

- создать локальный репозиторий в текущей папке:

```
user@waez MINGW64 /d/GitReposit
$ git init
Initialized empty Git repository in D:/GitReposit/.git/
```

- посмотреть статус текущего репозитория:

```
user@waez MINGW64 /d/GitReposit (master)
$ git status
On branch master

No commits yet

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

- что такое ветка и какая ветка является обычно основной:  
*Ветка в Git - это линия разработки. Обычно основной веткой является master или main.*
- добавить файл в контекст, который будет коммититься:

```
user@waez MINGW64 /d/GitReposit (master)
$ echo privet>text.txt

user@waez MINGW64 /d/GitReposit (master)
$ git add text.txt
warning: in the working copy of 'text.txt', LF will be replaced by CRLF the next
time Git touches it
```

- создать коммит на основе текущего контекста и указать для него комментарий:

```
user@waez MINGW64 /d/GitReposit (master)
$ git commit -m "Main character name"
[master (root-commit) 172ffc6] Main character name
1 file changed, 1 insertion(+)
create mode 100644 text.txt
```

- создать коммит, включающий изменения всех наблюдаемых файлов и указать для него комментарий:

```
user@waez MINGW64 /d/GitReposit (master)
$ echo sobaka | tee -a text.txt
sobaka

user@waez MINGW64 /d/GitReposit (master)
$ echo slonim>belarus.txt

user@waez MINGW64 /d/GitReposit (master)
$ git add belarus.txt
warning: in the working copy of 'belarus.txt', LF will be replaced by CRLF the next time Git touches it

user@waez MINGW64 /d/GitReposit (master)
$ git commit -a -m "sobaka & slonim"
warning: in the working copy of 'text.txt', LF will be replaced by CRLF the next time Git touches it
[master 032eead] sobaka & slonim
2 files changed, 2 insertions(+)
create mode 100644 belarus.txt
```

- посмотреть протокол(лог) коммитов:



- 

```
MINGW64:/c/Games/git
Artem@DESKTOP-J8ICEQP MINGW64 /c/Games/git (master)
$ git log
commit 60e42956a84199fcdc6e429/dfceadac5e78e16c (HEAD -> master)
Author: Artem-cellq <artempob311@g,ail.com>
Date: Wed Nov 27 20:30:40 2024 +0300

