TECNICHE DI RAPPRESENTAZIONE E MODELLIZZAZIONE DEI DATI

— Part 1 —

(2 CFU out of 6 total CFU)

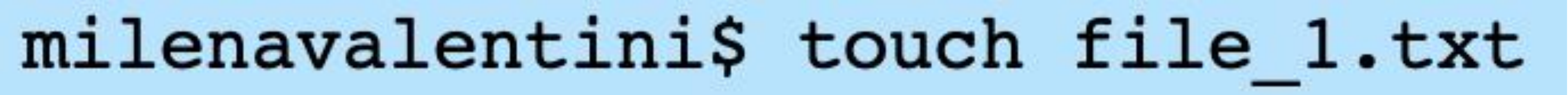
Link moodle: https://moodle2.units.it/course/view.php?id=11703

Teams code: 0ftoqj8

Dr. Milena Valentini | 29/09/2023

Command line

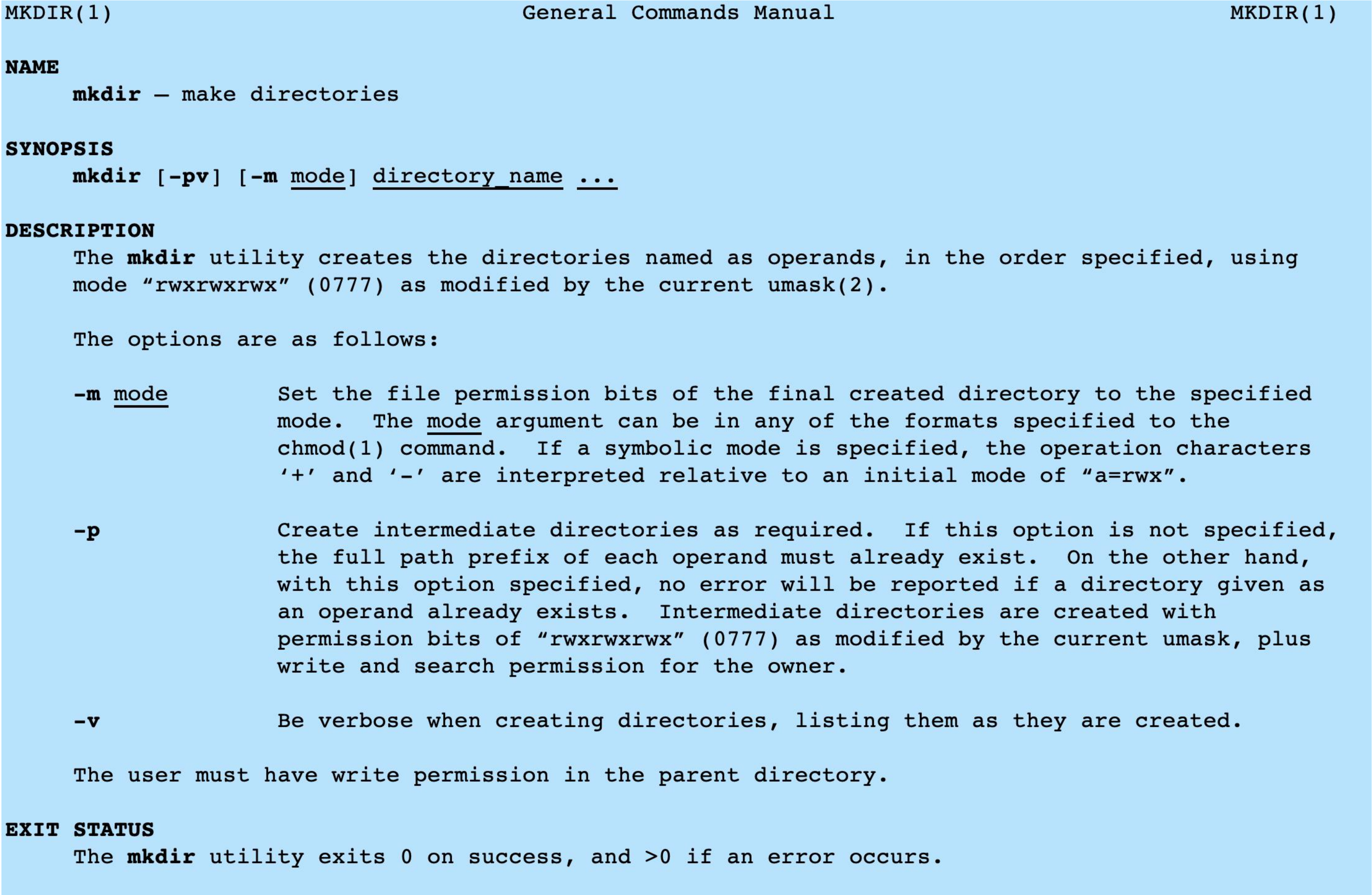
The **touch** command creates a file



```
milenavalentini$ touch file_1.txt
```

Command line

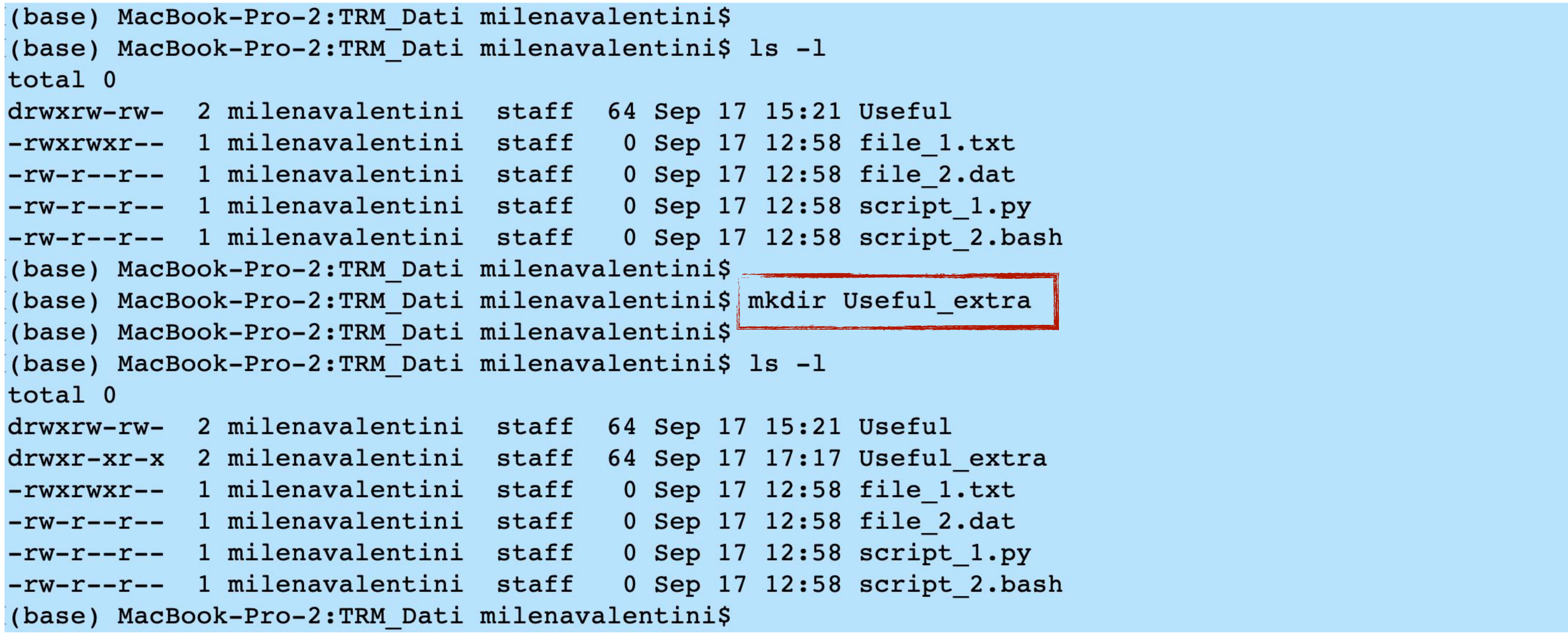
The **mkdir** command creates a directory



```
MKDIR(1)                    General Commands Manual                   MKDIR(1)

NAME
     mkdir – make directories

SYNOPSIS
     mkdir [-pv] [-m mode] directory_name ...

DESCRIPTION
     The mkdir utility creates the directories named as operands, in the order specified, using
     mode “rwxrwxrwx” (0777) as modified by the current umask(2).

     The options are as follows:

     -m mode        Set the file permission bits of the final created directory to the specified
                    mode.  The mode argument can be in any of the formats specified to the
                    chmod(1) command.  If a symbolic mode is specified, the operation characters
                    ‘+’ and ‘-’ are interpreted relative to an initial mode of “a=rwx”.

     -p             Create intermediate directories as required.  If this option is not specified,
                    the full path prefix of each operand must already exist.  On the other hand,
                    with this option specified, no error will be reported if a directory given as
                    an operand already exists.  Intermediate directories are created with
                    permission bits of “rwxrwxrwx” (0777) as modified by the current umask, plus
                    write and search permission for the owner.

     -v             Be verbose when creating directories, listing them as they are created.

     The user must have write permission in the parent directory.

EXIT STATUS
     The mkdir utility exits 0 on success, and >0 if an error occurs.
```

Command line

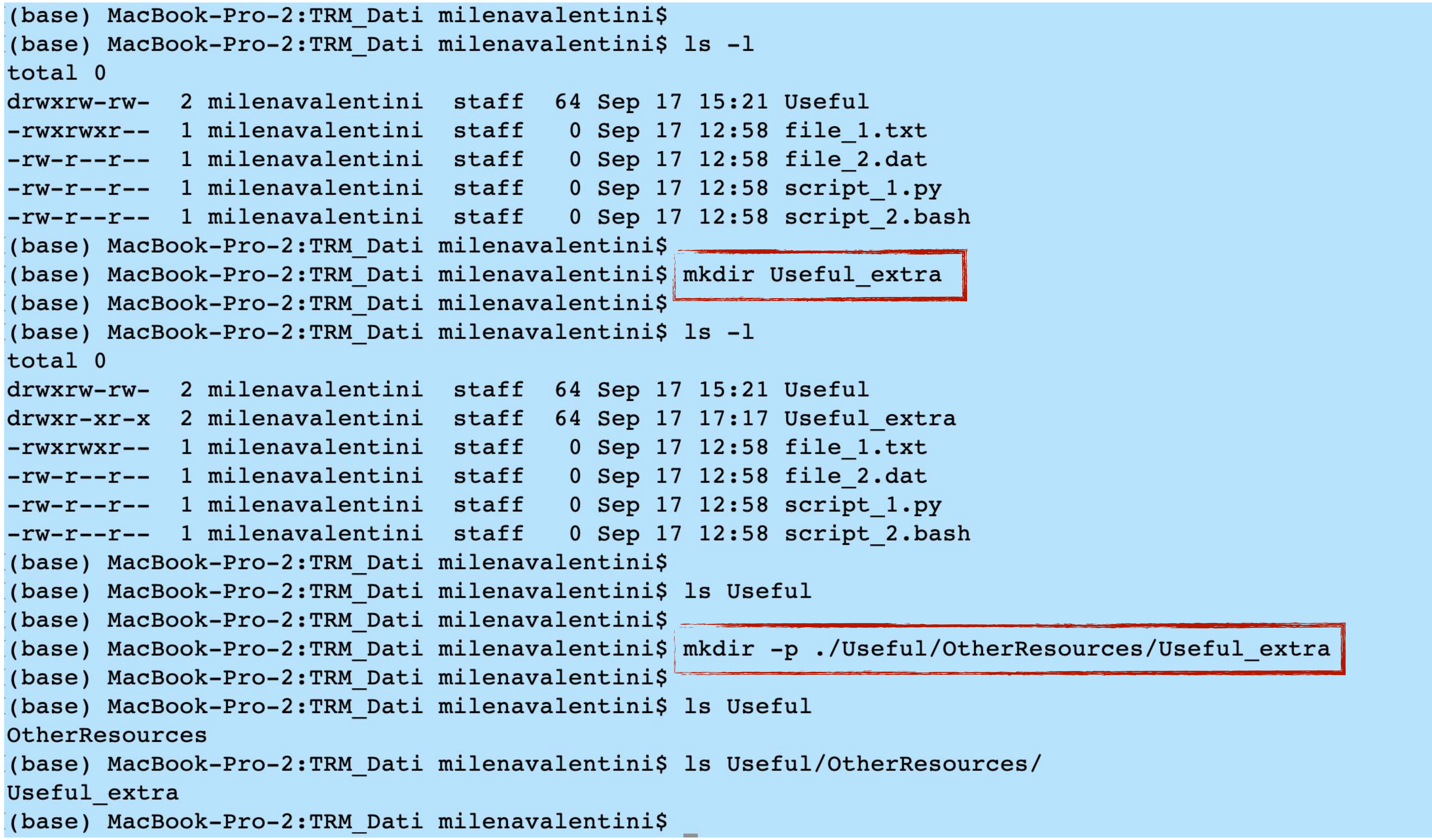
The **mkdir** command creates a directory — examples:



