

**KADI**



**TRAINING ON 28TH APRIL 2025, KAPITI FARM:**

Prepared by Levis Muse, Silas Issa and Mathew Mutuku

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>MORNING SESSION:</b>	<b>2</b>
Recap for Sunday	2
GIT	2
Terminal	2
Configuration:	3
Exercise	5
Deleting files	5
Branches	6
COLLABORATIONS	7
SECRET FILES	7
<b>AFTER TEA SESSION :</b>	<b>9</b>
VIRTUAL ENVIRONMENTS	9
Creating an environment	9
PACKAGES	10
EXTENSIONS	12
Jupyter	12
AutoDoc	12
<b>LUNCH BREAK</b>	<b>13</b>
<b>AFTERNOON SESSION:</b>	<b>13</b>
DATA	13
OPENING WITH JUPYTER NOTEBOOK	13
Kernel selection	13
Alternate way Jupyter	14
CLONING DATA FROM REPO	14
BRANCHING	14
PYTHON FUNCTIONS	15
Open ()	15
Help ()	16
Exercise	16
<b>Evening Session</b>	<b>17</b>
DATA TYPES	17
Loop Functions	17
Example 1	18
Example 2	18

# MORNING SESSION:

Morning walk

Recap for Sunday

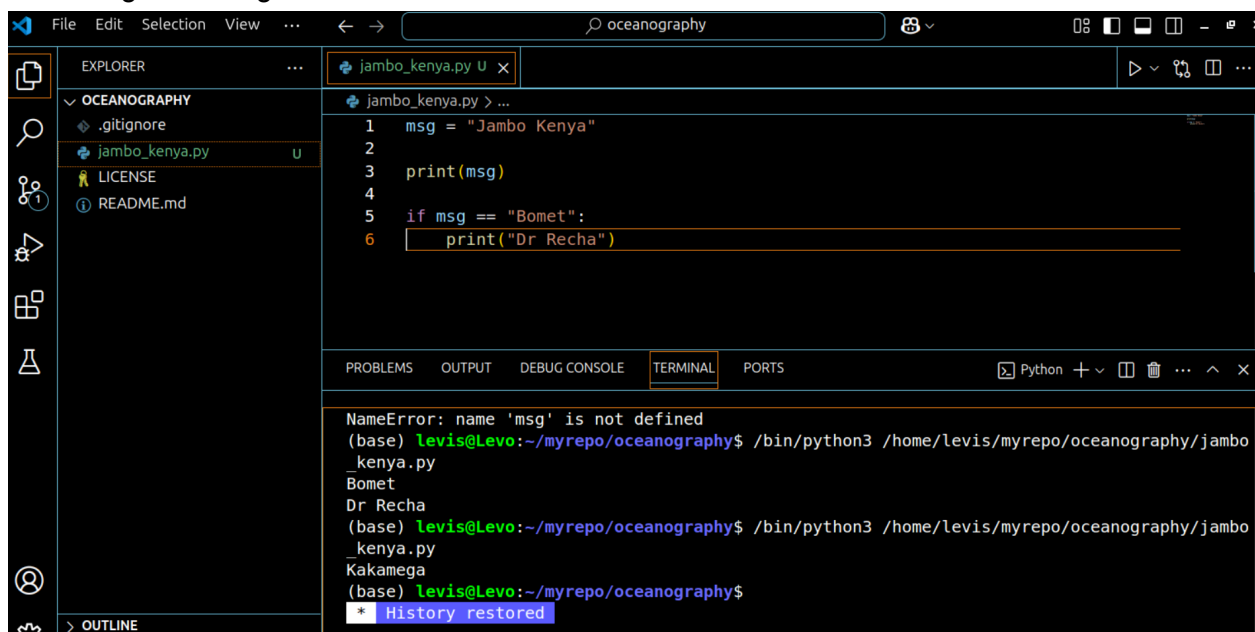
## GIT

The purpose was to integrate GIT into the visual studio workflow.

## Terminal

The layout of visual studio was explained with its main components including the explorer, terminal, source control and extensions.

Accessing and using the terminal is shown



The screenshot displays the Visual Studio IDE interface. On the left, the Explorer pane shows a project named 'OCEANOGRAPHY' with files: '.gitignore', 'jambo\_kenya.py' (marked with a 'u' for untracked), 'LICENSE', and 'README.md'. The main editor window shows the code for 'jambo\_kenya.py':

```
1 msg = "Jambo Kenya"
2
3 print(msg)
4
5 if msg == "Bomet":
6     print("Dr Recha")
```

Below the editor, the TERMINAL pane is active, showing the output of running the script with Python 3:

```
(base) levis@Levo:~/myrepo/oceanography$ /bin/python3 /home/levis/myrepo/oceanography/jambo_kenya.py
Bomet
Dr Recha
(base) levis@Levo:~/myrepo/oceanography$ /bin/python3 /home/levis/myrepo/oceanography/jambo_kenya.py
Kakamega
(base) levis@Levo:~/myrepo/oceanography$
```

A status bar at the bottom of the terminal indicates '\* History restored'.

The terminal gives an additional feature to run commands in visual studio.

