

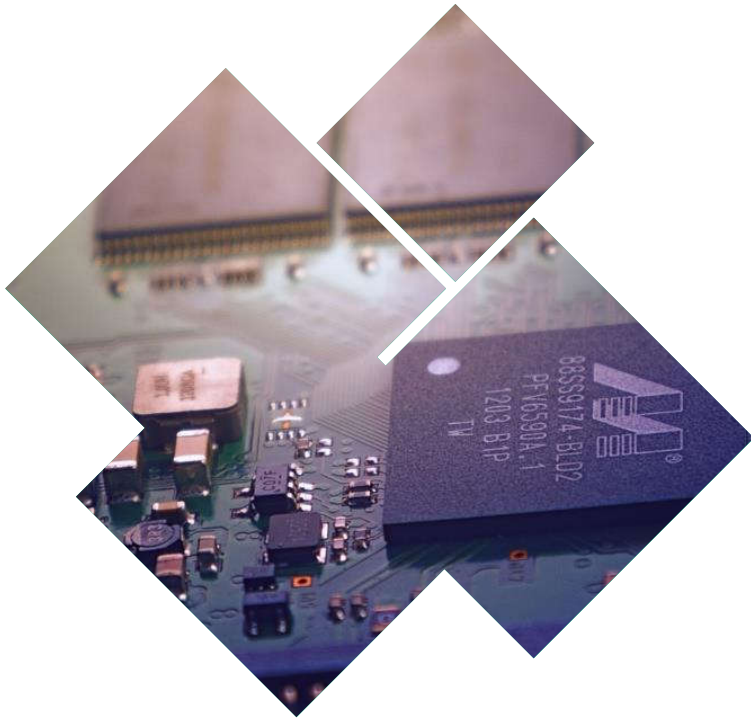
An introduction to GLT

112

PROPOSED BY

OUAMANI FADOUA

Outline



1

Common problems

2

What Git is?

3

How to use Git?

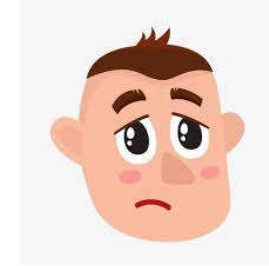
4

Git and GitHub

2



Common problems



- ✓ Your code has an error.
 - ✓ The fix seems obvious but unfortunately it leads to new problems.
 - ✓ Step by step you go along a chain of small code changes until nothing works anymore. Now you're lost. :(
 - ✓ There are new features that you would like to add but the implementation will be very messy if you integrate it into the current code.
 - ✓ You come up with a better architecture and start implementing it.
 - ✓ First step: Create a copy of the old code as a reference.
 - ✓ Then another copy of some intermediate solution. Then another copy
 - ✓ It's Monday. C++ does not compile on Mondays. :D
 - ✓ This is a mystery, since yesterday evening everything was still fine. :D
 - ✓ Maybe you changed one last thing before going to bed, but there's no way to recall what that was.



Common problems

- ✓ Handling of Many versions of the same document.
- ✓ First day as a developer and you have to clone a project
- ✓ Software development team working on a same project
- ✓ Keep trace of work history and recover deleted files



Common problems

- ✓ When the file was modified?
- ✓ What changed?
- ✓ Why it was modified?
- ✓ Who did the change?



What Git is?

- ✓ Open Source **Distributed Version Control System**.
 - ✓ **Control System:** This basically means that Git is a content tracker.
 - ✓ It can be used to store content
 - ✓ it is mostly used to store code due to the other features it provides.
 - ✓ **Version Control System:**
 - ✓ The code which is stored in Git keeps changing as more code is added.
 - ✓ Also, many developers can add code in parallel.
 - ✓ So Version Control System helps in handling this by maintaining a history of what changes have happened. Also, Git provides features like **branches** and **merges**.
 - ✓ **Distributed Version Control System:**
 - ✓ Git has a remote repository which is stored in a server (remote repository) and a local repository which is stored in the computer of each developer.
 - ✓ This means that the code is not just stored in a central server, but the full copy of the code is present in all the developers' computers.



Why a version control system like Git is needed to work on your project?



- ✓ You will be multiple classmates working in parallel to develop a program.
 - ✓ So a version control system like Git is needed to ensure there are no code conflicts between you and your classmates working on the same project.
- ✓ Additionally, the requirements in such projects change often.
 - ✓ So a version control system allows you to revert and go back to an older version of the code.
- ➔ Version control is an essential tool for ALL programmers.
- ➔ No matter if you end up programming in C++, Python or Javascript: In any professional environment, you will use version control.



How to use git?

- ✓ This link has details on how to install Git in multiple operating systems:

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

- ✓ Verify if Git is installed by using the following command in the command prompt:

```
git --version
```

```
ASUS@DESKTOP-3EHVP19 MINGW64 ~  
$ git --version  
git version 2.30.1.windows.1
```

- ✓ Create your local Git repository

- ✓ In your computer, create a folder for your project. Let's call the project folder "simple-git-demo".

