

Основные операции, которые могут быть выполнены с системой контроля версий Git. Обзор базовых команд.

При работе я использовала окно терминала Git Bash, GitHub Desktop, папка с подготовленным репозиторием, справочный материал.

Для инициализации системы контроля версий Git используется команда **git init**.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git init
Reinitialized existing Git repository in C:/Users/Aleksandr/Documents/GitHub/ep-2/.git/
```

С помощью команд **ls**, **ls -la** мы можем увидеть скрытую инициализированную только что папку.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ ls

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ ls -la
total 5
drwxr-xr-x 1 Aleksandr 197121  0 сен 27 14:09 ./
drwxr-xr-x 1 Aleksandr 197121  0 сен 27 14:09 ../
drwxr-xr-x 1 Aleksandr 197121  0 сен 27 15:59 .git/
-rw-r--r-- 1 Aleksandr 197121 66 сен 27 14:09 .gitattributes
```

Внутри этой скрытой папки будут храниться элементы системы контроля версий Git.

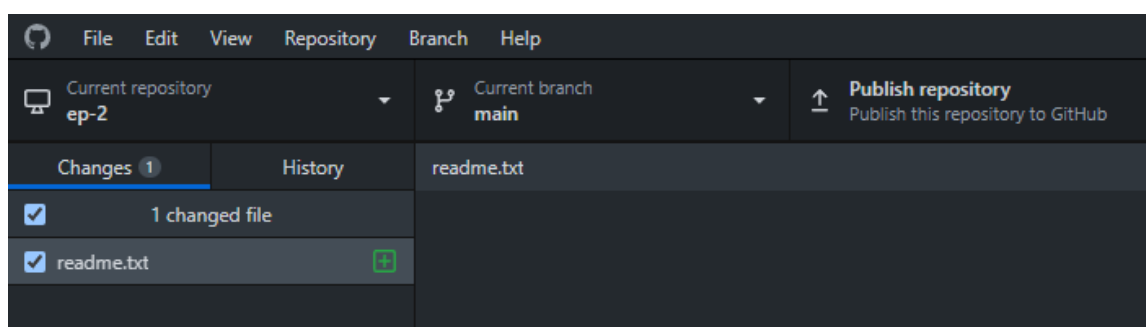
Чтобы определить статус репозитория необходимо ввести команду **git status**.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
nothing to commit, working tree clean
```

Так как я еще ни с чем не работала, данный каталог пустой, ничего подтверждать не надо, никаких действий совершать не надо.

Создадим файл **readme.txt** с помощью команды **touch**.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ touch readme.txt
```



С помощью команды **nano** можно ввести какой-то текст в существующий файл.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ nano readme.txt
```



Проверим статус:

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
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)
```

Можно увидеть, что файл был создан, однако он не отслеживается. Отслеживать файл можно с помощью команды **git add**

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git add readme.txt
warning: LF will be replaced by CRLF in readme.txt.
The file will have its original line endings in your working directory
```

Теперь файл является отслеживаемым.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   readme.txt
```

Зафиксируем данный файл с помощью команды `git commit -m "[комментарий]"`

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git commit -m "First file"
[main 6dc69fc] First file
1 file changed, 2 insertions(+)
create mode 100644 readme.txt
```

Теперь пока не добавлено новых файлов, нам нечего отслеживать или фиксировать.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
nothing to commit, working tree clean
```

Если мы внесем какую-нибудь новую информацию в зафиксированный файл, то он после этого станет снова не отслеживаемым. Сделаем его вновь отслеживаемым.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ nano readme.txt

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
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")

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git add readme.txt
warning: LF will be replaced by CRLF in readme.txt.
The file will have its original line endings in your working directory
```

Для быстроты и удобства можно добавить маску, по которой можно закоммитить множество файлов. Например:

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git add '*.txt'

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   readme.txt
```

