MANIPAL GLOBAL SKILLS ACADEMY

REPORT ON
"GIT"

SUBMITTED BY

B YAJNESH KUMAR MGSA_081

UNDER THE GUIDANCE OF

MRS. PRIYADARSHINI

TABLE OF CONTENTS

INTRODUCTION

1.1 GIT AND GIT HUB

Git is a free and open-source distributed version control system for tracking changes in source code and is used for software development. It was created by Linus Torvalds (creator and developer of Linux) in 2005. It allows you to record changes in your file over time so that you can access specific versions of your file later on. Moreover, you can also coordinate with several people working on a single file or files with the help of git. So git is basically a content tracker.

GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command-line tool, GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, such as wikis and basic task management tools for every project.

OBJECTIVES

In this report you will get to know about the
-Learn what version control is
-Understand distributed version control systems, like Git

Version control

A version control system (VCS) is a program or set of programs that tracks changes to a collection of files. One goal of a VCS is to easily recall earlier versions of individual files or of the entire project. Another goal is to allow several team members to work on a project, even on the same files, at the same time without affecting each other's work.

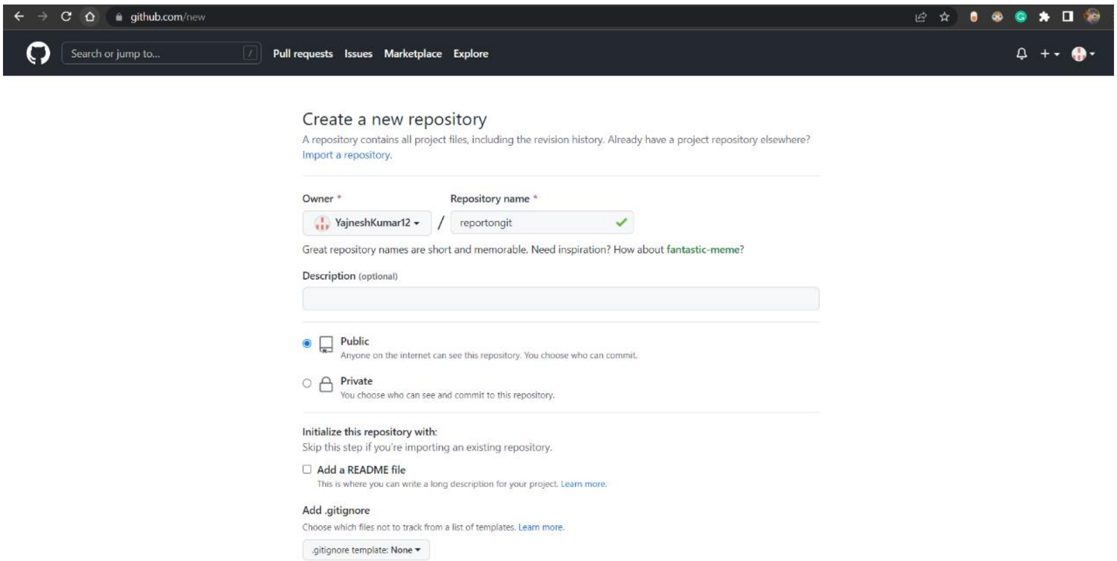
Distributed version control

Git is distributed, which means that a project's complete history is stored both on the client *and* on the server. You can edit files without a network connection, check them in locally, and sync with the server when a connection becomes available. If a server goes down, you still have a local copy of the project.

GIT COMMANDS

CREATING REPOSIORY IN GITHUB

To create a repository with the directory structure and files of an existing repository, you can see the below figure. In initialize some things like, naming the project, choosing the visibility etc. After performing these steps click Create Repository button.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

YajneshKumar12 / reportongit ✓

Great repository names are short and memorable. Need inspiration? How about fantastic-meme?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

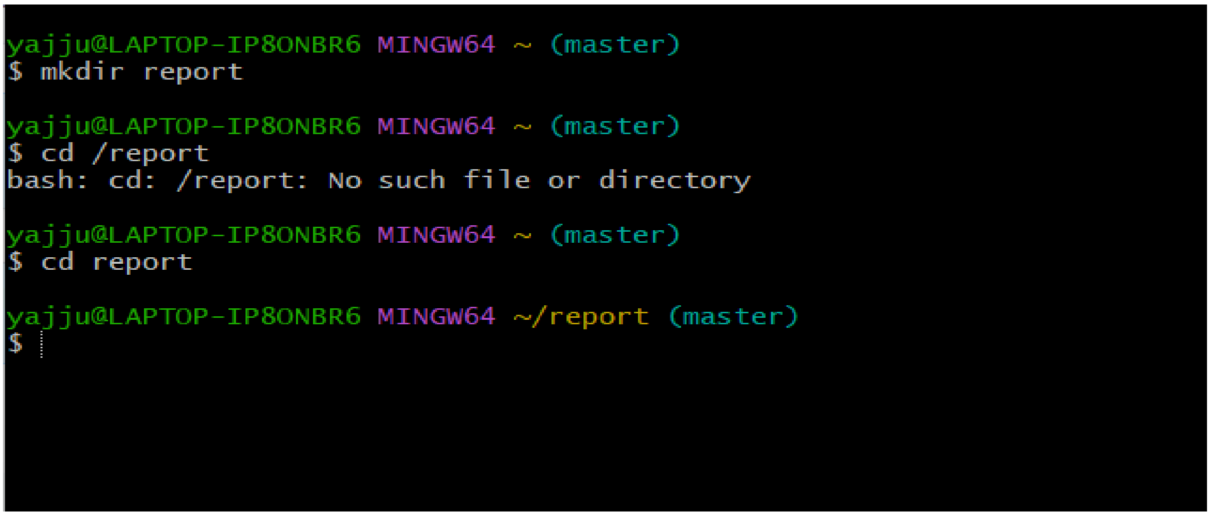
□ Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

.gitignore template: None

CREATING NEW DIRECTORY IN LOCAL REPOSITORY

Local repository is nothing but the files which are in the local PC and using the **mkdir** command we can create the new directory or folder in the certain path we want and **cd** command is used to change the directory.

