

Git guide

By Eliska Vrzalova


Git init

Git initialize a new empty repository, we are now on the master branch.

After we do git init command .git folder is created in the folder where we have initialized our repository.

We can initialize our repository on local machine (this case) or we can initialize our repository through GitHub

Repository can be initialized (using command git init) only once, if we initialized our repository through GitHub and want to start in on desktop on local machine, we need to use git clone and paste the URL of our repository from GitHub

 MINGW64:/c:/Users/C00301768/Desktop/git one Vrzalova

```
C00301768@CD-A307-30 MINGW64 /c:/Users/C00301768/Desktop/git one Vrzalova
$ git init
Initialized empty Git repository in C:/Users/C00301768/Desktop/git one Vrzalova/.git/

C00301768@CD-A307-30 MINGW64 /c:/Users/C00301768/Desktop/git one Vrzalova (master)
$ |
```

Git add

Git add command adds files in the working directory to the stages area.

If we have added .gitignore file git add command excluded those files that match lines from .gitignore file

Git add . – adds all the files

Git add filename (for ex. git guide txt.txt) – adds only the file we stated in the command.

If I modify my file after I added it – I should add them again and git will ignore the previous version of that file

In the screen shots I did git status first so I can see which files are not tracked by git (red color).

After I added the files and did git status again I was able to see that the file are now tracked and Git knows about them (green color).

```

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    git Guide word.docx
    git guide txt.txt
    ~$t Guide word.docx
    ~WRL0005.tmp

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

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
    add

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git add .

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git add .

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   git Guide word.docx
    new file:   git guide txt.txt
    new file:   ~$t Guide word.docx
    new file:   ~WRL0005.tmp

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$

```

Git status

Command gist status lists files.

It shows me whether the files are modified, untracked, added

It list files on the branch we are on at the moment (master in this case)

If git status shows me untracked files and I want hem to be tracked, I use git add command

```

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    git Guide word.docx
    git guide txt.txt
    ~$t Guide word.docx
    ~WRL0005.tmp

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

```

```

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git add .

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   git Guide word.docx
        new file:   git guide txt.txt
        new file:   ~$t Guide word.docx
        new file:   ~WRL0005.tmp

```

Git commit

Git commit command creates a new revision point on local repository. It is saving changes

It creates a snapshot of the staged changes – for example in the screenshot I can see that I have added 4 files so far

Words after – m are comment, a label for this commit that needs to be in “ ” if it contains spaces.

```

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git commit -m "first commit"
[master (root-commit) 369b649] first commit
Committer: (Student C00301768) Eliska Vrzalova <C00301768@setu.ie>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

4 files changed, 33 insertions(+)
create mode 100644 git Guide word.docx
create mode 100644 git guide txt.txt
create mode 100644 ~$t Guide word.docx
create mode 100644 ~WRL0005.tmp

```

Git remote

Command git remote is making your repository visible on github. Until now all of my git work was on my local one drive. Things that are remote are on Github server.

It allows me to save my work else where than on my local machine.

Remote= remote repository

Origin = the default name of the remote repository on Github

Master = the default name of the initial brach of a repository

Origin master = default branch of repository on github

```

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git remote add origin https://github.com/PeterLowe/git-one-EliskaVrzalova.git

C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git remote -v
origin https://github.com/PeterLowe/git-one-EliskaVrzalova.git (fetch)
origin https://github.com/PeterLowe/git-one-EliskaVrzalova.git (push)

```

Git push

Updates the remote repository on Github.

We should push everytime we want to share the piece of work we have done with others. Good thing is also push small and often

```
C00301768@CD-A307-30 MINGW64 /c/Users/C00301768/Desktop/git one Vrzalova (master)
$ git push origin master
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 98.52 KiB | 32.84 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PeterLowe/git-one-EliskaVrzalova.git
 * [new branch]      master -> master
```

Git branch

Git branch command creates a new branch. We need to type in git branch and name of the new branch. I have named it newBranch. My command looked like : git branch newBranch.

Now I have to branched – master and newBranch

```
C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (master)
$ git branch
* master
  newBranch
```

Git checkout

Makes the node the current head.

Very similar to switch command. In my screenshot I switch from newBranch to master and back. I did a git commit command before switching because I had changes on my master branch at the moment I wanted to switch to newBranch. – it ensured that the working directory matched the node This command will not work if there are modified files

```
C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (newBranch)
$ git checkout newBranch
Already on 'newBranch'

C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (newBranch)
$ git checkout master
Switched to branch 'master'

C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (master)
$ git branch
* master
  newBranch

C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (master)
$ git status
On branch master
nothing to commit, working tree clean

C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (master)
$ git checkout newBranch
Switched to branch 'newBranch'
```

Git merge

This command takes content from one branch and merge it into another branch. These branches still remain separate it is not like they become one.

There can be a merge conflict (for ex when 2 people were working separately and both of them created a variable of the same name- git then does not know what to do what to choose)

We should merge small and often.

We should merge only when it is ready for everyone to do so.

```
C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (master)
$ git checkout newBranch
Switched to branch 'newBranch'

C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (newBranch)
$ git merge master
Merge made by the 'ort' strategy.
  git guide txt.txt      | 60 ++++++-----
  ~$t Guide word.docx   | Bin 162 -> 0 bytes
  ~WRL0005.tmp          | 0
  3 files changed, 57 insertions(+), 3 deletions(-)
  delete mode 100644 ~$t Guide word.docx
  delete mode 100644 ~WRL0005.tmp

C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (newBranch)
$ git status
On branch newBranch
nothing to commit, working tree clean

C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (newBranch)
$ git commit -m "newBranch merged with master"
On branch newBranch
nothing to commit, working tree clean
```

Git log

This command displays entire commit history. We can see all the git commit commands we have done so far.

```
C00301768@CD-A003-10 MINGW64 /c/Users/c00301768/Desktop/git one Vrzalova (master)
$ git log
commit bb4aa1fee935098fbef774cf088c188ad39a7c10 (HEAD -> master)
Author: (Student C00301768) Eliska Vrzalova <C00301768@setu.ie>
Date:   Fri Oct 20 17:38:54 2023 +0100

    before checkout

commit 36d6d7fc476c50fc8cea8006d73223a09ec92471
Author: (Student C00301768) Eliska Vrzalova <C00301768@setu.ie>
Date:   Fri Oct 20 17:29:02 2023 +0100

    branch created

commit 369b64916156406a402a1520bb8826eb6cb6e696 (origin/master)
Author: (Student C00301768) Eliska Vrzalova <C00301768@setu.ie>
Date:   Fri Oct 20 10:14:57 2023 +0100

    first commit
```