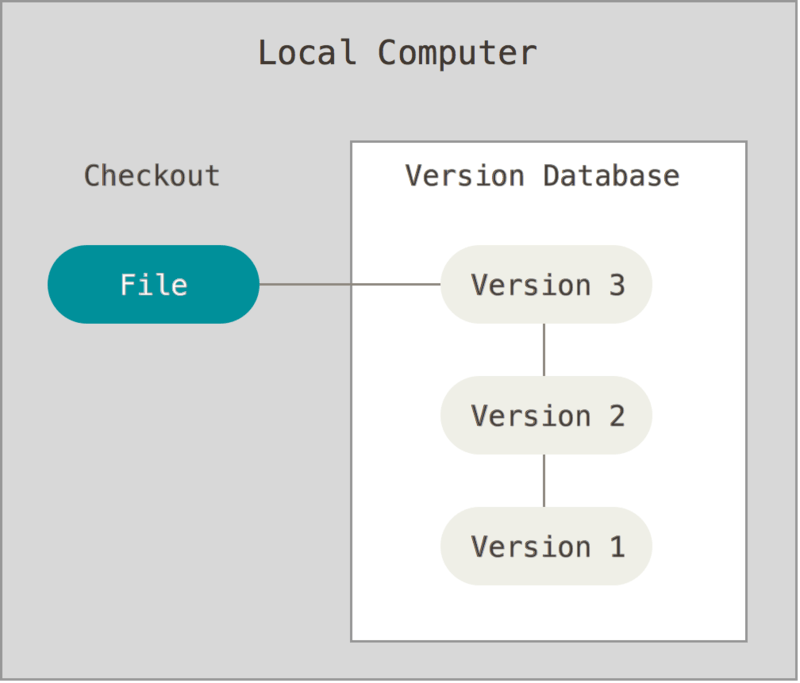
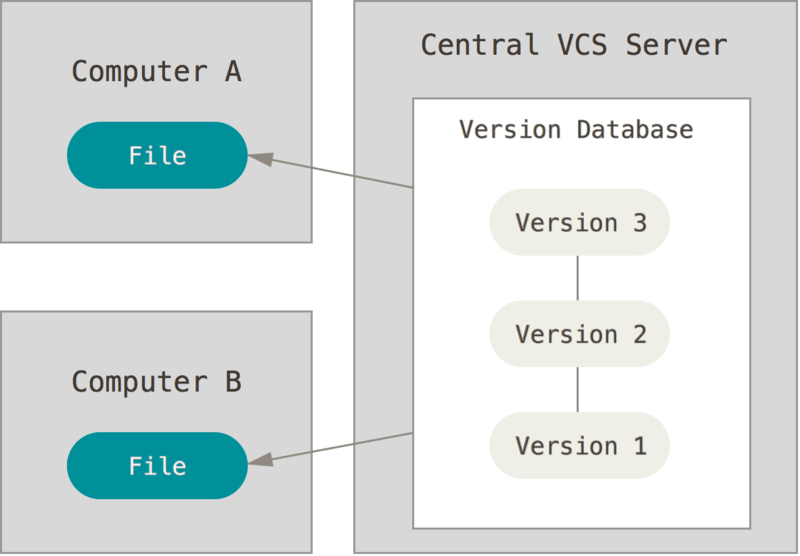
# Системи контролю версій

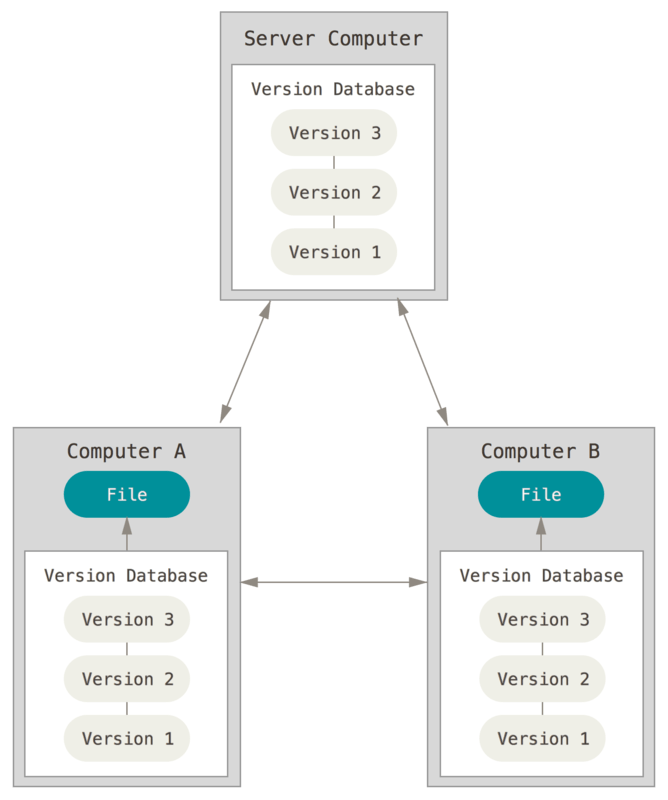
### Локальні



### Централізовані



### Розподілені



[Онлайн підручник](https://git-scm.com/book/uk/v2/%D0%92%D1%81%D1%82%D1%83%D0%BF-%D0%9F%D1%80%D0%BE-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%83-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8E-%D0%B2%D0%B5%D1%80%D1%81%D1%96%D0%B9)