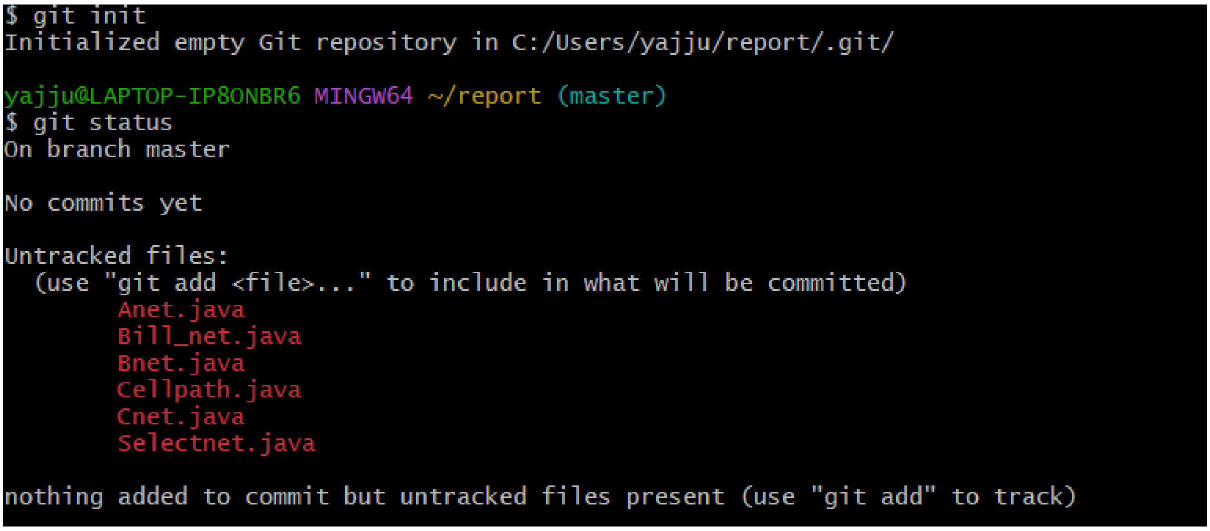


```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~ (master)
$ mkdir report

yajju@LAPTOP-IP8ONBR6 MINGW64 ~ (master)
$ cd /report
bash: cd: /report: No such file or directory

yajju@LAPTOP-IP8ONBR6 MINGW64 ~ (master)
$ cd report

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$
```

GIT INIT AND STATUS COMMAND

Using **git init** command we can create the new git reposiroty in local repository that is .git file will be generated in the certain path we want and it is used to initialize the existing repository or empty repository.

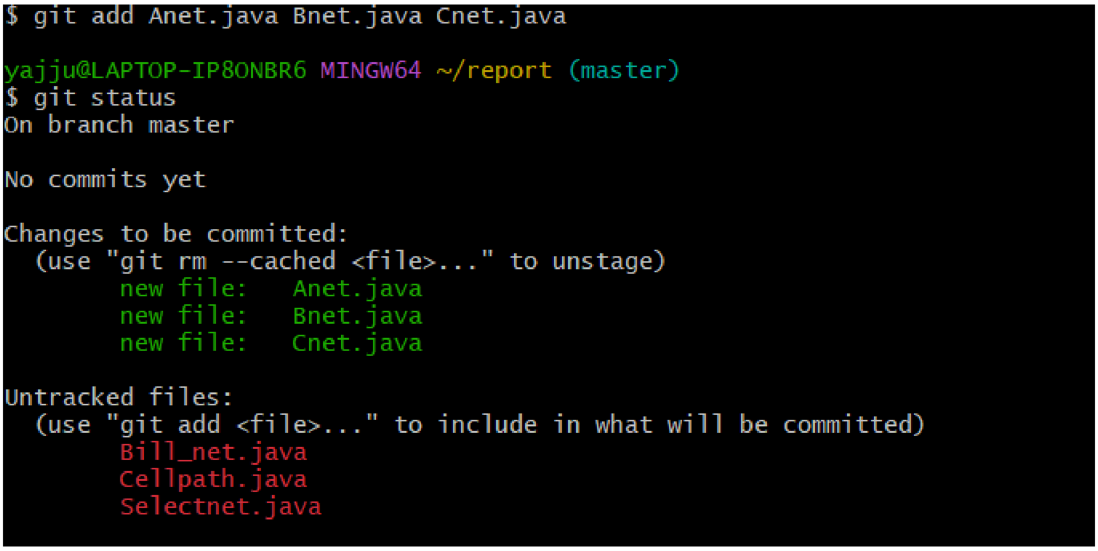
Using **git status** command we can check the status of the directory and if any changes made in the directory will displayed in it.



```
$ git init
Initialized empty Git repository in C:/Users/yajju/report/.git/

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Anet.java
        Bill_net.java
        Bnet.java
        Cellpath.java
        Cnet.java
        Selectnet.java

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

GIT ADD COMMAND WITH STATUS AFTER ADDING FILE

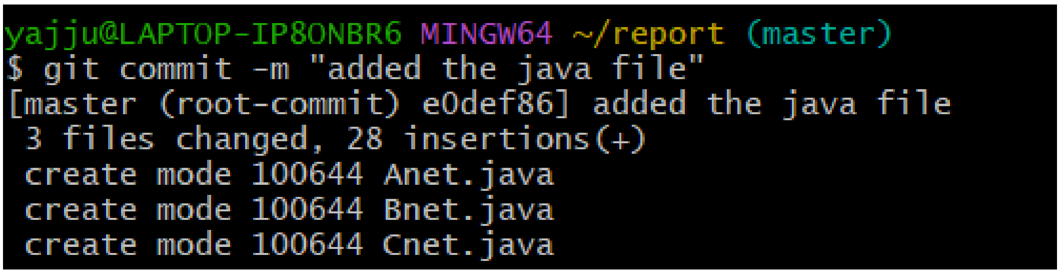
In this figure you can see the files that I want to add, that is Anet, Bnet, Cnet so by using the **git add files** I have added the files to the staging area and we can check the status of the added files using the **git status** command.