The following command

```
> git status
```

```

• (.venv) (base) levis@Levo:~/myrepo/oceanography$ git status
On branch main
Your branch is up to date with 'origin/main'.

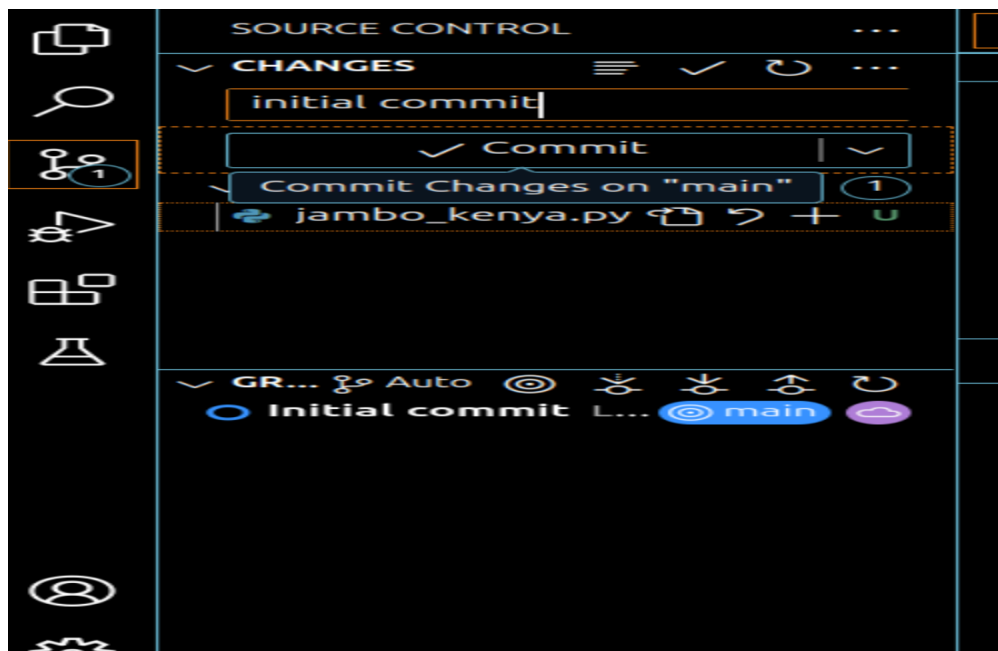
nothing to commit, working tree clean
○ (.venv) (base) levis@Levo:~/myrepo/oceanography$

```

## Source control

The ability to track changes and be able to work on pending changes to the remote repo.

- Make sure the file changes have been saved
- Stage a file changes
- Message for accompany the changes
- Commit a file
- Sync changes
- Once completed look at the Graph for the branch



Click **+** next to your file name to stage

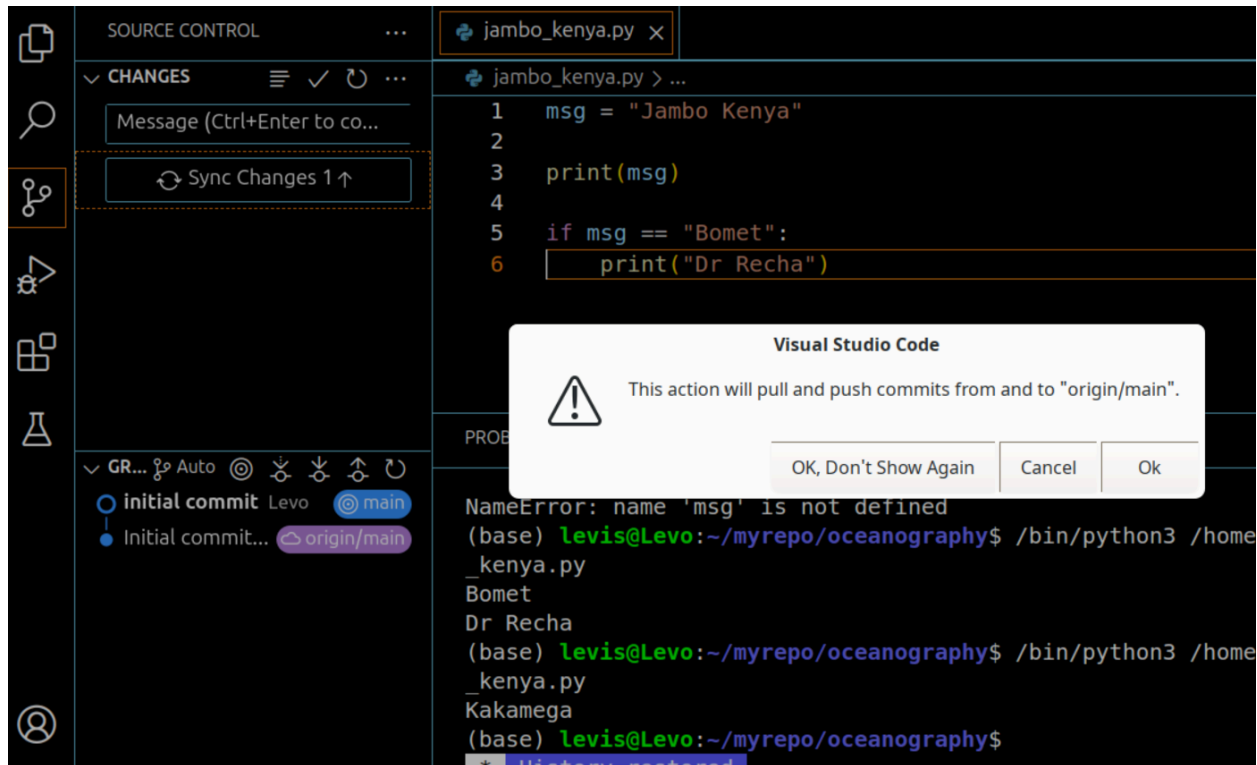
Then write initial commit message in the message box: write meaningful commit messages  
Commit the file