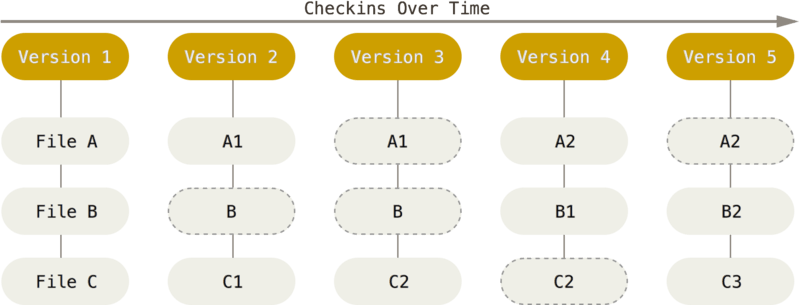
### Репозиторій

Дає змогу:

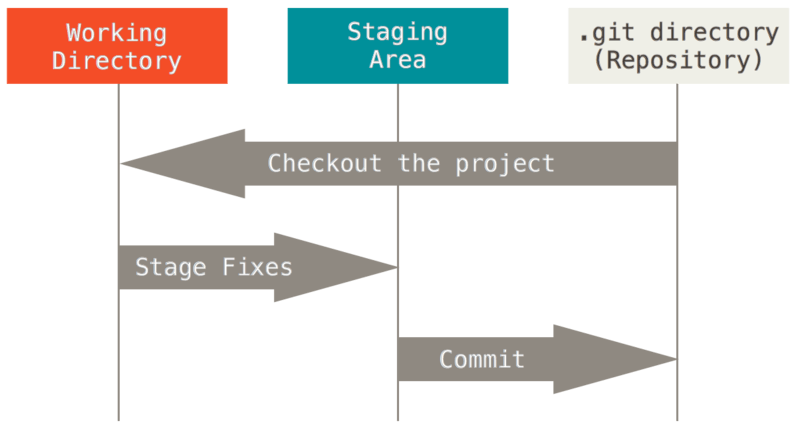
* зберігати ваш код;
* запам’ятовувати історію змін до вашого коду (та дозволяти у будь-який момент побачити хто саме зробив зміни, коли зробив зміни);
* у будь-який момент повернутися до будь-якої версії коду;
* з легкістю об’єднувати зміни різних версій, станів та розробників;
* розробляти проект командою розробників, одночасно працюючи над одними і тими ж модулями і навіть рядками коду;
* готувати ваш код до релізів та публікувати ваш код для зовнішнього світу;
* вести статистику змін до коду;
* ще багато іншого

Для взаємодії з репозиторієм потрібне програмне забезпечення, наприклад, [git-клієнт](https://git-scm.com/downloads/guis). Це програма, яка вміє створити репозиторій, додати до нього файли (чи вилучити), відслідковувати стан файлів, що додані до репозиторію, зберігати їхні версії. Якщо працювати виключно з клієнтом, то будемо використовувати *локальне* сховище. Можна використати альтернативний підхід: створити обліковий запис на [Github](https://github.com/) і використовувати виключно веб-інтерфейс - працювати зі сховищем через браузер. Тоді не потрібно встановлювати на свій комп'ютер додаткових програм, а сховище буде *централізованим*. Найчастіше використовують третій варіант: встановлюють клієнт на свій комп’ютер і створюють обліковий запис на хабі. Тоді є змога щодня працювати зі своїм локальним сховищем, а після отримання суттєвих результатів зберігати їх у сховищі в хмарі. Таким чином користувачі працюють з *розподіленим* сховищем.