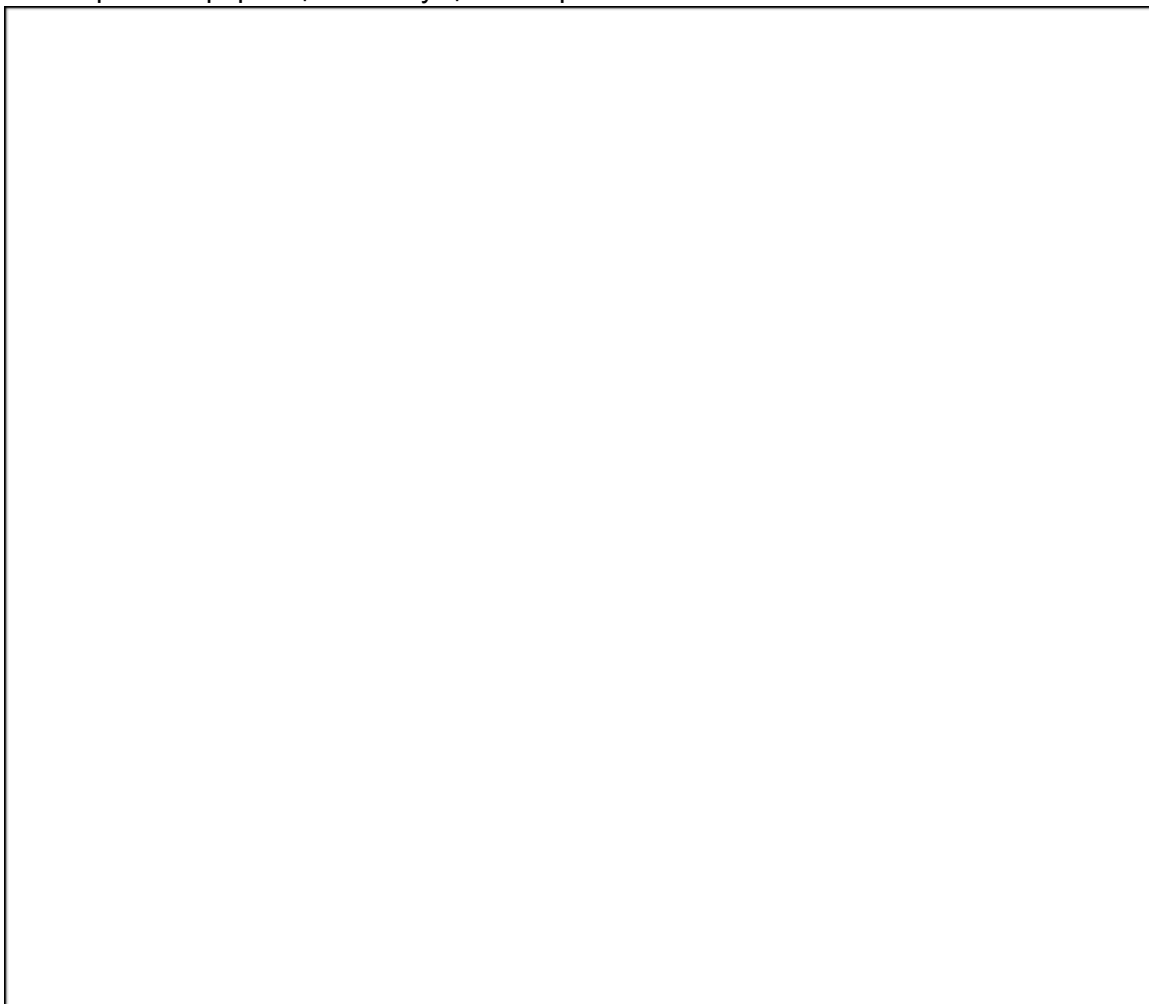
    sobaka & slonim

commit fcfcb1898943b3e24303735bb7cf17ee5f59c5d7
Author: Artem-cellq <artempob311@g,ail.com>
Date: Wed Nov 27 20:25:55 2024 +0300

    Main character name
```

-

- посмотреть информацию о текущих настройках:



```
user@waez MINGW64 /d/GitReposit (master)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=D:/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
gui.recentrepo=C:/Users/user/Desktop/Git учусь
user.name=waez
user.email=onewaezwork@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
```

- убрать файл из контекста:

```
user@waez MINGW64 /d/GitReposit (master)
$ git reset belarus.txt
```

- посмотреть изменения в файле по сравнению с последним коммитом:

```
Artem@DESKTOP-J8ICEQP MINGW64 /c/Games/git (master)
$ git log text.txt
commit 60e42956a84199fcdc6e4297dfceadac5e78e16c (HEAD -> master)
Author: Artem-cellq <artempob311@g,ail.com>
Date:   Wed Nov 27 20:30:40 2024 +0300

    sobaka & slonim

commit fcfcb1898943b3e24303735bb7cf17ee5f59c5d7
Author: Artem-cellq <artempob311@g,ail.com>
Date:   Wed Nov 27 20:25:55 2024 +0300

    Main character name

Artem@DESKTOP-J8ICEQP MINGW64 /c/Games/git (master)
$ git diff text.txt
```

- убрать изменения относительно последнего коммита из файла:

```
user@waez MINGW64 /d/GitReposit (master)
$ echo poka>text.txt
```

```
user@waez MINGW64 /d/GitReposit (master)
$ git checkout text.txt
Updated 1 path from the index
```

- добавить в контекст коммита все измененные и созданный файлы:

```
user@waez MINGW64 /d/GitReposit (master)
$ git add .
```

- изменить глобальные/локальные настройки:

## ГЛОБАЛЬНЫЕ

```
user@waez MINGW64 /d/GitReposit (master)
$ git config --global user.email "onewaezwork@gmail.com"
```

```
user@waez MINGW64 /d/GitReposit (master)
$ git config --global user.name "waez"
```

## ЛОКАЛЬНЫЕ

