

# Обзор базовых операций GIT

## Инициализация git-репозитория

Для инициализации git-репозитория необходимо перейти в директорию папки, в которой вы хотите создать репозиторий и написать в cmd следующее: «*git init*»

```
D:\GIT>git init
Initialized empty Git repository in D:/GIT/.git/
```

Рисунок 1 Инициализация git-репозитория

В директории появится скрытая папка `.git`:

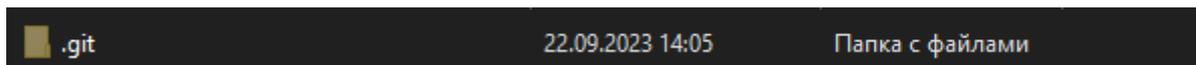


Рисунок 2 Скрытая папка `.git`, появляющаяся в директории после инициализации репозитория

## Получение статуса git-репозитория

Получения статуса репозитория на стандартной ветке `master` осуществляется при помощи команды «*git status*»

```
D:\GIT>git status
On branch master

No commits yet

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

Рисунок 3 Получение статуса пустого репозитория

## Взаимодействие с файлами и изменение их статуса

Создание новой папки `readme.txt` с текстом: “Hello!”

```
D:\GIT>echo "Hello!" > readme.txt
```

Git автоматически обнаруживает новые файлы в директории репозитория:

```
D:\GIT>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)
```

Рисунок 4 Состояние `untracked` созданного файла `readme.txt`

Для отслеживания файла необходимо ввести команду: «*git add (название файла)*»

```
D:\GIT>git add readme.txt
```

```
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   readme.txt
```

Рисунок 5 Новый статус readme.txt – отслеживаемый

Для фиксации состояния репозитория используется команда: «**git commit -m (комментарий)**»

```
D:\GIT>git commit -m "First file"
[master (root-commit) b4a5c3c] First file
1 file changed, 1 insertion(+)
 create mode 100644 readme.txt
```

Рисунок 6 Фиксация состояния репозитория с комментарием First file

```
D:\GIT>git status
On branch master
nothing to commit, working tree clean
```

Рисунок 7 Состояние репозитория

При модификации файла git это определить и присвоит файлу статус modified:

```
D:\GIT>echo "World" > readme.txt
```

Рисунок 8 Изменение содержания файла readme.txt

```
D:\GIT>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")
```

Рисунок 9 Состояние репозитория: файл readme.txt был модифицирован

Для отслеживания нескольких файлов одинакового формата (например txt) можно использовать команду: «**git add „\*.формат”**»

```
D:\GIT>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

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

Рисунок 10 Созданные копии файла readme.txt файлы readme now.txt и readme after.txt

```
D:\GIT>git add "*.txt"
```

```
D:\GIT>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
```

Рисунок 11 Состояние "отслеживание" для всех txt файлов после git add "\*.txt"

```
D:\GIT>git commit -m "All files with txt extension were added"
[master ef733af] 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
```

Рисунок 12 Фиксация состояния репозитория после добавления новых файлов

Просмотр историй фиксаций репозитория можно при помощи команд: «*git log*» и «*git log – summary*» (более подробно). История фиксаций отражена в обратном порядке, т.е. самая первая фиксация будет в самом низу.

```
D:\GIT>git log
commit ef733af86c7878af556e4ff4a870675e0fd61bf0 (HEAD -> master)
Author: 1R1DIUM <grishalyzhnikov2004@gmail.com>
Date:   Fri Sep 22 14:38:37 2023 +0300

    All files with txt extension were added

commit b4a5c3cb4e0f7c790a0cc5a044f086301ad3104b
Author: 1R1DIUM <grishalyzhnikov2004@gmail.com>
Date:   Fri Sep 22 14:27:13 2023 +0300

    First file
```

```
D:\GIT>git log --summary
commit ef733af86c7878af556e4ff4a870675e0fd61bf0 (HEAD -> master)
Author: 1R1DIUM <grishalyzhnikov2004@gmail.com>
Date:   Fri Sep 22 14:38:37 2023 +0300

    All files with txt extension were added

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

commit b4a5c3cb4e0f7c790a0cc5a044f086301ad3104b
Author: 1R1DIUM <grishalyzhnikov2004@gmail.com>
Date:   Fri Sep 22 14:27:13 2023 +0300

    First file

    create mode 100644 readme.txt
```