## Configuration:

On the terminal configure the following

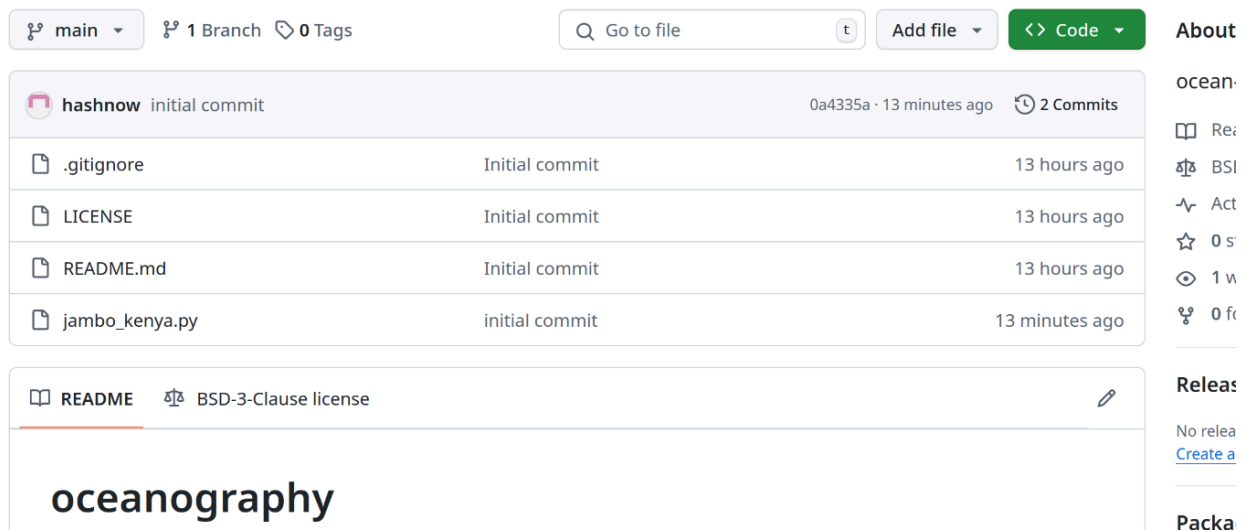
- **Set username:** Use the command `git config --global user.name "Levo"`.
- **Set email:** Use the command `git config --global user.email "levisjmuse@gmail.com"`.

Sync changes



Visit the website of GIT.