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls -l
total 0
drwxrw-rw-  2 milenavalentini  staff  64 Sep 17 15:21 Useful
-rwxrwxr--  1 milenavalentini  staff   0 Sep 17 12:58 file_1.txt
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 file_2.dat
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 script_1.py
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 script_2.bash
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ mkdir Useful_extra
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls -l
total 0
drwxrw-rw-  2 milenavalentini  staff  64 Sep 17 15:21 Useful
drwxr-xr-x  2 milenavalentini  staff  64 Sep 17 17:17 Useful_extra
-rwxrwxr--  1 milenavalentini  staff   0 Sep 17 12:58 file_1.txt
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 file_2.dat
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 script_1.py
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 script_2.bash
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

Command line

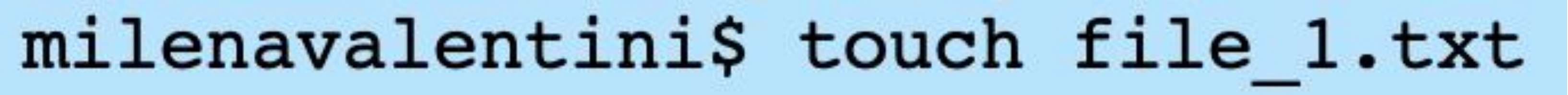
The **mkdir** command creates a directory — examples:



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls -l
total 0
drwxrw-rw-  2 milenavalentini  staff  64 Sep 17 15:21 Useful
-rwxrwxr--  1 milenavalentini  staff   0 Sep 17 12:58 file_1.txt
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 file_2.dat
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 script_1.py
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 script_2.bash
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ mkdir Useful_extra
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls -l
total 0
drwxrw-rw-  2 milenavalentini  staff  64 Sep 17 15:21 Useful
drwxr-xr-x  2 milenavalentini  staff  64 Sep 17 17:17 Useful_extra
-rwxrwxr--  1 milenavalentini  staff   0 Sep 17 12:58 file_1.txt
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 file_2.dat
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 script_1.py
-rw-r--r--  1 milenavalentini  staff   0 Sep 17 12:58 script_2.bash
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls Useful
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ mkdir -p ./Useful/OtherResources/Useful_extra
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls Useful
OtherResources
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls Useful/OtherResources/
Useful_extra
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ 
```

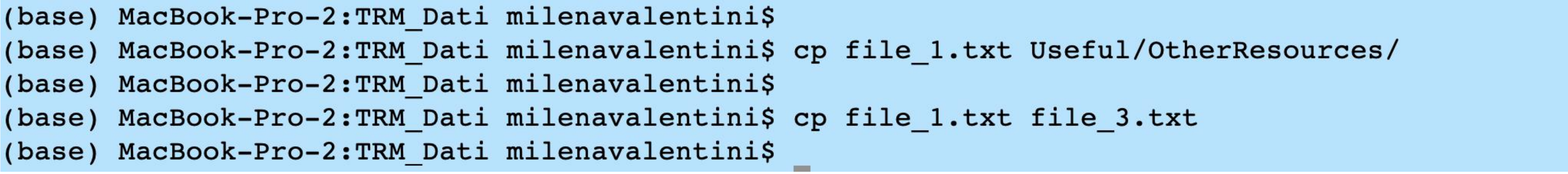
Command line

The **touch** command creates a file



```
milenavalentini$ touch file_1.txt
```

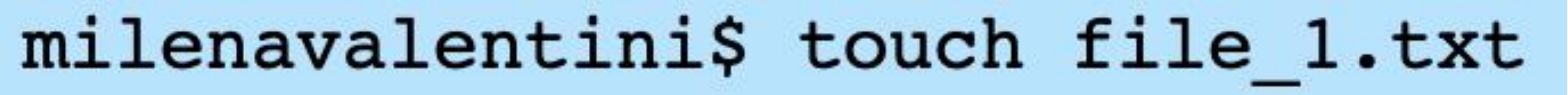
The **cp** command copies one file from a directory to another
or a given file into another file



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cp file_1.txt Useful/OtherResources/
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cp file_1.txt file_3.txt
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ 
```

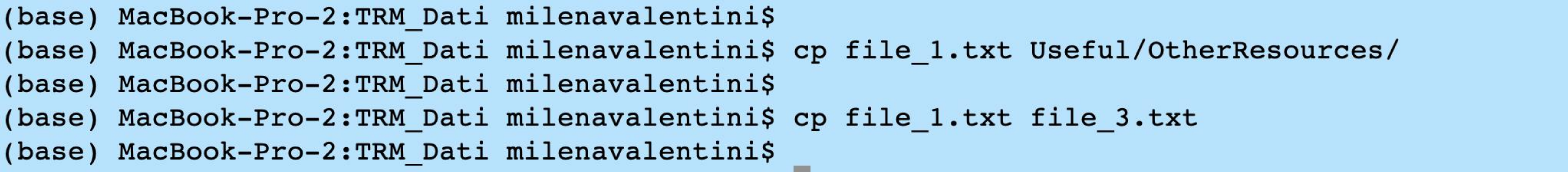
Command line

The **touch** command creates a file



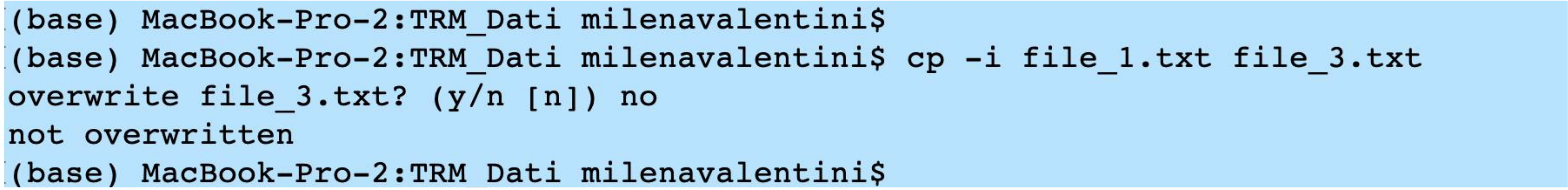
```
milenavalentini$ touch file_1.txt
```

The **cp** command copies one file from a directory to another
or a given file into another file



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cp file_1.txt Useful/OtherResources/
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cp file_1.txt file_3.txt
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

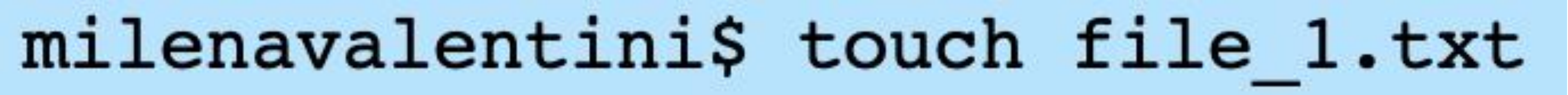
cp overwrites (e.g. if file_3.txt is not empty, it will then be the same as file_1.txt)



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cp -i file_1.txt file_3.txt
overwrite file_3.txt? (y/n [n]) no
not overwritten
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

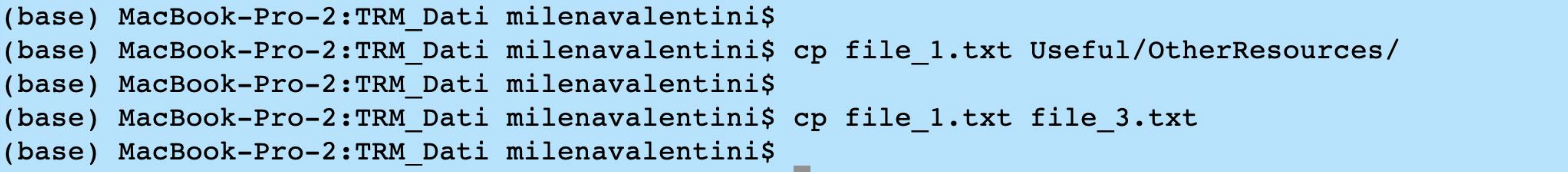
Command line

The **touch** command creates a file



```
milenavalentini$ touch file_1.txt
```

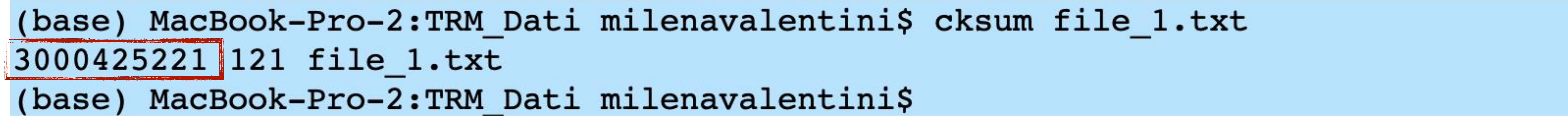
The **cp** command copies one file from a directory to another
or a given file into another file



