Инициализация репозитария:

* git init Инициализировать репозиторий можно с помощью команды.
* git status Проверить статус, или состояние, репозитория поможет команда.
* rm -rf удаление файлов (папок)
* -r ключ (от англ. recursive — «рекурсивно») позволяет удалять папки вместе с их содержимым;
* -f ключ (от англ. force — «заставить») избавит вас от вопросов вроде «Вы точно хотите удалить этот файл? А этот? И этот тоже?».
  + - Если вы ошиблись и случайно инициализировали не ту папку, можно «разгитить» её — удалить скрытую подпапку .git.

Подготовка к сохранению:

* git add Команда позволяет подготовить файл к сохранению.
* git add --all Команда подготовит к сохранению сразу все файлы.
* git add . С помощью можно добавить в репозиторий текущую папку со всеми файлами.

**Коммит -** само сохранение, или фиксация состояния файлов, (от англ. commit — «совершать», «фиксировать»). «Сделать коммит» значит сохранить текущую версию файла.

**Выполнить коммит — git commit**

Сообщение коммита выполняет те же функции — улучшает понимание и упрощает навигацию. Оно пишется после ключа -m в кавычках

### Просмотреть историю коммитов — git log

**Проверить наличие директории хранения ключей** и файлов в ней можно с помощью следующей команды - $ ls -la .ssh/ *# вывели список созданных ключей*

**Инструкция по генерации SSH-ключа**

1. Для генерации SSH-пары можно использовать программу ssh-keygen. Откройте терминал и введите следующую команду.

Скопировать код BASH

$ ssh-keygen -t ed25519 -C "электронная почта, к которой привязан ваш аккаунт на GitHub"

Укажите место хранения ключей. Простой вариант — сделать домашний каталог пользователя путём по умолчанию. Для этого нажмите Enter.

Enter a file in which to save the key (C:\Users\<имя\_пользователя>\.ssh\):[Press enter]

1. Теперь осталось проверить, что ключи действительно сгенерировались. Для этого вызовите эту команду.

Скопировать кодBASH

ls -a ~/.ssh

# скопировать содержимое ключа в буфер обмена:

$ clip < ~/.ssh/id\_rsa.pub

# для ed25519:

$ clip < ~/.ssh/id\_ed25519.pub

### Привязать удалённый репозиторий к локальному — git remote add

$ cd ~/dev/first-project

$ git remote add origin [git@github.com:%ИМЯ\_АККАУНТА%/first-project.git](mailto:git@github.com:%25ИМЯ_АККАУНТА%25/first-project.git)

### Убедиться, что репозитории связаны, — git remote -v

$ git remote -v

origin git@github.com:%ИМЯ\_АККАУНТА%/%ИМЯ-ПРОЕКТА%.git (fetch)

origin git@github.com:%ИМЯ\_АККАУНТА%/%ИМЯ-ПРОЕКТА%.git (push)

# Синхронизируем локальный и удалённый репозитории

### Отправить изменения на удалённый репозиторий — git push

$ git push -u origin main # Если команда приведёт к ошибке, попробуйте

# заменить main на master.