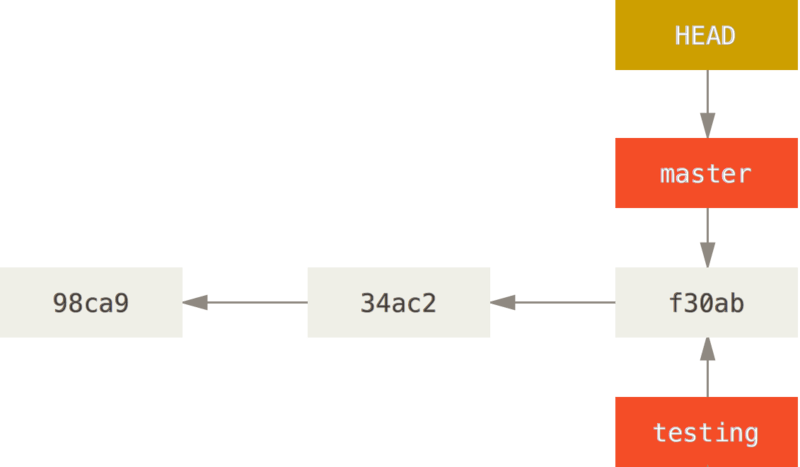
## Технології Git



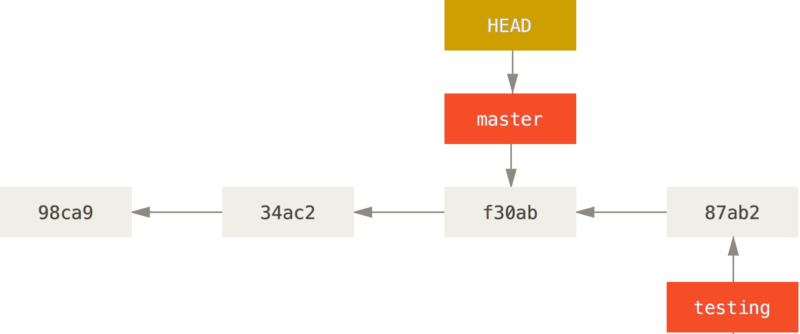
## Три стани



## Гілки Git



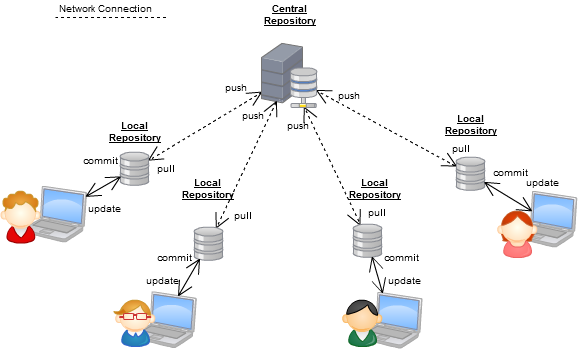
## Змінені гілки



* створення гілки додає вказівник
* робота в гілці продовжує ланцюжок, переміщує вказівник
* перемикання між гілками – переставляння *HEAD*
* паралельна робота в різних гілках спричиняє розгалуження ланцюжка
* злиття гілок – об’єднання *master* з вказівником нової гілки, або нова версія на основі двох різних галузок та спільного предка
* безконфліктне злиття відбувається автоматично, конфлікти редагують вручну в редакторі файлів або за допомогою програм-помічників
* вилучення гілки (вказівника) після злиття

### Тактика використання гілок

* *master* існує постійно, тому використовується для зберігання стабільної версії
* *develop* довготривала гілка для накопичення етапів розробки, стабільний етап зливають з master, гілку зберігають для наступних етапів
* *topic* короткотривала для тематичних розробок, досліджень, може влитися в develop, а може бути повністю скасована
* *hotfix* «миттєве» відгалуження, що зникає одразу після вирішення проблеми



# Платформа Github

# Що це таке?

Це платформа для завантаження, зберігання і поширення коду, контролю версій коду та організації спільної розробки (і не тільки коду). Вона дає вам змогу працювати разом зі співавторами над одним проектом, причому віддалено, з будь-якого місця, де є доступ до інтернету.

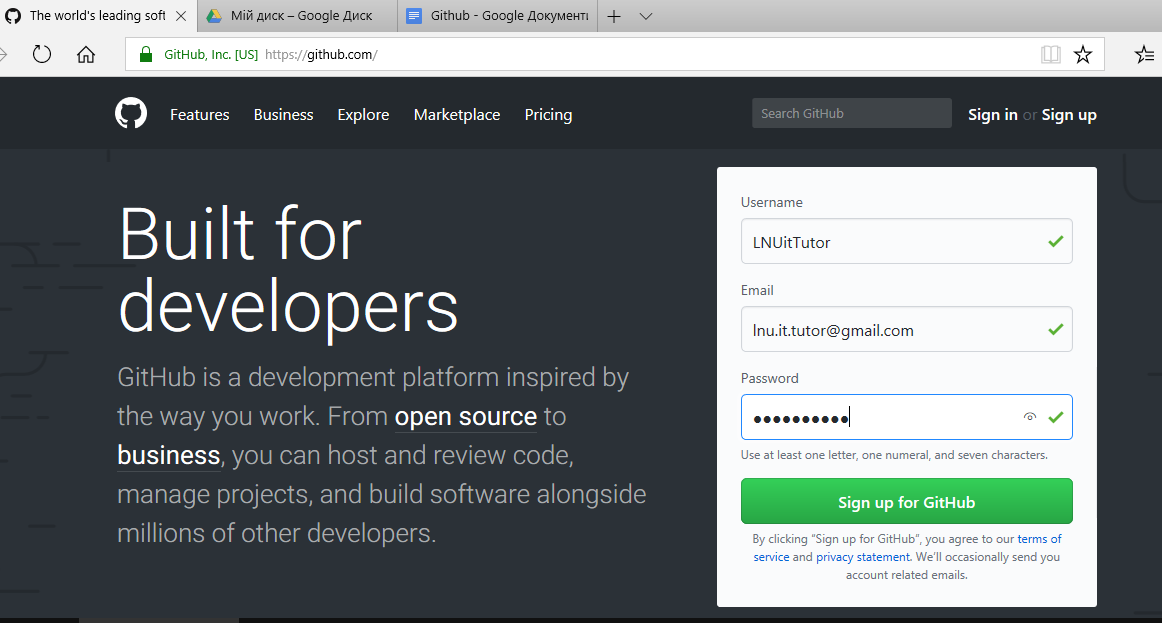
У найпростішому випадку вам достатньо мати обліковий запис на Github.com та “живий” інтернет-браузер. Інсталювати програму Git не доведеться, бо хаб володіє вбудованою системою контролю версій.

Базовими потяттями є *repository (сховище)*, *branches (галузки, вітки)*, *commits (записи, занесення до сховища)*, *Pull Requests (запити завантаження)*.

