

Все действия выполнены в Git Bash

- 1) `git init` – команда для создания пустого репозитория Git;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2
$ git init
Initialized empty Git repository in C:/Space/IDEs/Git/Git_practice2/.git/
```

- 2) `ls -la` – команда используется для того, чтобы убедиться, что репозиторий создан;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ ls -la
total 8
drwxr-xr-x 1 User 197121 0 сен 23 13:37 ./
drwxr-xr-x 1 User 197121 0 сен 23 13:37 ../
drwxr-xr-x 1 User 197121 0 сен 23 13:37 .git/
```

- 3) `git status` – команда для проверки статуса репозитория, отображающая, на какой ветке находится пользователь;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git status
On branch master

No commits yet

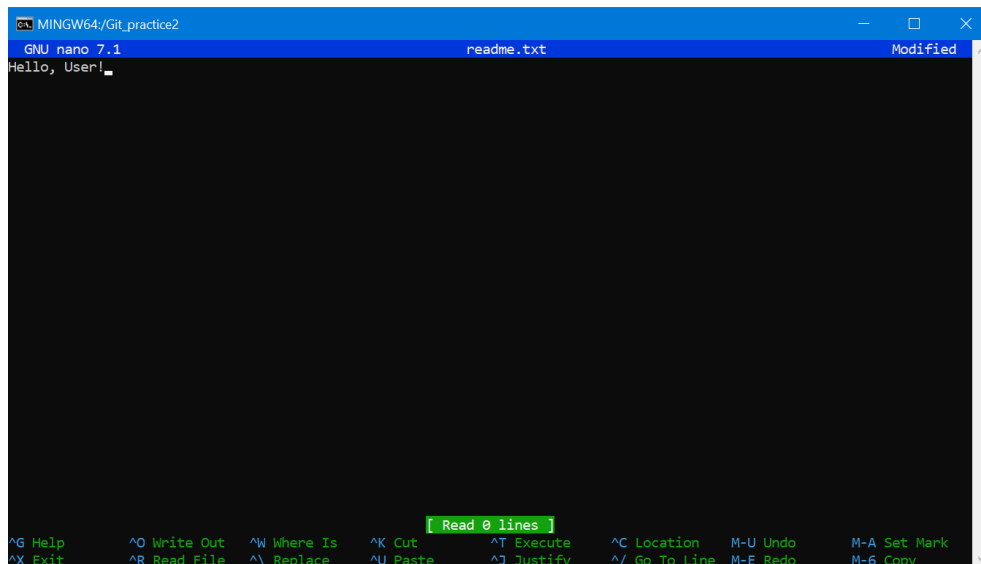
nothing to commit (create/copy files and use "git add" to track)
```

- 4) `touch readme.txt` (`touch` + название файла) – команда для создания текстового документа в репозитории;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ touch readme.txt
```

- 5) nano readme.txt (nano + название файла) – команда для введения или изменения текста в определенном файле. Для того, чтобы сохранить изменения и выйти нужно нажать клавиши Ctrl+X, а затем нажать “Y” и Enter;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ nano readme.txt
```



Используем git status для фиксации изменений.

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git status
On branch master

No commits yet

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

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

- 6) git add readme.txt (git add + название файла) – команда для отслеживания файла.

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git add readme.txt
warning: in the working copy of 'readme.txt', LF will be replaced by CRLF the next time Git touches it
```

После вводим команду `git status` для проверки статуса нового файла.

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   readme.txt
```

- 7) `git commit -m "First file"` (“комментарий”) – команда для фиксации статуса;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git commit -m "First file"
[master (root-commit) 74950ef] First file
 1 file changed, 1 insertion(+)
 create mode 100644 readme.txt
```

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git status
On branch master
nothing to commit, working tree clean
```

- 8) Необходимо выполнить команду `nano` для изменения текста в файле.
Далее выполнить команду `git status`. После проверки статуса

замечены изменения и отмечены красным цветом. Необходимо снова использовать команду `git add`;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ nano readme.txt

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   readme.txt

no changes added to commit (use "git add" and/or "git commit -a")

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git add readme.txt
warning: in the working copy of 'readme.txt', LF will be replaced by CRLF the next time Git touches it
```

- 9) `git add '*.txt'` – команда для добавления всех файлов формата `'*.txt'`.
Необходимо набрать команду `git status` для проверки добавления файлов;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   readme.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        readme after.txt
        readme now.txt

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git add '*.txt'
warning: in the working copy of 'readme after.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'readme now.txt', LF will be replaced by CRLF the next time Git touches it

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   readme after.txt
        new file:   readme now.txt
        modified:   readme.txt
```

- 10) Фиксация и комментирование с помощью команды `git commit -m`;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git commit -m "All files with txt extension were added"
[master 4a5a4c2] All files with txt extension were added
3 files changed, 3 insertions(+), 1 deletion(-)
create mode 100644 readme after.txt
create mode 100644 readme now.txt
```

- 11) `git log` – команда, позволяющая просмотреть историю всех фиксаций;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git log
commit 4a5a4c2dfcd1ae51e4edf6bdd5d2a6d41246e3e6 (HEAD -> master)
Author: julia1gin <124470751+julia1gin@users.noreply.github.com>
Date: Sat Sep 23 13:55:17 2023 +0300

    All files with txt extension were added

commit 74950ef8bcf85b15da396be13cd174bd06dfa0b9
Author: julia1gin <124470751+julia1gin@users.noreply.github.com>
Date: Sat Sep 23 13:44:50 2023 +0300

    First file
```

- 12) `git log --summary` – команда делает то же, что и `git log`, но с большим количеством деталей;

