyr-Kachurivskyi/Learn\_BS

Home

Send feedback

Filter

<> Start writing code

Start a new repository for Volodymyr-Kachurivskyi

A repository contains all of your project's files, revision history, and collaborator discussion.

Repository name \*

name your new repository...

Public

Anyone on the internet can see this repository

Private

You choose who can see and commit to this repository

Create a new repository

Introduce yourself with a profile README

Share information about yourself by creating a profile README, which appears at the top of your profile page.

Volodymyr-Kachurivskyi / README.md

Create

1 - 👋 Hi, I'm @Volodymyr-Kachurivskyi

2 - 🤖 I'm interested in ...

3 - 📖 I'm currently learning ...

4 - 🤝 I'm looking to collaborate on ...

5 - 📧 How to reach me ...

6 - 🗨️ Pronouns: ...

7 - ⚡ Fun fact: ...

8

Use tools of the trade

Simplify your development workflow with a GUI

Install GitHub Desktop to visualize, commit, and push changes without ever touching the

Get AI-based coding suggestions

Try GitHub Copilot free for 30 days, which suggests entire functions in real time, right

GitHub Models

Join the Limited Public Beta

Try, test, and deploy a range of popular AI models with an interactive playground, for free, right on GitHub.

Learn more

UNIVERSE'24

Check out our full schedule of AI, DevEx, and security sessions.

Get tickets now to our global developer event on Oct. 29-30.

Get tickets





# Створення репозиторію на GitHub

Dashboard

Type to search

Top repositories

New

Find a repository...

Volodymyr-Kachurivskyi/Learn\_BS

Home

Send feedback

Filter

<> Start writing code

Start a new repository for Volodymyr-Kachurivskyi

A repository contains all of your project's files, revision history, and collaborator discussion.

Repository name \*

name your new repository...

☐ Public

Anyone on the internet can see this repository

☒ Private

You choose who can see and commit to this repository

Create a new repository

Introduce yourself with a profile README

Share information about yourself by creating a profile README, which appears at the top of your profile page.

Volodymyr-Kachurivskyi / README.md

Create

1 - 👋 Hi, I'm @Volodymyr-Kachurivskyi

2 - 🐼 I'm interested in ...

3 - 📖 I'm currently learning ...

4 - 🤝 I'm looking to collaborate on ...

5 - 📧 How to reach me ...

6 - 🗨️ Pronouns: ...

7 - ⚡ Fun fact: ...

8

Use tools of the trade

Simplify your development workflow with a GUI

Install GitHub Desktop to visualize, commit, and push changes without ever touching the

Get AI-based coding suggestions

Try GitHub Copilot free for 30 days, which suggests entire functions in real time, right

GitHub Models

Join the Limited Public Beta

Try, test, and deploy a range of popular AI models with an interactive playground, for free, right on GitHub.

Learn more

UNIVERSE'24

Check out our full schedule of AI, DevEx, and security sessions.

Get tickets now to our global developer event on Oct. 29-30.