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cp file_1.txt Useful/OtherResources/
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cp file_1.txt file_3.txt
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ 
```

The **rsync** commands behaves similarly to the cp command: it deals with file transfer and allows you to only transfer the differences between two sets of files using a checksum-search algorithm.
It’s especially useful when network/ssh connections are involved.

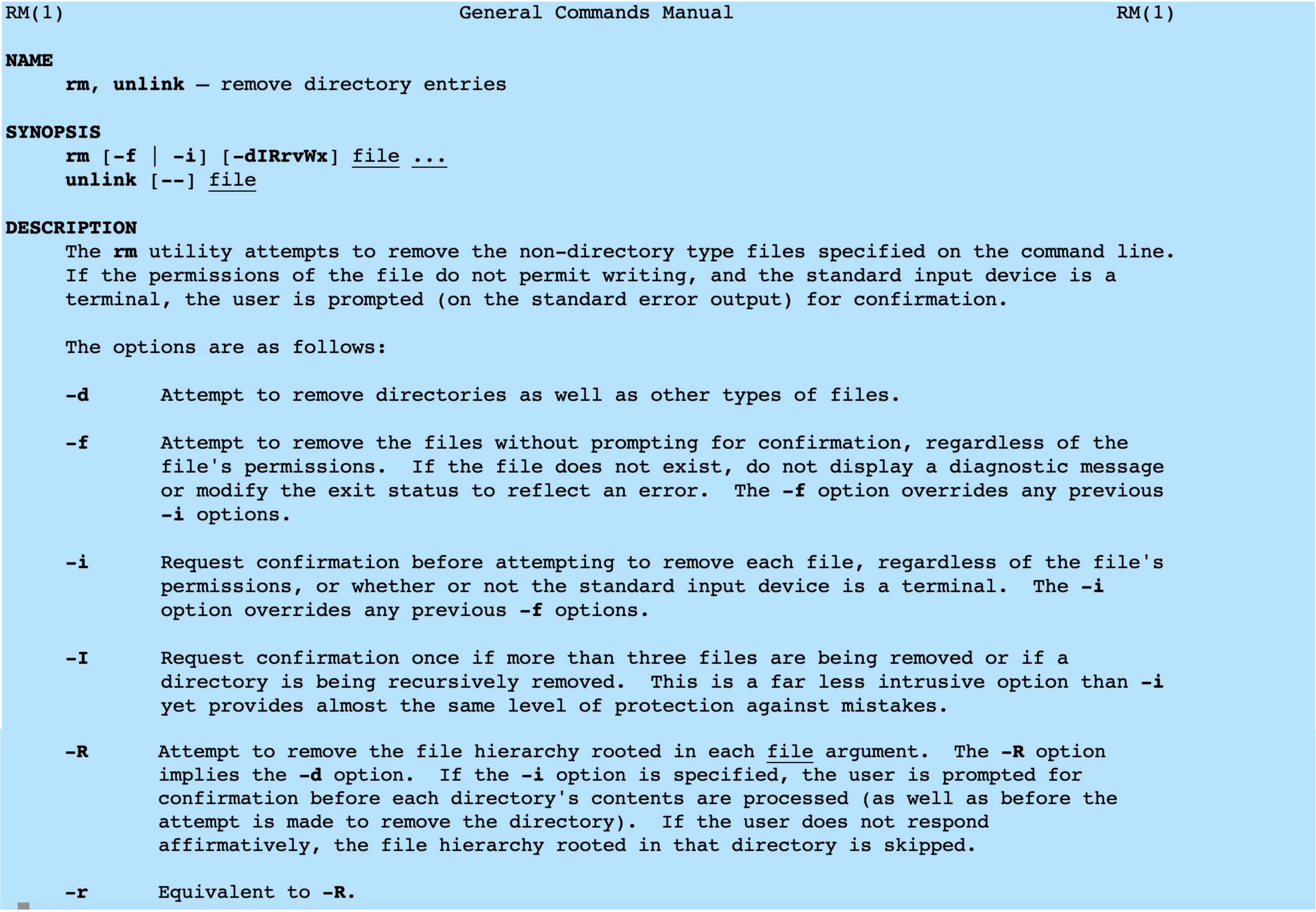
The **cksum** command displays a check value, the total number of octets in the file, and the filename itself.



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cksum file_1.txt
3000425221 121 file_1.txt
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

How to manipulate files

rm removes files and directories



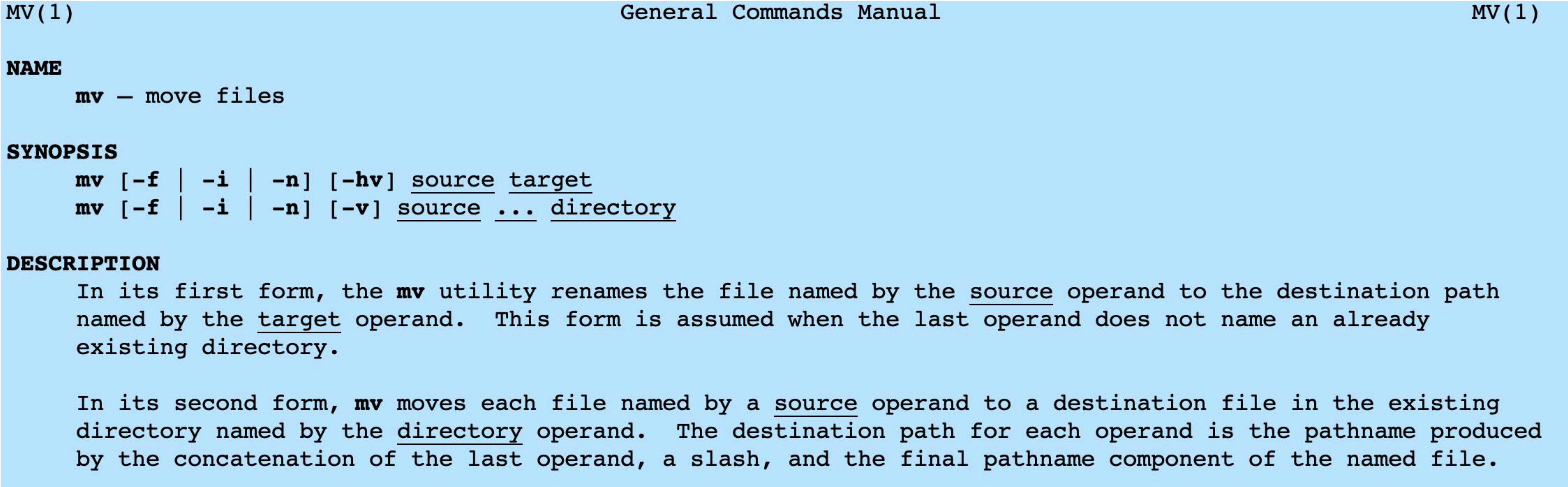
```
RM(1)                          General Commands Manual                          RM(1)

NAME
     rm, unlink – remove directory entries

SYNOPSIS
     rm [-f | -i] [-dIRrvWx] file ...
     unlink [--] file

DESCRIPTION
     The rm utility attempts to remove the non-directory type files specified on the command line.
     If the permissions of the file do not permit writing, and the standard input device is a
     terminal, the user is prompted (on the standard error output) for confirmation.

     The options are as follows:

     -d      Attempt to remove directories as well as other types of files.

     -f      Attempt to remove the files without prompting for confirmation, regardless of the
             file's permissions.  If the file does not exist, do not display a diagnostic message
             or modify the exit status to reflect an error.  The -f option overrides any previous
             -i options.

     -i      Request confirmation before attempting to remove each file, regardless of the file's
             permissions, or whether or not the standard input device is a terminal.  The -i
             option overrides any previous -f options.

     -I      Request confirmation once if more than three files are being removed or if a
             directory is being recursively removed.  This is a far less intrusive option than -i
             yet provides almost the same level of protection against mistakes.

     -R      Attempt to remove the file hierarchy rooted in each file argument.  The -R option
             implies the -d option.  If the -i option is specified, the user is prompted for
             confirmation before each directory's contents are processed (as well as before the
             attempt is made to remove the directory).  If the user does not respond
             affirmatively, the file hierarchy rooted in that directory is skipped.

     -r      Equivalent to -R.
```

rmdir removes an empty directory

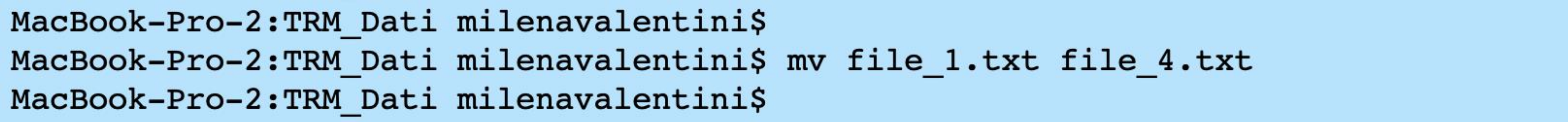
rm -rf forces to recursively delete a non empty directory

How to manipulate files

The **mv** command moves files

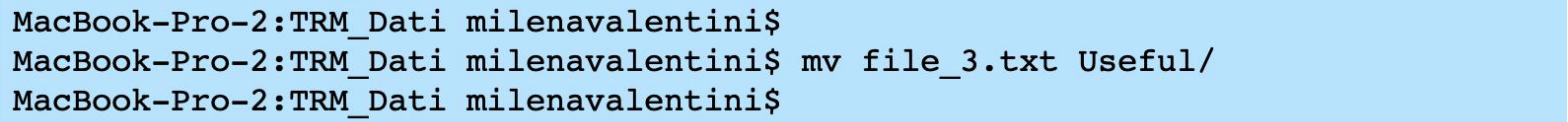


```
MV(1)                        General Commands Manual                       MV(1)

NAME
     mv – move files

SYNOPSIS
     mv [-f | -i | -n] [-hv] source target
     mv [-f | -i | -n] [-v] source ... directory

DESCRIPTION
     In its first form, the mv utility renames the file named by the source operand to the destination path
     named by the target operand.  This form is assumed when the last operand does not name an already
     existing directory.

     In its second form, mv moves each file named by a source operand to a destination file in the existing
     directory named by the directory operand.  The destination path for each operand is the pathname produced
     by the concatenation of the last operand, a slash, and the final pathname component of the named file.
```



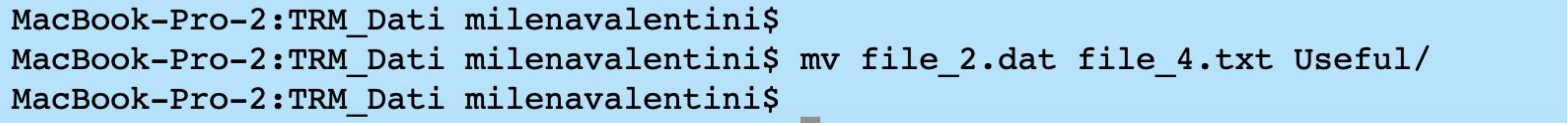
```
MacBook-Pro-2:TRM_Dati milenavalentini$
MacBook-Pro-2:TRM_Dati milenavalentini$ mv file_1.txt file_4.txt
MacBook-Pro-2:TRM_Dati milenavalentini$
```

equivalent to: cp file_1.txt file_4.txt; rm file_1.txt