## Взаимодействие с удаленным репозиторием

Для того, чтобы «привязать» локальный репозиторий к удалённому, необходимо ввести команду: «*git remote add* (Произвольное название удаленного репозитория) (ссылка на удаленный репозиторий на github)»

```
D:\GIT>git remote add origin https://github.com/maaaaageeer/EgorMager.herzen.practice.git
```

Рисунок 13 «Привязка» локального репозитория к удаленному

Для загрузки файлов из локального репозитория в удаленный, необходимо ввести команду: «*git push -u (название удаленного репозитория) (ветку, которую надо загрузить)*»

```
D:\GIT>git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 20 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 507 bytes | 507.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/maaaaageeer/EgorMager.herzen.practice/pull/new/master
remote:
To https://github.com/maaaaageeer/EgorMager.herzen.practice.git
* [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Рисунок 14 Загрузка ветки master в удаленный репозиторий

 readme after.txt	All files with txt extension were added	18 minutes ago
 readme now.txt	All files with txt extension were added	18 minutes ago
 readme.txt	All files with txt extension were added	18 minutes ago

Рисунок 15 Загруженные файлы из локального репозитория на сайте github.com

Для загрузки с удаленного репозитория на локальный репозиторий, необходимо ввести команду: «*git pull (название удаленного репозитория) (ветка, куда загрузить)*»

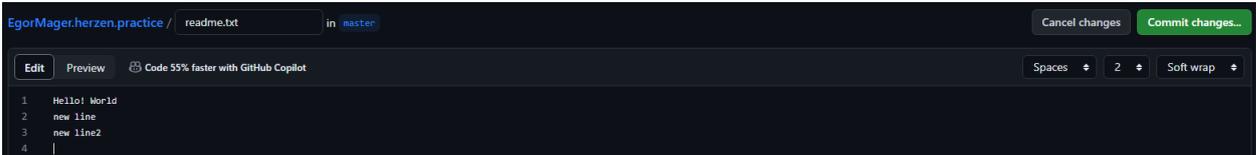


Рисунок 16 Изменение информации в файле readme.txt на удаленном репозитории

```
D:\GIT>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 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 654 bytes | 7.00 KiB/s, done.
From https://github.com/maaaaageeer/EgorMager.herzen.practice
* branch                master      -> FETCH_HEAD
   ef733af..a065bba     master      -> origin/master
Updating ef733af..a065bba
Fast-forward
 readme.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

Рисунок 17 Загрузка новой версии ветки master с удаленного репозитория

## Просмотр изменений в файле

Чтобы посмотреть изменения в файле, нужно ввести команду: «*git diff HEAD*», где HEAD – последнее состояние текущей ветки (master)

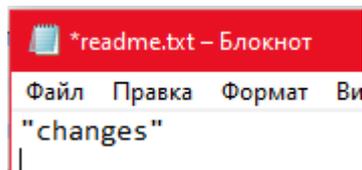


Рисунок 18 Изменение файла readme.txt

```
D:\GIT>git diff HEAD
diff --git a/readme.txt b/readme.txt
index 6bf98da..0b5b031 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,3 +1 @@
-Hello! World
-new line
-new line2
+"changes\n"
```

Рисунок 19 Просмотр изменений в ветке master

## Взаимодействие с каталогами

```
D:\GIT>mkdir folder
D:\GIT>cd folder
D:\GIT\folder>echo > bye.txt
```

Рисунок 20 Создание каталога с файлом bye.txt

```
D:\GIT\folder>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)
  ./

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

Рисунок 21 Статус репозитория: каталог не отслеживается

```
D:\GIT\folder>git add bye.txt
```

Рисунок 22 Отслеживание файла в каталоге

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

Рисунок 23 Состояние репозитория после отслеживания