Сховище - місце, де можна зберігати ідеї, ресурси, провадити обговорення та обмін. Відкриті сховища є безкоштовними. Приватні сховища комерційних продуктів - платні.

# Створення облікового запису

Як тільки ви зайдете на github.com, вас зустріне запрошення створити обліковий запис: “*Sign up for GitHub*”. Зареєстровані користувачі для входу використовують *Sign in*.



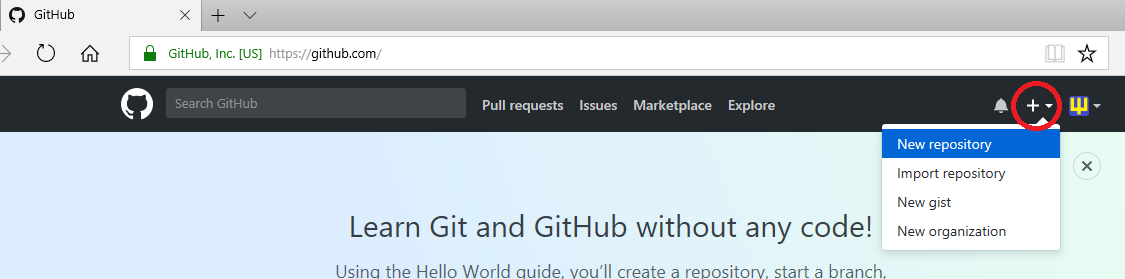
Додайте декілька слів про себе, підтвердіть вашу електронну адресу, і обліковий запис готовий!

# Створення сховища

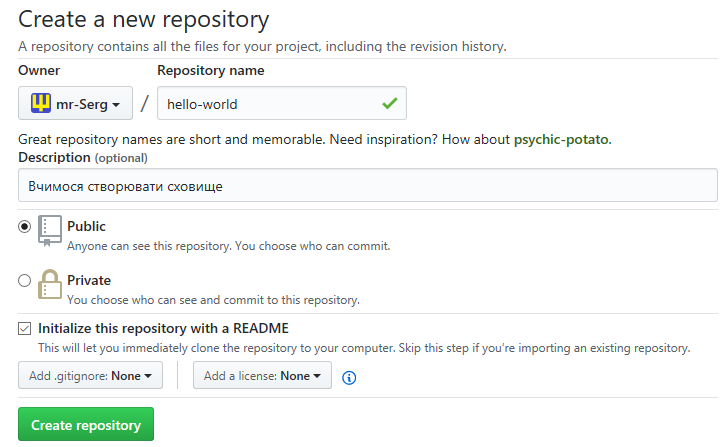
Одне сховище зазвичай використовують для одного проекту. Сховище може містити папки, файли, зображення, відео, таблиці та набори даних - все, чого потребує ваш проект. Рекомендовано включати до проекту файл *ReadMe*, чи будь-який інший з поясненнями про мету створення проекту.GitHub створює такий файл автоматично разом зі створенням сховища, а також надає низку загальних опцій, наприклад, файл ліцензії.

## Послідовність кроків

1. Натисніть “+” біля вашого аватару в правому верхньому кутку екрана і виберіть “New repository”



1. Введіть назву сховища та його короткий опис.
2. Відзначте “**Initialize this repository with a README**”
3. Клацніть “**Create repository**”



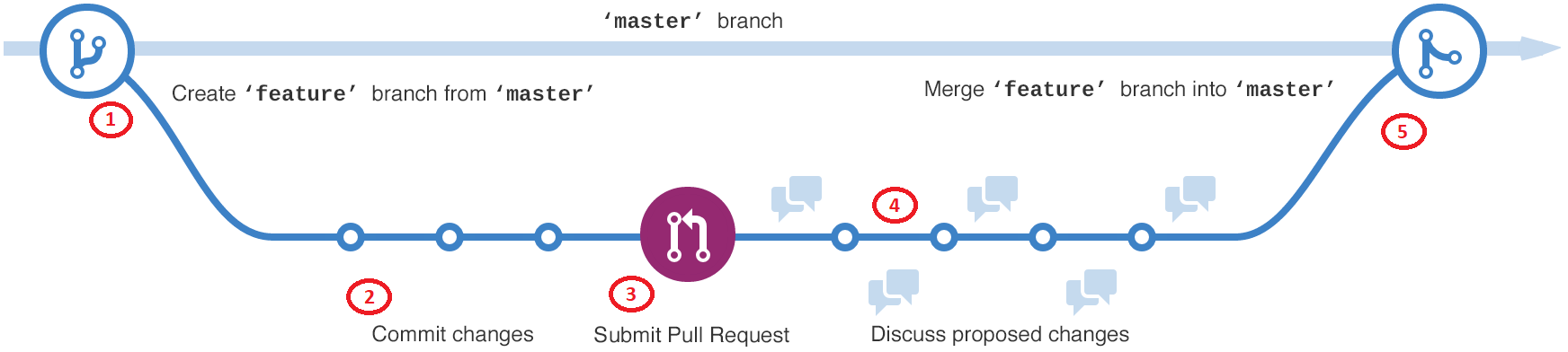
# Створення вітки

Галуження - це спосіб одночасної роботи з кількома версіями проекту.

За замовчуванням ваше сховище має одну вітку (чи галузку), що називається *master*. Вона є головною і остаточною галузкою. Ми виконаємо галуження заради експерименту. У відгалуженні виконаємо редагування, а тоді запишемо їх до *master*.