```
MacBook-Pro-2:TRM_Dati milenavalentini$
MacBook-Pro-2:TRM_Dati milenavalentini$ mv file_3.txt Useful/
MacBook-Pro-2:TRM_Dati milenavalentini$
```

1st form of the manual



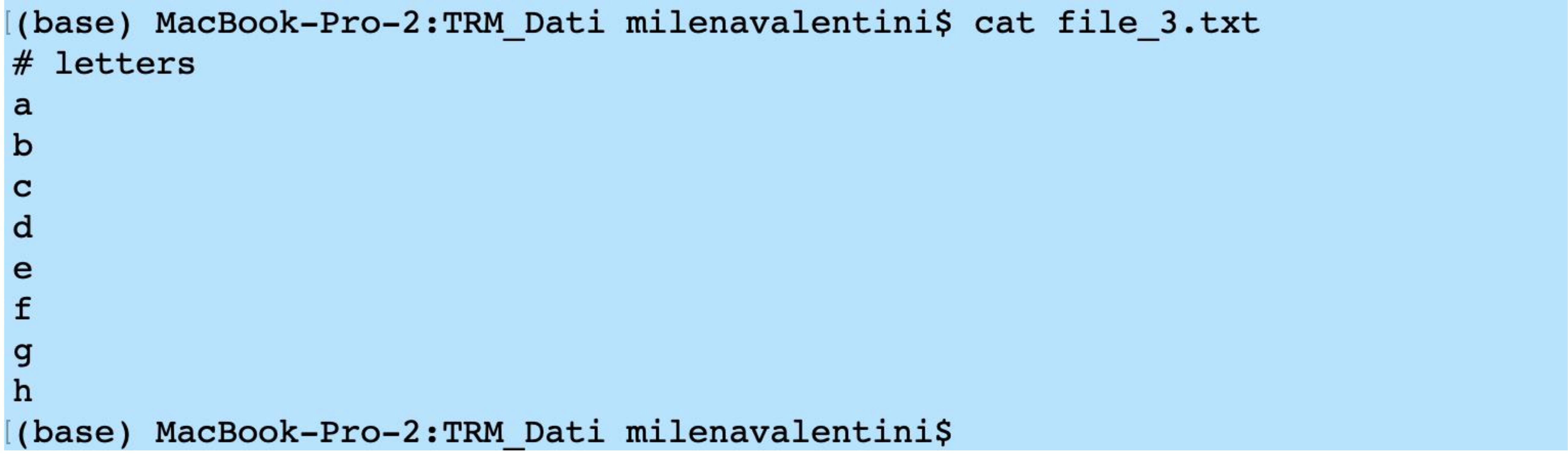
```
MacBook-Pro-2:TRM_Dati milenavalentini$
MacBook-Pro-2:TRM_Dati milenavalentini$ mv file_2.dat file_4.txt Useful/
MacBook-Pro-2:TRM_Dati milenavalentini$
```

2nd form of the manual

How to manipulate files

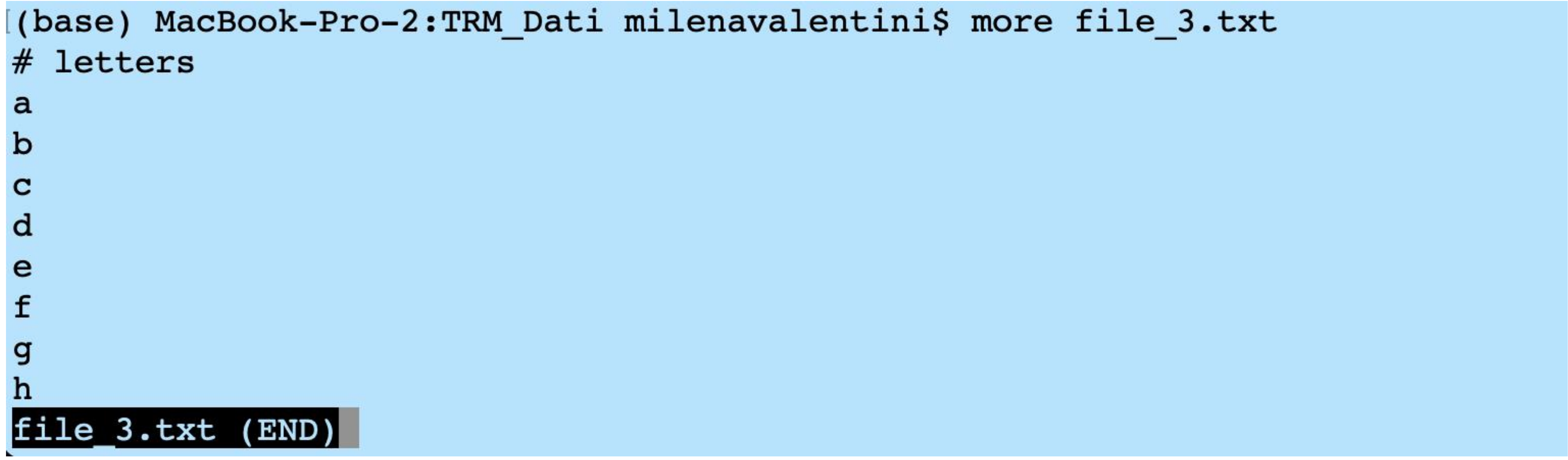
The **cat** command is used to display a text file or to concatenate multiple files into a single file.

By default, the **cat** command prints outputs to the standard output.



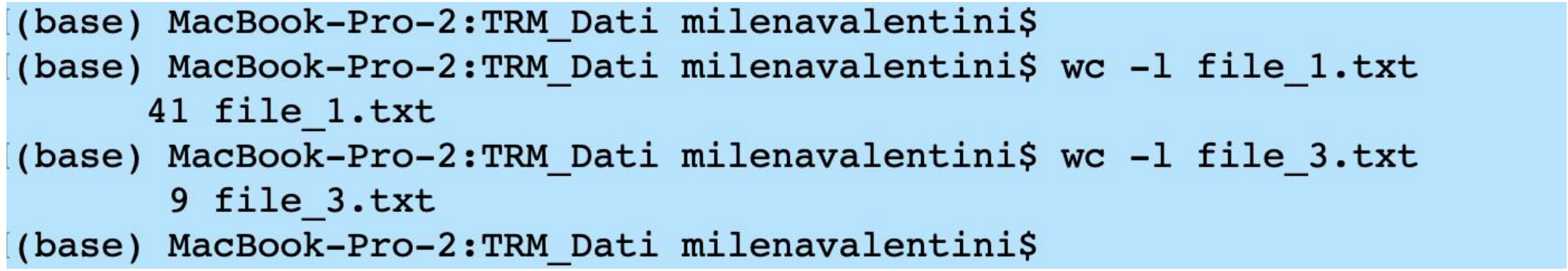
```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cat file_3.txt
# letters
a
b
c
d
e
f
g
h
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

To simply inspect the file content, you can also use to command **more**



```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ more file_3.txt
# letters
a
b
c
d
e
f
g
h
file_3.txt (END)
```

If you want to know how many lines a file is made of, you can use to command **wc -l** (it counts lines or words)

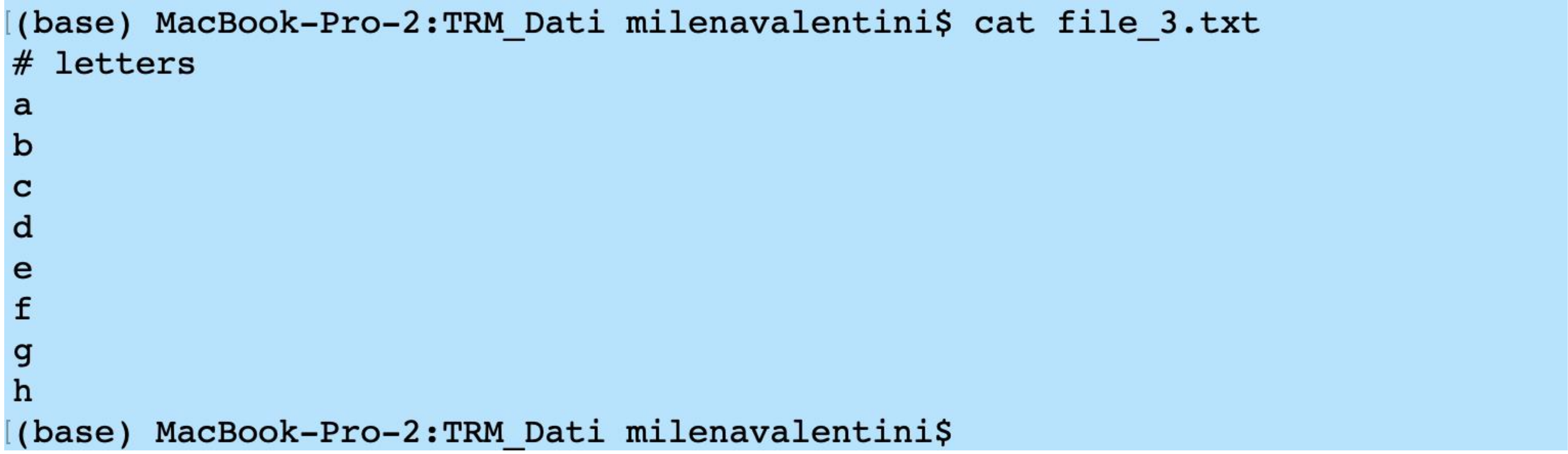


```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ wc -l file_1.txt
      41 file_1.txt
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ wc -l file_3.txt
       9 file_3.txt
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

How to manipulate files

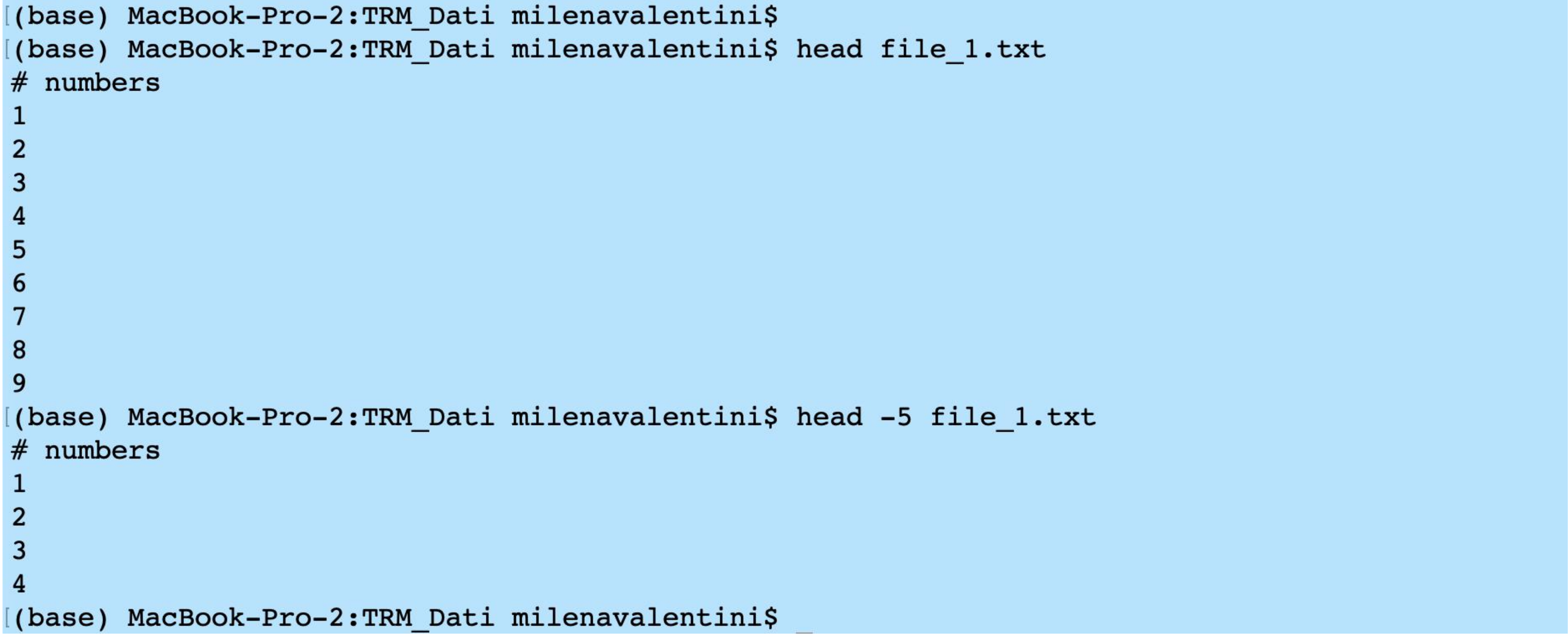
The **cat** command is used to display a text file or to concatenate multiple files into a single file.

By default, the **cat** command prints outputs to the standard output.



