

{Secure Systems and Programmable Networks: Tools for the Future}

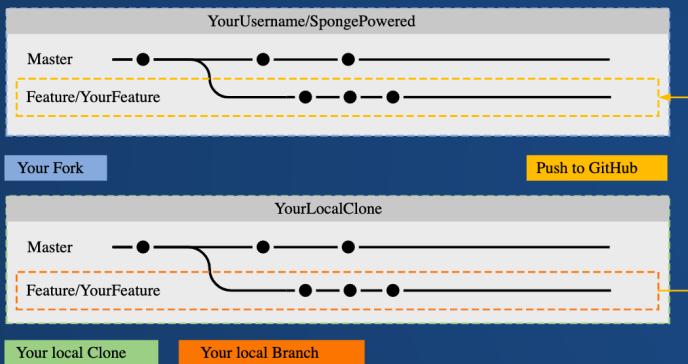


Università degli Studi di Catania
Dipartimento di Matematica e Informatica
2025





GIT



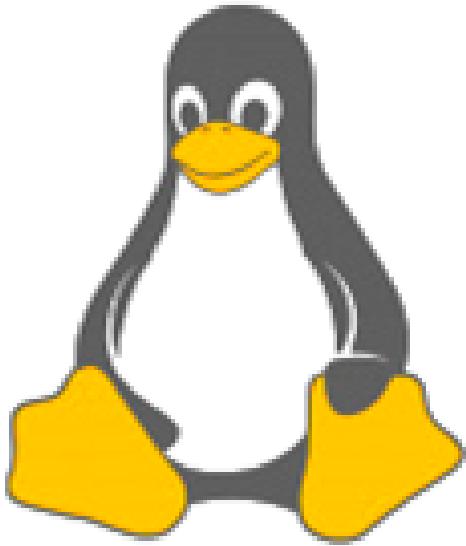
Git

A way to manage your project



2005





Linux™



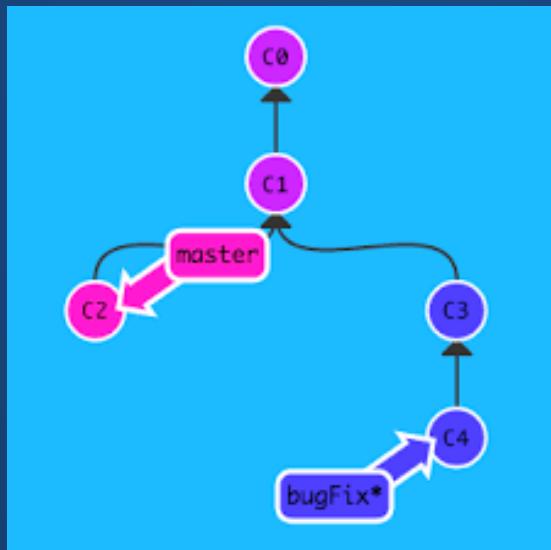
<https://ohmygit.org/>



<https://github.com/firstcontributions/first-contributions>



Learn to make your first open source contribution on GitHub in 5 minutes



<https://learngitbranching.js.org/>

```
1 $ git config
2
3 $ git add
4 $ git rm
5 $ git mv
6 $ git commit -m 'desc commit'
7
8 $ git checkout -b branch_name # craete a new branch
9 $ git checkout branch_name
10 $ git merge REMOTE BRANCH # ex. git merge origin master
11
12 $ git reset
13 $ git revert
14
15 $ git status
16 $ git log
17 $ git diff
18
19 $ git init
20 $ git clone
21 $ git remote → git remote add, git remote -v, git remote rm
22 $ git fetch
23
24 $ git pull
25 $ git push
```

How to install GIT



Linux (or WSL), via package manager:

```
$ apt install git
```



Mac via Homebrew or MacPort:

```
$ brew install git
```

```
$ port install git
```