```
user@waez MINGW64 /d/GitReposit (master)
$ git config user.email "onewaezwork@gmail.com"
```

```
user@waez MINGW64 /d/GitReposit (master)
$ git config user.name "waez"
```

- переписать имя пользователя:

```
user@waez MINGW64 /d/GitReposit (master)
$ git config --global user.name "waez"
```

- посмотреть существующие ветки:

```
user@waez MINGW64 /d/GitReposit (master)
$ git branch
* master
```

- создать новую ветку:

```
user@waez MINGW64 /d/GitReposit (master)
$ git branch artem

user@waez MINGW64 /d/GitReposit (master)
$ git branch
  artem
* master
```

- переключиться на другую ветку:

```
user@waez MINGW64 /d/GitReposit (master)
$ git checkout artem
Switched to branch 'artem'
```



- создать новую ветку и сразу же переключиться на неё:



```
user@waez MINGW64 /d/GitReposit (artem)
$ git checkout -b ANIKV
Switched to a new branch 'ANIKV'

user@waez MINGW64 /d/GitReposit (ANIKV)
$
```

- удалить ветку/удалить ветку, даже если она не примержена:



```
user@waez MINGW64 /d/GitReposit (ANIKV)
$ git branch -D artem
Deleted branch artem (was 032eead).
```

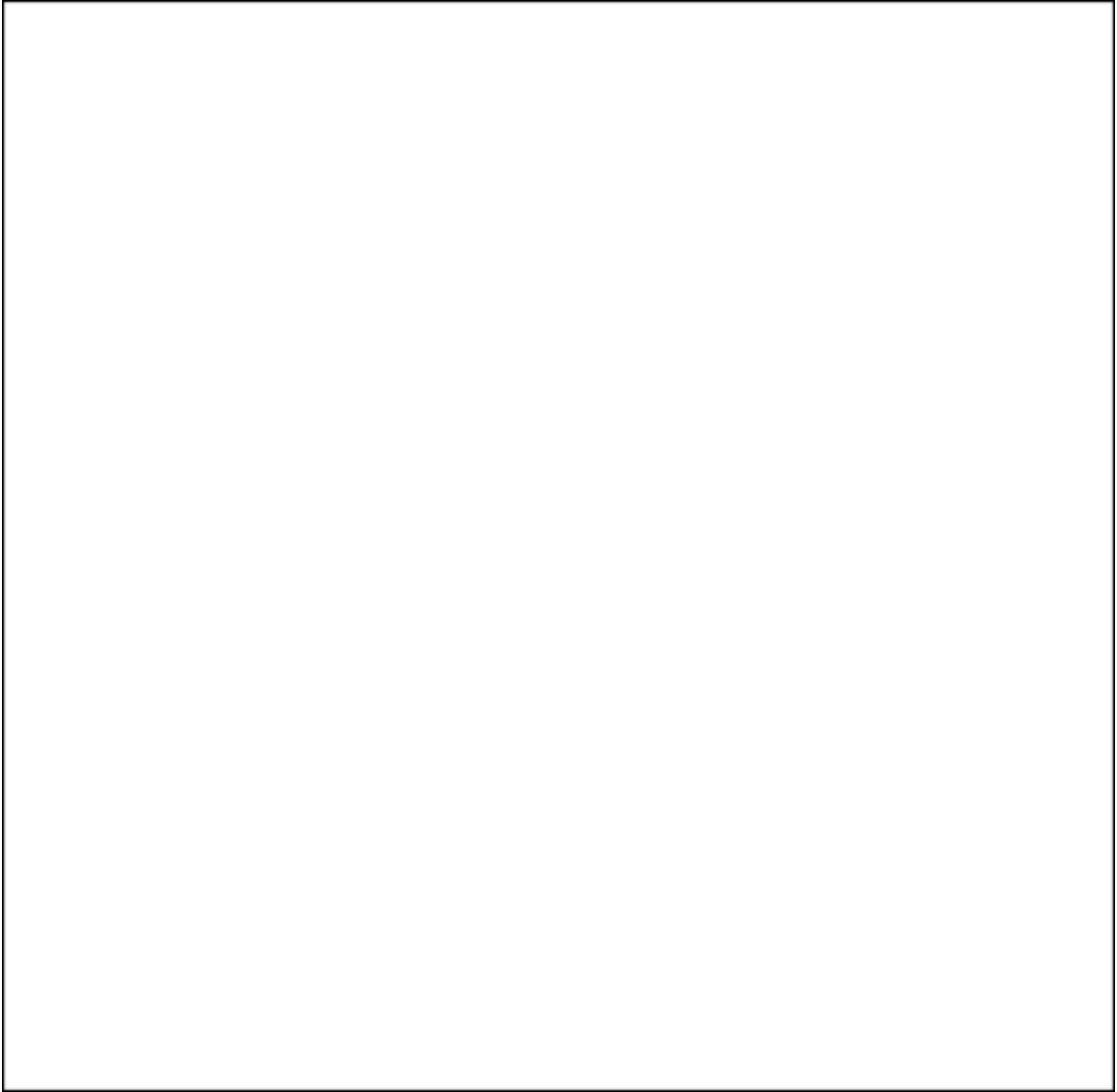
- примержить изменения из указанной ветки в текущую:



```
user@waez MINGW64 /d/GitReposit (ANIKV)
$ git merge master
Already up to date.
```

- в каком случае могут появиться конфликты? сделать конфликт:  
*Конфликты могут возникнуть в случаях, когда несколько членов команды работают над одним и тем же репозиторием Git, и каждый из них вносит изменения в одни и те же файлы. Конфликты могут возникнуть при слиянии (merge) или ребейзе (rebase) веток, а также при попытке отправить изменения*

*в удаленный репозиторий (push), когда другой пользователь уже сделал изменения в том же файле.*



```
user@waez MINGW64 /d/GitReposit (ANIKV)
$ echo "Tesno">tracks.txt

user@waez MINGW64 /d/GitReposit (ANIKV)
$ git add tracks.txt
warning: in the working copy of 'tracks.txt', LF will be replaced by CRLF the next time Git touches
t

user@waez MINGW64 /d/GitReposit (ANIKV)
$ git commit -m "Tesno"
[ANIKV dbe08fe] Tesno
1 file changed, 1 insertion(+)
create mode 100644 tracks.txt

user@waez MINGW64 /d/GitReposit (ANIKV)
$ git checkout master
Switched to branch 'master'

user@waez MINGW64 /d/GitReposit (master)
$ echo "Ogni">tracks.txt

user@waez MINGW64 /d/GitReposit (master)
$ git add tracks.txt
warning: in the working copy of 'tracks.txt', LF will be replaced by CRLF the next time Git touches
t

user@waez MINGW64 /d/GitReposit (master)
$ git commit -m "Ogni"
[master 0f7fdc2] Ogni
1 file changed, 1 insertion(+)
create mode 100644 tracks.txt

user@waez MINGW64 /d/GitReposit (master)
$ git merge ANIKV
Auto-merging tracks.txt
CONFLICT (add/add): Merge conflict in tracks.txt
Automatic merge failed; fix conflicts and then commit the result.
```

- как посмотреть в каких файлах конфликты:

```
user@waez MINGW64 /d/GitReposit (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both added:      tracks.txt

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

- как устранить конфликты:

```
user@waez MINGW64 /d/GitReposit (master|MERGING)
$ echo "Ogni">>tracks.txt

user@waez MINGW64 /d/GitReposit (master|MERGING)
$ git add tracks.txt
warning: in the working copy of 'tracks.txt', LF will be replaced by CRLF the next time Git touches it

user@waez MINGW64 /d/GitReposit (master|MERGING)
$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   tracks.txt
```

- как переключиться на указанный коммит:

```
user@waez MINGW64 /d/GitReposit (master)
$ git checkout ANIKV
Switched to branch 'ANIKV'
```