Коли ви створюєте відгалуження *master*, ви фіксуєте стан (робите миттєвий знімок) вітки *master* на момент відгалуження. Якщо хтось інший зробить зміни в галузці *master*, поки ви працюєте зі своєю віткою, ви зможете завантажити всі зміни, зроблені без вас.

На діаграмі зображено можливий сценарій розвитку подій:



1 - від вітки *master* відгалузилась вітка *feature*, призначена для чогось особливого

2 - триває паралельна робота над обома галузками, автори заносять зміни

3 - отримання оновлень з *master* у *feature*

4 - узгодження запропонованих змін

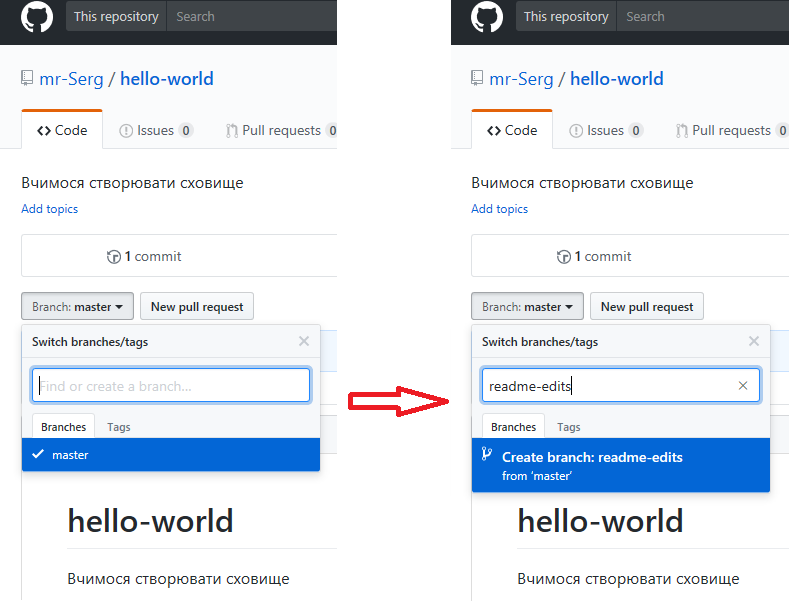
5 - злиття галузок

Вітки зберігають різні версії файлів проекту. Після погодження та злиття залишається остаточна версія.

## Послідовність кроків

1. Перейдіть до вашого нового сховища *hello-world*.
2. Клацніть на стрілці спадного списку вибору віток і введіть в текстовому полі назву нової вітки
3. Натисніть *Create branch: назва-вітки*.

Тепер у вас дві однакові галузки проекту. Вони стануть різними, як тільки ми внесемо зміни в одну з них.

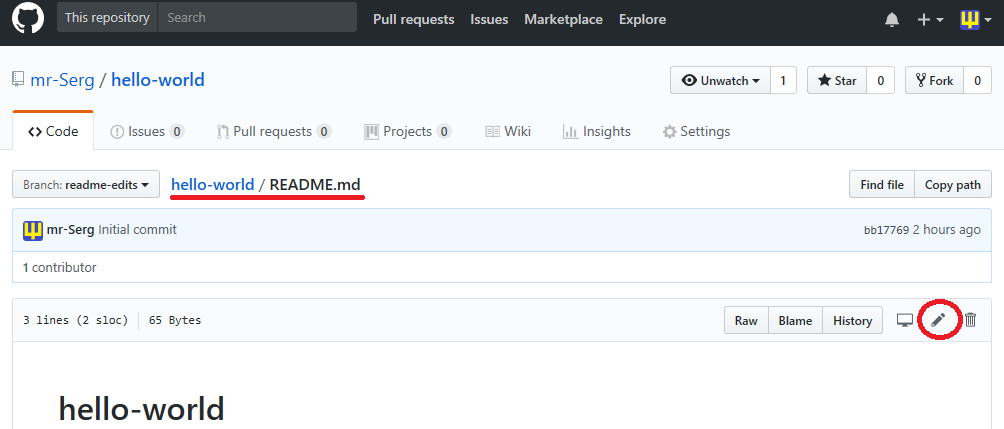


# Внесення і зберігання змін

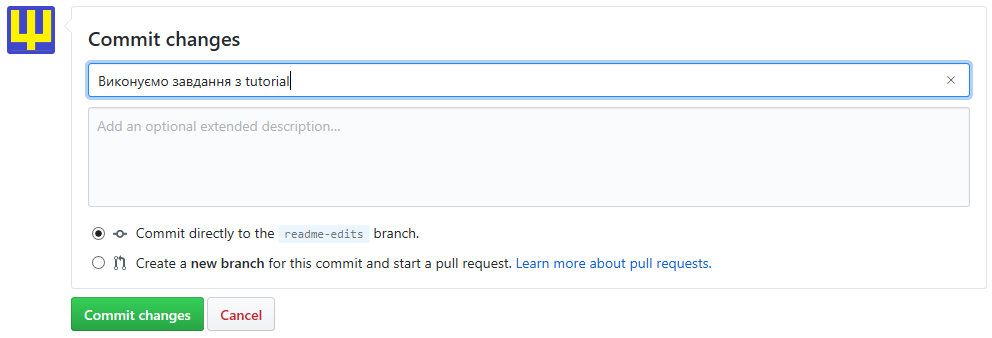
Зміни на Github називаються *commits*. Кожна збережена зміна має пов’язане з нею повідомлення, яке пояснює мету зроблених змін. Повідомлення змін складають історію ваших змін. Завдяки ним інші співавтори розумітимуть, що ви зробили і навіщо.