```
$ git add Anet.java Bnet.java Cnet.java

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Anet.java
        new file:   Bnet.java
        new file:   Cnet.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Bill_net.java
        Cellpath.java
        Selectnet.java
```

GIT COMMIT COMMAND

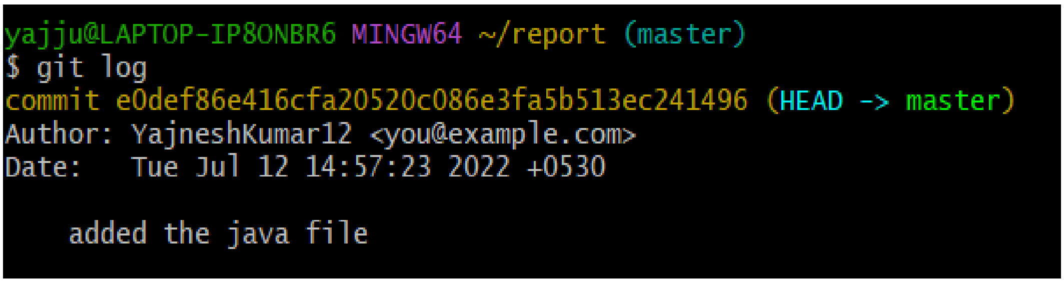
Using the **git commit -m “message”** command we can see the changes made in the project like adding files etc. It can be used to keep track of our progress and changes as we work with the message written during the commit.



```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git commit -m "added the java file"
[master (root-commit) e0def86] added the java file
 3 files changed, 28 insertions(+)
 create mode 100644 Anet.java
 create mode 100644 Bnet.java
 create mode 100644 Cnet.java
```

GIT LOG

Using the **git log** command we can check on the repository once the commit is done.

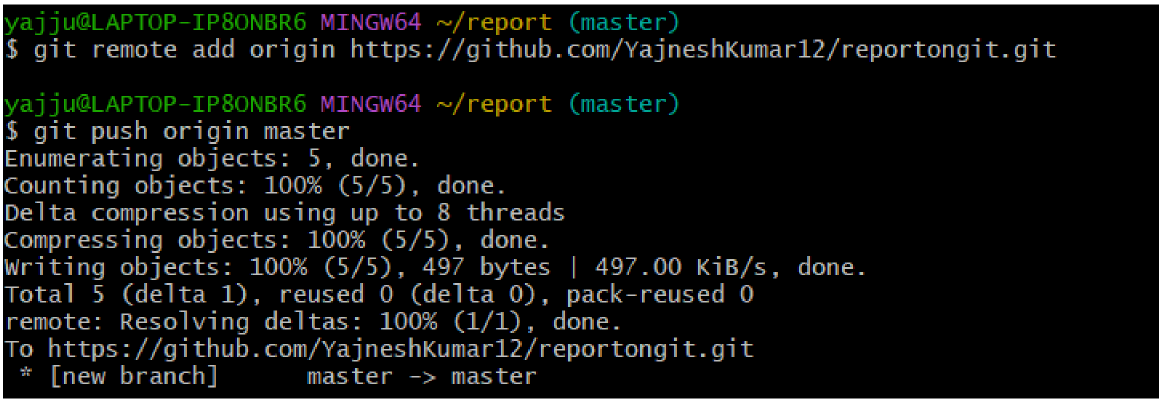


```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git log
commit e0def86e416cfa20520c086e3fa5b513ec241496 (HEAD -> master)
Author: YajneshKumar12 <you@example.com>
Date:   Tue Jul 12 14:57:23 2022 +0530