You will see the file uploaded in your remote repository



Git status

```
• (base) levis@Levo:~/myrepo/oceanography$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Using the terminal commands

>git commit - m "feat(text.txt)"

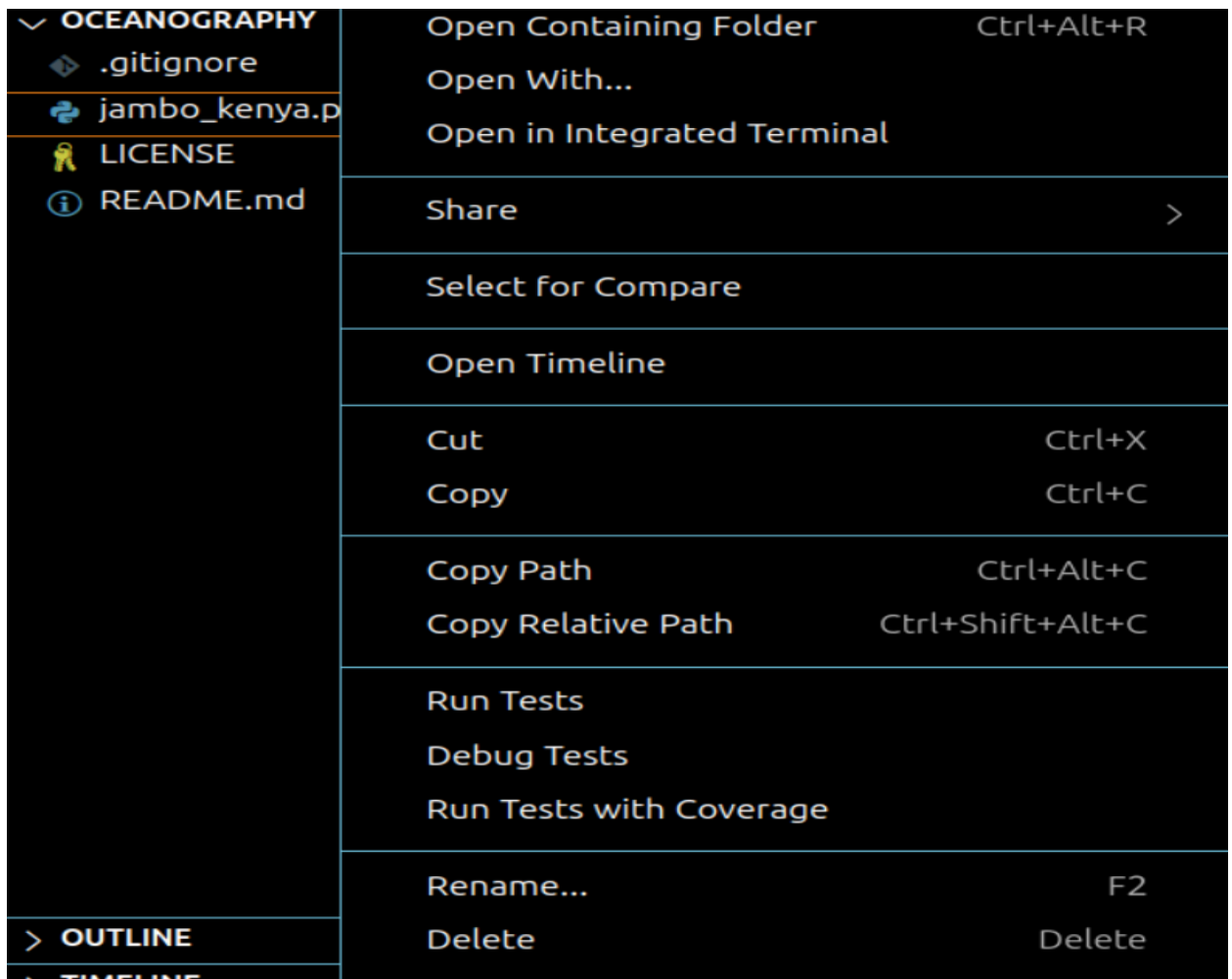
> git push

## Exercise

1. Clone python data gym repository: <https://github.com/KADI-Project/Python-Data-Gym>
2. Forke a classmate repo
3. Create and merge a branch

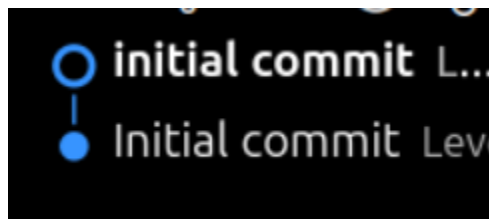
## Deleting files

- Move to trash
- Delete



Check the source control to commit the changes

## Branches



- ❖ Create new branches
- ❖ Merge branches
- ❖ Main branch

## COLLABORATIONS

Wiki page

Github handles

<https://github.com/hashnow/>

Watch: if changes are made

Create a fork

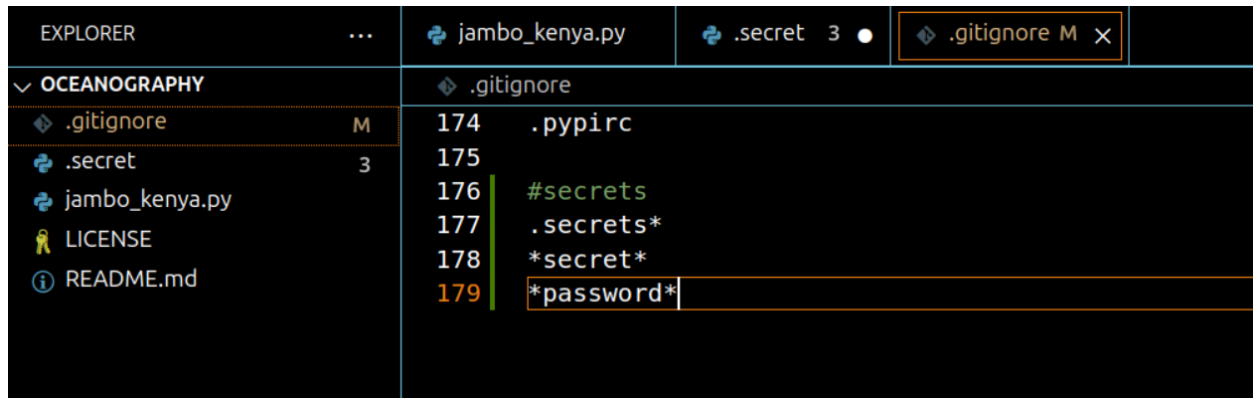
Security details

## SECRET FILES

File name .secret

Open gitignore

Add the secret file words



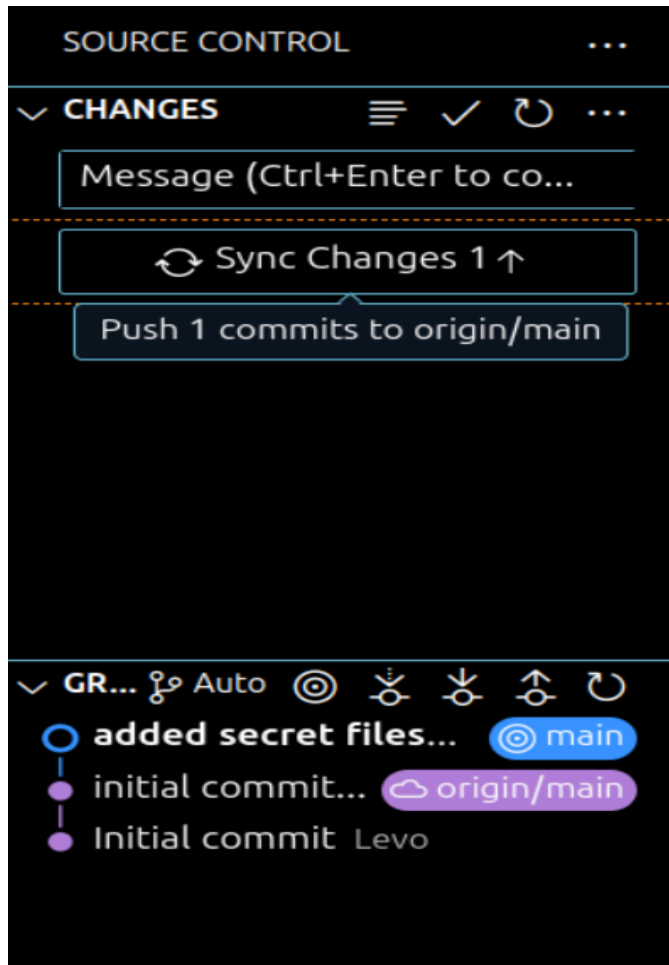
The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the file structure of a project named 'OCEANOGRAPHY'. The files listed are '.gitignore' (marked as modified with an 'M'), '.secret' (with a count of 3), 'jambo\_kenya.py', 'LICENSE', and 'README.md'. The main editor area on the right shows the content of the '.gitignore' file. The file contains the following lines: '.pypirc', '.secrets\*', '\*secret\*', and '\*password\*'. The line '\*password\*' is currently selected, and the cursor is at the end of the line.

