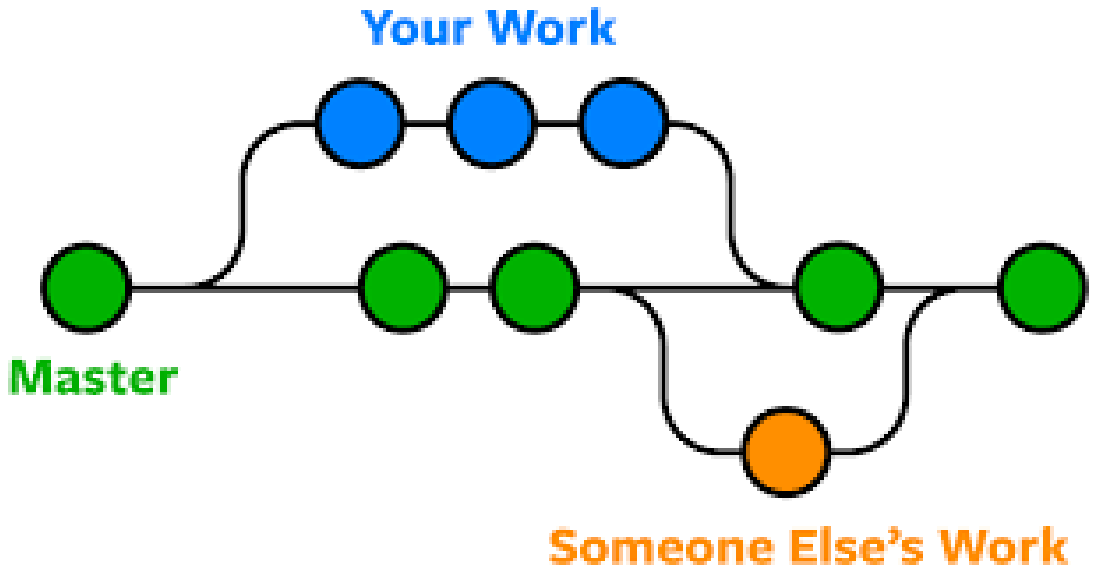
Git Fundamentals
Day 1:
Introduction

Course Overview:

- ✓ Introduction to Git from basic principles
- ✓ You will have the knowledge and understanding to add version control to your projects
- ✓ Get insights of how an effective workflow with Git should look like



Course Contents:

Introduction

- Motivation
- Basics
- Installation
- Bash commads

Workflow

- Adding changes
- Navigate
- Compare
- Ignore

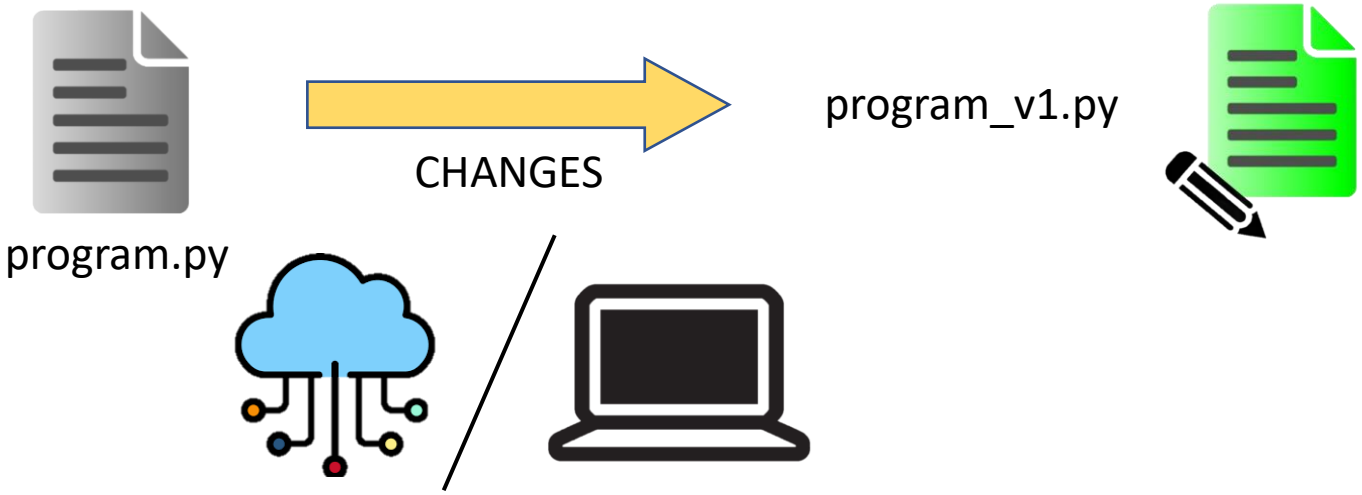
Contributing

- Branches
- Remote Repositories
- GitHub

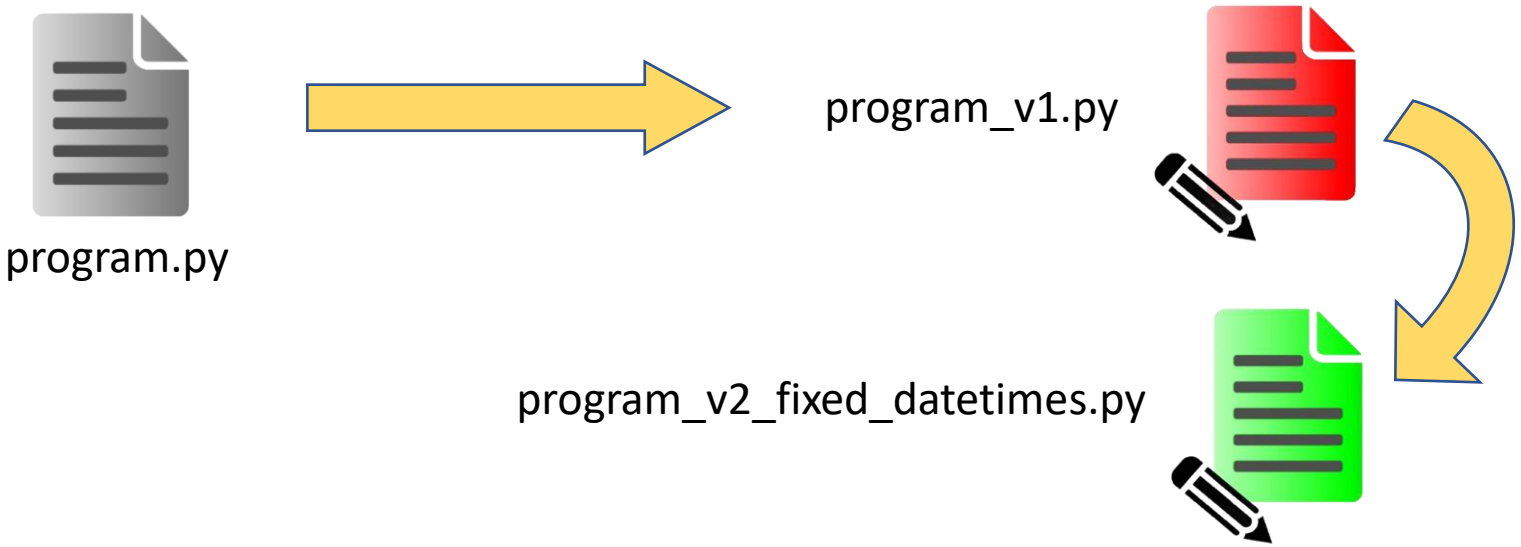
What is Git?

- Version control system (VCS)
- Created by Linus Torvalds for use in the development of Linux kernel
- Often used by programmers to leverage codebases
- Git is not the same as GitHub!

