



Ветки и слияния

Ветки

Когда мы прописываем `git init` автоматически создается основная ветка `master/main`, но мы можем создавать свои ветки по желанию. Каждый разработчик обычно имеет свою собственную ветку.

Для создания новой ветки и перехода к ней

```
git checkout -b <branch_name>
```

Просто создание ветки

```
git branch <branch_name>
```

Для перехода между ветками

```
git checkout <branch_name>
```

Можно использовать `switch` вместо `checkout`

Чтобы перейти ветке или создать новую ветку

```
git switch <branch_name>  
git switch -c <branch_name>
```

Список веток

```
git branch
```

Переименовать ветку

```
git branch -m <new_branch>
```

Удаление ветки (нельзя удалять ветку в которой ты находишься или не слитую)

```
git branch -d <branch_name>
```

Чтобы удалить ветку не слитую в другую ветку используем `-D` вместо `-d`

Слияние

Для объединения веток нужно перейти в ветку, с которой необходимо провести слияние, например из ветки `work` перейти в ветку `main`, а далее

```
git merge work
```

Существует два типа слияния:

1. **Fast-forward merge (прямое)** - Когда целевая ветка не имеет новых коммитов после ответвления. Git просто перемещает указатель ветки на более новый коммит.
2. **Recursive (рекурсивное)** - Определяет общий предок, сравнивает различия, создаёт новый коммит слияния.

Во время слияния могут возникать конфликты при изменении одних и тех же фрагментов кода. В таком случае конфликт придётся решать вручную, оставляя нужный участок кода, а после завершая коммитом.

Для того, чтобы отменить слияние во время конфликта

```
git merge --abort
```

`git rebase` — это перенос коммитов из одной ветки поверх другой. То есть, он перезаписывает историю так, будто коммиты создавались прямо "на вершине" целевой ветки. Получаются новые коммиты с теми же изменениями (другие SHA)

```
git rebase main
```

С помощью **rebase** можно переписать историю

Чтобы посмотреть, что можно сделать с последними тремя коммитами

```
git rebase -i HEAD~3
```

```
# Возможные команды:  
# pick — оставить как есть  
# squash — объединить с предыдущим  
# reword — изменить сообщение коммита  
# drop — удалить коммит
```

Изменения

Как изменить последний коммит, например добавить файлы. (в случае локального изменения)

```
git add file  
git commit --amend
```

Для того, чтобы перенести конкретный коммит в другую ветку, мы используем **cherry pick** (если не хотим объединять ветку целиком)

Например, случайно сделали коммит не в той ветке (сделал в **main**, а хотел в **work**)

```
git log --oneline  
git checkout work  
git cherry-pick <hash from git log>
```