```

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git log --summary
commit 4a5a4c2dfcd1ae51e4edf6bdd5d2a6d41246e3e6 (HEAD -> master)
Author: julia1gin <124470751+julia1gin@users.noreply.github.com>
Date: Sat Sep 23 13:55:17 2023 +0300

    All files with txt extension were added

create mode 100644 readme after.txt
create mode 100644 readme now.txt

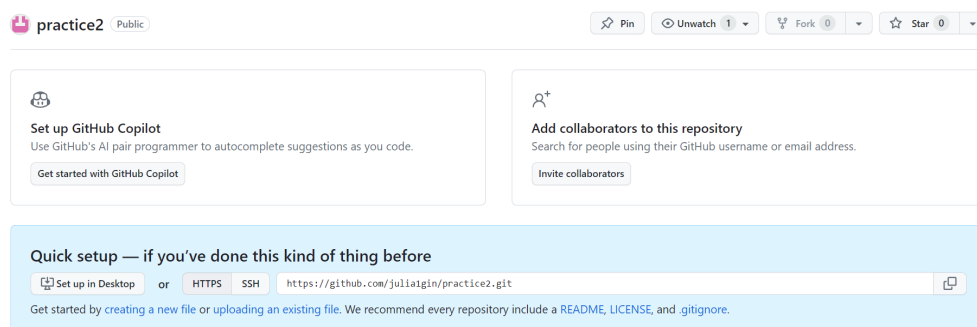
commit 74950ef8bcf85b15da396be13cd174bd06dfa0b9
Author: julia1gin <124470751+julia1gin@users.noreply.github.com>
Date: Sat Sep 23 13:44:50 2023 +0300

    First file

create mode 100644 readme.txt

```

- 13) `git remote add origin` + ссылка на репозиторий – команда, позволяющая добавить удаленный репозиторий



```

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git remote add origin https://github.com/julia1gin/practice2.git

```

- 14) `git push -u origin master` (`git push -u` + имя репозитория + название ветки) – команда, позволяющая выгружать файлы в удаленный репозиторий;

```

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 471 bytes | 235.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/julia1gin/practice2.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

```

- 15) `git pull origin master` (`git pull` + имя репозитория + название ветки)
 – команда, позволяющая узнать последние изменения в удаленном репозитории. В данном случае в файле `readme.txt` была произведена замена слова, о чем сообщают последние две строки;

```

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 1 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 639 bytes | 91.00 KiB/s, done.
From https://github.com/julia1gin/practice2
 * branch            master      -> FETCH_HEAD
  4a5a4c2..0f5d6ab  master      -> origin/master
Updating 4a5a4c2..0f5d6ab
Fast-forward
 readme.txt | 2 +
 1 file changed, 1 insertion(+), 1 deletion(-)

```

- 16) Для начала при помощи команды `naipo` необходимо совершить изменения в файле. Затем ввести команду `git diff HEAD`, которая позволит увидеть различия между зафиксированной и нынешней версией файла;

```

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ nano readme.txt

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git diff HEAD
diff --git a/readme.txt b/readme.txt
index d07e7c0..930d8aa 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,3 @@
  Hello, User!
+Hello, User!
+Hello, User!

```

- 17) `mkdir` + название каталога – команда, добавляющая каталог. После необходимо добавить файл, используя команду `touch` + название папки/название файла. Далее проверка статуса;

```

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ mkdir folder

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ touch folder/bye.txt

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   readme.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        folder/

no changes added to commit (use "git add" and/or "git commit -a")

```

- 18) `git diff --staged` – команда, позволяющая сравнить состояние репозитория с тем, каким он был до изменений;


```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git diff --staged
```

- 19) `git reset` + название каталога – команда, позволяющая отменить изменения;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git reset folder/bye.txt
Unstaged changes after reset:
M      readme.txt

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git diff --staged
```

- 20) `git checkout --` + имя файла – команда, позволяющая вернуть файл в первоначальное состояние;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git checkout -- readme.txt

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      folder/

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

- 21) `cat` + имя файла – команда, выводящая изначальный текст файла в КОНСОЛИ;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ cat readme.txt
Hello, User!
```

- 22) `git branch` + название ветки – команда, позволяющая создать новую ветку. Команда `git branch` позволяет узнать, в какой ветке сейчас находится пользователь;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git branch clean_up

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git branch
clean_up
* master
```

- 23) `git checkout` + название ветки – команда, позволяющая переключиться на другую ветку;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git checkout clean_up
Switched to branch 'clean_up'

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (clean_up)
$ git branch
* clean_up
master
```

- 24) `rm -r` + название каталога – команда, позволяющая удалить каталог и его файлы данной ветки. После удаления необходимо зафиксировать изменения;

```

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (clean_up)
$ rm -r folder

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (clean_up)
$ git commit -m 'Deleted folder and files'
On branch clean_up
nothing to commit, working tree clean

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (clean_up)
$ git status
On branch clean_up
nothing to commit, working tree clean

```

- 25) `git merge` + имя ветки – команда слияния двух веток;

```

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git merge clean_up
Updating 0f5d6ab..ebca750
Fast-forward
 readme after.txt | 1 -
 readme now.txt   | 1 -
 2 files changed, 2 deletions(-)
 delete mode 100644 readme after.txt
 delete mode 100644 readme now.txt

```

- 26) `git branch -d` + название ветки – команда, позволяющая удалить ненужную ветку;

```

User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git branch -d clean_up
Deleted branch clean_up (was ebca750).

```

- 27) `git push` – команда, позволяющая загрузить изменения в удаленный репозиторий;

```
User@DESKTOP-U3DTBP2 MINGW64 /Git_practice2 (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (1/1), done.
Writing objects: 100% (2/2), 248 bytes | 248.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/julia1gin/practice2.git
0f5d6ab..ebca750 master -> master
```