    added the java file
```

GIT REMOTE AND PUSH COMMAND

Using the **git remote add origin HTTPS** command we can add, view and delete the connection to the remote repository using the HTTPS or SSH. As you can see in the figure I have added the remote repository using the HTTPS.

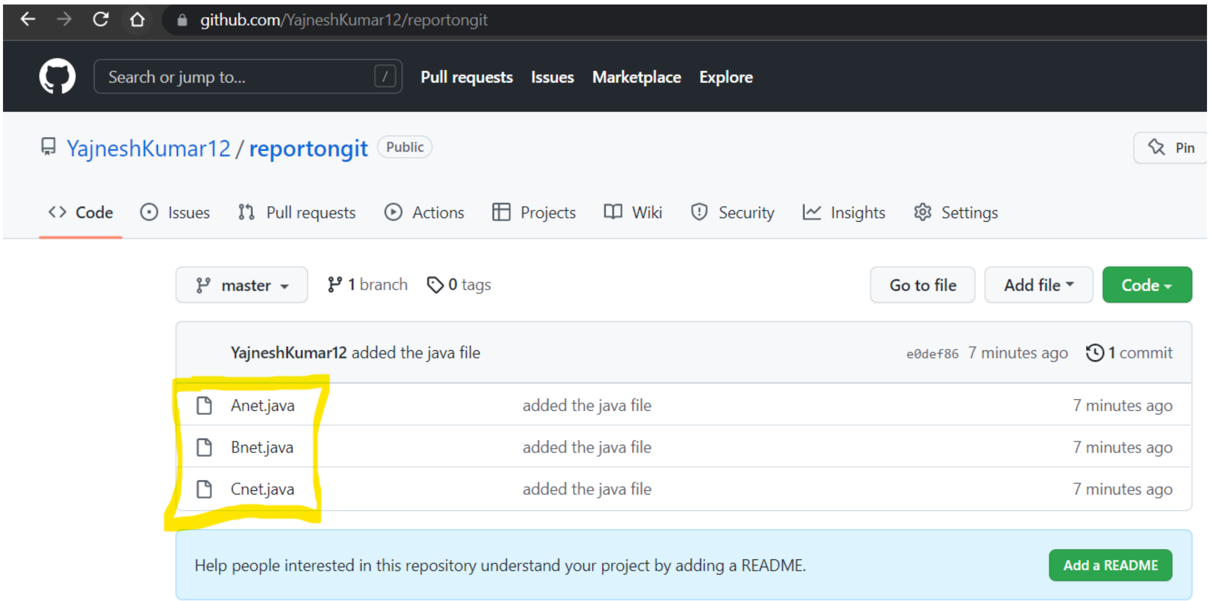
Using the **git push origin master** we can upload the files to the remote repository from the local repository, so use the push command we need to connect to remote repository using the **git remote add origin HTTPS**



```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git remote add origin https://github.com/YajneshKumar12/reportongit.git

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 497 bytes | 497.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/YajneshKumar12/reportongit.git
 * [new branch]      master -> master
```

AFTER USE OF PUSH COMMANDS FILES ADDED IN THE REMOTE REPOSITORY

After using the git push commands you can see the below figure which files are uploaded to the remote repository.



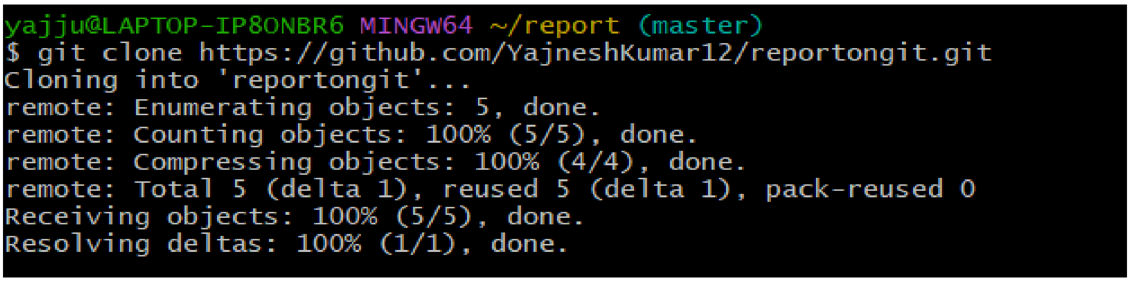
YajneshKumar12 added the java file e0def86 7 minutes ago 1 commit

Anet.java	added the java file	7 minutes ago
Bnet.java	added the java file	7 minutes ago
Cnet.java	added the java file	7 minutes ago

Help people interested in this repository understand your project by adding a README.

GIT CLONE COMMAND

Using **git clone HTTPS** command used to copy an existing Git repository into a new local directory. The Git clone command will create a new local directory for the repository, copy all the contents of the specified repository, create the remote tracked branches, and checkout an initial branch locally.

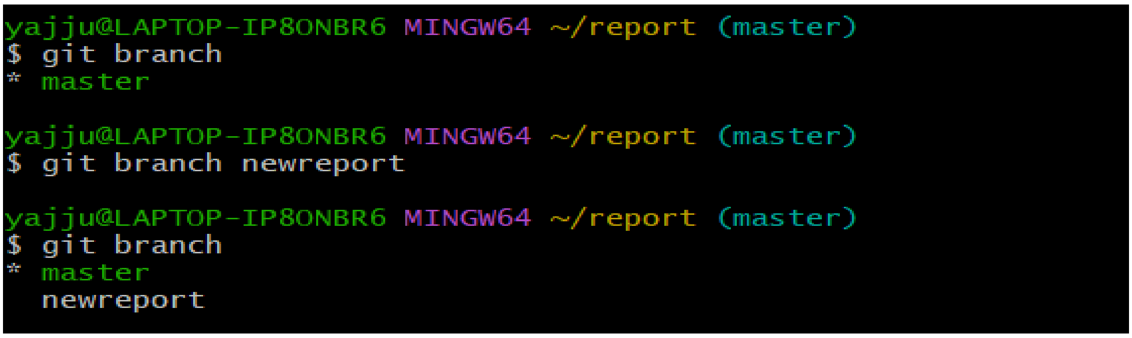


```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git clone https://github.com/YajneshKumar12/reportongit.git
Cloning into 'reportongit'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 1), reused 5 (delta 1), pack-reused 0
Receiving objects: 100% (5/5), done.
Resolving deltas: 100% (1/1), done.
```

GIT BRANCH COMMAND

Using **git branch** command we can check on the status of the branch and how many branch we in the repository

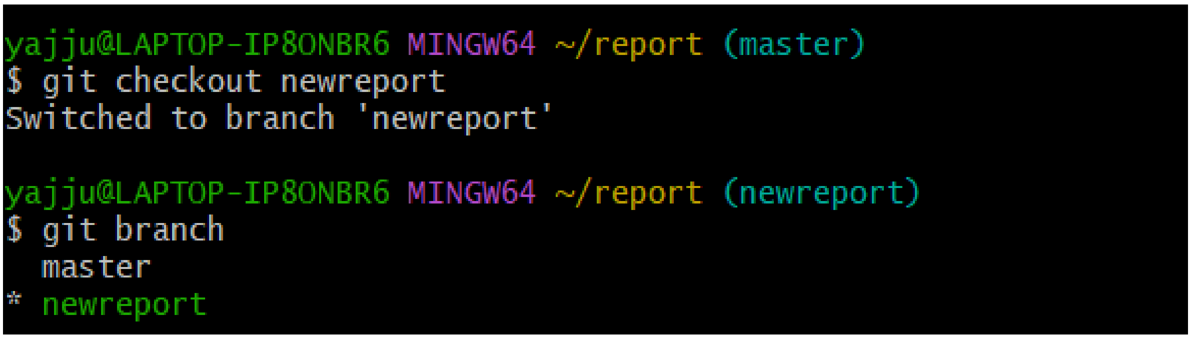
Using the **git branch branchname** we can create the new branch name and we can check on the branch created using the git branch and below figure specifies it.