- ✓ Go into your project folder and add a local Git repository to the project using the following commands:

```
cd simple-git-demo  
git init
```

```
ASUS@DESKTOP-3EHVP19 MINGW64 ~  
$ cd d:\simple-git-demo  
  
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo  
$ git init  
Initialized empty Git repository in D:/simple-git-demo/.git/  
  
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)  
$
```

- ➔ The git init command adds a local Git repository

- ➔ A hidden .git repository



How to use git?

- ✓ Before you actually use git, it is a good idea to set up some **global configuration**.
- ✓ It will be used to identify you as the author of code changes:
 - ✓ `git config --global user.name "your name"`
 - ✓ `git config --global user.email "first.last@inst.tn"`

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git config --global user.name "Fadoua"

ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git config --global user.email "fadoua.ouamani@ensi-uma.tn"

ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$
```



How to use git?

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git status
On branch master

No commits yet

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

ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ |
```

Staging and Committing the code:

- ✓ Committing is the process in which the code is added to the **local repository**.
- ✓ Before committing the code, it has to be in the **staging area**.
- ✓ The staging area is there to keep track of all the files which are to be committed.
- ✓ Any file which is not added to the staging area will not be committed.
- ➔ This gives the developer control over which files need to be committed.



How to use git?

Let's Add some Small Code now

- ✓ Create a file called demo.txt in the project folder and add the following text into it:
Initial Content

Staging

- ✓ Use the following command for staging the file:

git add demo.txt

- ✓ To add multiple files

git add file1 file2 file3

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git add demo.txt
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ |
```

- ✓ To add all the files inside your project folder to the staging area

git add .

Committing

- ✓ Use the following command to commit the file:

git commit -m "Initial Commit"

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git commit -m "Initial Commit"
[master (root-commit) 482ce8e] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 demo.txt
```

- ➔ "Initial Commit" : relevant commit message to indicate what code changes were done in that particular commit.



How to use git?

Git status

- ✓ Now modify the demo.txt file and add the following snippet:
Initial Content Adding more Content
- ✓ Use git status to find out information regarding what files are modified and what files are there in the staging area

git status

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ 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:   demo.txt

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

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git add demo.txt

ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   demo.txt
```

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git commit -m "demo.txt file is modified"
[master 3df7b82] demo.txt file is modified
1 file changed, 1 insertion(+), 1 deletion(-)

ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git status
On branch master
nothing to commit, working tree clean
```



How to use git?

Git log

- ✓ Use git log to print out all the commits which have been done up until now.
git log
- ✓ The log shows the author of each commit, the date of the commit, and the commit message.

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git log
commit 3df7b82b1b23896e38d2895a43eacd61b8988269 (HEAD -> master)
Author: Fadoua <fadoua.ouamani@ensi-uma.tn>
Date: Thu Feb 18 22:39:48 2021 +0100

    demo.txt file is modified

commit 482ce8ef18daa485a9db3f9fdc49b01278f02201
Author: Fadoua <fadoua.ouamani@ensi-uma.tn>
Date: Thu Feb 18 22:29:02 2021 +0100

    Initial Commit
```



How to use git?

Branches

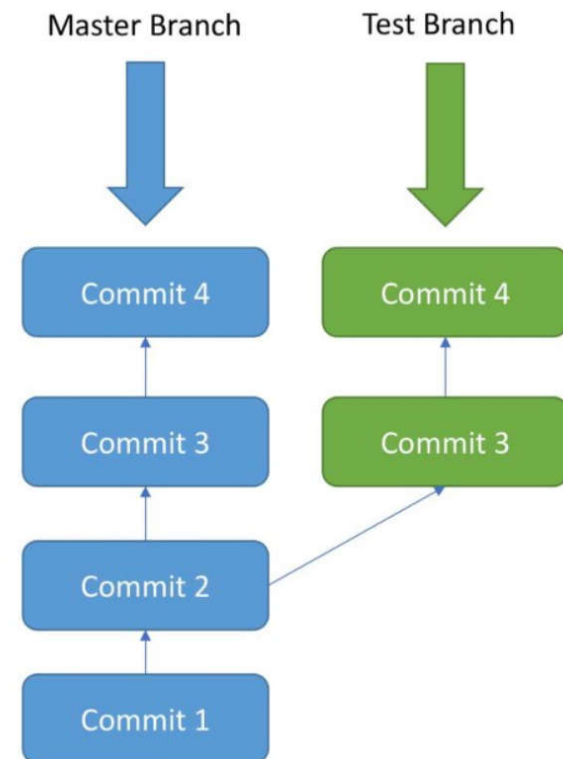
What is a branch?

✓ A branch is nothing but a pointer to the latest commit in the Git repository.

➔ So currently our master branch is a pointer to the second commit “demo.txt file is modified”.

Why are multiple branches needed?

➔ Multiple branches are needed to support multiple parallel developments.





How to use git?

Create a New Branch in Local

- ✓ Create a new branch called **test** using the following command:

git branch test

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git branch test

ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ |
```

- ➔ This command creates the test branch.
- ➔ We are still in the context of the master branch.
- ➔ In order to switch to the test branch. use the following command:

git checkout test

- ➔ Now we are in the test branch.
- ➔ List all the branches in local

git branch

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git checkout test
Switched to branch 'test'

ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (test)
$ |
```

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (test)
$ git branch
master
* test
```



How to use git?

