

1. создать локальный репозиторий в текущей папке

```
hudson@savaiche MINGW64 /d/Git
$ git init
Initialized empty Git repository in D:/Git/.git/
```

2. посмотреть статус текущего репозитория

```
hudson@savaiche MINGW64 /d/Git (master)
$ git status
On branch master

No commits yet

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

3. что такое ветка и какая ветка является обычно основной

*Ветка в Git — это отдельная линия разработки. Обычно основной веткой является **main** или **master**.*

4. добавить файл в контекст, который будет коммититься

```
hudson@savaiche MINGW64 /d/Git (master)
$ touch hello.txt

hudson@savaiche MINGW64 /d/Git (master)
$ git add hello.txt
```

5. создать коммит на основе текущего контекста и указать для него комментарий

```
hudson@savaiche MINGW64 /d/Git (master)
$ git commit -m "Created hello.txt"
[master (root-commit) da6d74b] Created hello.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hello.txt
```

6. создать коммит, включающий изменения всех наблюдаемых файлов и указать для него комментарий

```
hudson@savaiche MINGW64 /d/Git (master)
$ echo "hello friends!" > hello.txt

hudson@savaiche MINGW64 /d/Git (master)
$ echo "pepsi,sprite, svoyak" > drinks.txt

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

hudson@savaiche MINGW64 /d/Git (master)
$ git commit -a -m "edit first docx & created drinks.txt"
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the next time Git touches it
[master 900a700] edit first docx & created drinks.txt
2 files changed, 2 insertions(+)
create mode 100644 drinks.txt
```

## 7. посмотреть протокол(лог) коммитов

```
hudson@savaiche MINGW64 /d/Git (master)
$ git log
commit 900a700e450b0d4c16402e403dc8ff493a807d27 (HEAD -> master)
Author: savaiche <hudsonried@gmail.com>
Date:   Mon Oct 21 22:55:09 2024 +0300

    edit first docx & created drinks.txt

commit da6d74b6d045302db6f7c5b4233784131e64de9b
Author: savaiche <hudsonried@gmail.com>
Date:   Mon Oct 21 22:50:08 2024 +0300

    Created hello.txt
```

## 8. посмотреть информацию о текущих настройках

```
hudson@savaiche MINGW64 /d/Git (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=C:/Program Files/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
user.name=savaiche
user.email=hudsonried@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
```

## 9. убрать файл из контекста

```
hudson@savaiche MINGW64 /d/Git (master)
$ git reset drinks.txt
```

## 10. посмотреть изменения в файле по сравнению с последним коммитом

```
hudson@savaiche MINGW64 /d/Git (master)
$ echo "Это второоооооой" > hello.txt

hudson@savaiche MINGW64 /d/Git (master)
$ git diff hello.txt
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the next time Git touches it
diff --git a/hello.txt b/hello.txt
index afa0ce4..2193412 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,1 @@
-hello friends!
+Это второоооооой
```

11. убрать изменения относительно последнего коммита из файла

```
hudson@savaiche MINGW64 /d/Git (master)
$ git checkout -- hello.txt
```

12. добавить в контекст коммита все измененные и созданный файлы

```
hudson@savaiche MINGW64 /d/Git (master)
$ echo "52, ДА ЗДРАВСТВУЙ САНКТ-ПЕТЕРБУРГ!!!" >> hello.txt
echo "52, ДА ЗДРАВСТВУЙ САНКТ-ПЕТЕРБУРГ!" >> hello.txt
git commit -m "modified hello.txt!" >> hello.txt

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

13. изменить глобальные/локальные настройки

- Глобальные

```
hudson@savaiche MINGW64 /d/Git (master)
$ git config --global user.email "hudsonied@gmail.com"

hudson@savaiche MINGW64 /d/Git (master)
$ git config --global user.name "savaiche"
```

- Локальные

```
hudson@savaiche MINGW64 /d/Git (master)
$ git config user.email "hudsonied@gmail.com"

hudson@savaiche MINGW64 /d/Git (master)
$ git config user.name "savaiche"
```

14. переписать имя пользователя

```
hudson@savaiche MINGW64 /d/Git (master)
$ git config --global user.name "savaiche"
```

15. просмотреть существующие ветки

```
hudson@savaiche MINGW64 /d/Git (master)
$ git branch
* master
```

16. создать новую ветку

```
hudson@savaiche MINGW64 /d/Git (master)
$ git branch goida
```

17. переключиться на другую ветку

```
hudson@savaiche MINGW64 /d/Git (master)
$ git checkout goida
Switched to branch 'goida'
```

18. создать новую ветку и сразу же переключиться на неё

```
hudson@savaiche MINGW64 /d/Git (goida)
$ git checkout -b CURSEDZXC52
Switched to a new branch 'CURSEDZXC52'
```