```
174 .pypirc
175
176 #secrets
177 .secrets*
178 *secret*
179 *password*
```

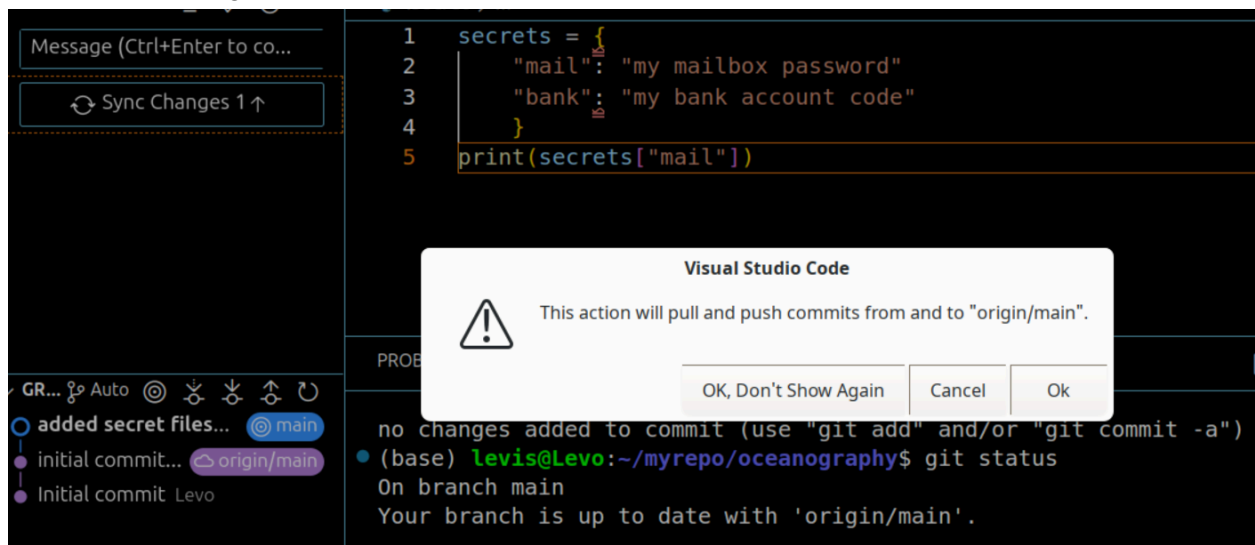
Commit the changes in the source control

Observe the graph





Confirm the changes



> git status

**Do not commit .secrets or .venv**

# AFTER TEA SESSION :

## VIRTUAL ENVIRONMENTS

One project = one environment

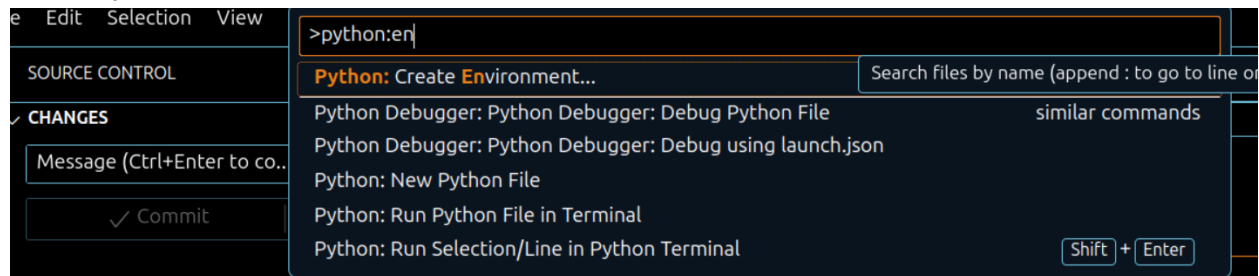
Create little boma with everything that you need: that is an environment with all the libraries that you need.

### Creating an environment

1. Venv
2. Conda

On the search box type: >python :envir

Select python: creat



Choose “venv” and complete the process

Activate the environment

linux : Source .venv/bin/activate

Powershell (PS): .venv\Scripts\Activate.ps1

### Access different shells



> deactivate – to leave the environment

Make sure you have (.venv)

## PACKAGES

Use: pip install

>pip install requests

> pip install polars

> pip install matplotlib

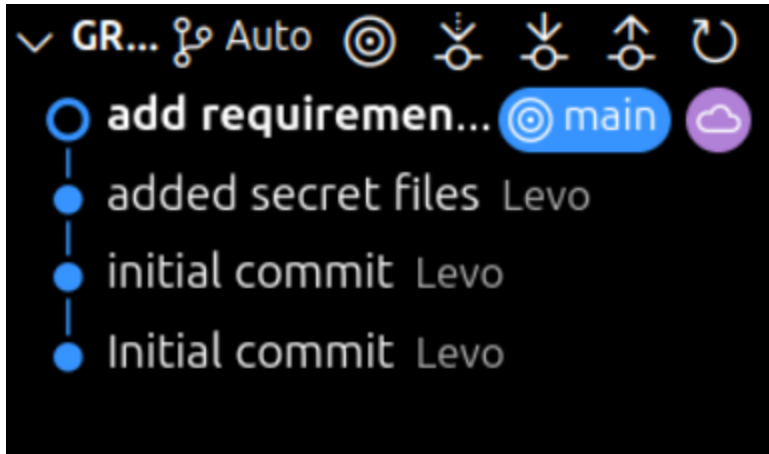
> pip install pandas

> pip freeze > requirements.txt

In the venv/lib path see libraries installed