Get tickets



# Стан файлу у репозиторії

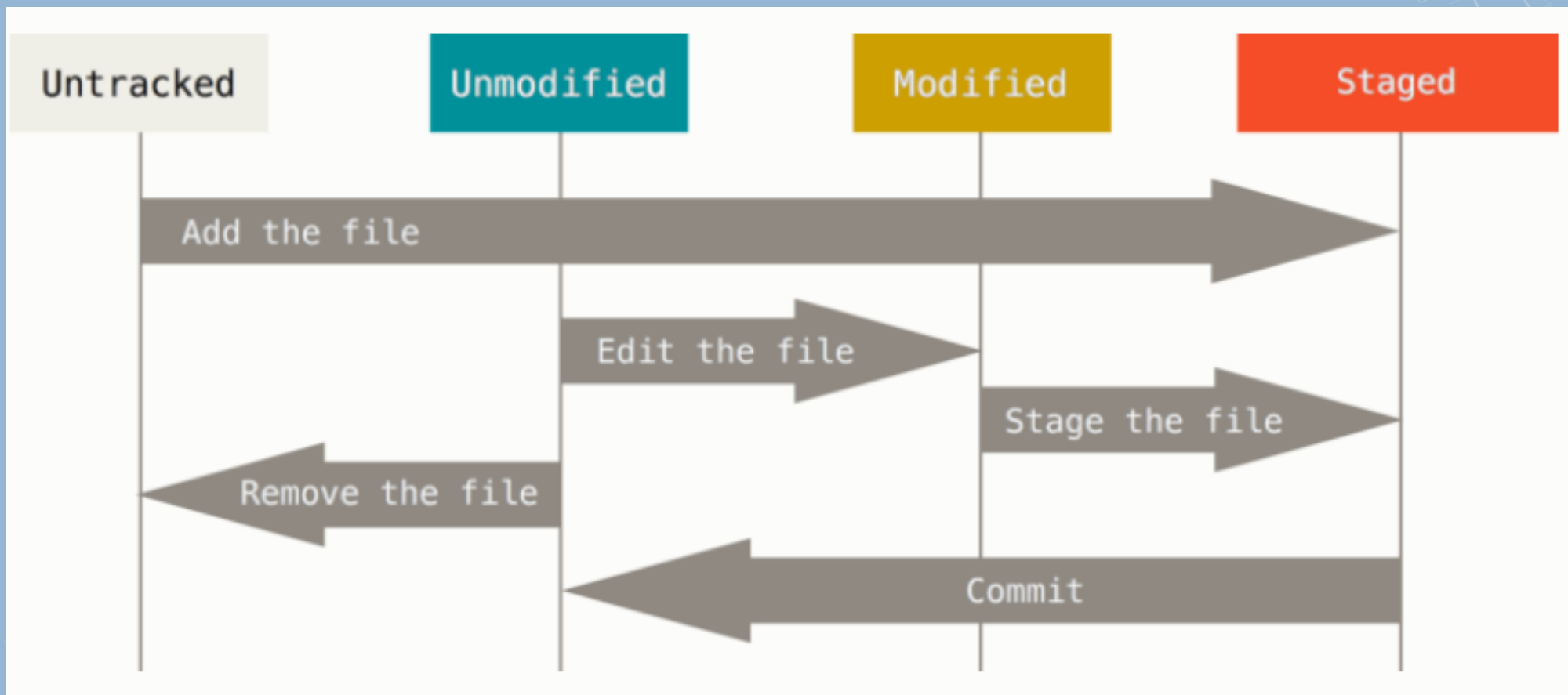
Файл вашої робочої директорії може бути в одному з двох станів:

- **контрольований (*tracked*)**
- **неконтрольований (*untracked*)**.

Контрольовані файли — це файли, що були в останньому знімку. Вони можуть бути не зміненими, зміненими або індексованими. Якщо стисло, контрольовані файли — це файли, про які Git щось знає.



# Стан файлу у репозиторії





# Команди роботи з репозиторієм у терміналі

## Створення Git-репозиторію

Зазвичай Git репозиторій отримують одним з двох способів:

1. Беруть **локальну директорію**, що наразі не під контролем версій, та перетворюють її на сховище Git
2. Звідкілясь **клонують** існуючий Git репозиторій.

У будь-якому разі Ви отримуєте на локальній машині готове до роботи Git сховище.



# Команди роботи з репозиторієм у терміналі

## Створення Git-репозиторію з локального репозиторію

У терміналі активуємо директорію

```
$ cd /d/TestGit
```

```
$ git init
```

```
$ git init
Initialized empty Git repository in D:/TestGit/.git/
User@ghost-11112001 MINGW64 /d/TestGit (master)
$
```

Це створить новий підкаталог `.git`, який містить всі необхідні файли вашого репозиторія — скелет Git-репозиторія





# Команди роботи з репозиторієм у терміналі

## Перевірка статусу ваших файлів

Щоб дізнатись, в якому стані ваші файли, варто скористатись командою **git status**.

```
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

User@ghost-11112001 MINGW64 /d/TestGit (master)
$
```