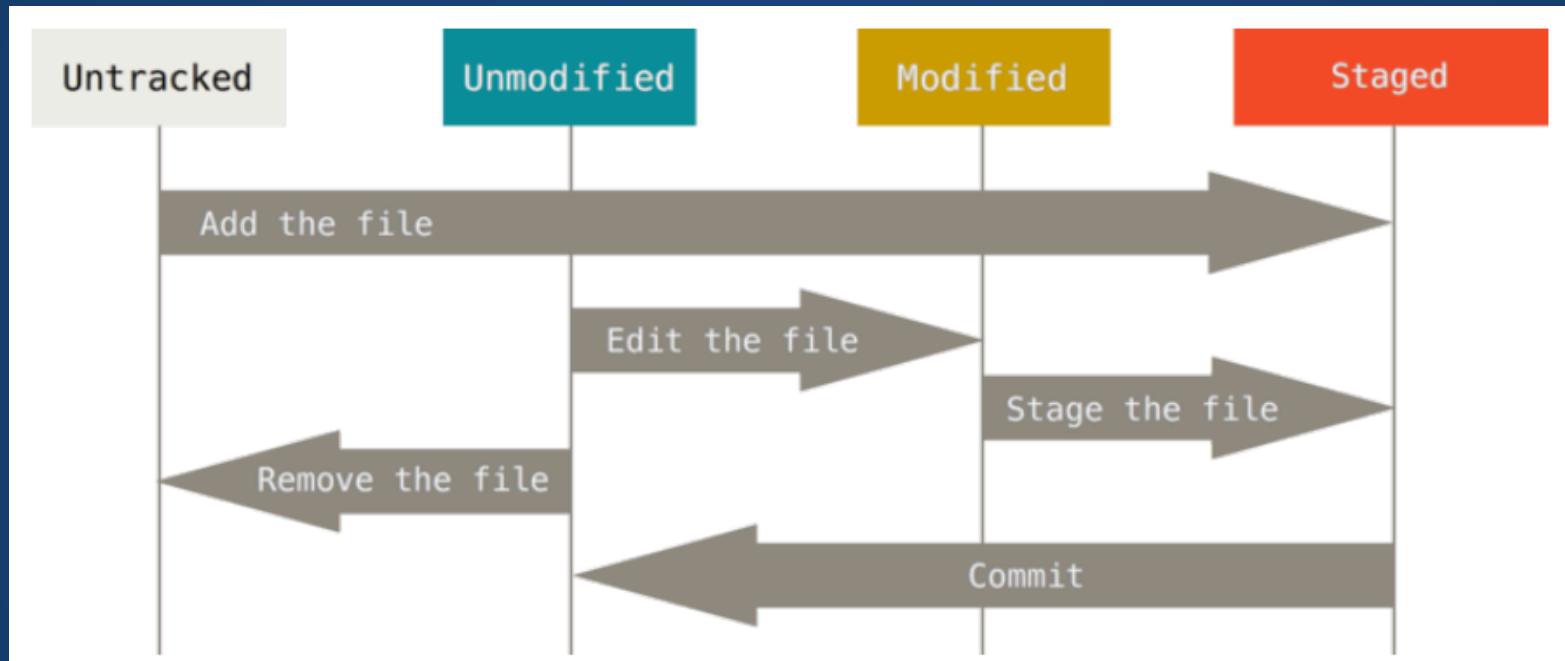
Windows

<https://gitforwindows.org/>

```
1  
2 $ git config --global user.name "Stefano Borzì"  
3 $ git config --global user.email "stefano@example.com"  
4 $ git config --global core.editor nano  
5  
6  
7 $ git config --list  
8 user.name=Stefano Borzì  
9 user.email=stefano@example.com  
10 core.editor=nano  
11 ...  
12  
13  
14 $ git config user.name  
15 Stefano Borzì  
16
```

```
1 $ git init
2
3 $ git add
4 $ git rm
5 $ git mv
6
7 $ git status
8 $ git diff
9
10 $ git commit -m 'desc commit'
11
12 $ git log
```

git status



The diagram illustrates the relationship between a git commit history and its corresponding changelog. On the left, a dark-themed terminal window shows a commit graph with colored lines and arrows. A blue line labeled 'feat:' connects a commit to a merge commit 'M'. A red line labeled 'fix:' connects another commit to the same merge commit 'M'. A green line labeled 'improvement:' connects a third commit to the merge commit 'M'. The merge commit 'M' has a blue arrow pointing to it from the text 'v3.4.4'. Another blue arrow points to it from the text 'v3.4.3'. On the right, a light-themed browser window displays a 'Changelog' page. It lists three entries:

- v3.4.4 12/06/2019**
 - feat:* Image gallery upload
 - fix:* Fixed MacOS 10.13.1. installation bugs
 - improvement:* Added links to management icons
- v3.4.3 10/06/2019**
 - improvement:* editing and deleting photos on mobile
 - fix:* Fixed broken links on footer
- v3.4.2 10/06/2019**

Conventional Commits

```
1 fix: for a fix          ex. fix(main): windows build
2 feat: implementing a new feature ex. feat(home): add footer
3 docs: for documentation.   ex. doc(contribution): add contribution guidelines
4 refactor: for refactoring purposes ex. refactor(tests): replace all "pippo" variables
5 test: adding unit-tests, e2e-etest  ex. test(lessons): add unit-tests for lesson
6 chore: minor improvements    ex. chore(merge): solve conflict
```

git diff

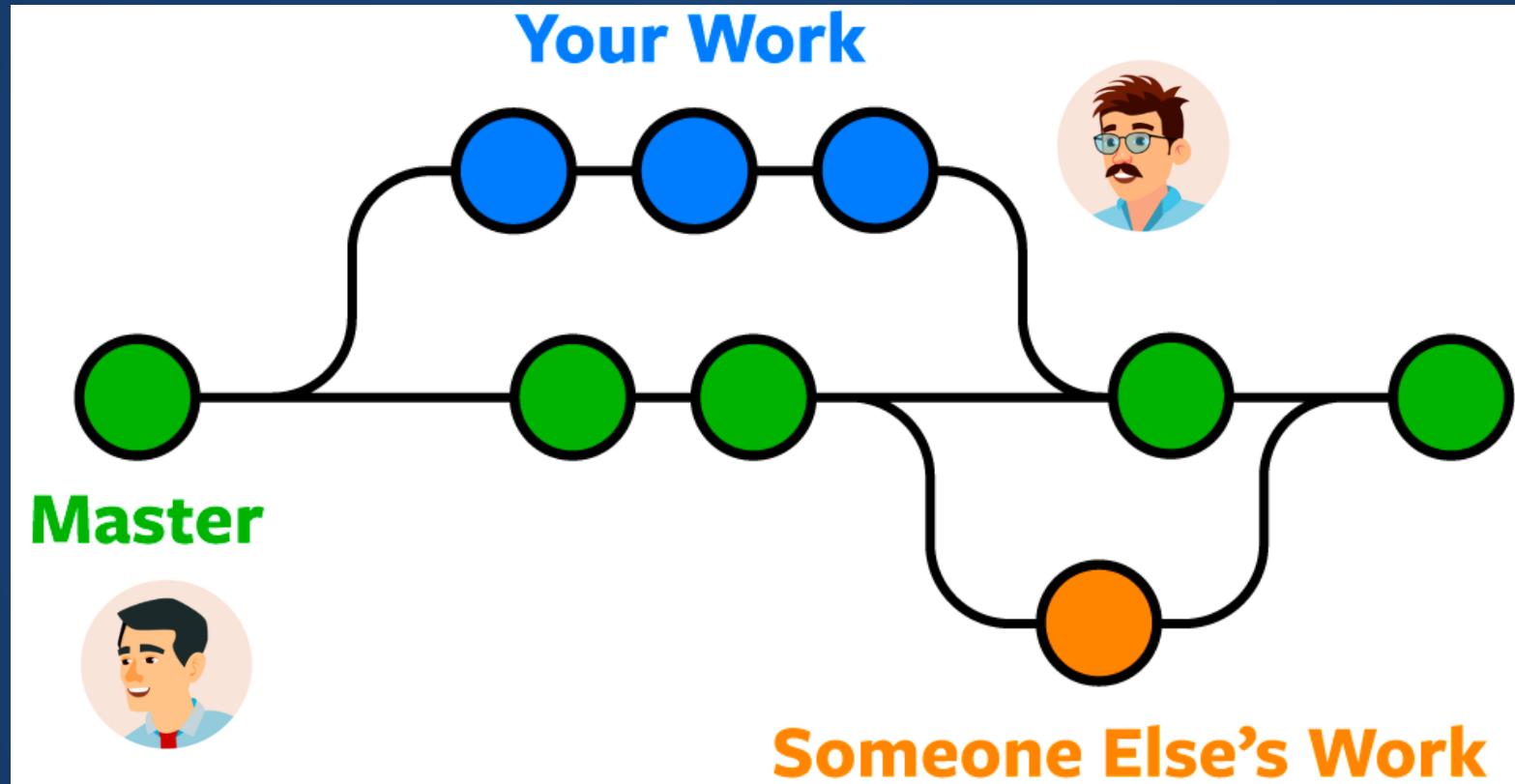
```
6 lines - 2 Removals          Copy all      5 lines + 2 Additions      Copy all

1 This is a test              1 This is not a test
2 This is a simple test       2 This is a simple test
3 Test                         3 Test
4 UNICT                        4 UNICT DMI
5 DMI
6 GitHub
```

