

Ответы на вопросы по Git:

Отказаться от изменений в файле (файлах) находящихся в рабочем каталоге (work directory):

Используем команду `git checkout -- <file>` для отмены изменений в определенном файле или `git checkout --` . для отмены изменений во всех файлах в рабочем каталоге. эта команда безвозвратно отменит изменения, которые еще не были добавлены в индекс.

Вернуть файл из индекса в рабочий каталог:

Используем команду `git reset HEAD <file>` для удаления файла из индекса, но сохранения изменений в рабочем каталоге.

Отменить ошибочный commit:

Если мы хотим отменить последний commit, используем команду `git reset HEAD~`. Это удалит последний коммит из истории, но изменения останутся в рабочем каталоге. Если мы хотим изменить коммит и снова его закоммитить, используем `git commit --amend`.

Исправить commit отправленный на сервер GitHub:

Если мы уже отправили неправильный коммит на сервер GitHub, то необходимо быть осторожным с его исправлением. Не рекомендуется изменять историю коммитов на общедоступных ветках. Вместо этого можно создать новый коммит с исправлениями и отправить его на сервер. В этом случае другим разработчикам не придется работать с измененной историей.

Создать новую ветку и переключиться в нее:

Используем команду `git checkout -b <branch_name>` для создания новой ветки и переключения на нее. Если ветка уже существует, мы можем переключиться на нее с помощью `git checkout <branch_name>`.

Как создать Git репозиторий из локального каталога на своем компьютере и отправить его на GitHub:

Инициализировать локальный репозиторий: Перейти в каталог с нашим проектом и выполнить команду `git init`, чтобы создать новый Git репозиторий.

Добавить файлы в индекс: Используем команду `git add .`, чтобы добавить все файлы в индекс (отслеживание изменений).

Создать коммит: Выполняем команду `git commit -m "Первый коммит"`, чтобы создать первый коммит с нашими файлами.

Создать репозиторий на GitHub: Зайти на сайт <https://github.com/> и создать новый репозиторий без инициализации README файла, если он уже есть в нашем локальном репозитории.

Связать локальный репозиторий с репозиторием на GitHub: Используем команду `git remote add origin <URL репозитория на GitHub>`, чтобы связать наш локальный репозиторий с репозиторием на GitHub.

Загрузить данные на GitHub: Выполняем команду `git push -u origin master`, чтобы загрузить наш локальный репозиторий на GitHub. После первой загрузки можно будет использовать просто `git push`.

Как получить на локальный компьютер изменения с GitHub:

Связать локальный репозиторий с репозиторием на GitHub (если не сделано ранее): Используем команду `git remote add origin <URL репозитория на GitHub>`.

Получить изменения с GitHub: Выполним команду `git pull origin master` (или другая ветка, если мы работаем с другой веткой) для получения изменений с GitHub и их объединения с нашим локальным репозиторием.

Команды `git status`, `git log`, `git remote -v`, `git branch -v`:

`git status`: Показывает состояние рабочего каталога и индекса. Позволяет увидеть, какие файлы были изменены, добавлены или удалены.

`git log`: Показывает историю коммитов. Позволяет увидеть все сделанные коммиты, их авторов, даты и сообщения коммитов.

`git remote -v`: Показывает список удаленных репозиториях и их URL. Позволяет узнать, на какой сервер ссылается наш локальный репозиторий.

`git branch -v`: Показывает список веток в нашем репозитории и последние коммиты на каждой из них. Позволяет увидеть актуальность веток и последние изменения.