GitBash

```
PS D:\Learn_BS> git status
On branch main
Your branch is up to date with 'origin/main'.
```

Visual Studio Code

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
Lectures/Lecture#3-Git.pptx
```

```
Lectures/~$Lecture#3-Git.pptx
```

```
nothing added to commit but untracked files present (use "git add" to track)
```



# Команди роботи з репозиторієм у терміналі

## Статус ваших файлів

***new file*** - це файли, які тільки що про індексували (під контролем Git);

***deleted*** - це файли, які були видалені з локального сховища;

***modified*** - це файли, які були змінені, але їх зміна ще не була зафіксована;

***untracked files*** - це файли, які ще не проіндексовані (не під контролем Git).



# Команди роботи з репозиторієм у терміналі

Для додавання файлів під контроль Git: **git add**

**git add index.php**

Якщо потрібно додати відразу всі файли сховища в індекс, використовуйте команду:

**git add .**





# Команди роботи з репозиторієм у терміналі

## Фіксація індексованих змін

Для фіксації змін застосовують команду: **git commit -m**

**git commit -m “Коментар до коміту”**

Дуже зручно, якщо ви бажаєте відразу і додати файли в індекс і зафіксувати зміни, зробити це не двома окремими командами, а просто однією командою так:

**git commit -a -m “Коментар до коміту”**



# Команди роботи з репозиторієм Git

Створення  
локального  
репозиторію

`git init`

добавлення  
в індекс

`git add`

фіксація  
змін

`git commit`

під'єднання  
віддаленого  
репозиторію

`git remote add`

заливка файлів на  
віддалений  
репозиторій

`git push`



# Завантаження проекту

Далі для роботи з *віддаленим репозиторієм* потрібно до нього спочатку підключитися командою:

```
git remote add folder URL
```

де *folder* - назва вашого сховища,

*URL* - адреса вашого сховища.

Щоб подивитися список усіх віддалених репозиторіїв, скористайтесь командою:

```
git remote -v
```

Для додавання інформації в віддалений репозиторій (наприклад, в GitHub, Bitbucket і т.д.) введіть команду:

```
git push
```

```
PS D:\Learn_BS> git remote -v
origin https://github.com/Volodymyr-Kachurivskyi/Learn_BS.git (fetch)
origin https://github.com/Volodymyr-Kachurivskyi/Learn_BS.git (push)
```



# Команди роботи з віддаленим репозиторієм

Якщо працюєте в команді і хочете приєднатися до роботи над уже існуючим проектом, можна просто клонувати віддалений репозиторій собі командою:

```
git clone URL
```

URL - адреса сховища.



# Завантаження проекту

Для спільної роботи над проектом вам потрібно буде “забирати” собі зміни та напрацювання інших учасників проекту. Щоб вилучити данні у свій репозиторій спочатку використовуємо команду:

**git fetch**

При цьому йде з'єднання з віддаленим репозиторієм, забираються зміни, яких у нас ще немає на локальному репозиторії і зберігаються у папці .git. Але цих змін поки немає в нашому робочому каталозі. Для того, щоб вони з'явилися, потрібно далі ввести команду:

**git pull**

# Конфігурування репозиторію

Крім конфігурування URL-адреси віддаленого репозиторію, вам може знадобитися встановити глобальні параметри Git, наприклад ім'я користувача або електронну адресу.

Команда **git config** дозволяє настроїти інсталяцію Git (або окремий репозиторій) з командного рядка.





# Конфігурування репозиторію

Вкажіть ім'я автора, яке використовуватиметься для всіх комітів у поточному репозиторії. Зазвичай для налаштування конфігураційних установок для поточного користувача використовується прапор `--global`.

```
git config --global user.name <name>
```

Ця команда визначає ім'я автора, яке буде використовуватися для всіх комітів, виконаних поточним користувачем.

Додавання аргументу або виконання команди без параметра конфігурації призведе до встановлення значення для поточного локального репозиторію.—  
local user.name

```
git config --local user.email <email>
```