```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git branch
* master

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git branch newreport

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git branch
* master
  newreport
```

GIT CHECKOUT

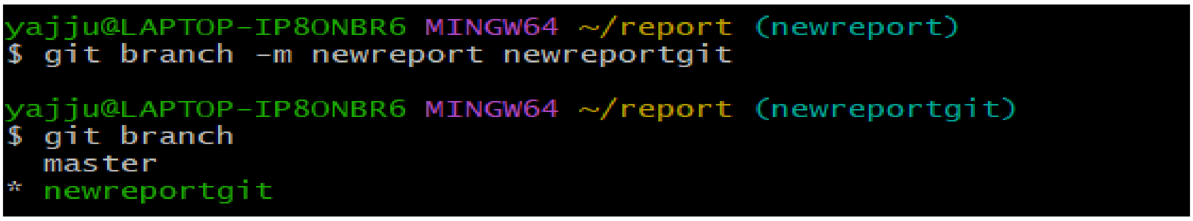
Using the **git checkout branchname** we can switch to different branch with the name exists in it and for example we can switch from the branch master to the branch newreport where the given specifies it.



```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git checkout newreport
Switched to branch 'newreport'

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (newreport)
$ git branch
  master
* newreport
```

GIT RENAME BRANCH

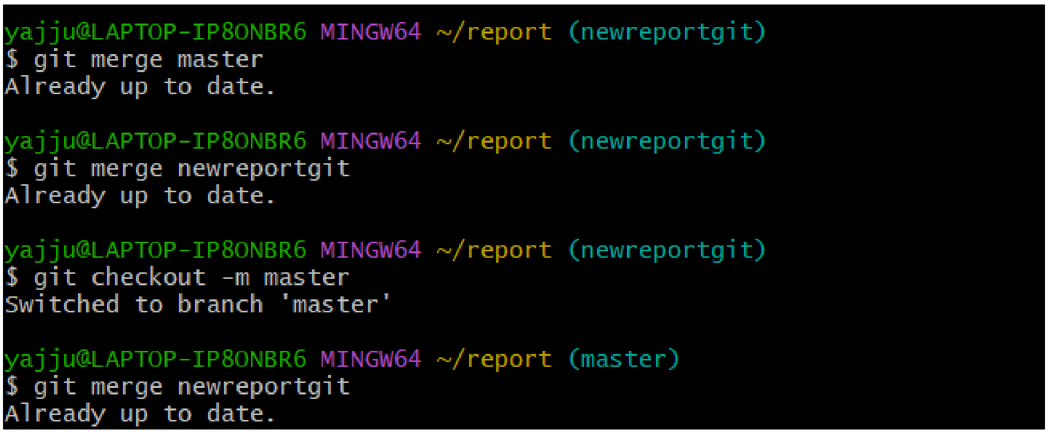
Using the **git branch -m oldbranchname newbranchname** we can rename the existting branch name with the new name and below figure spefies the example.



```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (newreport)
$ git branch -m newreport newreportgit

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (newreportgit)
$ git branch
  master
* newreportgit
```

GIT MERGING

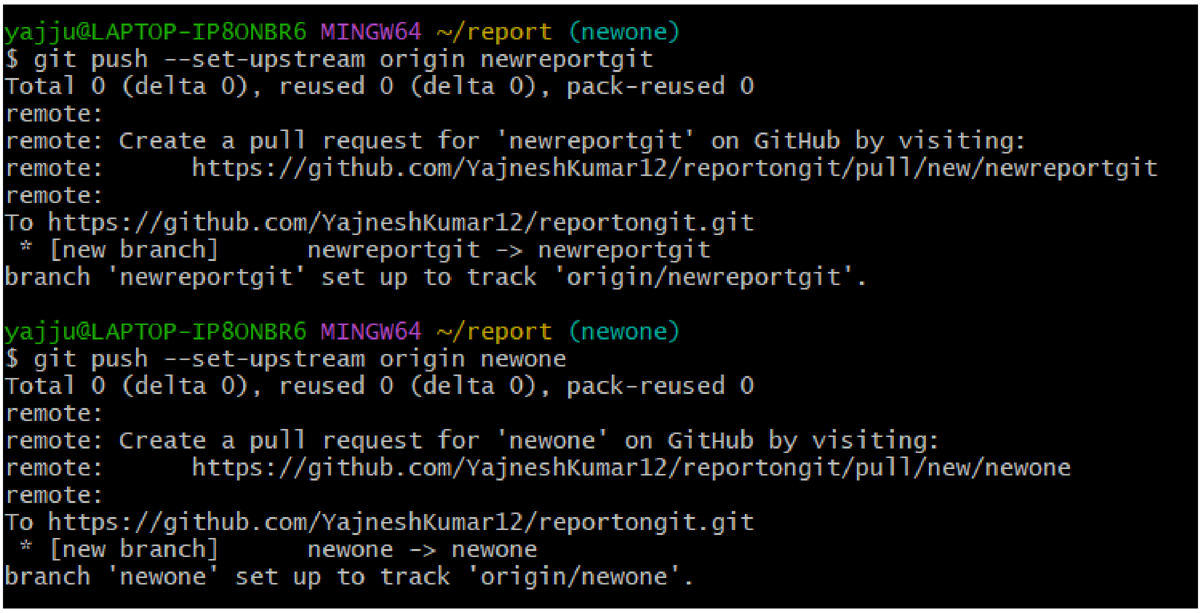
Using the **git merge branchname** we can merge the branch with the existing files present in the old branch.