```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cat file_3.txt
# letters
a
b
c
d
e
f
g
h
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

To see the first part of a file, or its first n lines, use the **head** command:

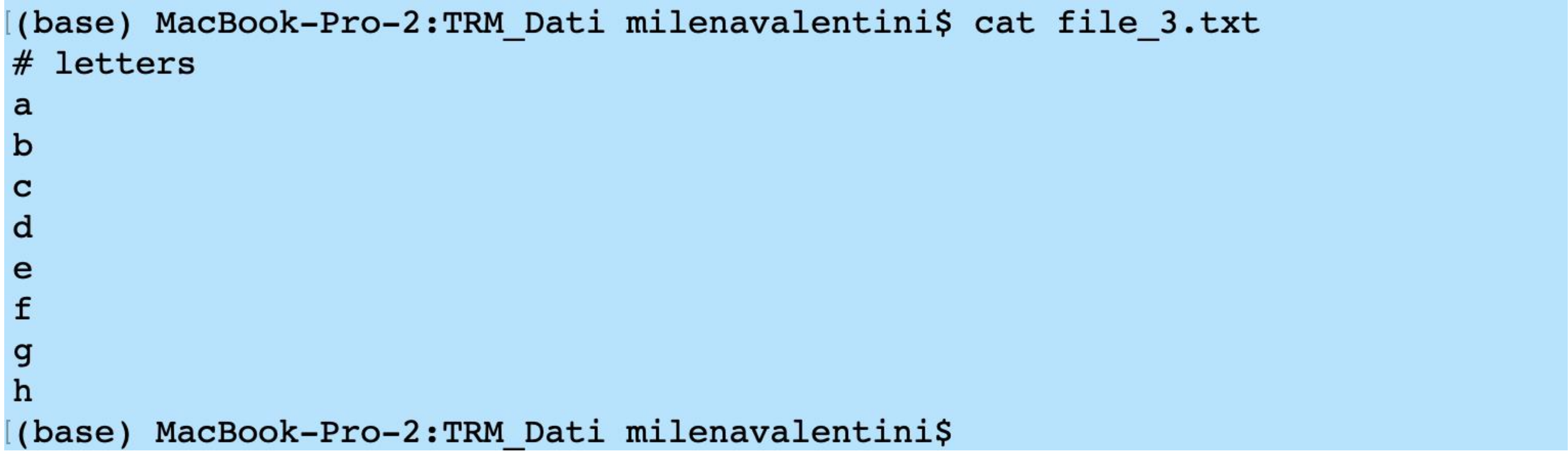


```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ head file_1.txt
# numbers
1
2
3
4
5
6
7
8
9
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ head -5 file_1.txt
# numbers
1
2
3
4
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ _
```

How to manipulate files

The **cat** command is used to display a text file or to concatenate multiple files into a single file.

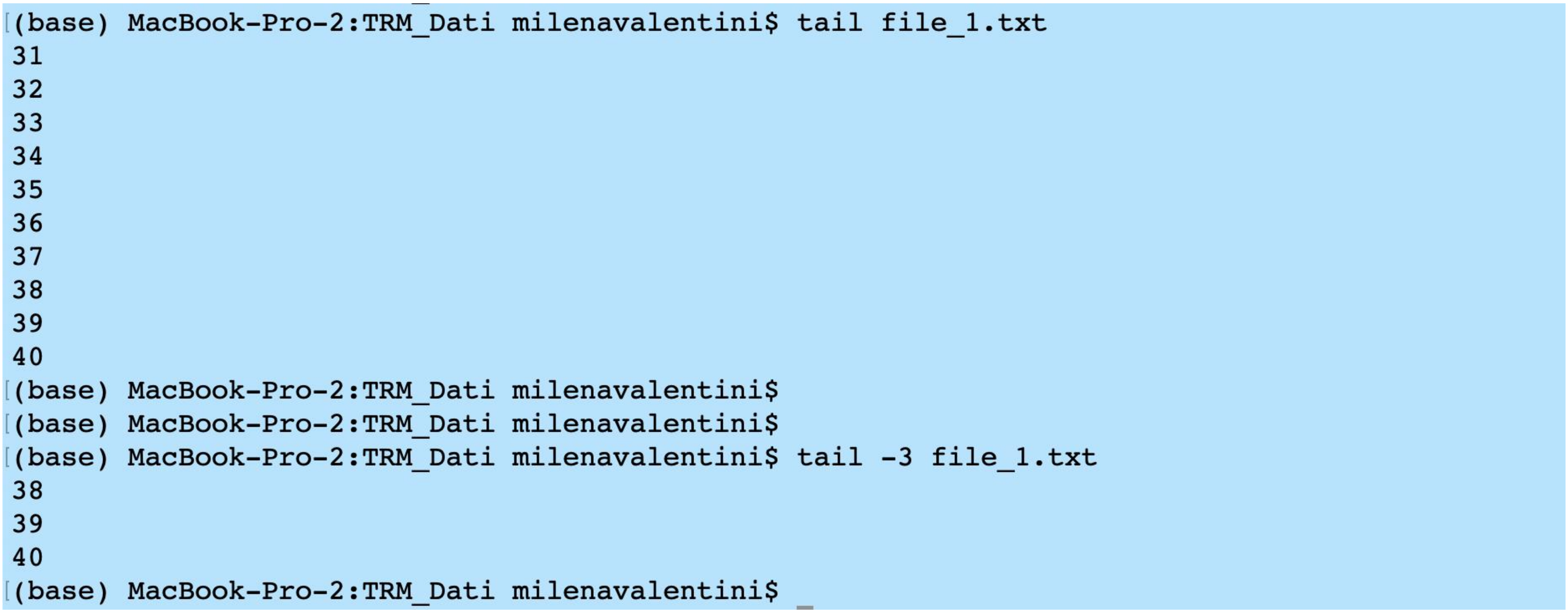
By default, the **cat** command prints outputs to the standard output.



```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cat file_3.txt
# letters
a
b
c
d
e
f
g
h
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

To see the first part of a file, or its first n lines, use the **head** command:

To access the bottom part of a file, or its last n lines, use the **tail** command:

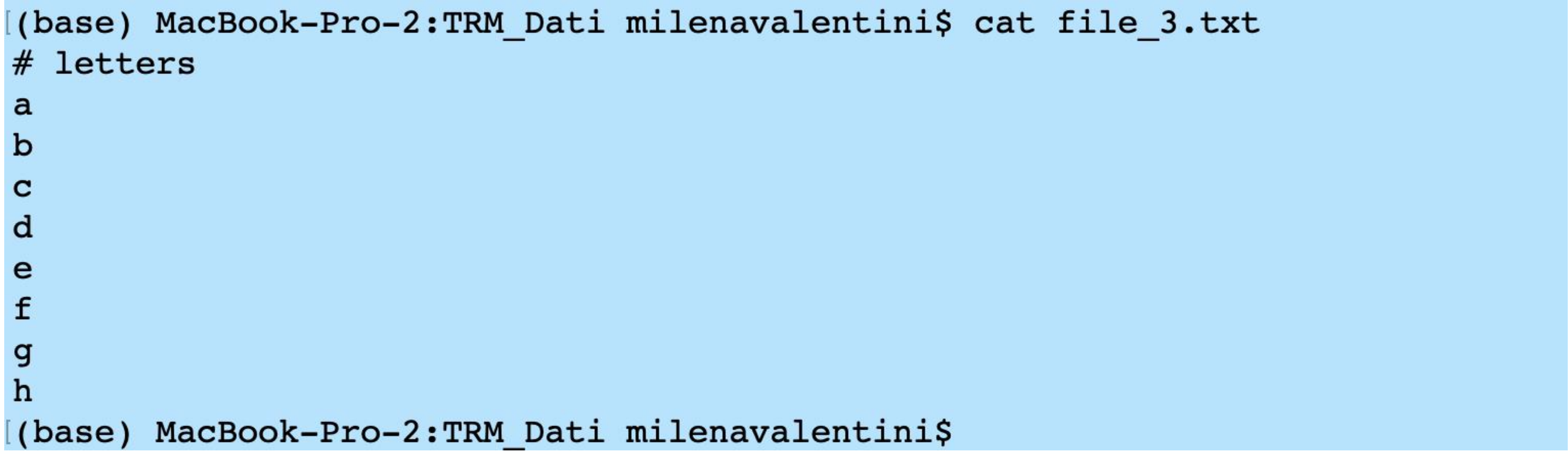