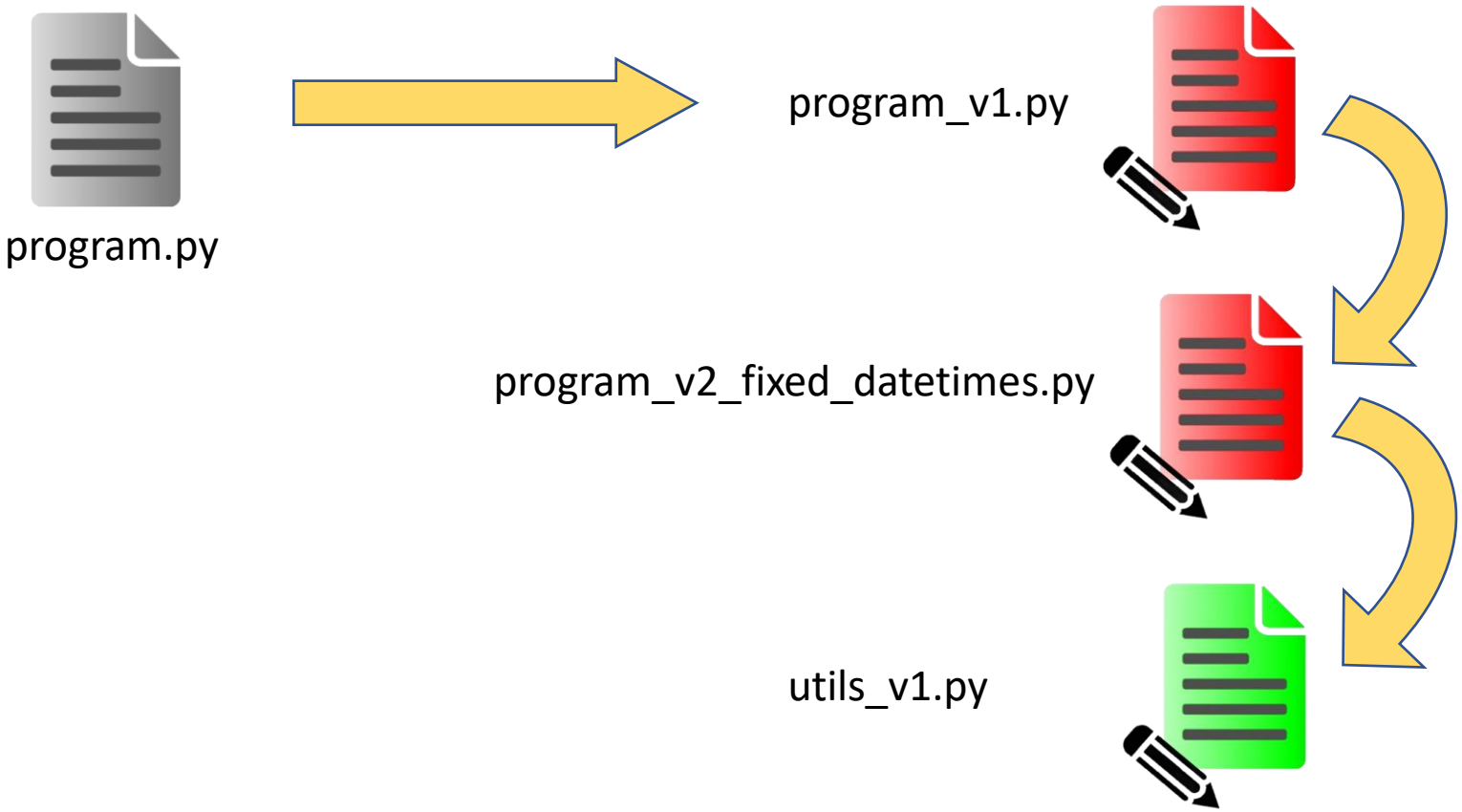
Why version control?



