Lab environment and tools

About this activity

This is a description of the lab environment used for Part 2: Testing. You should install the tools and use them.

Git and GitHub

Using Git is optional in the assignment, but it is also highly recommended as a learning experience. If you for some reason fail in using Git, then you can always go back and just "copy the files" with your partner.

The article "**Work with Git (https://gitlab.com/mikael-roos/oopython/-/blob/main/public/doc/work-with-git.md)** " provides an overview of the Git commands to use and it has a **video playlist (https://www.youtube.com/playlist?list=PLEtyhUSKTK3iTFcdLANJq0TkKo246XAlv)** that introduce the basics on how to install and use Git in the terminal using Git Bash and how to connect the repo with a web service like GitHub.

Bash terminal

You should use a bash terminal when you work with the exercises in the course. On Linux/Mac you already have a terminal. On Windows you can use the terminal Git Bash you get while installing Git.

There is a video showing "**Use various bash terminals on Windows (Git Bash, Cygwin, Debian/WSL2) (https://www.youtube.com/watch?v=kialYZs6Oyc&list=PLEtyhUSKTK3gHj087mUjPfXyqMvSy2Rwz&index=2)** ".

Python

Check that you have the latest version of Python 3 installed. There is a video showing how to "**Install python on Windows and run in cmd terminal and in git bash terminal (https://www.youtube.com/watch?v=PeM9UxEGH0o&list=PLEtyhUSKTK3hOCnMrPKGOu3_VjUAkhsgG&index=2)** ".

Python virtual environment (venv)

Work through the article and exercise "**Work in a Python virtual environment (https://gitlab.com/mikael-roos/oopython/-/blob/main/public/doc/python-venv.md)** " that shows how to get going with a Python virtual environment. The article includes a video showing how its done.

The basic idea is to install all needed into the working directory of your project. This means you do not need to install the Python development environment as a global installation to your system, the installation is only done into the specific project directory. This allows you to have several different installations with different versions of packages at the same time.

Make and Makefiles

The Make command and the Makefiles are in general used to compile, build and run programs and tasks. If you have not yet heard about Makefiles, then read the short **introduction to Makefiles (https://www.gnu.org/software/make/manual/html_node/Introduction.html)** .

Ensure that you have it installed in your terminal, it should already be installed on Linux/Mac.

You can check your current version like this.

```
$ make --version
GNU Make 4.3
```

Install make on Windows

If you are using Git Bash on Windows, then check out this post about how to **install make on Windows (https://stackoverflow.com/a/32127632)** . The Windows package manager `chocolatey` is one way to do this. You will then install the make command within Windows and it can then be used from Git Bash.

1. **Install the Windows packet manager Chocolatey (https://chocolatey.org/install)** .
2. **Install GNU make (https://community.chocolatey.org/packages/make)** using `choco install make` using PowerShell (you might need to run the terminal as admin).
3. Open a new window for Git Bash and check that it works be checking what version you have using `make --version`.

If you need assistance there is a video here.

- **Install GNU Make on Windows using Chocolatey package manager (https://www.youtube.com/watch?v=5TavcolACQY&list=PLEtyhUSKTK3hOCnMrPKGOu3_VjUAkhsgG&index=3)**

(https://www.youtube.com/watch?v=5TavcolACQY&list=PLEtyhUSKTK3hOCnMrPKGOu3_VjUAkhsgG&index=3)

Work with an example Python development repo

When you are all done installing the lab environment, then proceed to check it out if it works together with a Python development repo.

In the article/exercise "**Work with an example Python development repo (https://gitlab.com/mikael-roos/oopython/-/blob/main/public/doc/work-with-a-example-python-development-repo.md)** " (video included) you will clone a repo with a Python development repository that has a few example programs of object oriented Python program to learn from.

When you work through the article you will also use the Makefile and work in a venv.

Notice how the repo is structured, the intention is that you should work in the same directory structure when you work with your assignment.