checkout, revert, reset

```
1 $ git checkout [COMMIT]  
2  
3 $ git revert [COMMIT]  
4  
5 $ git reset
```

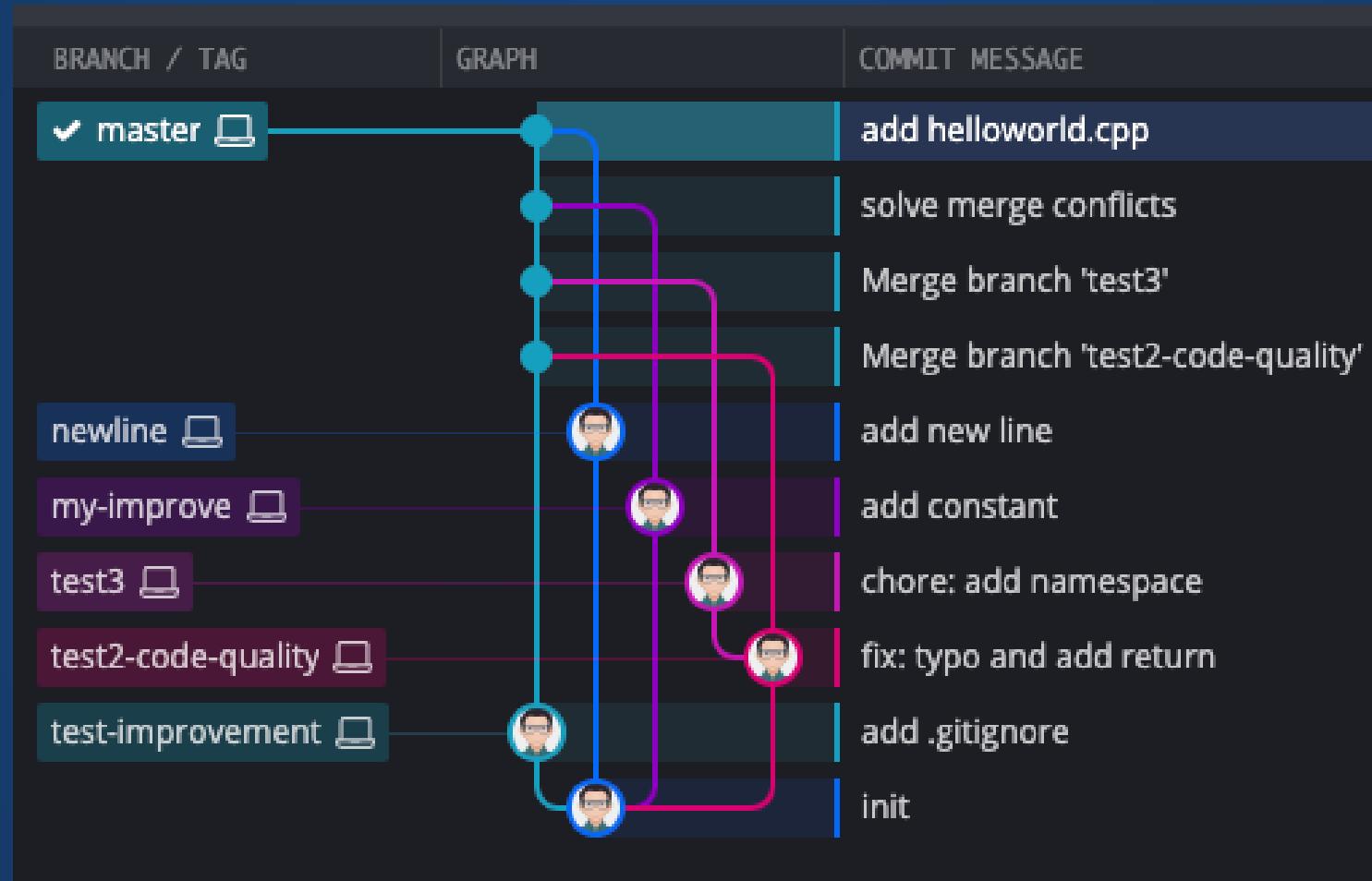
Branches and Merges



Branches and Merges

```
1 # craete a new branch and checkout  
2 $ git checkout -b branch_name  
3  
4 $ git checkout branch_name  
5  
6 $ git merge [REMOTE] BRANCH  
7 # ex. git merge origin master
```