```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (newreportgit)
$ git merge master
Already up to date.

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (newreportgit)
$ git merge newreportgit
Already up to date.

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (newreportgit)
$ git checkout -m master
Switched to branch 'master'

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git merge newreportgit
Already up to date.
```

GIT MERGE COMMAND TO MERGE IN REMOTE REPO

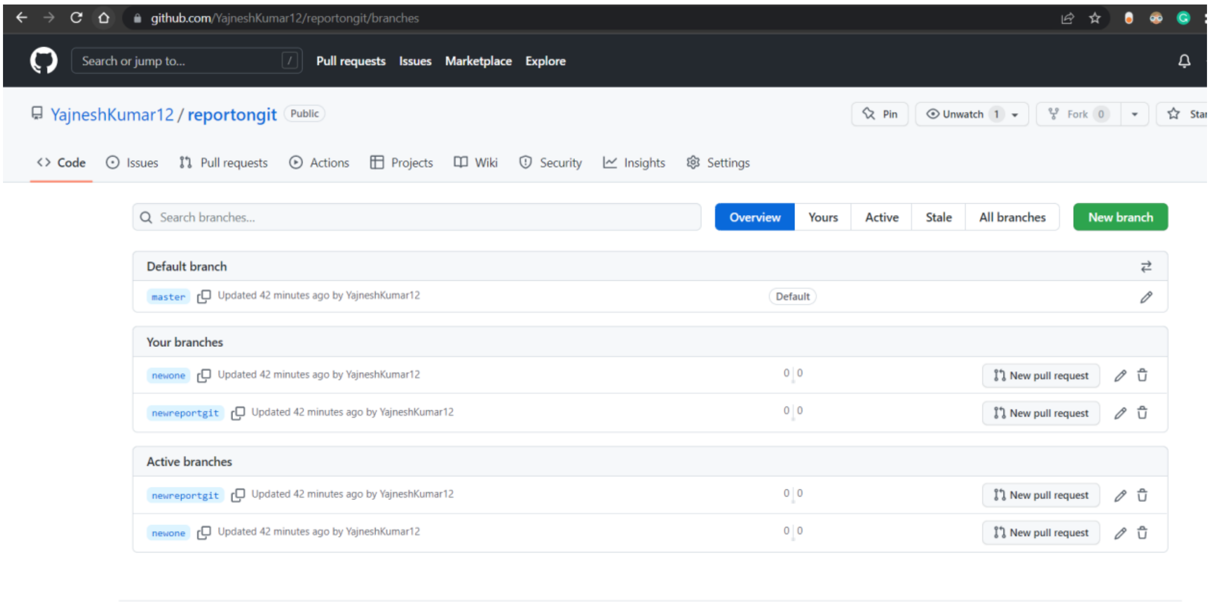
Using the **git push –set upstream origin branchname** we can merge the branch from the one branch to another and it’s file contents exists in the both the branch and below figure specifies the example of the merging branch in the remote repository.



```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (newone)
$ git push --set-upstream origin newreportgit
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'newreportgit' on GitHub by visiting:
remote:      https://github.com/YajneshKumar12/reportongit/pull/new/newreportgit
remote:
To https://github.com/YajneshKumar12/reportongit.git
 * [new branch]      newreportgit -> newreportgit
branch 'newreportgit' set up to track 'origin/newreportgit'.

yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (newone)
$ git push --set-upstream origin newone
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'newone' on GitHub by visiting:
remote:      https://github.com/YajneshKumar12/reportongit/pull/new/newone
remote:
To https://github.com/YajneshKumar12/reportongit.git
 * [new branch]      newone -> newone
branch 'newone' set up to track 'origin/newone'.
```

AFTER MERGING

After merging the branch in the remote repository you can see the below figure where the merged branch and branches which are created can be seen in the below figure.



Default branch

master Updated 42 minutes ago by YajneshKumar12 Default

Your branches

newone Updated 42 minutes ago by YajneshKumar12 0 | 0 New pull request

newreportgit Updated 42 minutes ago by YajneshKumar12 0 | 0 New pull request

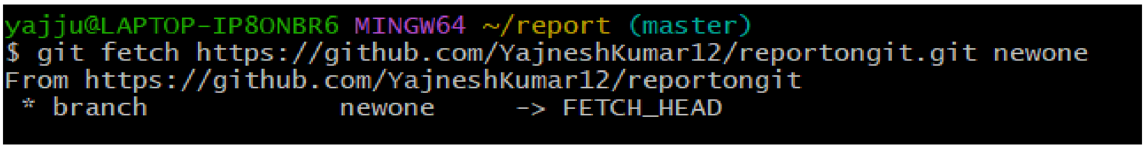
Active branches

newreportgit Updated 42 minutes ago by YajneshKumar12 0 | 0 New pull request

newone Updated 42 minutes ago by YajneshKumar12 0 | 0 New pull request

GIT FETCH

Using the **git fetch HTTPS** used to download the contents from the remote repository and below figure specifies it.

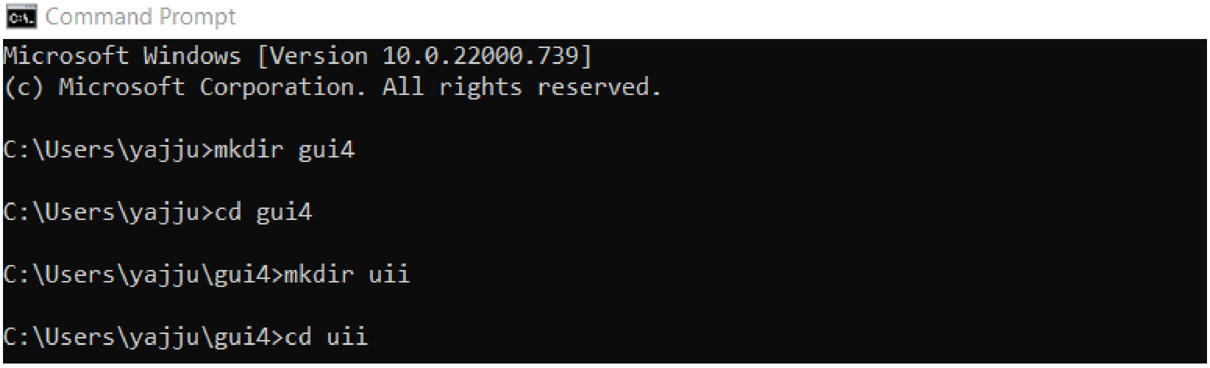