```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ tail file_1.txt
31
32
33
34
35
36
37
38
39
40
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ tail -3 file_1.txt
38
39
40
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ _
```

How to manipulate files

The **cat** command is used to display a text file or to concatenate multiple files into a single file.

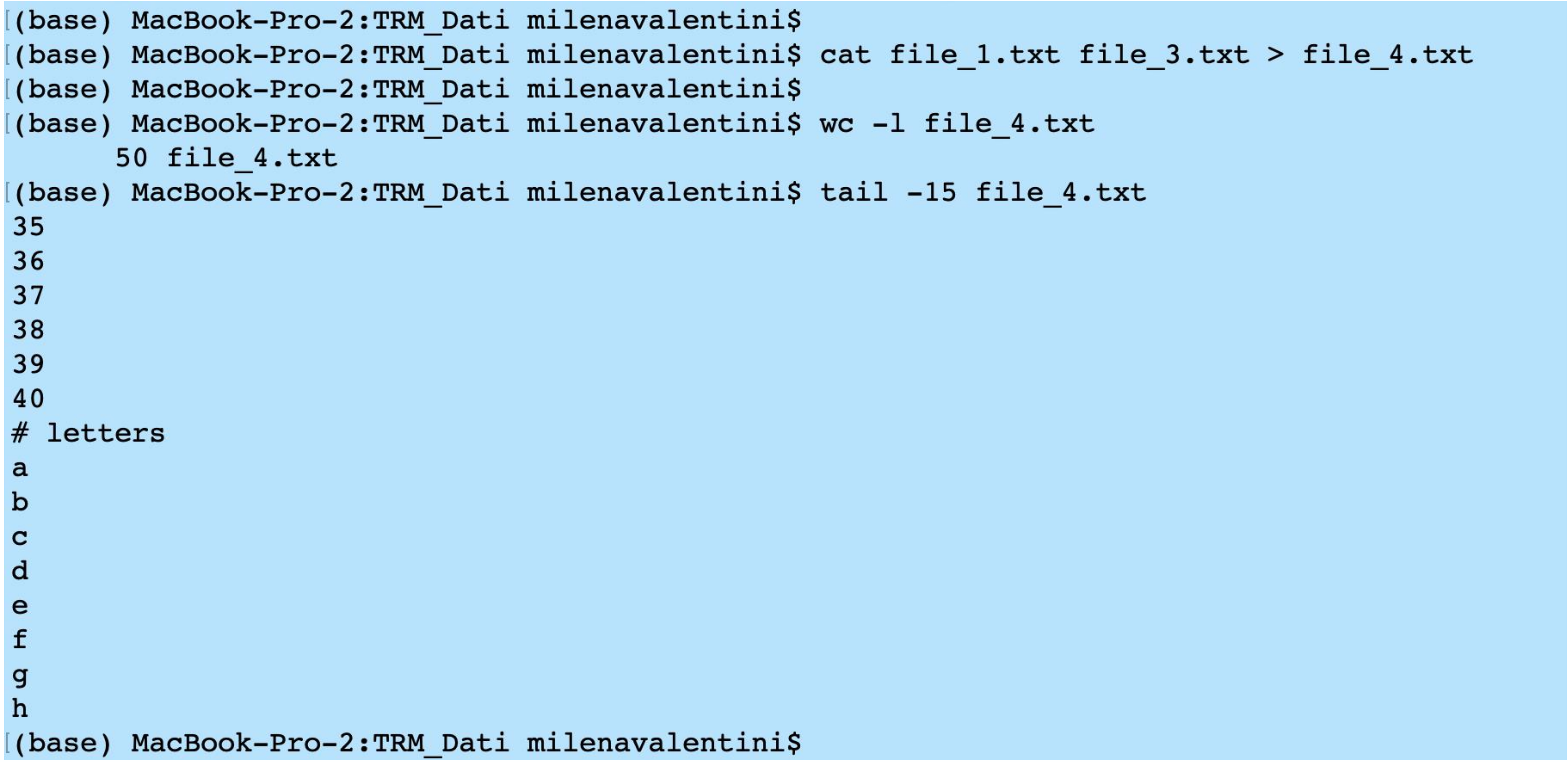
By default, the **cat** command prints outputs to the standard output.



```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cat file_3.txt
# letters
a
b
c
d
e
f
g
h
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

The **cat** command takes in one or more filenames as its arguments.
The files are concatenated
in the order they appear in the argument list.

As for almost every command, the **cat** command generates the output to standard output,
which can be redirected to a file (using the UNIX direction operator > ; use >> to append)

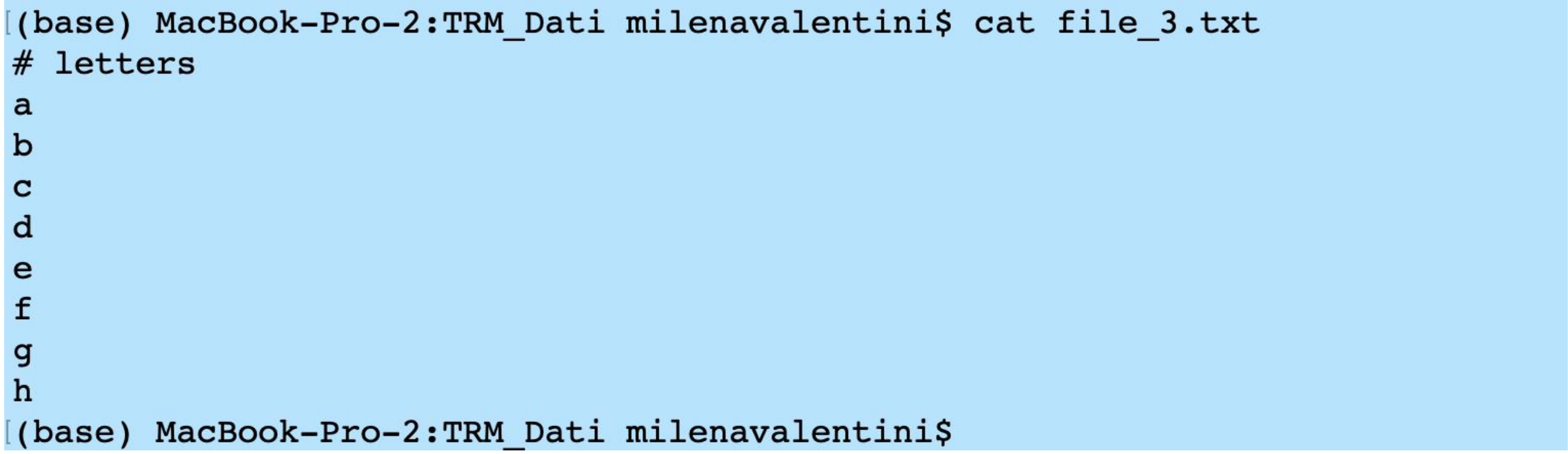


```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cat file_1.txt file_3.txt > file_4.txt
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ wc -l file_4.txt
      50 file_4.txt
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ tail -15 file_4.txt
35
36
37
38
39
40
# letters
a
b
c
d
e
f
g
h
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

How to manipulate files

The **cat** command is used to display a text file or to concatenate multiple files into a single file.

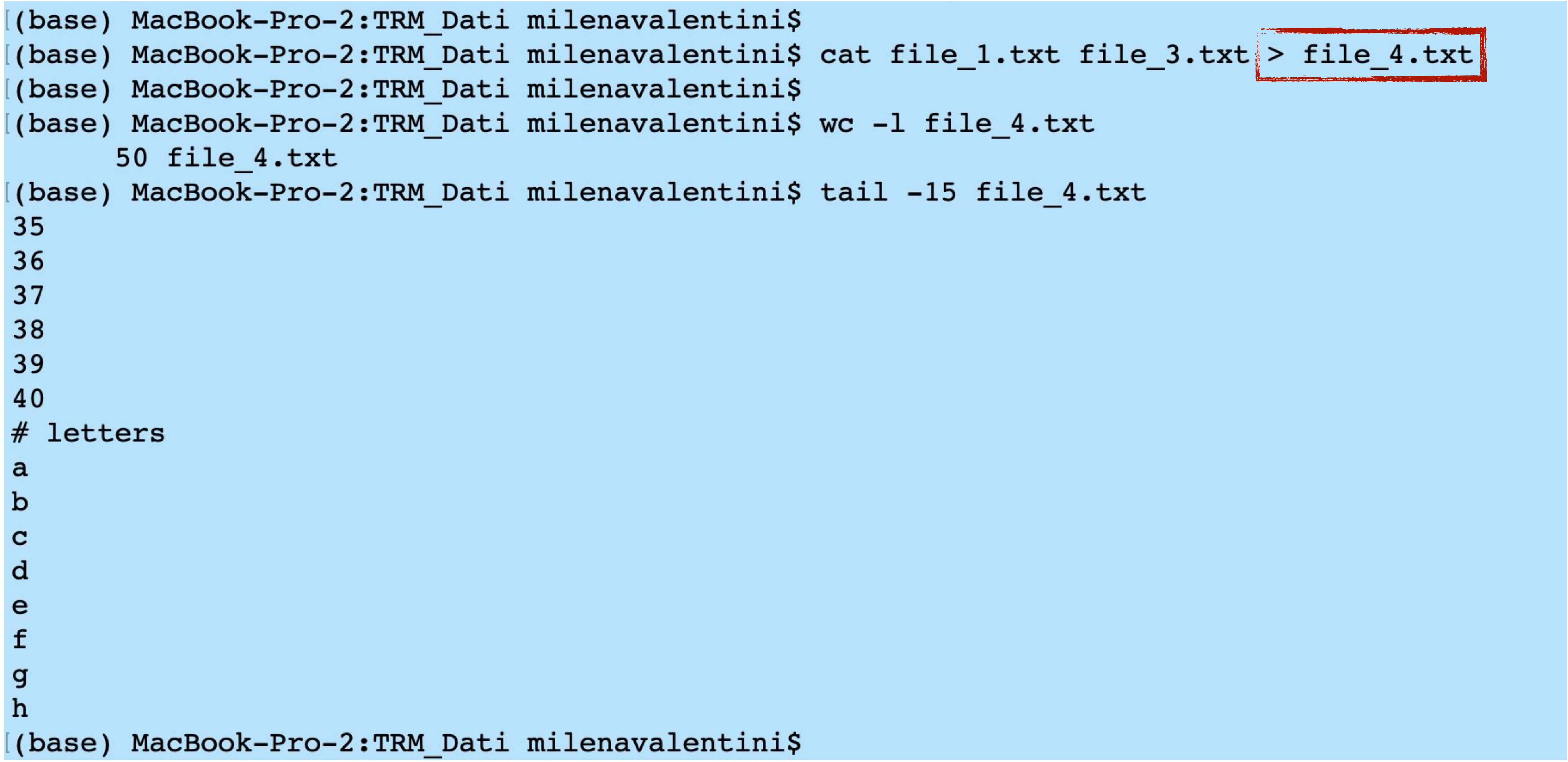
By default, the **cat** command prints outputs to the standard output.



```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cat file_3.txt
# letters
a
b
c
d
e
f
g
h
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

The **cat** command takes in one or more filenames as its arguments.
The files are concatenated
in the order they appear in the argument list.