Создадим копии файла `readme.txt` – `readme after.txt`, `readme now.txt`. Можно увидеть, что отслеживается лишь исходный файл, а две его копии – нет.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
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
```

Исправим это, сделав все файлы отслеживаемыми. Используем маску.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git add '*.txt'
warning: LF will be replaced by CRLF in readme after.txt.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in readme now.txt.
The file will have its original line endings in your working directory

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   readme after.txt
    new file:   readme now.txt
    modified:   readme.txt
```

Теперь все файлы отслеживаются. Зафиксируем все файлы.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git commit -m "All files with txt extension were added"
[main 8a57cc5] All files with txt extension were added
 3 files changed, 3 insertions(+), 2 deletions(-)
 create mode 100644 readme after.txt
 create mode 100644 readme now.txt
```

Для того, чтобы посмотреть историю всех коммитов (от самого нового, до самого старого), необходимо использовать команду **git log**.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git log
commit 8a57cc5d89fa62949c85ac03270ba5e5582e1168 (HEAD -> main)
Author: Emma Muller <umnykote@mail.ru>
Date:   Tue Sep 27 17:28:08 2022 +0300

    All files with txt extension were added

commit 6dc69fcac3c407fc176993f931f5337874b43d724
Author: Emma Muller <umnykote@mail.ru>
Date:   Tue Sep 27 17:05:10 2022 +0300

    First file

commit 8b80c968b4c00b1ce0c1b8dade7f5163a15dae8e
Author: Emma Muller <umnykote@mail.ru>
Date:   Tue Sep 27 14:09:35 2022 +0300

    Initial commit
```

С помощью команды `git log --summary` на экран выведется больше информации, чем при `git log`.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git log --summary
commit 8a57cc5d89fa62949c85ac03270ba5e5582e1168 (HEAD -> main)
Author: Emma Muller <umnykote@mail.ru>
Date: Tue Sep 27 17:28:08 2022 +0300

    All files with txt extension were added

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

commit 6dc69fcac3c407fc176993f931f537874b43d724
Author: Emma Muller <umnykote@mail.ru>
Date: Tue Sep 27 17:05:10 2022 +0300

    First file

    create mode 100644 readme.txt

commit 8b80c968b4c00b1ce0c1b8dade7f5163a15dae8e
Author: Emma Muller <umnykote@mail.ru>
Date: Tue Sep 27 14:09:35 2022 +0300

    Initial commit

    create mode 100644 .gitattributes
```

Добавление удаленного репозитория. Чтобы добавить удаленный репозиторий, необходимо написать команду `git remote add [название (чаще всего origin)] [ссылка на репозиторий]`. Я заранее создала репозиторий, поэтому имею готовую ссылку.

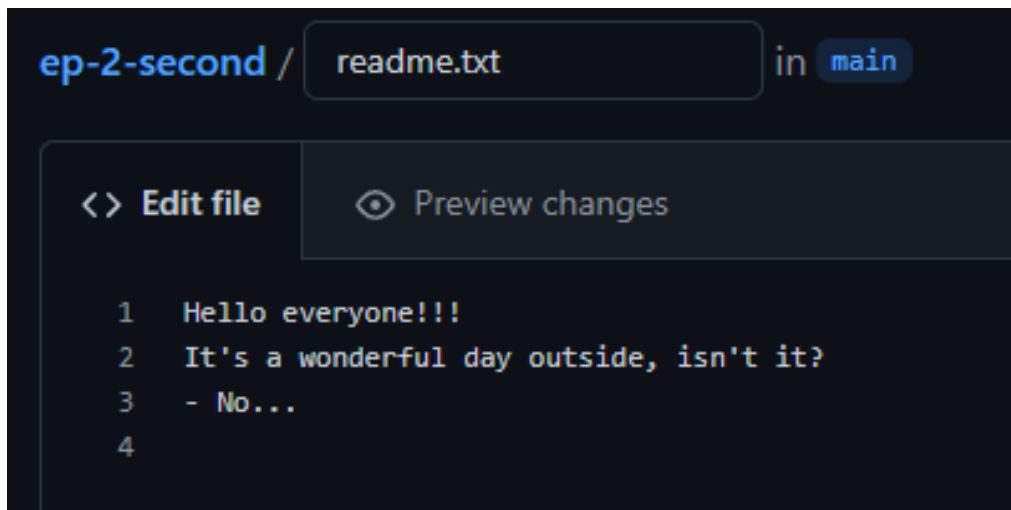
```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git remote add origin https://github.com/gabyshut/ep-2-second.git
```

После этого будет предложено ввести имя пользователя и специальный ключ, который можно создать в настройках аккаунта (я делала на GitHub). После ввода ключа создается удаленный репозиторий, связанный с созданным на платформе GitHub. Импортируем все файлы репозитория «ep-2» в удаленный репозиторий с помощью команды `git push -u [название удаленного репозитория] main (master)`.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git push -u origin main
Logon failed, use ctrl+c to cancel basic credential prompt.
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 884 bytes | 442.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0)
To https://github.com/gabyshut/ep-2-second.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

После этого в созданном на GitHub репозитории появятся все созданные ранее файлы, хранящиеся в репозитории «ep-2». Они доступны для просмотра, если репозиторий публичный.

Внесем изменения в файле readme.txt в удаленном репозитории.



Чтобы изменения также были произведены и в репозитории «ep-2» необходимо использовать команду `git pull origin main (master)`.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 695 bytes | 99.00 KiB/s, done.
From https://github.com/gabyshut/ep-2-second
 * branch            main          -> FETCH_HEAD
   8a57cc5..72d09aa    main          -> origin/main
Updating 8a57cc5..72d09aa
Fast-forward
 readme.txt | 2 ++
 1 file changed, 2 insertions(+)
```