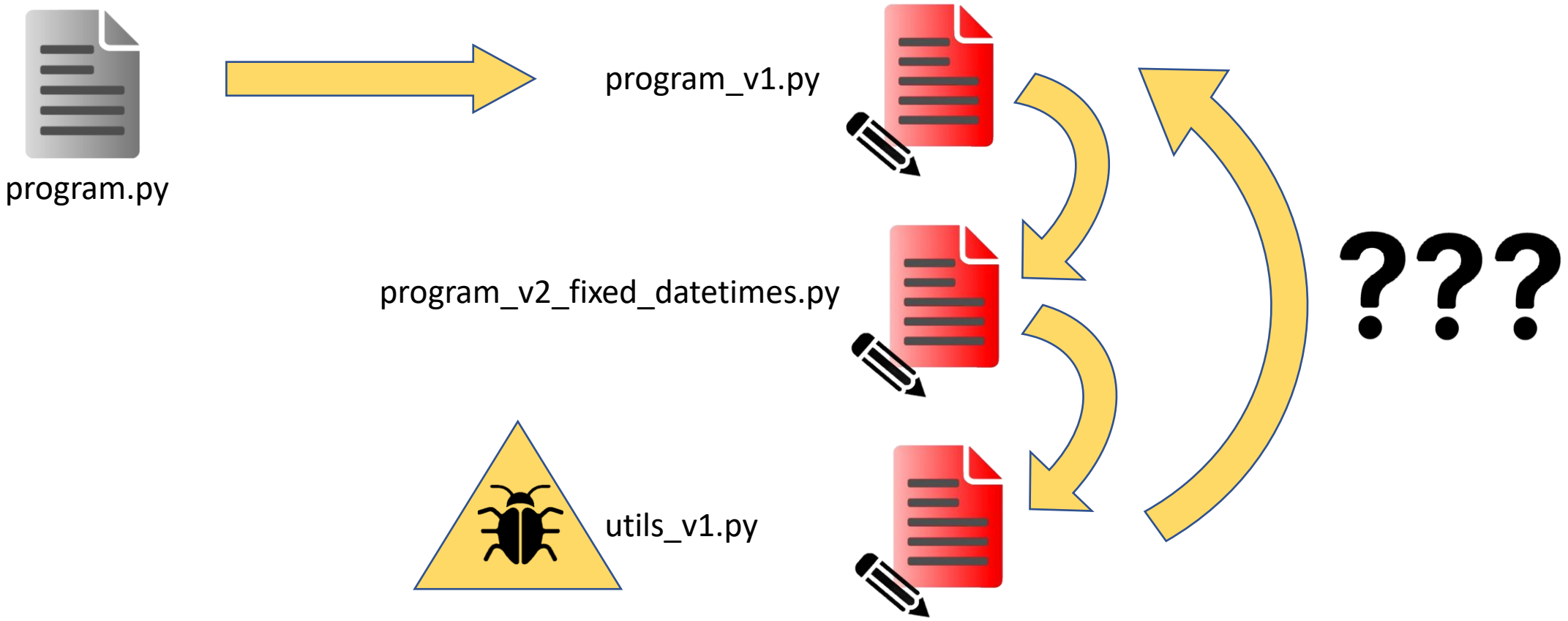
Why version control?



Why version control?

