


浙江大学爱丁堡大学联合学院

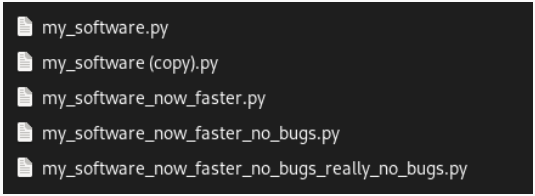
ZJU-UoE Institute

Introduction to Version Control

Nicola Romanò - nicola.romano@ed.ac.uk

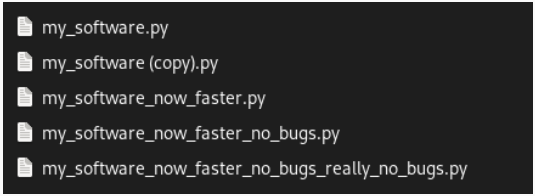
Importance of Code Versioning

- When writing code you often change files multiple times.
- It's tempting to keep multiple copies of the same file with different names, for example to keep track of different versions before and after a change.
- This can lead to confusion and errors.

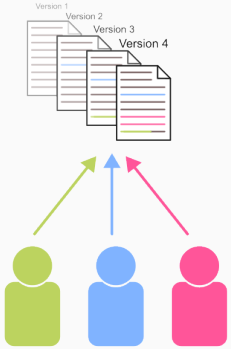


Importance of Code Versioning

- When writing code you often change files multiple times.
- It's tempting to keep multiple copies of the same file with different names, for example to keep track of different versions before and after a change.
- This can lead to confusion and errors.



- A version control system keeps track of changes and allows you to go back to previous versions.
- It also enables collaboration and sharing of code with others.



Git and GitHub

Today we are going to talk about two tools for version control:

- GitHub is a platform for hosting code repositories and collaborating on projects.
- It uses Git, a version control system, to track changes to code over time.
- Enables collaboration by allowing multiple people to contribute to the same project.

What is Git?

- Git is a version control system that lets you track changes in your project.
- It creates snapshots of your project over time.
- You can use Git locally to manage versions, and then use GitHub to share and collaborate.

Setting Up your system

Installing Git and registering on GitHub

https://git-scm.com/

https://www.github.com/

Installing Git and registering on GitHub

https://git-scm.com/

https://www.github.com/

Check GIT installation

```
$> git --version
git 2.46.2
```

Configure GIT

```
$> git config --global user.name "Your name"
$> git config --global user.email "email@domain.com"
```

Some Git nomencalture

When using Git, you will encounter some terms that are important to understand:

- **Repository**: a directory that contains your project, including the files/directories in your project, as well as the history of changes to those files.

Some Git nomencalture

When using Git, you will encounter some terms that are important to understand:

- **Repository**: a directory that contains your project, including the files/directories in your project, as well as the history of changes to those files.
- **Staging**: the process of preparing files to be committed. This is done before you commit changes to your repository.
- **Commit**: a snapshot of your project at a specific point in time.
- **Push**: sending your changes from the local computer to a remote repository.
- **Pull**: getting changes from a remote repository to your local computer.

Some Git nomencalture

When using Git, you will encounter some terms that are important to understand:

- **Repository**: a directory that contains your project, including the files/directories in your project, as well as the history of changes to those files.
- **Staging**: the process of preparing files to be committed. This is done before you commit changes to your repository.
- **Commit**: a snapshot of your project at a specific point in time.
- **Push**: sending your changes from the local computer to a remote repository.
- **Pull**: getting changes from a remote repository to your local computer.
- **Branch**: a parallel version of your repository that allows you to work on a feature or bugfix without affecting the main project.
- **Pull request**: a request to merge changes from one branch to another.

Summary of git commands