

Лабораторна робота №1. Git та GitHub. Linux, Command Line.

Мета: Ознайомитись з основними концепціями та інструментами систем контролю версій Git та платформи GitHub, а також навчитися ефективно використовувати їх для керування кодом та спільної розробки.

Вимоги та самостійна робота:

- Зареєструвати акаунт на <https://github.com/> (не з пошти ЧНУ).
- Розібратись з <https://pages.github.com/>.
- Основи git <https://devzone.org.ua/post/git-za-30-khvylyn>.
- Закріпити свої знання про git <https://www.webfx.com/blog/web-design/git-tips/>.

Практичні завдання:

Завдання 1.

1. Пройдіть курс <https://githowto.com/>.

Завдання 2.

1. Пройдіть усі рівні тут <https://learngitbranching.js.org/>.
2. Зробіть скріншот своїх завершених завдань.

Завдання 3.

1. Створіть репозиторій та назвіть його **web-technologies**.
2. Створіть README.md для репозиторію.
3. Опишіть свої враження від вивченого матеріалу у README.md (що для вас було новим, що вас здивувало, що ви плануєте використовувати в майбутньому).
4. Надішліть pull-request сюди <https://github.com/Computer-Science-Department-ChNU/git-learning> пропонуючи зміни.

Як зробити pull-request

1. Форкніть репозиторій <https://github.com/Computer-Science-Department-ChNU/git-learning> через GitHub web-interface.
2. Клонуйте **свій** форк локально на свій комп'ютер: **git clone https://github.com/Computer-Science-Department-ChNU/git-learning.git**.
3. Додайте репозиторій <https://github.com/Computer-Science-Department-ChNU/git-learning> як upstream: **git remote add upstream https://github.com/Computer-Science-Department-ChNU/git-learning.git**.
4. Перейдіть до гілки main (**git checkout main**) і потім створіть нову гілку, ім'я на ваш роздум (aka feature branch): **git checkout -b BRANCH_NAME**.

5. Внесіть деякі зміни до свого локального сховища. Це може бути що завгодно, насправді. Якщо ви знайшли помилку в README - чудово! Подбайте про те, щоб дати своєму PR значуще (осмислене) ім'я та опис.
6. Внесіть зміни до новоствореної гілки (***git commit -m "Your commit message"***).
7. Перейдіть до гілки main: ***git checkout main***.
8. Витягніть останні зміни з гілки upstream main: ***git pull upstream main***.
9. Об'єднайте головну гілку зі своєю гілкою: ***git checkout BRANCH_NAME && git merge main***.
10. Вирішіть будь-які конфлікти мержу, якщо такі є (Resolve merge conflicts).
11. Надішліть гілку до вашого віддаленого сховища: ***git push --set-upstream origin BRANCH_NAME***
12. Зробіть pull-request з вашого репозиторію до репозиторію <https://github.com/Computer-Science-Department-ChNU/git-learning> через GitHub web-interface.
13. Якщо вас попросять виправити merge конфлікт, зверніться до додаткових матеріалів для отримання відповідної інформації.
14. Зауважте, що ваш PR можуть не розглянути швидко.

Завдання 4.

1. Пройдіть всі модулі (4 modules)
<https://linuxsurvival.com/linux-tutorial-introduction/>.
2. Зробіть скріншот своїх завершених модулів і помістіть скріншот в теку task_linux_cli вашого репо **web-technologies**.

Примітка для користувачів Windows: Цей курс забезпечує віртуальне середовище для введення команд. Однак, якщо ви хочете пізніше використовувати ці команди на практиці, вам потрібно буде працювати з Linux CLI, оскільки багато команд не підтримуються Windows cmd або PowerShell. Ви повинні вже встановити git з <https://git-scm.com/downloads>. Він пропонує Git Bash, який можна використовувати для введення команд bash.

Додаткові матеріали.

Вивчіть додаткові матеріали нижче, щоб покращити свої навички. Якщо ви вважаєте, що це вплине на вашу загальну ефективність курсу, подумайте над тим, щоб повернутись до них пізніше, наприклад коли ви виконаєте всі обов'язкові завдання:

1. <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/about-merge-conflicts>
2. <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/resolving-a-merge-conflict-using-the-command-line>
3. <https://ohshitgit.com/>
4. <https://github.com/k88hudson/git-flight-rules>
5. <https://skills.github.com/>

Контрольні запитання:

1. Що таке коміт у Git і які основні компоненти він містить?
2. Як створити нову гілку у локальному репозиторії Git і переключитися на неї?
3. Яка команда використовується для синхронізації локального репозиторію з віддаленим репозиторієм на GitHub?
4. Що таке pull request (PR) і які його основні цілі при роботі з GitHub?
5. Як вирішити конфлікти при злитті гілок у Git?
6. Що таке форк репозиторію і як він використовується в контексті GitHub?
7. Як можна переглянути історію комітів у Git?