```
D:\GIT>git diff --staged
diff --git a/folder/bye.txt b/folder/bye.txt
new file mode 100644
index 0000000..cf53638
--- /dev/null
+++ b/folder/bye.txt
@@ -0,0 +1 @@
+<90><A5><A6><A8><AC> <A2> <A4><A0> <AA><AE><AC><A0><AD><A4> <AD><A0> <ED><AA> (ECHO) <A2><AA><AB><EE>
diff --git a/readme.txt b/readme.txt
index 6bf98da..0b5b031 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,3 +1 @@
-Hello! World
-new line
-new line2
+"changes\n"
```

Рисунок 24 Более расширенное сравнение состояний репозитория при помощи «git diff --staged»

## Удаление файла из ветки репозитория

Для удаления файла из ветки нужно ввести команду: «git reset (файл)»

```
D:\GIT>git reset folder/bye.txt
D:\GIT>git diff
```

Рисунок 25 Удаление файла bye.txt из ветки репозитория master

```
D:\GIT>git diff --staged
diff --git a/readme.txt b/readme.txt
index 6bf98da..0b5b031 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,3 +1 @@
-Hello! World
-new line
-new line2
+"changes\n"
```

Рисунок 26 Сравнение состояний репозитория

## Отмена изменений

При случайном удалении или изменении файла можно использовать команду: «git checkout -- (имя файла)»

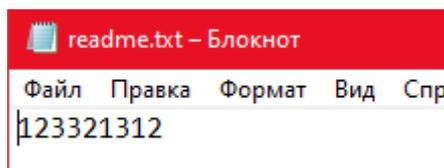


Рисунок 27 «Случайное» изменение содержимого файла

```
D:\GIT>git checkout -- readme.txt
```

Рисунок 28 Отмена изменений в файле

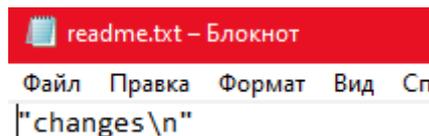


Рисунок 29 Восстановленное содержимое файла

## Создание и взаимодействие с ветками

Для того, чтобы создать новую ветку, необходимо ввести: «*git branch (название новой ветки)*»

```
D:\GIT>git branch clean_up
```

Рисунок 30 Создание новой ветки «clean\_up»

Для просмотра текущей ветки нужно ввести: «*git branch*»

```
D:\GIT>git branch
clean_up
* master
```

Рисунок 31 Просмотр текущей ветки (master)

Для переключения ветки введите: «*git checkout (название существующей ветки)*»

```
D:\GIT>git checkout clean_up
Switched to branch 'clean_up'
M       readme.txt

D:\GIT>git branch
* clean_up
master
```

Рисунок 32 Переключение ветки (текущая теперь clean\_up)

Для удаления файлов из отслеживаемых нужно использовать команду: «*git rm (название файла)*»

```
D:\GIT>git rm "readme after.txt"
rm 'readme after.txt'

D:\GIT>git rm "readme now.txt"
rm 'readme now.txt'
```

Рисунок 33 Исключение из отслеживаемых файлов readme after.txt и readme now.txt

```
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
deleted:   readme after.txt
deleted:   readme now.txt
```

Рисунок 34 Статус репозитория после исключения двух файлов из отслеживаемых

```
D:\GIT>git commit -m "Deleted folder and files"
[master 02a9c7c] Deleted folder and files
2 files changed, 2 deletions(-)
delete mode 100644 readme after.txt
delete mode 100644 readme now.txt
```

Рисунок 35 Фиксация изменений в ветке clean\_up

Для слияния веток нужно ввести: «*git merge (название ветки)*»

```
D:\GIT>git merge clean_up
Updating a065bba..b13646d
Fast-forward
 readme.txt | 4 +---
1 file changed, 1 insertion(+), 3 deletions(-)
```

Рисунок 36 Слияние веток master и clean\_up

Удаление ветки происходит при помощи команды: «*git branch -d (название ветки)*»

```
D:\GIT>git branch -d clean_up  
Deleted branch clean_up (was b13646d).
```

Рисунок 37 Удаление ветки *clean\_up*

```
D:\GIT>git push  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.  
Delta compression using up to 20 threads  
Compressing objects: 100% (1/1), done.  
Writing objects: 100% (2/2), 258 bytes | 258.00 KiB/s, done.  
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/maaaaageer/EgorMager.herzen.practice.git  
b13646d..02a9c7c master -> master
```

Рисунок 38 Загрузка изменённой ветки *master*, которая содержит один файл *readme.txt*



Рисунок 39 Обновленное состояние удаленного репозитория