In the source control commit and sync the changes for requirements file



## EXTENSIONS

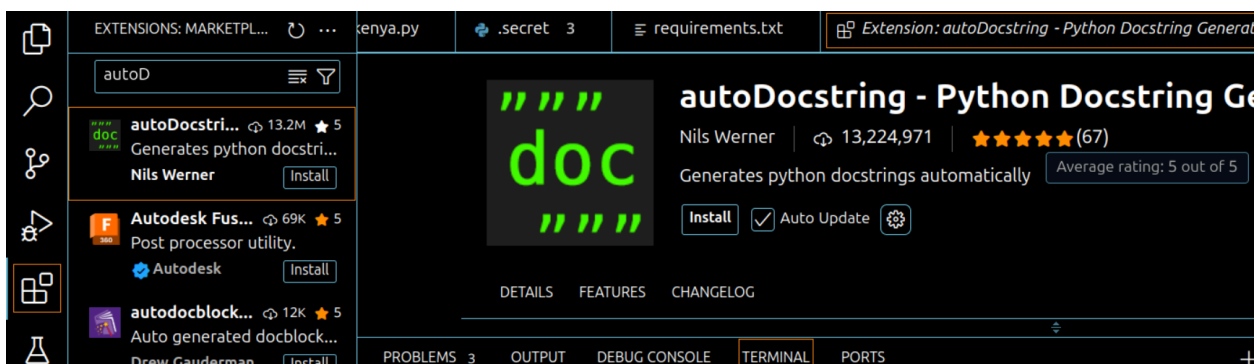
### Jupyter

Install jupyter from the extension



### AutoDoc

Install AutoDoc



# LUNCH BREAK

## AFTERNOON SESSION:

### DATA

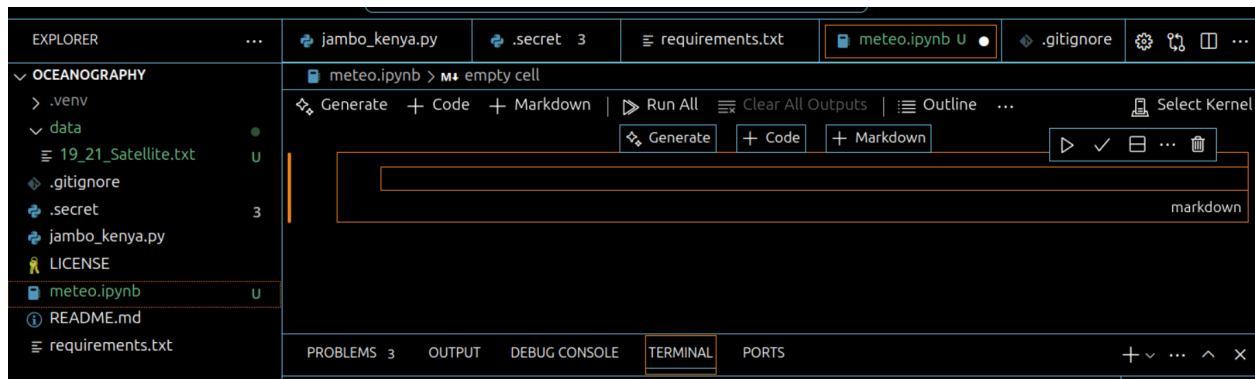
Create a data folder

- ❖ Data formats
  - ★ CSV, TXT
  - ★ Excel
  - ★ netcdf
- ❖ Reading csv files
  - ★
- ❖

### OPENING WITH JUPYTER NOTEBOOK

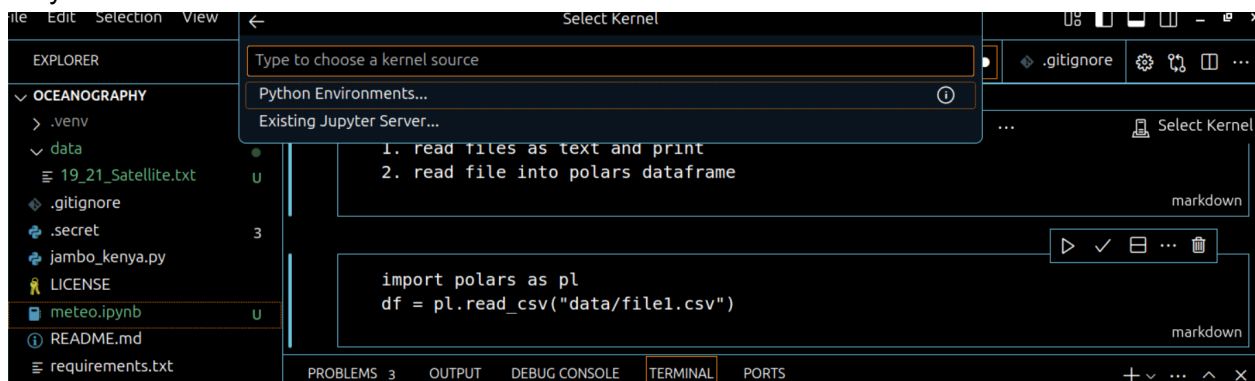
Save a file with ipynb extensions

meteo.ipynb



Kernel selection

> Python environment

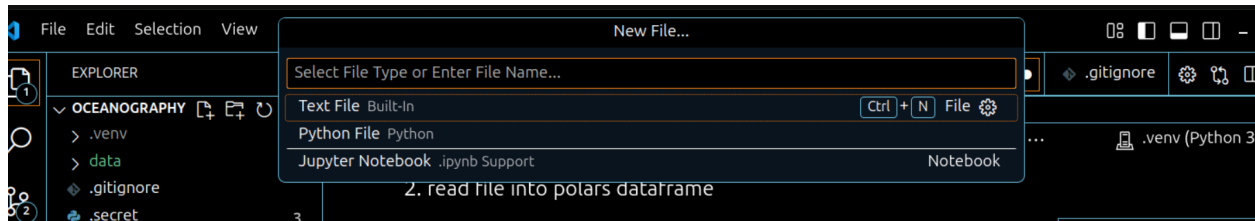


1. Markdown – text
2. Code – writing code

Alternate way Jupyter

Another way to open jupyter notebook

FILE -> NEW FILE ->



## CLONING DATA FROM REPO

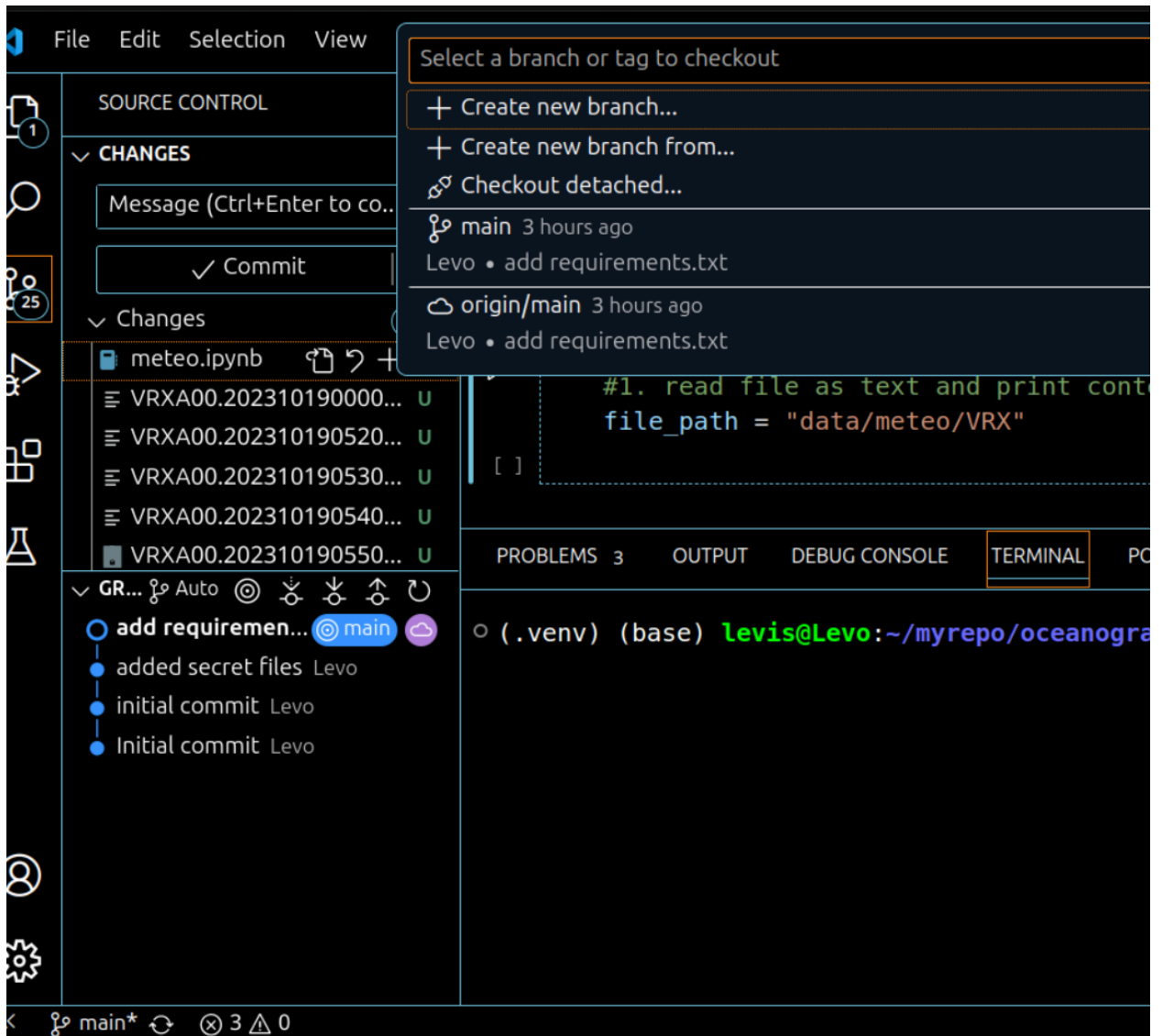
Procedure

1. FILE
2. NEW WINDOW
3. CLONE GIT REPO
4. Clone from github
5. Choose the repo /name/path/

## BRANCHING

Procedure to branch a collaborators work

1. SELECT main branch bottom left
2. Create new branch
3. Branch name



## PYTHON FUNCTIONS

Open ()

Read file as text and print content