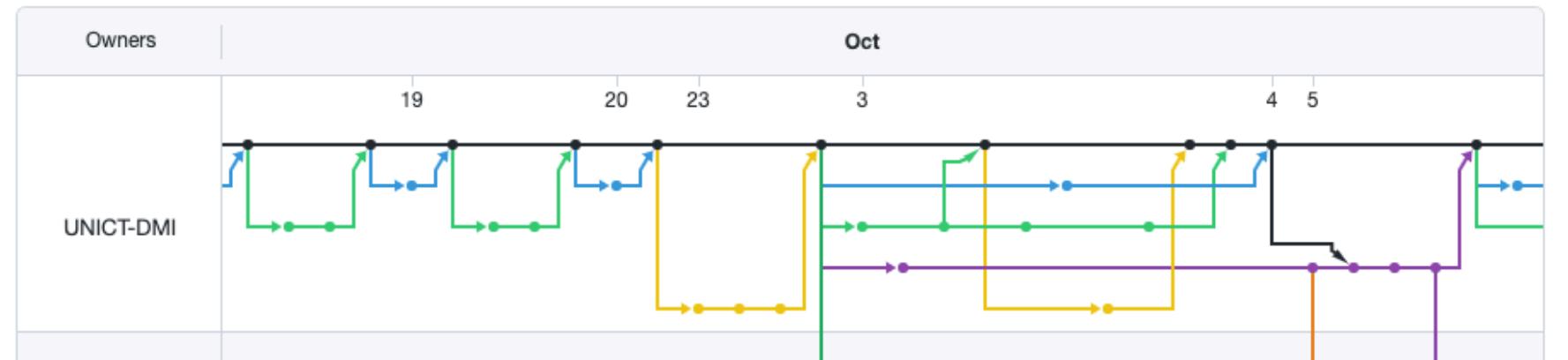
Branches and Merges



DMI Bot - network graph

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



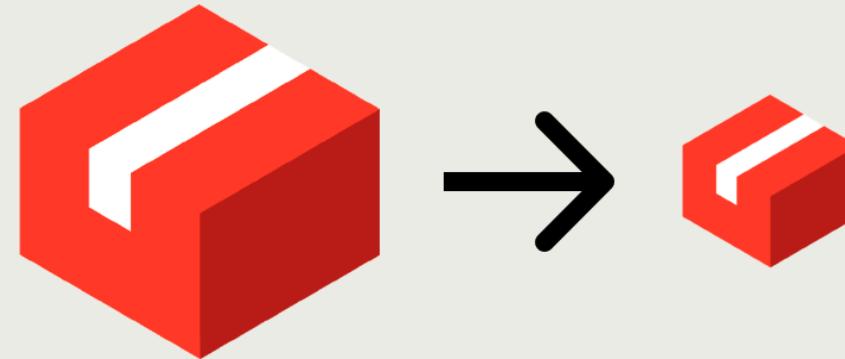
.gitignore

<https://git-scm.com/docs/gitignore>

```
1 # See http://help.github.com/ignore-files/ for more about ignoring files.
2
3 # compiled output
4 /dist
5 /tmp
6 *.js.map
7
8 # dependencies
9 /node_modules
10
11 # IDEs and editors
12 /.idea
13 .project
14 .classpath
15 .c9/
16 *.launch
17 .settings/
18 *.sublime-workspace
19 .vscode/*
20
21 # System Files
22 .DS_Store
23 Thumbs.db
```

GIT LFS

 **Git Large File Storage**



GIT LFS



Git LFS at Light Speed
by Lars Schneider

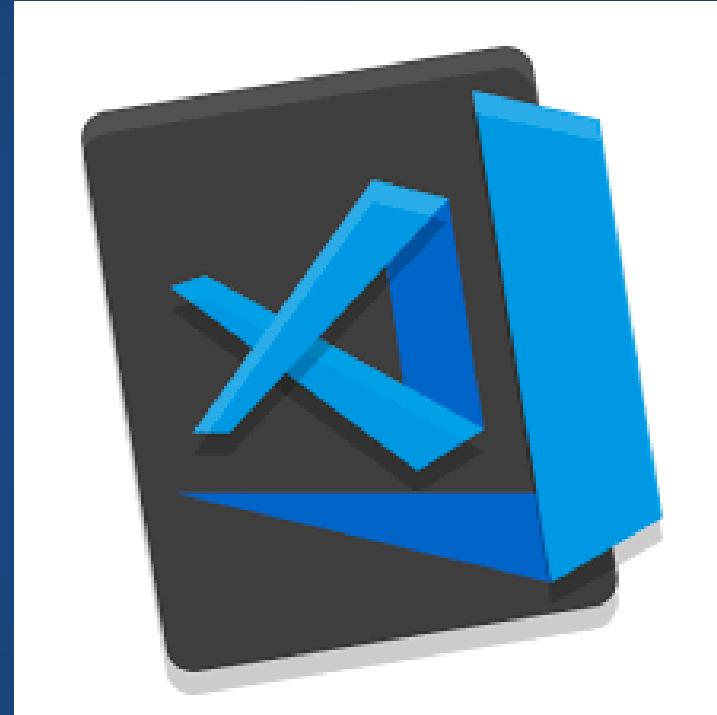
Visual Studio Code plugins

Git related

- Git Lens
- Git History

Generic plugins

- sonarlint
- change-case



Exercise

```
$ wget https://bit.ly/3ucdNHK
```

```
1 /*  
2 Make a BRANCH per each task.  
3  
4 - fix warning and errors  
5 - fix the code  
6 - make N dynamic, let the user choose N  
7 - remove unused  
8 - make some improvements on your own  
9 */
```

all the commands shown so far

```
1 $ git config  
2  
3 $ git init  
4  
5 $ git add  
6 $ git rm  
7 $ git mv  
8 $ git commit -m 'desc commit'  
9  
10 $ git checkout -b branch_name # craete a new branch  
11 $ git checkout branch_name  
12 $ git merge BRANCH # ex. git merge feature-1  
13  
14 $ git reset  
15 $ git revert  
16  
17 $ git status  
18 $ git log  
19 $ git diff
```