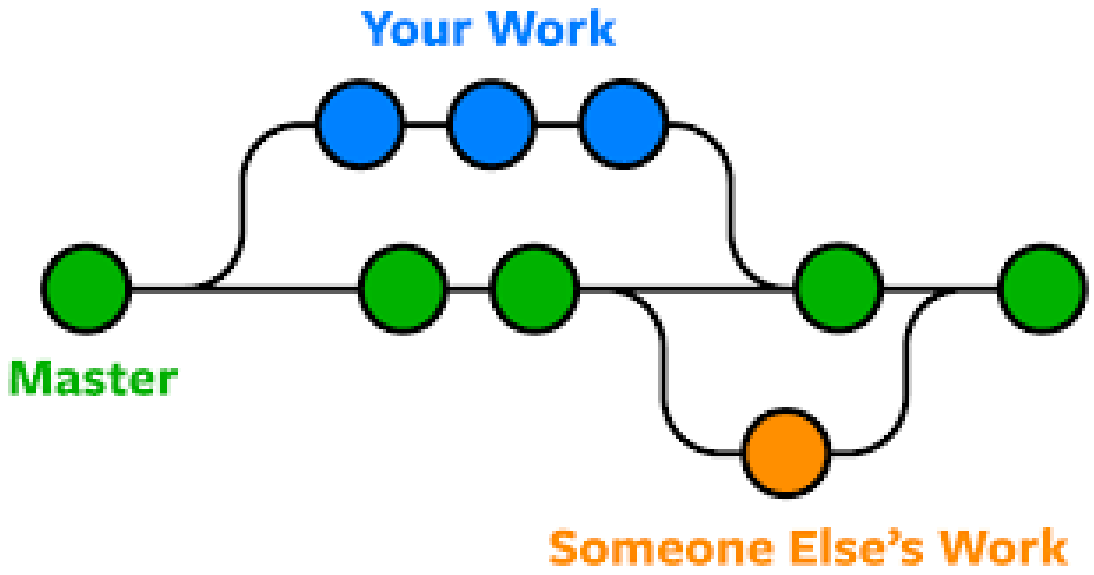


Why version control?



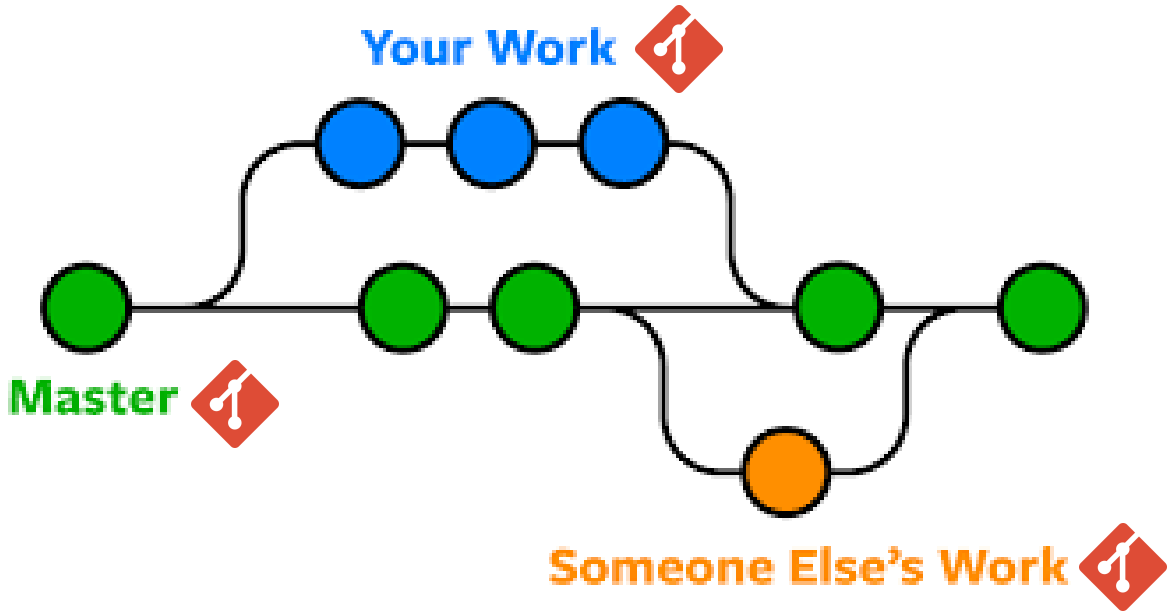
Why version control?

- Traceability
  - What went wrong? When?
- Consistency
  - Which version to production?
- Shareability
  - How to integrate with others?



Why Git?

