

1. Выберите какой-нибудь проект на изучаемом вами языке программирования, с которым вы будете тренироваться работать в Git, и инициализируйте в папке этого проекта локальный репозиторий.

Создадим проект в папке **bash**

```
mkdir bash
cd bash
echo "printf 'Hello world!\n'" > hello.sh
echo "local README.md" > README.md
```

Создадим локальный репозиторий

```
git init
git branch -M main
```

Зафиксируем изменения в репозитории

```
git add .
git commit -m "local README.md"
```

2. Создайте непустой удалённый репозиторий (например, с файлом README.md) с именем, соответствующим имени этого проекта.

Создам удаленный репозиторий <https://github.com/allseenn/bash.git> без веббраузера

```
curl -u allseenn https://api.github.com/user/repos -d
'{"name":"bash","private":false}'
```

Используя утилиту **curl** необходимо указать параметры:

- **u** - имя пользователя
- **name** - имя репозитория
- **private** - приватность/публичность репозитория
- **PAT** - для авторизации запросит personal access token. Можно задать один раз в личном кабинете. Т.к. с 13.08.2021 github отключил авторизацию по

паролю.

Далее создадим временный локальный репозиторий в папке **temp**

```
mkdir temp
cd temp
git init
git branch -m main
```

Во временной папки **temp** создадим заполненный README.md

```
echo "remote README.md" > README.md
```

Зафиксируем результат и отправим на удаленный репозиторий **bash**.

```
git add .
git commit -m "remote README.md"
git remote add origin https://github.com/allseenn/bash.git
git push -u origin main
```

Временную папку **temp** удалим (пайп для совместимости с windows)

```
cd ..
rm -r temp || rm -r -fo temp
```

*Таким образом обошлись полностью без веббраузера. Это полезно, например, на пих-машинах без графического интерфейса, либо используя в скриптах для автоматизации процессов.*

3. Подключите свой проект к этому удалённому репозиторию и отправьте в него код этого проекта. Самостоятельно разрешите конфликты и проблемы, если они возникнут при выполнении данного задания.

Подключим локальный репозиторий **bash** к удаленному git remote add origin <https://github.com/allseenn/bash.git>

```
git remote add origin https://github.com/allseenn/bash.git
git branch -u origin/main main
```

```
git push -u origin main
```

## Имеем ошибку

```
To https://github.com/allseenn/bash.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/allseenn/bash.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

## Проверяем статус

```
git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean
```

## Получаем информацию из удаленного репозитория

```
git fetch origin
git merge origin/main --allow-unrelated-histories
```

## Появляется информация о конфликте слияние и о проблемном файле

```
Auto-merging README.md
CONFLICT (add/add): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

## При просмотре файла видим различия

```
cat .\README.md
<<<<<<< HEAD
local README.md
=====
```

```
remote README.md
>>>>>> origin/main
```

С помощью любимого консольного редактора **vim** удаляем ненужную информацию и фиксируем изменения

```
vim README.md
git add .
git commit -m "edited local README.md"
```

Отправляем изменения на сервер

```
git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 414 bytes | 414.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/allseenn/bash.git
  1c9ec0d..916d87e  main -> main
```

Все в порядке. Наш локальный рабочий репозиторий с баш-скриптом успешно опубликован на гитхабе. Конфликтов нет.

```
git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
```

*Описанные выше команды были использованы на системе Windows 11 Enterprise с установленным git version 2.41.0.windows.2, с помощью командного интерпретатора PowerShell 7.3.5. Данные команды совместимы с интерпретатором bash, в том числе на nix системах.*