19. удалить ветку/удалить ветку, даже если она не примержена

```
hudson@savaiche MINGW64 /d/Git (master)
$ git branch -D CURSEDZXC52
Deleted branch CURSEDZXC52 (was 900a700).
```

20. примержить изменения из указанной ветки в текущую

```
hudson@savaiche MINGW64 /d/Git (goida)
$ git merge master
Updating 900a700..a40d728
Fast-forward
 hello.txt | 1 +
 1 file changed, 1 insertion(+)
```

21. 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.

```
hudson@savaiche MINGW64 /d/Git (goida)
$ echo "SVOAYK, VOZDYX, KONOPLYANKA" > zapretka.txt

hudson@savaiche MINGW64 /d/Git (goida)
$ git add zapretka.txt
warning: in the working copy of 'zapretka.txt', LF will be replaced by CRLF the next time Git touches i
t

hudson@savaiche MINGW64 /d/Git (goida)
$ git commit -m "created conflict"
[goida 0fb22f5] created conflict
1 file changed, 1 insertion(+)
create mode 100644 zapretka.txt

hudson@savaiche MINGW64 /d/Git (goida)
$ git checkout master
Switched to branch 'master'

hudson@savaiche MINGW64 /d/Git (master)
$ echo "ЭЩКЕPEEEEE" > zapretka.txt

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

hudson@savaiche MINGW64 /d/Git (master)
$ git commit -m "created conflict in two branch"
[master 01ab14c] created conflict in two branch
1 file changed, 1 insertion(+)
create mode 100644 zapretka.txt

hudson@savaiche MINGW64 /d/Git (master)
$ git merge goida
Auto-merging zapretka.txt
CONFLICT (add/add): Merge conflict in zapretka.txt
Automatic merge failed; fix conflicts and then commit the result.
```

22. как посмотреть в каких файлах конфликты

```
hudson@savaiche MINGW64 /d/Git (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:      zapretka.txt

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

## 23. как устранить конфликты

```
hudson@savaiche MINGW64 /d/Git (master|MERGING)
$ echo "zazaza" >> zapretka.txt

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

hudson@savaiche MINGW64 /d/Git (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:   zapretka.txt
```

## 24. как переключиться на указанный коммит

```
hudson@savaiche MINGW64 /d/Git (master)
$ git checkout 0fb22f5
Note: switching to '0fb22f5'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 0fb22f5 created conflict
```

## 25. сделать ребазирование(rebase) текущей ветки

```
hudson@savaiche MINGW64 /d/Git (master)
$ git rebase goida
Auto-merging zapretka.txt
CONFLICT (add/add): Merge conflict in zapretka.txt
error: could not apply 01ab14c... created conflict in two branch
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 01ab14c... created conflict in two branch
```

## 27. отменить ребазирование во время конфликтов

```
hudson@savaiche MINGW64 /d/Git (master|REBASE 1/1)
$ git rebase --abort
```

28. пропустить текущий конфликтный коммит и перейти к следующему

```
hudson@savaiche MINGW64 /d/Git (master)
$ git rebase goida
Auto-merging zapretka.txt
CONFLICT (add/add): Merge conflict in zapretka.txt
error: could not apply 01ab14c... created conflict in two branch
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 01ab14c... created conflict in two branch

hudson@savaiche MINGW64 /d/Git (master|REBASE 1/1)
$ git rebase --skip
Successfully rebased and updated refs/heads/master.
```

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

```
hudson@savaiche MINGW64 /d/Git (master)
$ git remote add origin https://github.com/savaiche/lab_git.git

hudson@savaiche MINGW64 /d/Git (master)
$ git push origin master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (13/13), 1.07 KiB | 1.07 MiB/s, done.
Total 13 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/savaiche/lab_git.git
 * [new branch]      master -> master
```

30. забрать изменения из репозитория, для которого были созданы удалённые ветки по умолчанию

```
hudson@savaiche MINGW64 /d/Git (master)
$ git fetch origin
```

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

```
hudson@savaiche MINGW64 /d/Git (master)
$ git fetch origin master
From https://github.com/savaiche/lab_git
 * branch            master      -> FETCH_HEAD
```

32. создание копии репозитория

```
hudson@savaiche MINGW64 /d/Git (master)
$ git clone https://github.com/savaiche/lab_git.git
Cloning into 'lab_git'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 13 (delta 1), reused 13 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (1/1), done.
```



33. переименовать последний коммит

```
hudson@savaiche MINGW64 /d/Git (master)
$ git commit --amend -m "change name last commit"
[master 339f91e] change name last commit
Date: Mon Oct 21 23:23:49 2024 +0300
1 file changed, 1 insertion(+)
create mode 100644 zapretka.txt
```

34. скрыть изменения по сравнению с последним коммитом

```
hudson@savaiche MINGW64 /d/Git (master)
$ git stash
No local changes to save
```