Чтобы сделать какие-либо изменения в файле без лишних коммитов, необходимо ввести `git diff HEAD` и редактировать файл. Например:

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git diff HEAD

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ nano readme.txt

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git diff HEAD
diff --git a/readme.txt b/readme.txt
index 64b969e..5594b9e 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,3 +1,4 @@
 Hello everyone!!!
 It's a wonderful day outside, isn't it?
 - No...
+- why is that so?!
```

HEAD – указатель на последнюю зафиксированную версию.

Создадим каталог (с помощью команды **mkdir**) и файл внутри него. Можно заметить, что каталог не отслеживается.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ mkdir folder

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ touch folder/bye.txt

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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")
```

Чтобы отслеживать данный каталог, необходимо произвести такой ряд действий:

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git add folder/bye.txt

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ cd ..
```

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub (master)
$ ls
ep-2/  praktika/

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub (master)
$ cd ep-2

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   folder/bye.txt

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
```

Теперь каталог отслеживается.



Чтобы сравнить состояние репозитория с тем, которое было, необходимо выполнить операцию `git diff --staged`. Эта команда позволяет посмотреть на те изменения, которые были сделаны в стадии staged.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git diff --staged
diff --git a/folder/bye.txt b/folder/bye.txt
new file mode 100644
index 0000000..e69de29
```

Если мы хотим отменить это действие мы можем выполнить операцию `git reset (folder/bye.txt)`.

Чтобы создать ветку, необходимо ввести команду `git branch [название]`. Чтобы посмотреть, в какой ветке мы работаем, введем команду `git branch`.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git branch clean_up

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git branch
  clean_up
* main
```

Чтобы переключиться на другую ветку, надо ввести команду `git checkout [название ветки]`.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (main)
$ git checkout clean_up
Switched to branch 'clean_up'
A       folder/bye.txt

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (clean_up)
$ git branch
* clean_up
  main
```

Удалю каталог folder. Для этого введу команду `rm -r folder`.

```
Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (clean_up)
$ rm -r folder

Aleksandr@DESKTOP-1JIB8JC MINGW64 ~/Documents/GitHub/ep-2 (clean_up)
$ git commit -m 'Deleted folder and files'
[clean_up 516315c] Deleted folder and files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 folder/bye.txt
```

Основные команды были рассмотрены.