

1-Создаем локальный репозиторий добавляем несколько текстовых файлов и делаем первый коммит

```
Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2
$ git init
Initialized empty Git repository in C:/Users/Admin/Desktop/GIT/project2/.git/

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        text1.txt
        text2.txt
        text3.txt

nothing added to commit but untracked files present (use "git add" to track)

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git add .

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git status
On branch master


No commits yet

changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   text1.txt
        new file:   text2.txt
        new file:   text3.txt

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git commit -m "first commit"
[master (root-commit) 69659be] first commit
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 text1.txt
create mode 100644 text2.txt
create mode 100644 text3.txt

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git diff
```

2-Создаем файл .gitignore в который включим игнорирования ненужных файлов

 .gitignore

Файл Изменить Просмотр

*.log

3-Создаем новую ветку и переключаемся на нее.

Делаем изменения в text2 и создаем второй коммит

```
Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git checkout -b "test"
Switched to a new branch 'test'

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (test)
$ git commit -am "test change"
[test 468dc33] test change
1 file changed, 1 insertion(+)

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (test)
$ git log --oneline --graph --all
* 468dc33 (HEAD -> test) test change
* 69659be (master) first commit
```

4- Переключаемся между ветками и делаем их

слияние и 2 коммита делая по 1 изменению в файлах перед созданием коммита.

```
Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (test)
$ git checkout master
Switched to branch 'master'

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git merge test
Updating 69659be..468dc33
Fast-forward
 text2.txt | 1 +
1 file changed, 1 insertion(+)

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git log --oneline --graph --all
* 468dc33 (HEAD -> master, test) test change
* 69659be first commit

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git checkout test
Switched to branch 'test'

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (test)
$ git commit -am "TEST CHANGE"
[test 506b08d] TEST CHANGE
1 file changed, 1 insertion(+)

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (test)
$ git checkout master
git: 'checkout' is not a git command. See 'git --help'.

The most similar command is
  checkout

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (test)
$ git checkout master
Switched to branch 'master'

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git commit -am "TEST CHANGE master"
On branch master
nothing to commit, working tree clean

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git commit -am "TEST CHANGE master"
[master a5fe30d] TEST CHANGE master
1 file changed, 1 insertion(+)

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git log --oneline --graph --all
* a5fe30d (HEAD -> master) TEST CHANGE master
* 506b08d (test) TEST CHANGE
* 468dc33 test change
* 69659be first commit
```

5-Произошел конфликт

```
Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git merge test
Auto-merging text1.txt
CONFLICT (content): Merge conflict in text1.txt
Automatic merge failed; fix conflicts and then commit the result.
```

6-Изменяем тот же файл что и в основной ветки и разрешаем конфликт

```
Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master|MERGING)
$ git commit -am "resolve conflict"
[master 4a4baa5] resolve conflict

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git commit -am "resolve conflict"
On branch master
nothing to commit, working tree clean

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git log --oneline --graph --all
*   4a4baa5 (HEAD -> master) resolve conflict
| \
|  * 506b08d (test) TEST CHANGE
|  * | a5fe30d TEST CHANGE master
| /
* 468dc33 test change
* 69659be first commit
```

7-Создаем локальный репозиторий связываем его с локальным и передаем туда коммиты

```
Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git remote add origin https://github.com/SOSBorov/Lab-1

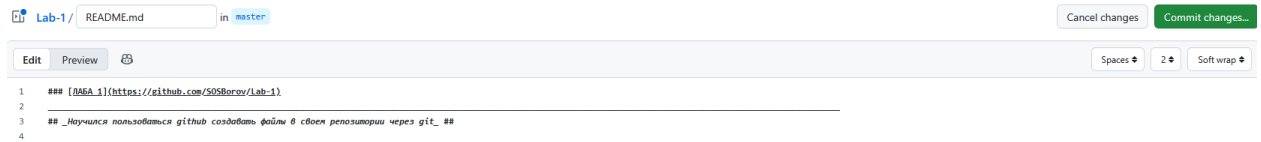
Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git remote -v
origin  https://github.com/SOSBorov/Lab-1 (fetch)
origin  https://github.com/SOSBorov/Lab-1 (push)

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git push -u original master
fatal: 'original' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
```

8-Создаем файл README.md в своем репозитории и пишем краткую информацию



9-В локальном репозитории получим изменение github командой pull, что бы скачать файл README.md

```
Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ git pull origin master
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (12/12), 3.69 KiB | 75.00 KiB/s, done.
From https://github.com/SOSBorov/Lab-1
 * branch                master       -> FETCH_HEAD
   e12f3d4..6cb697e      master       -> origin/master
Updating e12f3d4..6cb697e
Fast-forward
 .gitignore | 1 +
 README.md  | 3 +++
 2 files changed, 4 insertions(+)
 create mode 100644 README.md

Admin@DESKTOP-9437AGJ MINGW64 ~/Desktop/GIT/project2 (master)
$ |
```