```
yajju@LAPTOP-IP8ONBR6 MINGW64 ~/report (master)
$ git fetch https://github.com/YajneshKumar12/reportongit.git newone
From https://github.com/YajneshKumar12/reportongit
 * branch            newone     -> FETCH_HEAD
```

GIT COMMANDS IN THE CLI

MKDIR AND CD COMMAND

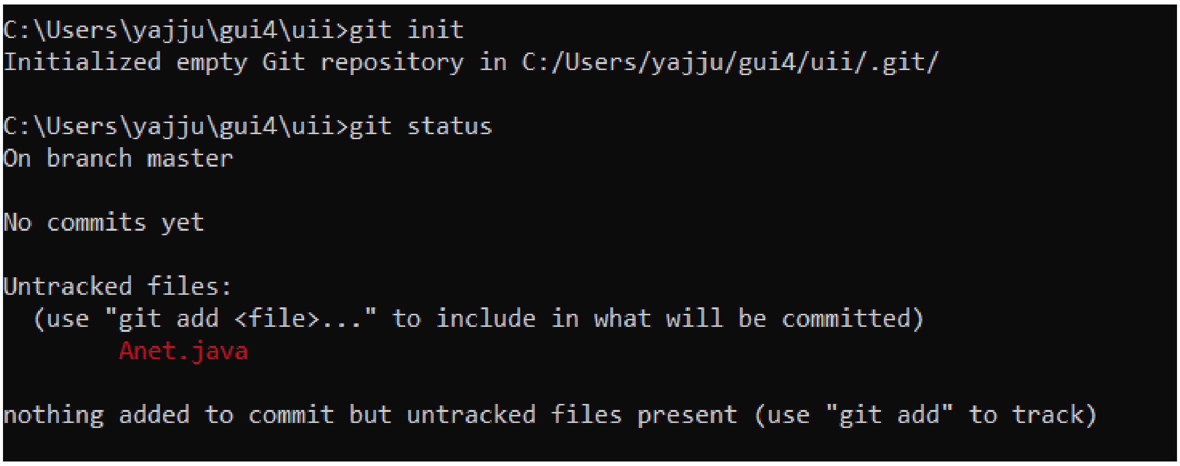
Here created the directory called gui4 and uii using the **mkdir** and **cd** command is used to change the directory to the certain directory we want.



```
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yajju>mkdir gui4

C:\Users\yajju>cd gui4

C:\Users\yajju\gui4>mkdir uii

C:\Users\yajju\gui4>cd uii
```

GIT INIT

Using **git init** command we can create the new git reposiroty in local repository that is .git file will be generated in the certain path we want and it is used to initialize the existing repository or empty repository. Here the **git init** creates new git repository and we can check on the Status of the local files where are the files are present and it tells the git that we want to add particular file in the next commit

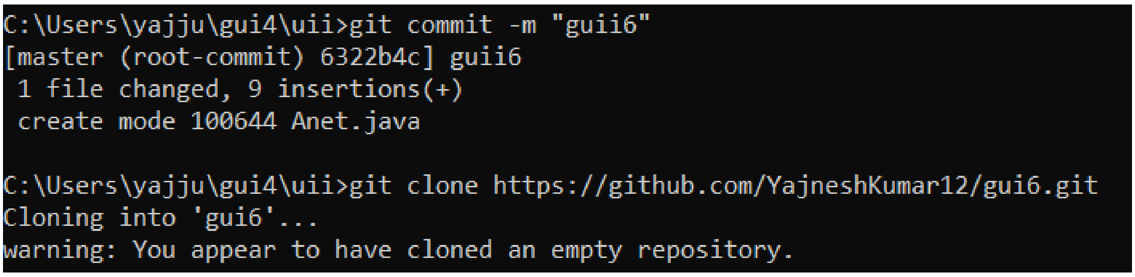


```
C:\Users\yajju\gui4\uii>git init
Initialized empty Git repository in C:/Users/yajju/gui4/uii/.git/

C:\Users\yajju\gui4\uii>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Anet.java

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

GIT COMMIT AND CLONE

Here committed the files with the message guii6 and also cloned it. Using the **git commit -m "message"** command we can see the changes made in the project like adding files etc. It can be used to keep track of our progress and changes as we work with the message written during the commit.