## Послідовність кроків

1. Клацніть на імені файла *README.md* - ви увійдете в файл, вигляд екрана зміниться



1. Клацніть на піктограмі з олівцем у правому верхньому куті вікна файла, щоб перейти в режим редагування
2. Опишіть докладніше мету проекту. Напишіть повідомлення про зміни.
3. Клацніть на кнопці “**Commit changes**”



Зміни буде внесено до файла у вітці, паралельній до початкової, так що два файли (дві версії) тепер відрізняються.

## Запит завантаження

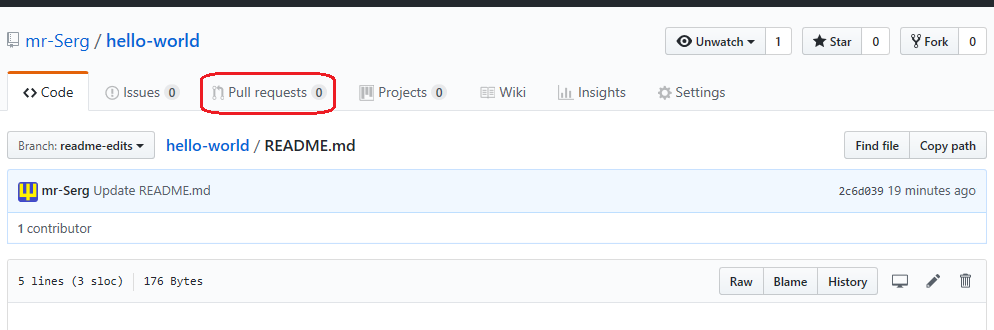
Серцевиною взаємодії на Github є *pull request*. Коли ви відкриваєте запит на зміни, ви пропонуєте зроблені вами зміни власникові котроїсь із віток, щоб він переглянув їх, схвалив і об’єднав з власною версією документа. Запит демонструє авторам відмінності їхніх версій, позначає різними кольорами зміни, доповнення і вилучення.

Як тільки ви зберегли зміни, ви можете зробити запит на збереження і розпочати обговорення змін, навіть, якщо роботу ще не завершено.

У повідомленні про зміни можна задіяти систему нагадувань Github, якщо використати посилання на певного користувача чи команду у вигляді “*@username Текст повідомлення (чи запитання)*”. Таким чином ви можете попросити про відповідь будь-кого зі співавторів незалежно від того, де він знаходиться: в одній кімнаті з вами чи в іншому часовому поясі.

Ви можете також об’єднувати власні версії документа.

## Послідовність кроків



1. Виберіть закладку “*Pull request*”, а тоді на сторінці запиту клацніть на кнопці “**New pull request**”.
2. У панелі “**Example Comparisons**” виберіть вітку *readme-edits* для порівняння з віткою *master*.
3. Перегляньте у вікні відмінностей, чим відрізняються версії файла - чи це те, що ви хотіли об’єднати. Якщо так, то натисніть кнопку “**Create Pull Request**”.
4. Назвіть свій запит, та дайте йому стислий опис. В описі ви можете використовувати смайлики та інші графічні символи: просто наберіть літеру ‘:’ і виберіть потрібне зі спадного списку.

# Об’єднання версій

Після всіх погоджень можна об’єднати дві версії файла в одну

## Послідовність кроків

1. Клацніть на кнопці “**Merge pull request**”.
2. Клацніть на кнопці “**Confirm merge**”.
3. Вилучіть паралельну вітку за допомогою “**Delete branch**”: вона більше не потрібна, оскільки зроблені в ній зміни внесено до початкової вітки.

# Співавторство

## Пошук сховища

Використовуйте імена користувачів, назви сховищ, ключові слова з опису репозиторію в рядку пошуку github, щоб знайти потрібний вам проект. Усі public сховища відкриті для читання всім зареєстрованим користувачам. На сайті публікують проекти в першу чергу для поширення.