- сделать ребазирование(rebase) текущей ветки:

```

user@waez MINGW64 /d/GitReposit (ANIKV)
$ echo BOY$>SALUKI.txt

user@waez MINGW64 /d/GitReposit (ANIKV)
$ git add SALUKI.txt
warning: in the working copy of 'SALUKI.txt', LF will be replaced by CRLF the next time Git touches i
t

user@waez MINGW64 /d/GitReposit (ANIKV)
$ git commit -m "SALUKI.txt"
[ANIKV b0eab99] SALUKI.txt
1 file changed, 1 insertion(+)
create mode 100644 SALUKI.txt

user@waez MINGW64 /d/GitReposit (ANIKV)
$ git checkout master
Switched to branch 'master'

user@waez MINGW64 /d/GitReposit (master)
$ git rebase ANIKV
Auto-merging tracks.txt
CONFLICT (add/add): Merge conflict in tracks.txt
error: could not apply 0f7fdc2... Ogni
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
hint: Disable this message with "git config advice.mergeConflict false"
Could not apply 0f7fdc2... Ogni

```

27. отменить ребазирование во время конфликтов:

```

user@waez MINGW64 /d/GitReposit (master|REBASE 1/1)
$ git rebase --abort

```

28. пропустить текущий конфликтный коммит и перейти к следующему:

```
user@waez MINGW64 /d/GitReposit (master)
$ git rebase ANIKV
Auto-merging tracks.txt
CONFLICT (add/add): Merge conflict in tracks.txt
error: could not apply 0f7fdc2... Ogni
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort"
hint: Disable this message with "git config advice.mergeConflict false"
Could not apply 0f7fdc2... Ogni

user@waez MINGW64 /d/GitReposit (master|REBASE 1/1)
$ git rebase --skip
Successfully rebased and updated refs/heads/master.
```

29. Как отправить изменения из локального репозитория для указанной ветки в удалённый(дистанционный) репозиторий

29.1 Отправка изменений в указанную ветку:

Убедитесь, что вы находитесь на нужной ветке:

**bash**

Копировать код:

**git checkout имя\_ветки**

Или:

**bash**

Копировать код:

**git switch имя\_ветки**



Отправьте изменения на удалённый репозиторий:

**bash**

Копировать код:

**git push origin имя\_ветки**

Замените origin на имя удалённого репозитория (по умолчанию это origin) и имя\_ветки на название ветки, в которую хотите отправить изменения.

Например, если вы хотите отправить изменения из ветки **feature-branch**, выполните:

**bash**

Копировать код:

**git push origin feature-branch**

**30.** Как забрать изменения из репозитория, для которого были созданы удаленные ветки по умолчанию.

30.1 Откройте проект в GitHub Desktop.

30.2 Переключитесь на нужную ветку.

30.3 Нажмите кнопку Fetch origin, чтобы забрать изменения.

30.4 Если в удалённой ветке есть новые изменения, которые отсутствуют в локальной, появится кнопка Pull, на которую можно нажать, чтобы забрать и объединить изменения.

**31.** Как забрать изменения удалённой ветки из репозитория по умолчанию, основной ветки

31.1 Убедитесь, что вы находитесь на ветке main.

31.2 Нажмите кнопку Fetch origin, чтобы забрать изменения.

31.3 Если есть новые изменения в origin/main, появится кнопка Pull. Нажмите на неё, 31.4 чтобы получить и объединить изменения с вашей локальной веткой.

**32.** Как клонировать репозиторий

32.1 Нажмите File > Clone Repository.

32.2 Вставьте URL репозитория в поле или выберите репозиторий из списка, если он находится в вашем аккаунте.

32.4 Выберите путь для сохранения репозитория на вашем компьютере.

Нажмите Clone.

**33.** Как переименовать последний коммит

Чтобы переименовать последний коммит в Git, используйте команду

**git commit --amend**

Эта команда позволяет внести изменения в последний коммит, включая изменение его сообщения.

**35.** Как скрыть изменения по сравнению с последним коммитом

Чтобы скрыть изменения в файлах по сравнению с последним коммитом в Git, вы можете использовать команду

**git checkout**

Или

**git restore.**

Эти команды позволят вам отменить все незакоммиченные изменения и вернуть файлы к состоянию последнего коммита.