```
nb > with open(file=file_path, mode='r') as fh:
+ (function) def open(
    file: FileDescriptorOrPath,
    mode: OpenTextMode = "r",
    buffering: int = -1,
    encoding: str | None = None,
    errors: str | None = None,
    newline: str | None = None,
    closefd: bool = True,
    opener: _Opener | None = None
) -> TextIOWrapper[_WrappedBuffer]
Open file and return a stream. Raise OSError upon failure.
file is either a text or byte string giving the name (and the path)
th open(file=file_path, mode='r') as fh:
    file_content = fh.readlines()
```

Encoding options: utf8, ascii

Mode: "r" read

## Help ()

For more information use the help function

> help(open)

Example code on printing the header

```
header = file_content[3]
print(header)
header_list = header.split()
print(header_list)
21] ✓ 0.0s Python
iii zzzzttrt tre200s0 uor200s0 prestas0 fa1010z0 da1010z0 rre150z0 ta1200s0 ua1200s0 pa1stas0 fkl010z0 dkl010z0 r
['iii', 'zzzztttt', 'tre200s0', 'uor200s0', 'prestas0', 'fa1010z0', 'da1010z0', 'rre150z0', 'ta1200s0', 'ua1200s0', 'pa1stas0', 'fkl010z0', 'dkl010z0']
```

## Exercise

1. Extract the header element for temperature(tre200s0)
2. Extract the corresponding temperature reading
3. Print the result as follows: tre200s

Solution