As for almost every command, the **cat** command generates the output to standard output, which can be redirected to a file (using the UNIX direction operator **>**; use **>>** to append)



```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cat file_1.txt file_3.txt > file_4.txt
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ wc -l file_4.txt
      50 file_4.txt
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ tail -15 file_4.txt
35
36
37
38
39
40
# letters
a
b
c
d
e
f
g
h
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

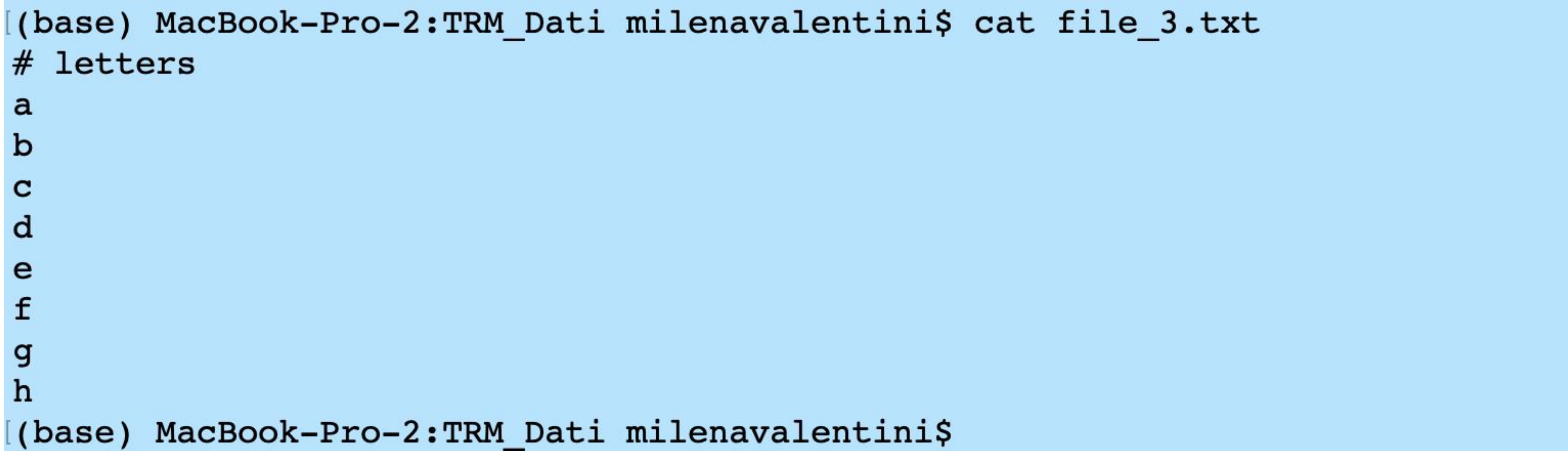
How to manipulate files

The **cat** command is used to display a text file or to concatenate multiple files into a single file.

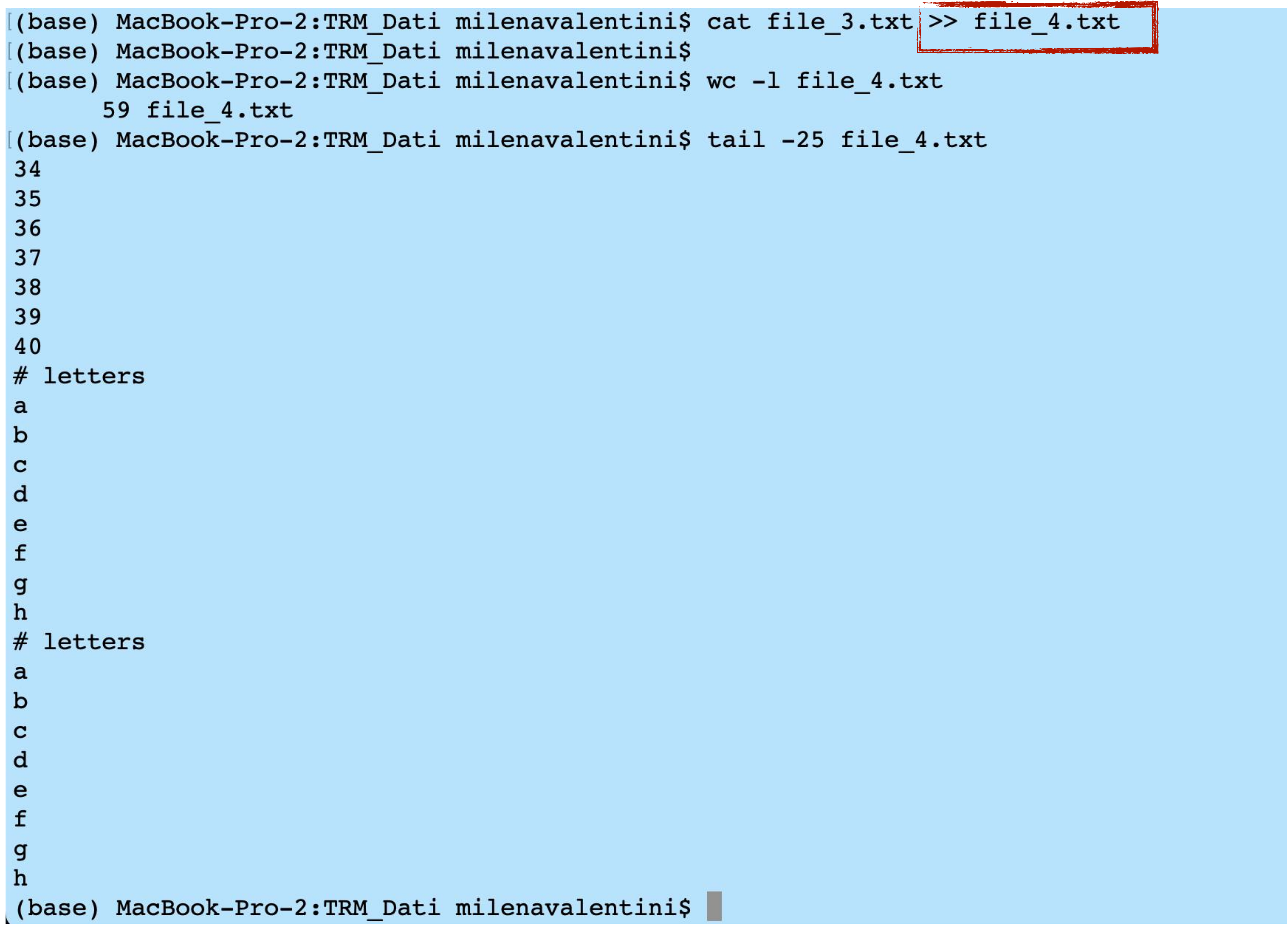
By default, the **cat** command prints outputs to the standard output.

The **cat** command takes in one or more filenames as its arguments.
The files are concatenated
in the order they appear in the argument list.

As for almost every command, the **cat** command generates the output to standard output, which can be redirected to a file (using the UNIX direction operator **>** ; use **>>** to append)



```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cat file_3.txt
# letters
a
b
c
d
e
f
g
h
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```

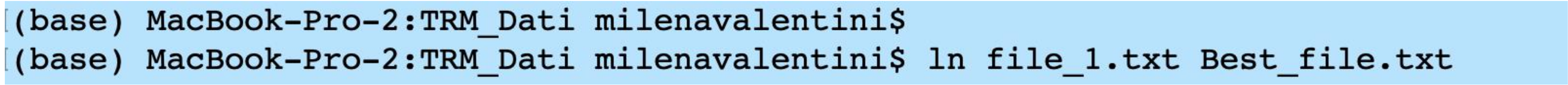


```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cat file_3.txt >> file_4.txt
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ wc -l file_4.txt
      59 file_4.txt
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ tail -25 file_4.txt
34
35
36
37
38
39
40
# letters
a
b
c
d
e
f
g
h
# letters
a
b
c
d
e
f
g
h
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ 
```

How to manipulate files

The **ln** command provides a given file with an alternative name.

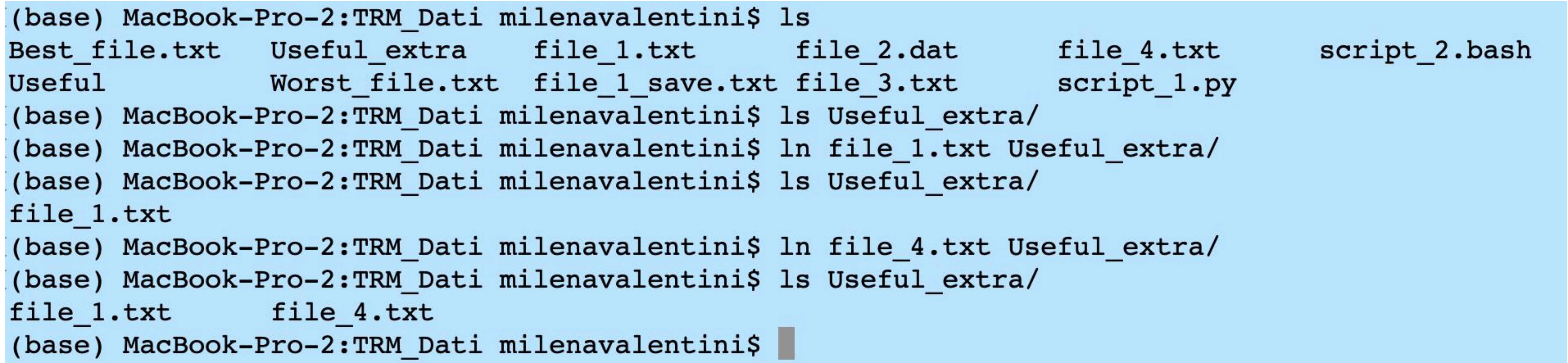
It links a file name to another one.
It is possible to link a file to another in the same directory or even to the same name in another directory.



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ln file_1.txt Best_file.txt
```

When linking a filename to another filename, only two arguments can be specified:
the source filename and the target filename.

When linking a filename to a directory, you can specify multiple filenames to be linked to the same directory.

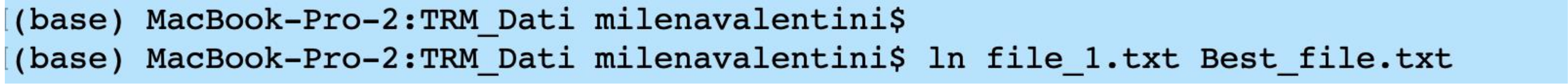


```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls
Best_file.txt   Useful_extra    file_1.txt      file_2.dat      file_4.txt      script_2.bash
Useful          Worst_file.txt  file_1_save.txt file_3.txt      script_1.py
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls Useful_extra/
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ln file_1.txt Useful_extra/
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls Useful_extra/
file_1.txt
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ln file_4.txt Useful_extra/
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ls Useful_extra/
file_1.txt      file_4.txt
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ 
```

How to manipulate files

The **ln** command provides a given file with an alternative name.

Link two files in the current directory:



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ln file_1.txt Best_file.txt
```

The flags that can be used with the **ln** command are as follows:

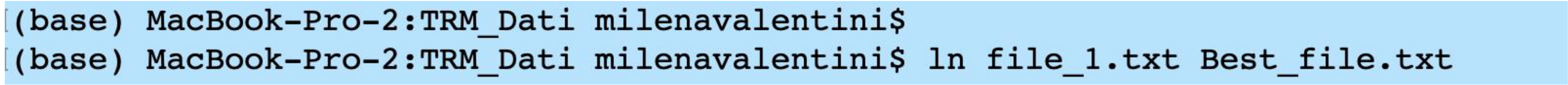
-s to create a soft link to another file or directory.
In a soft link, the linked file contains the name of the original file.
When an operation on the linked filename is done, the name of the original file in the link is used to reference the original file.

-f to ensure that the destination filename is replaced by the linked filename if the file already exists.

How to manipulate files

The **ln** command provides a given file with an alternative name.