- Free and open source
- Leading market option
- Lightweight and Fast
- Most IDEs have builtins for it



Tools:

Tools: VS Code

Popular IDE (Integrated Development Enviroment)

- Not important for the course
- Our default for editing messages
- Quick setup for GitHub

Tools: Git + CLI

We'll use a Linux like CLI (Command Line Interface):

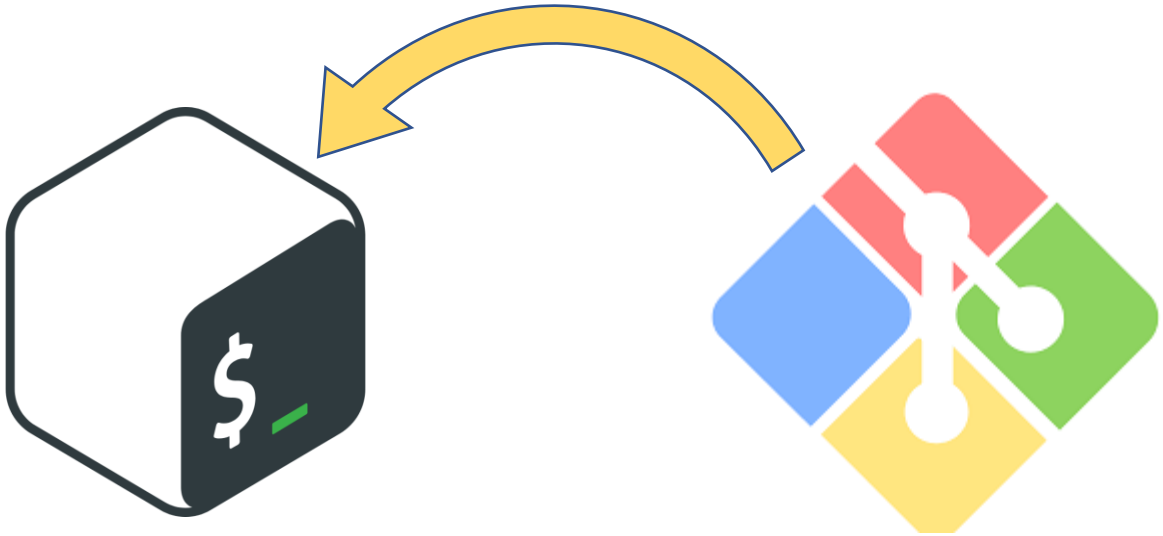
- Windows: Git Bash (Emulator)
- Mac: update to the latest version (most already have it installed)
- Linux: apt-get update && apt-get install git

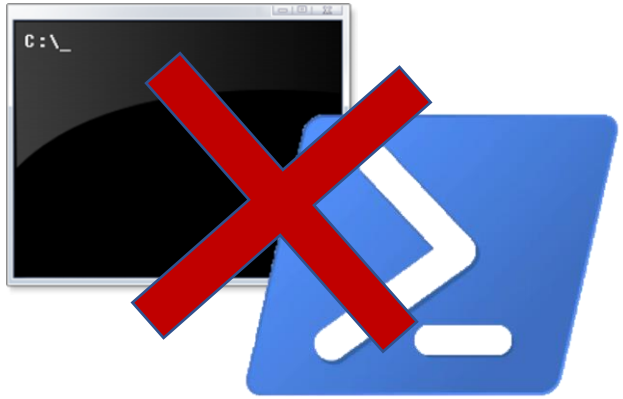
Tools: GitHub

Code hosting platform:

- Remote repositories
- Host our projects
- Collaborate with others

Tools: CLI crash course



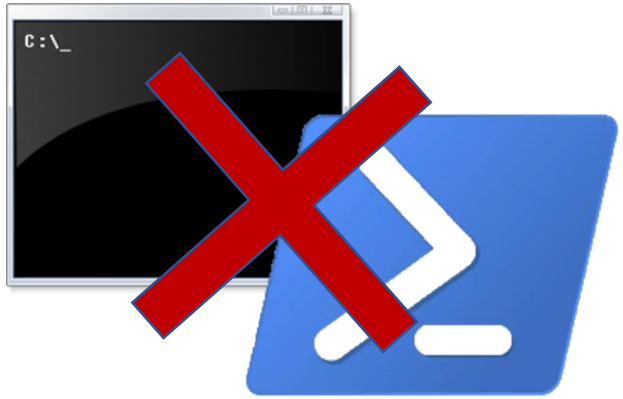


Tools: CLI crash course

Double quotes for paths with spaces

“/” instead of “\”