```
temperature = header_list[2]
print(temperature)
values = file_content[4]
values = values.split()
print(values)
temperature_value = values[2]
temperature_value = float(temperature_value)
print(int(temperature_value))

tre200s0
['187', '202310190530', '007.3', '098.6', '0662.9', '003.4', '329.6', '00000.00', '006.8', '100.0', '0662.7', '00
7

print(temperature, " : ", (temperature_value))

tre200s0 : 7.3
```

## Evening Session

### DATA TYPES

1. FLOAT → float()
2. INTEGER → int()

### Loop Functions

#### Creating

Loop are important if you have repetitive task

- For loop

## Example 1

```
> ~
    a = 1
    for i in range(10):
        a = a + i
        if a > 20:
            break
        print(i, ":", a)

35] ✓ 0.0s

.. 0 : 1
   1 : 2
   2 : 4
   3 : 7
   4 : 11
   5 : 16
```

## Example 2

```
import os
path = "data/meteo"
files = os.listdir(path)
datetimes = list()
temperatures = list()
for file in files:
    if ".zip" in file:
        continue
    else:
        #print(file)
        datetime = file.replace("VRXA00.", "")
        datetimes.append(datetime)
        with open(file=os.path.join(path, file)) as fh:
            file_content = fh.readlines()

            temp = file_content[4].split()[2]

            temperatures.append(temp)

print(temperatures)
print(datetimes)

✓ 0.0s

['008.0', '007.0', '007.8', '008.5', '008.1', '/', '/', '008.6', '/', '008.6', '005.1', '/', '007.3']
['202310190630', '202310190540', '202310190600', '202310190650', '202310190640', '202310190520', '202401111540', '202310190610']
```

The order is not preserved but the match should be preserved