Do Some Commits in the New Branch

- ✓ Modify demo.txt by adding the following snippet:
Initial Content Adding more Content Adding some Content from test Branch
- ✓ Now stage and commit using the following commands:
`git add demo.txt`
`git commit -m "Test Branch Commit"`
- ➔ This commit was done in the Test Branch, and now Test Branch is ahead of Master Branch by 1 commit
- ➔ as the test branch also includes the 2 commits from the master branch.

You can verify the commit history in Test Branch using:

`git log`

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (test)
$ git add demo.txt

ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (test)
$ git commit -m "test branch commit"
[test 0e0ff9c] test branch commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (test)
$ git log
commit 0e0ff9cbaf16d95e5f497e78d362e1fb178e6354 (HEAD -> test)
Author: Fadoua <fadoua.ouamani@ensi-uma.tn>
```




How to use git?

Merging

- ✓ Currently, Test Branch is ahead of the Master by 1 commit.
- ✓ We want all the code in the Test Branch to be brought back to the Master Branch.
- ➔ This is where git merge is very useful.
- ✓ In order to merge the code from the test branch into the master branch, follow these steps:

1) First go back to the master branch:

git checkout master

2) Then run the merge command:

git merge test

```
ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (test)
$ git checkout master
Switched to branch 'master'

ASUS@DESKTOP-3EHVP19 MINGW64 /d/simple-git-demo (master)
$ git merge test
Updating 3df7b82..0e0ff9c
Fast-forward
 demo.txt | 2 + -
 1 file changed, 1 insertion(+), 1 deletion(-)
```

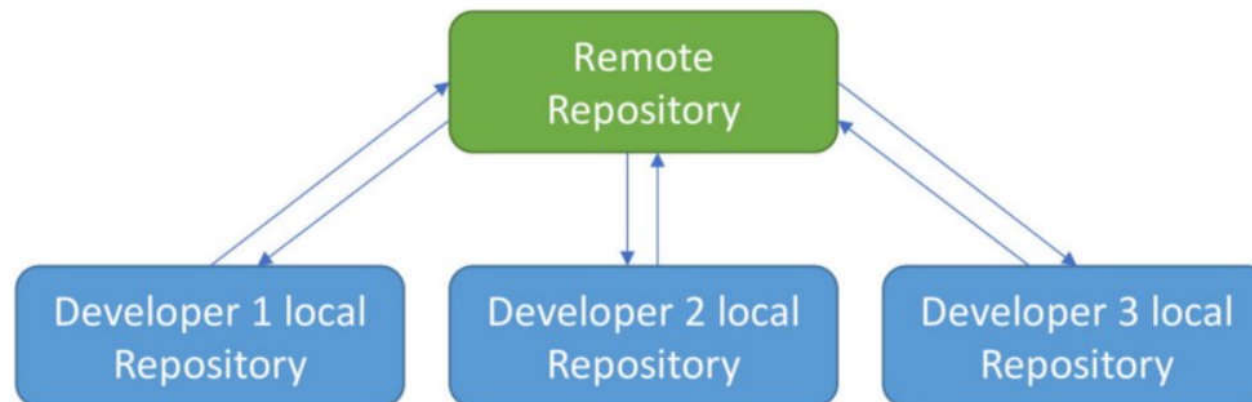
- ➔ In real projects, there will be conflicts when a merge is being done. Resolving the conflict is something which comes with experience, so as you work more with Git you will be able to get the hang of resolving conflicts.



How to use git?

The Remote Git Repository

- ✓ Until now, we have been working only in the local repository.
- ✓ Each developer will work in their local repository but eventually, they will **push** the code into a **remote repository**.
- ✓ Once the code is in the remote repository, other developers can see and modify that code.



Showing Remote and Local Repositories



Git and GitHub

- ✓ Here we will be using GitHub for the remote repository.
- ✓ Go to <https://github.com/> and create an account.
- ✓ After registering in the GitHub homepage, click on **Start a Project** to create a new Git repository. Give the repository a name “git-blog-demo” and click “Create Repository”
- ✓ Give the name as This will create a remote repository in GitHub, and when you open the repository, a page like the image below will open up:

The repository URL

<https://github.com/FadouaOuamani/git-blog-demo.git>

Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

[https://github.com/FadouaOuamani/git](https://github.com/FadouaOuamani/git-blog-demo.git)

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repo:

- ✓ In order to point your local repository to the remote repository:
`git remote add origin [repository url]`



Git and GitHub

Git Push

- ✓ In order to push all the code from the local repository into the remote repository, use the following command:
`git push -u origin master`
- ➔ This pushes the code from the master branch in the local repository to the master branch in the remote repository.



Git and GitHub

Additional Commands

Git Pull

- ✓ git pull is used to pull the latest changes from the remote repository into the local repository.
- ✓ The remote repository code is updated continuously by various developers, hence git pull is necessary:

`git pull origin master`

Git Clone

- ✓ git clone is used to clone an existing remote repository into your computer.
- `git clone [repository url]`