Other …



Tools: CLI crash course

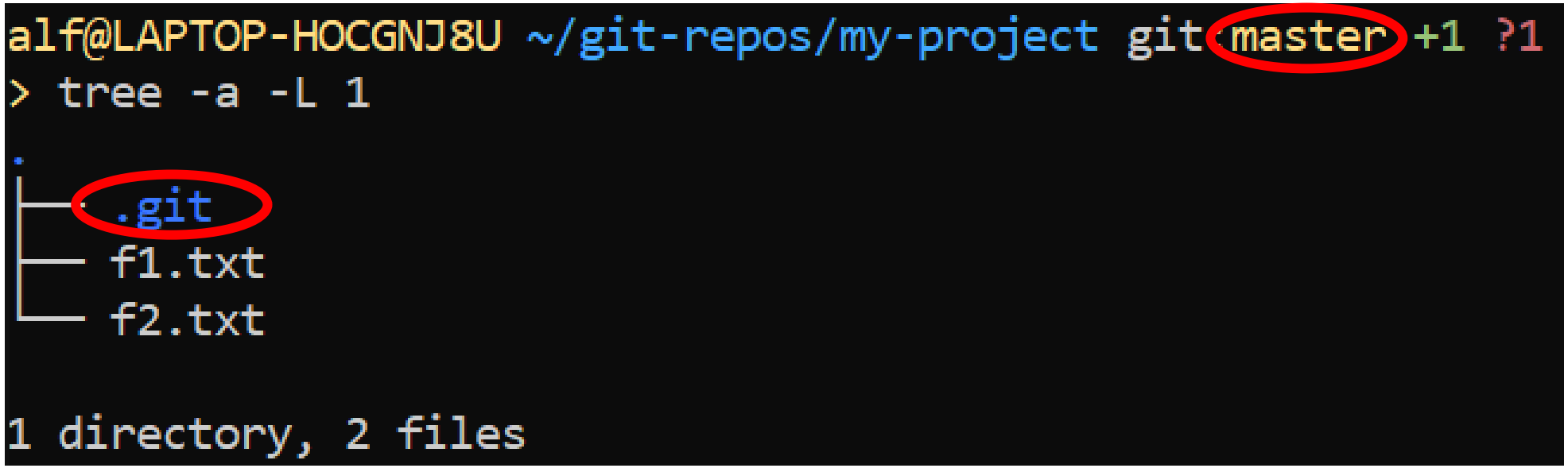
Command	Description	Common Args	Args Descriptions
pwd	print working directory		
ls	list directory	-a	view all (hidden files included)
echo	display message	-e > >>	enable interpretation "", overwrite, append (rhs with lhs)
printf	display string		
cd	change directory	. .. -	current, parent, previous

Tools: CLI crash course

touch	create file		
mkdir	create directory		
cat	print all contents		
cp	copy	-r	recursive (directories)
mv	rename / move	-r	recursive (directories)
rm	remove	-r -f	recursive (directories), force

First steps: Repository

A repository (“repo”) contains all the files and history of a project.



First steps: Status

To check if we are in a git repository we can use `git status` command.

```
alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/some-dir
$ git status
fatal: not a git repository (or any of the parent directories): .git
```

```
alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/my-project (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   program.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Dockerfile
        config.json
        data/
```

First steps: .git directory

Or look for an existing .git directory with ls

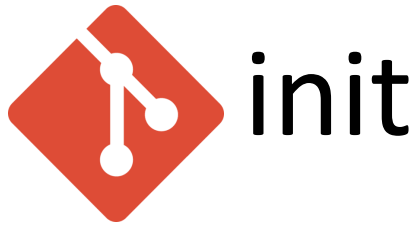
Note: Tipically files starting by “.” are hidden we use `ls –a` to view all files including hidden ones.

```
alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/my-project (master)
$ ls -a
./  ../  .git/  Dockerfile  config.json  data/  program.py
```