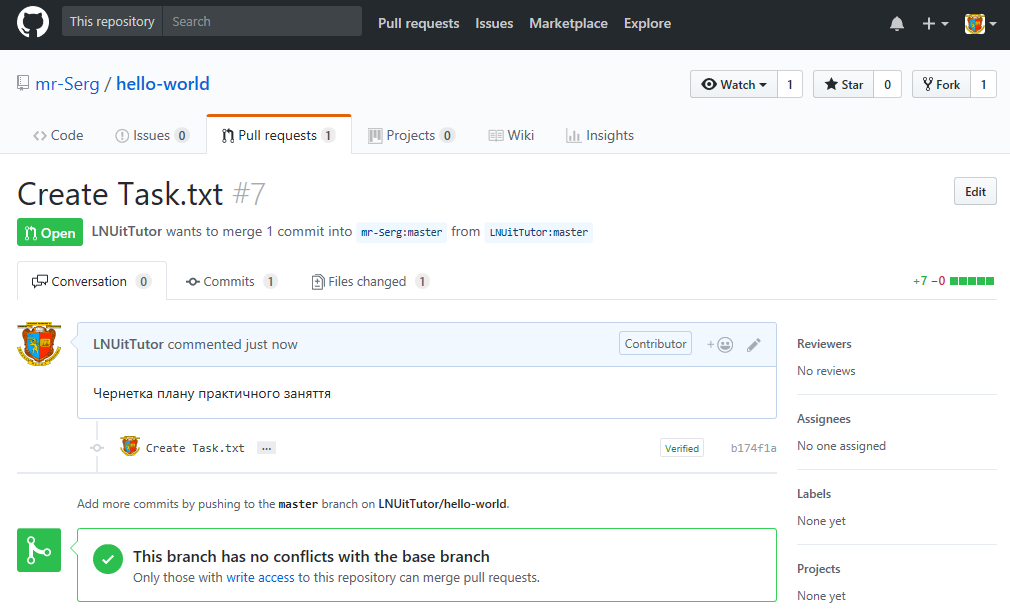
## Відгалуження проекту

Спроба внесення змін до чужого проекту призводить до виконання *fork*. Ви отримуєте копію чужого сховища, зафіксованого в момент доповнення чи зміни, стаєте її власником. Ваша копія матиме власну вітку *master*. Ви можете створювати додаткові вітки як звичайно, додавати файли, редагувати наявні тощо. Усі виконані вами правки ніяк не впливають на початкове сховище. Вони “існують у паралельній реальності”.

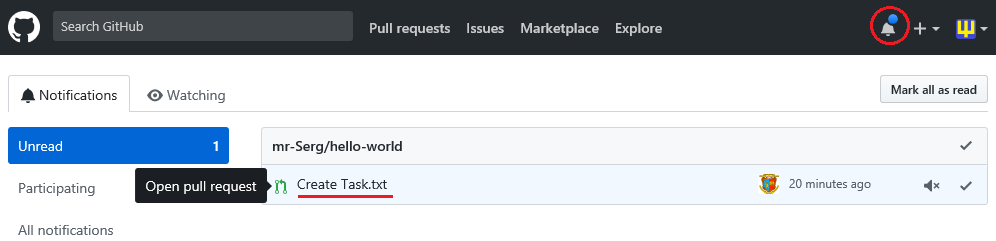
Внесення змін з відгалуження до початкового сховища можливе лише за згодою власника. Ви повинні створити запит на злиття версій (галузок). Власник початкового сховища отримає повідомлення про ваш запит, зможе розглянути ваші пропозиції про внесення змін чи доповнень.

Важливе місце в процесі займає обговорення. Сайт фіксує історію повідомлень.

### Запит на об’єднання



### Повідомлення про запит

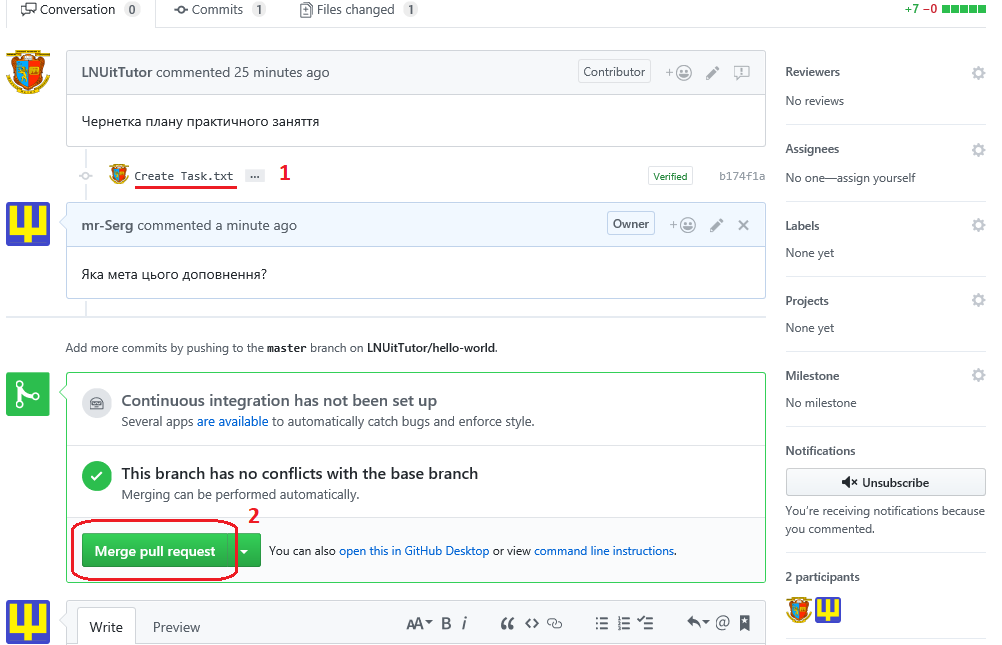


### Обговорення

github створює форум для обговорення можливих змін і доповнень. Повідомлення форуму дублюються листами на електронну почту учасників. Використовується зручна система позначень, режимів перегляду та редагування, коментування внесених змін тощо. З різних сторін форум виглядає по різному.

Власник початкового репозиторію може схвалити доповнення чи відхилити їх, коментувати нейтрально.

Власник відгалуження тільки коментує. Дати вказівку об’єднати галузки може тільки власник.



