

Гілкування і об'єднання

№ уроку: 6 **Курс:** Основи використання Git

Засоби навчання: Git console

Огляд, мета та призначення уроку

Навчитися працювати з гілками, створювати, видаляти та перемикатися між ними. Виконувати злиття гілок, вирішувати конфлікти злиття. Також навчимося зберігати зміни у гілці, не виконуючи коміт.

Вивчивши матеріал даного заняття, учень зможе:

- Створювати та видаляти гілки.
- Перемикатися між гілками.
- Виконувати злиття чи перебазування.
- Вирішувати конфлікти злиття.

Зміст уроку

1. Що таке гілка
2. Як створити гілку
3. Управління гілками
4. Робота з гілками
5. Вирішення конфліктів при виконанні merge
6. Rebase

Резюме

- **Гілка** — це показчик на коміт. За замовчуванням, ім'я основної гілки Git - це master. Як тільки ви почнете створювати коміти, гілка master завжди вказуватиме на останній коміт. Щоразу, при створенні коміту, показчик гілки master пересуватиметься на наступний коміт автоматично.
- Команда **git branch <name>** - створює нову гілку, тобто - новий показчик на поточний коміт.
- Показчик HEAD майже завжди вказує на одну з гілок, при виконанні коміту – він буде доданий до тієї гілки, де зараз знаходиться показчик HEAD.
- Для перемикання вказівника HEAD на іншу гілку використовується команда **git checkout <branch_name>**. Git дає можливість створити гілку та переключитися на неї однією командою - **git checkout -b <branch_name>**.
- Для видалення гілки необхідно переконатися, що HEAD зараз на неї не вказує і після цього виконати команду **git branch -d <branch_name>**.
- Переглянути список гілок, які існують у репозиторії, можна також командою **git branch**.
- Для злиття використовується команда **git merge <branch_name>** де <branch_name> - ім'я гілки зміни, яку ви хочете злити в поточну (на яку вказує HEAD під час виконання команди).
- Якщо коміт зливається з тим, який буде доступний рухаючись по історії прямо, то використовується спрощена процедура - просто переносячи показчик гілки вперед, оскільки немає розбіжностей у змінах. Така операція називається **"fast-forward"**.
- Альтернативною операції merge є операція **rebase** (перебазування). Вона бере зміни поточної гілки та застосовує їх поверх всього, що є у зазначеній.

Це працює наступним чином:

1. береться загальний батьківський знімок двох гілок (поточної та тієї, поверх якої ви виконуєте rebase);
2. визначається дельта кожного коміту поточної гілки та зберігається у тимчасовий файл;
3. поточна гілка встановлюється на останній коміт гілки, поверх якої ви виконуєте перебазування;
4. потім по черзі застосовуються дельти з тимчасових файлів.

Результат такий самий як при злитті, але історія комітів виходить чистішою. І, в цьому випадку, не буде конфліктів злиття.

- Недолік використання rebase, порівняно з merge. Якщо переміщувати таким чином коміти, які вже синхронізовані з віддаленим репозиторієм, з яким працюють інші розробники, це може призвести до серйозних проблем. І сюди ж можна зарахувати те, що хронологія змін не збережеться.
- Після операції rebase необхідно переключитися на гілку, в яку він виконувався і виконати злиття з перебазованою гілкою. Наприклад:
 1. `git rebase master`
 2. `git checkout master`
 3. `git merge bugfix`В результаті зміни гілки bugfix застосовані до гілки master.
- **Конфлікт злиття** відбувається, коли ту саму частину коду змінювали в обох гілках, які зливаються в одну. Якщо git самотійно не може вирішити, які зміни будуть результатом злиття – необхідно робити це вручну..

Закріплення матеріалу

- Якою командою можна видалити гілку?
- Яка команда дозволяє створити гілку і відразу на неї перейти?
- Яка операція є альтернативою команди злиття?
- Які коміти не слід переміщати командою git rebase?

Додаткове завдання

Завдання

Створіть у репозиторії нову гілку bugfix, зробіть коміт у гілці master, перейдіть на гілку bugfix, переконайтеся, що в ній немає останнього коміта в master.

Самостійна діяльність учня

Завдання 1

Створіть нову гілку (feature) і зробіть кілька комітів у ній, потім перейдіть на гілку master. Виконайте кілька комітів у гілці master. Виконайте злиття гілки feature у гілку master. Видаліть гілку feature.

Завдання 2

Повторіть кроки попереднього завдання до злиття, виконайте перемикання на гілку feature і перебазування на гілку master.

Рекомендовані ресурси

Офіційна документація Git
<https://git-scm.com/docs>