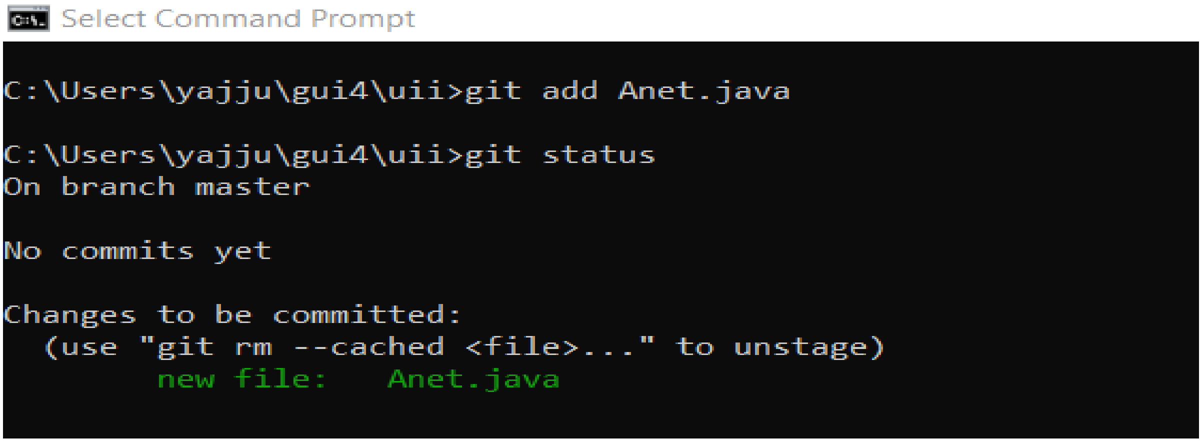
Using **git clone HTTPS** command used to copy an existing Git repository into a new local directory. The Git clone command will create a new local directory for the repository, copy all the contents of the specified repository, create the remote tracked branches, and checkout an initial branch locally.



```
C:\Users\yajju\gui4\uii>git commit -m "guii6"
[master (root-commit) 6322b4c] guii6
 1 file changed, 9 insertions(+)
 create mode 100644 Anet.java

C:\Users\yajju\gui4\uii>git clone https://github.com/YajneshKumar12/gui6.git
Cloning into 'gui6'...
warning: You appear to have cloned an empty repository.
```

GIT ADD

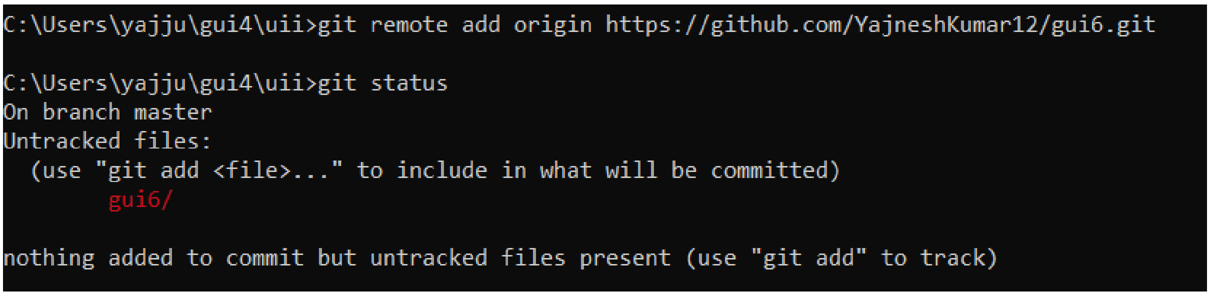
Using the **git add files** I have added the files to the staging area. Here added the files to the directory



```
C:\Users\yajju\gui4\uii>git add Anet.java

C:\Users\yajju\gui4\uii>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Anet.java
```

GIT REMOTE

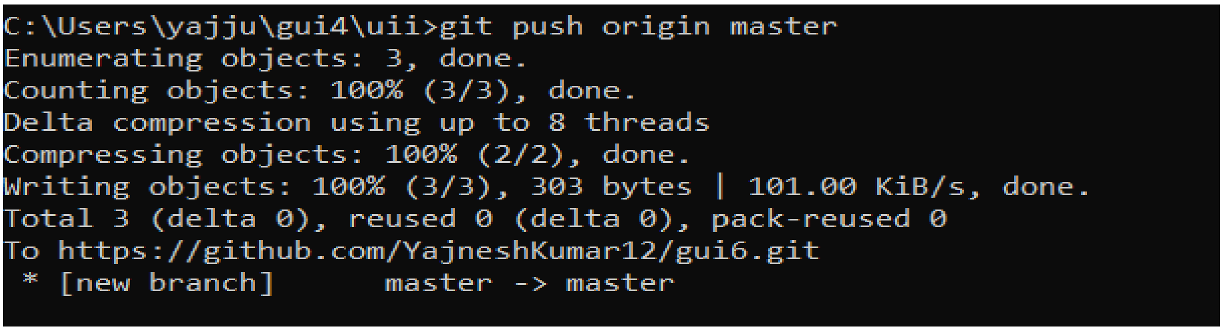
Using the **git remote add origin HTTPS** command we can add, view and delete the connection to the remote repository using the HTTPS or SSH. In this screenshot we can see that we have remotely added the origin of the github with the link HTTPS.



```
C:\Users\yajju\gui4\uii>git remote add origin https://github.com/YajneshKumar12/gui6.git

C:\Users\yajju\gui4\uii>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        gui6/

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

GIT PUSH

Using the **git push origin master** we can upload the files to the remote repository from the local repository. Here uploaded the files from the local repository to the remote repository



```
C:\Users\yajju\gui4\uii>git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 101.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/YajneshKumar12/gui6.git
 * [new branch]      master -> master
```