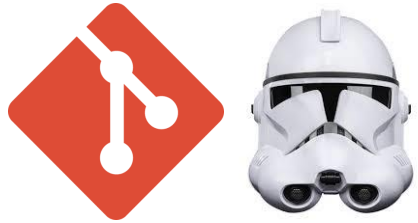
First steps: Initialize and clone

To create a git repository in a machine
2 options:

- Initialize an existent directory



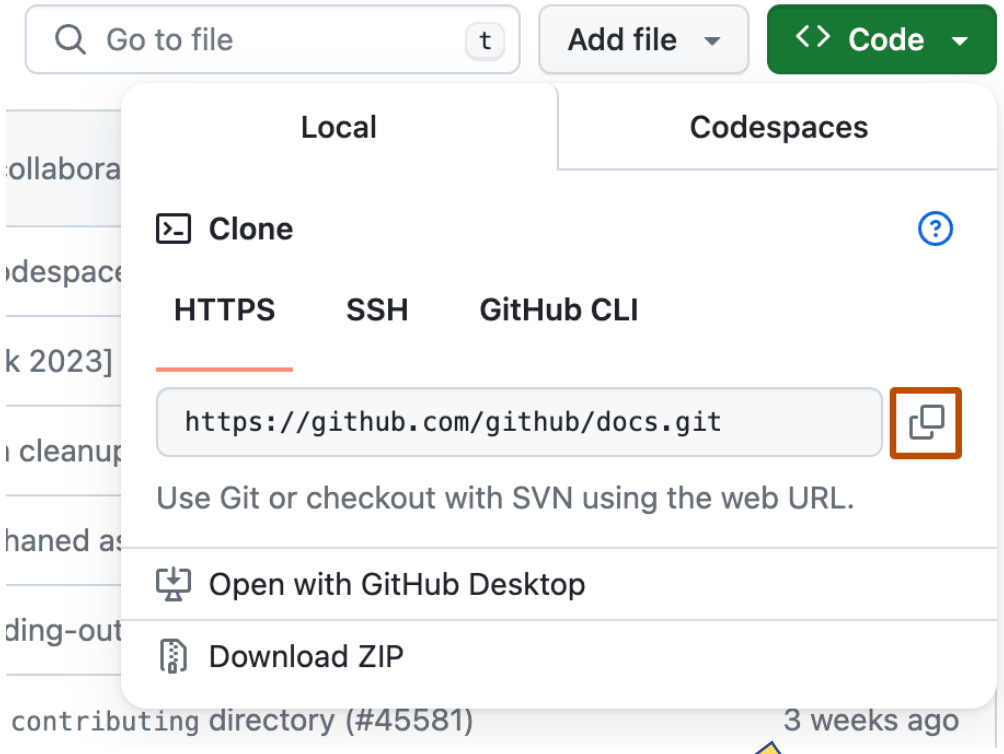
- Clone a remote repository



First steps: git init

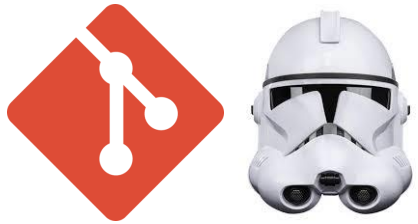
```
alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/my-project
$ git status
fatal: not a git repository (or any of the parent directories): .git

alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/my-project
$ git init
Initialized empty Git repository in C:/Users/alfre/Desktop/my-project/.git/

alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/my-project (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Dockerfile
        config.json
        data/
        program.py

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

First steps: git clone




```
alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop
$ git clone https://github.com/Alf-caput/simple-tests.git
Cloning into 'simple-tests'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 11 (delta 1), reused 8 (delta 0), pack-reused
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (1/1), done.

alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop
$ cd simple-tests/

alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/simple-tests (main)
$ ls -a
./  ../  .git/  README.md  f1
```

Clone notes:



When creating a remote repository (e.g. GitHub repository) allow GitHub to generate some files, useful ones:

- README.md
- .gitignore (predefined preset for some language)

TODO: Practice

- ✓ GitHub account
- ✓ Install and configure tools
- ✓ Bash commands
- ✓ Initialize repositories