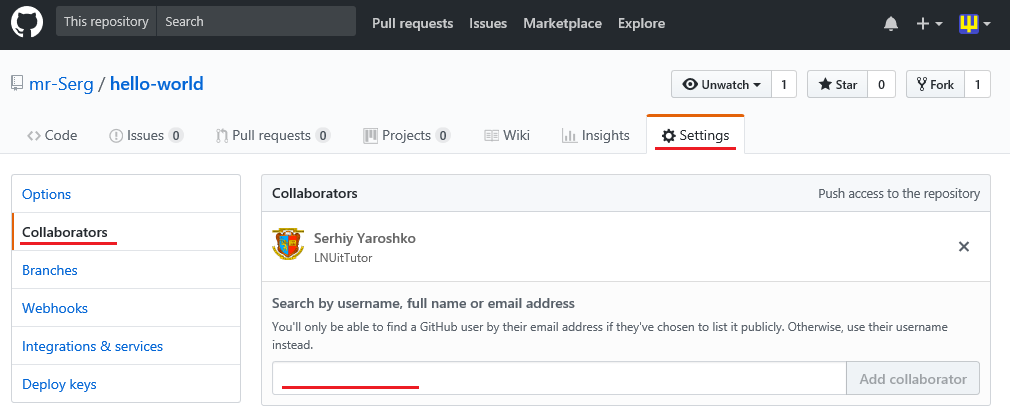
1 - перехід до режиму перегляду файла і коментування

2 - згода на об’єднання

Власник відгалуження, що робив запит на злиття, може закрити запит після того, як власник початкового сховища його схвалив.

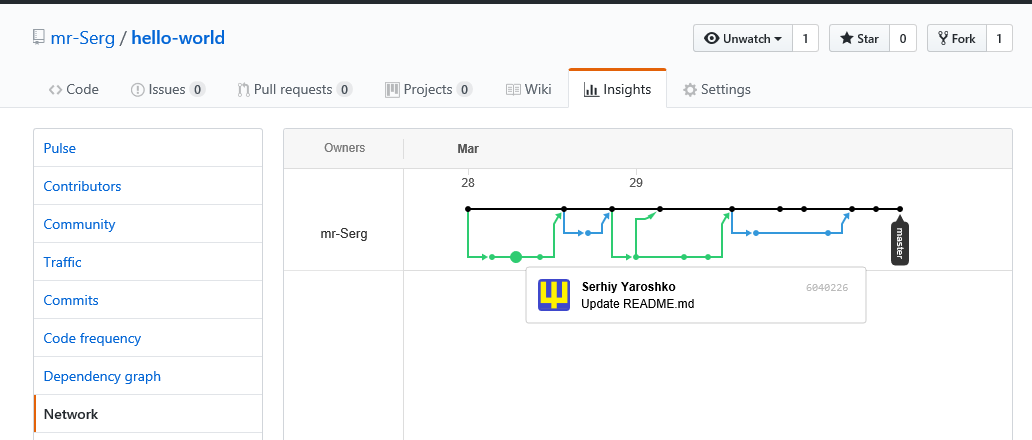
## Співпраця

Власник сховища може запросити зареєстрованого користувача до співавторства. Запрошений стає співавтором, як тільки прийме запрошення. Користувач завжди може відмовитись від співавторства. Зміни, зроблені співавтором, появляються у сховищі одразу.



## Статистика

Система детально відслідковує вклад кожного зі співавторів і траекторію проекту



[**Системи контролю версій**](#_gjdgxs) **1**

[Локальні](#_38xfq09afosp) 1

[Централізовані](#_7ovd5dso44n2) 1

[Розподілені](#_kk9f1u7v32bp) 2

[Репозиторій](#_nqqvz8m6y7v3) 2

[Технології Git](#_jcypswmopl15) 3

[Три стани](#_x513s3p43znj) 3

[Гілки Git](#_899p4namui2a) 3

[Змінені гілки](#_uzaty1tyqabv) 4

[Тактика використання гілок](#_dq2ir1quoih1) 4

[**Платформа Github**](#_numfz78qc1qx) **5**

[**Що це таке?**](#_30j0zll) **5**

[**Створення облікового запису**](#_1fob9te) **5**

[**Створення сховища**](#_3znysh7) **6**

[Послідовність кроків](#_2et92p0) 6

[**Створення вітки**](#_tyjcwt) **7**

[Послідовність кроків](#_3dy6vkm) 7

[**Внесення і зберігання змін**](#_1t3h5sf) **8**

[Послідовність кроків](#_4d34og8) 8

[Запит завантаження](#_2s8eyo1) 9

[Послідовність кроків](#_17dp8vu) 9

[**Об’єднання версій**](#_3rdcrjn) **10**

[Послідовність кроків](#_26in1rg) 10

[**Співавторство**](#_wrkxo7rveeoz) **10**

[Пошук сховища](#_bqoqnj1t13fz) 10

[Відгалуження проекту](#_r6i0cskeanz7) 10

[Запит на об’єднання](#_j8l351cx6gvh) 11

[Повідомлення про запит](#_7e314ydhx8q4) 11

[Обговорення](#_f3wqmvbyvb7) 11

[Співпраця](#_h9qlab1qae4g) 12

[Статистика](#_5myf1m5vqvt3) 13