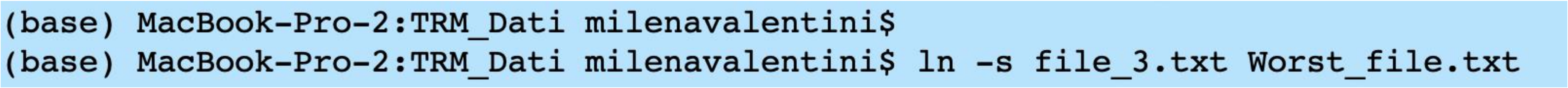
Link two files in the current directory:



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ln file_1.txt Best_file.txt
```

It links to file_1.txt
If one of the files is removed, the other
will not undergo changes.

To create a symbolic link of the first argument in the current directory:

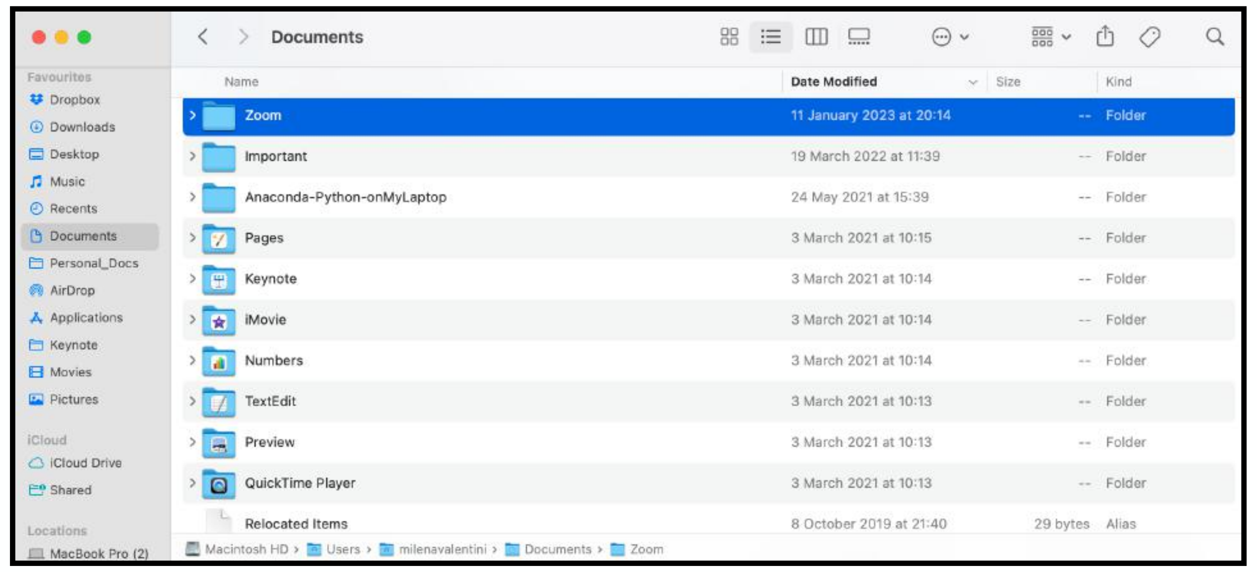


```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ ln -s file_3.txt Worst_file.txt
```

This linked file
only contains the name of file_3.txt
If you remove file_3.txt, you will be left with an orphan Worst_file.txt,
which points to nowhere.

Useful working tools

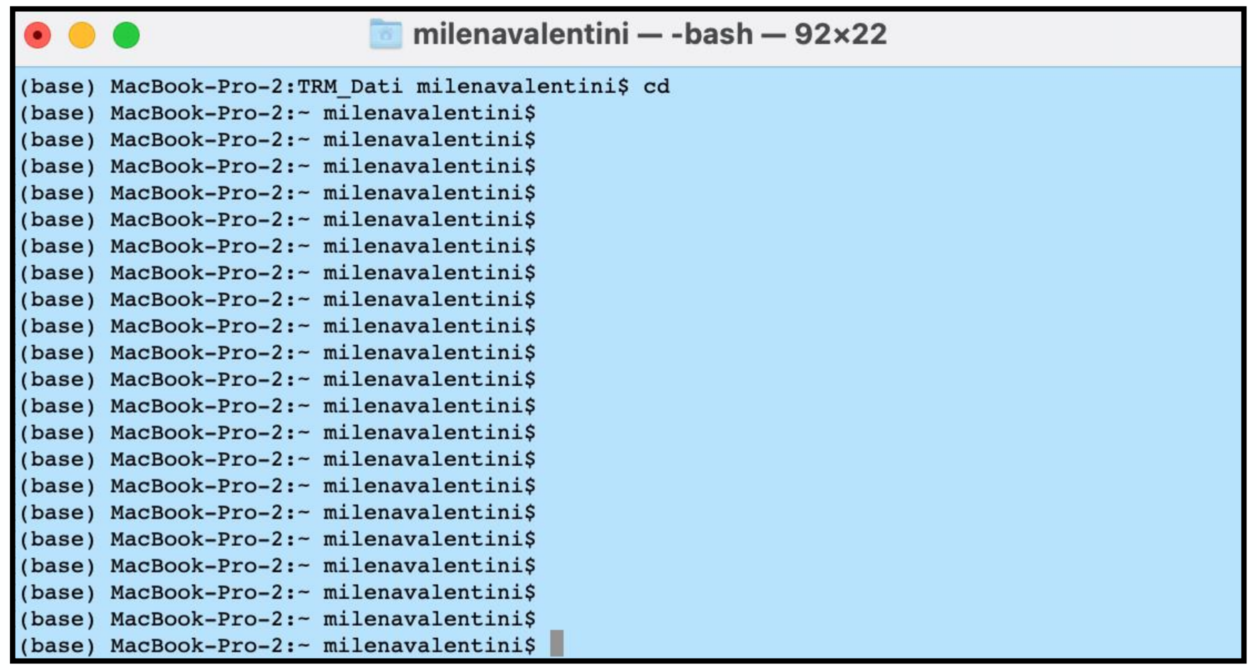
File browser or manager:
program of an Operative System (OS) which provides you with a user interface to manage folders and files



Name	Date Modified	Size	Kind
Zoom	11 January 2023 at 20:14	--	Folder
Important	19 March 2022 at 11:39	--	Folder
Anaconda-Python-onMyLaptop	24 May 2021 at 15:39	--	Folder
Pages	3 March 2021 at 10:15	--	Folder
Keynote	3 March 2021 at 10:14	--	Folder
iMovie	3 March 2021 at 10:14	--	Folder
Numbers	3 March 2021 at 10:14	--	Folder
TextEdit	3 March 2021 at 10:13	--	Folder
Preview	3 March 2021 at 10:13	--	Folder
QuickTime Player	3 March 2021 at 10:13	--	Folder
Relocated Items	8 October 2019 at 21:40	29 bytes	Alias

Macintosh HD › Users › milenavalentini › Documents › Zoom

Shell/terminal/console/command line prompt:
interface to interact with the computer
via command line
without relying on graphical unit interfaces



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ cd
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$
```

Text / code editor:
tool to edit code and file content

The text editor

vi: Visual Editor is the default editor that comes with the UNIX OS.

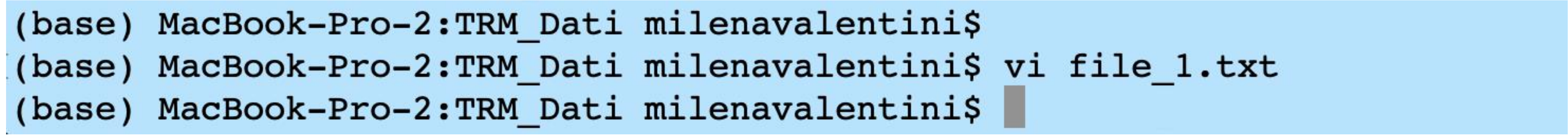
The vi editor is a full screen editor and has two modes of operation:

1. — *Command mode:* commands produce actions to be taken on the file, and
2. — *Insert mode:* entered text is written into the file.

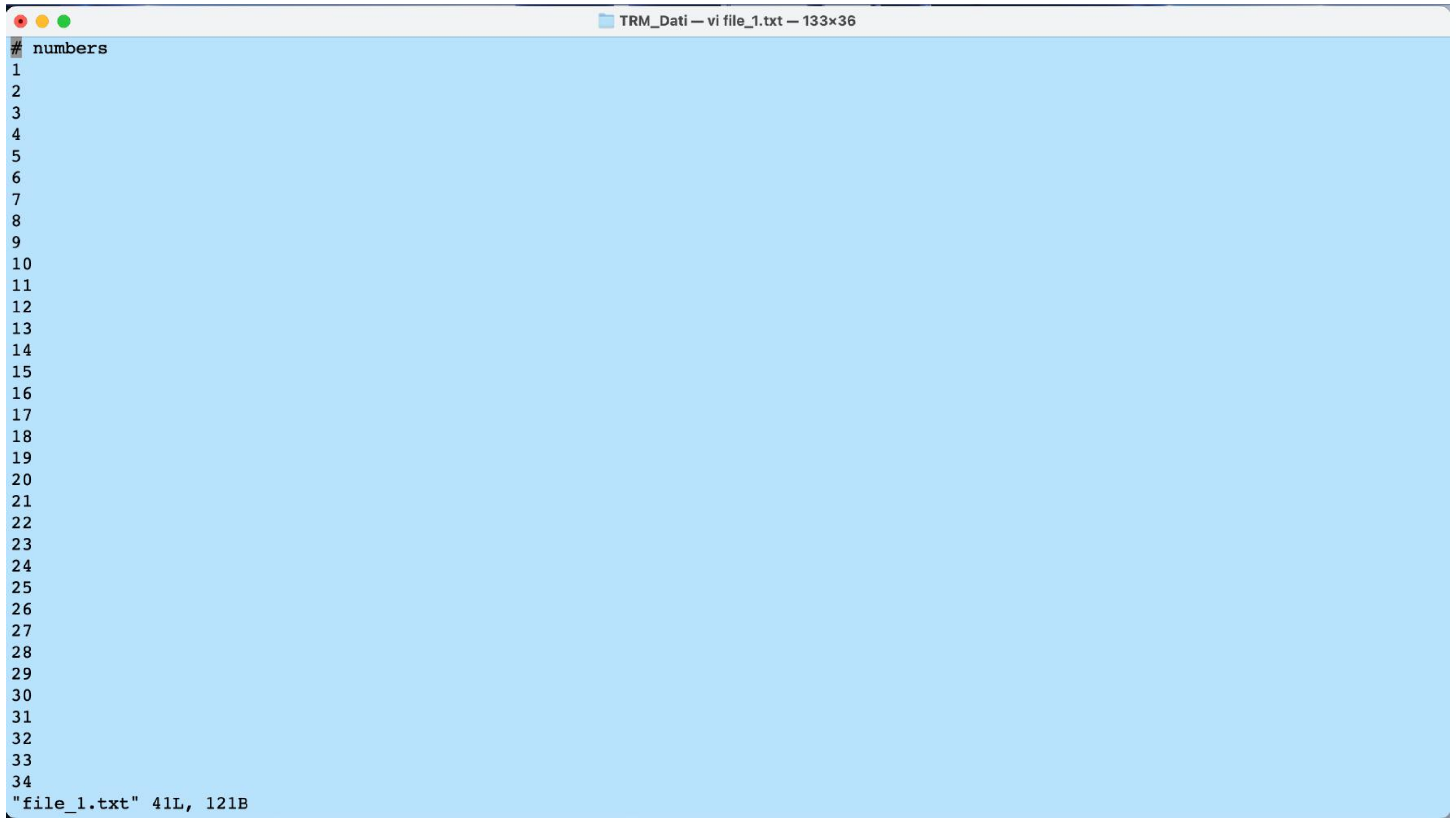
In the command mode, every character typed is a command;
the *i* character typed in the command mode makes the vi editor enter the insert mode.

In the insert mode, typed characters are added to the text in the file.
Press the escape key to exit the insert mode.

Several websites where useful manuals can be used, e.g.:
https://www.cs.colostate.edu/helpdocs/vi.html
https://vimdoc.sourceforge.net/htmldoc/usr_toc.html (vim)



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ vi file_1.txt
(base) MacBook-Pro-2:TRM_Dati milenavalentini$
```



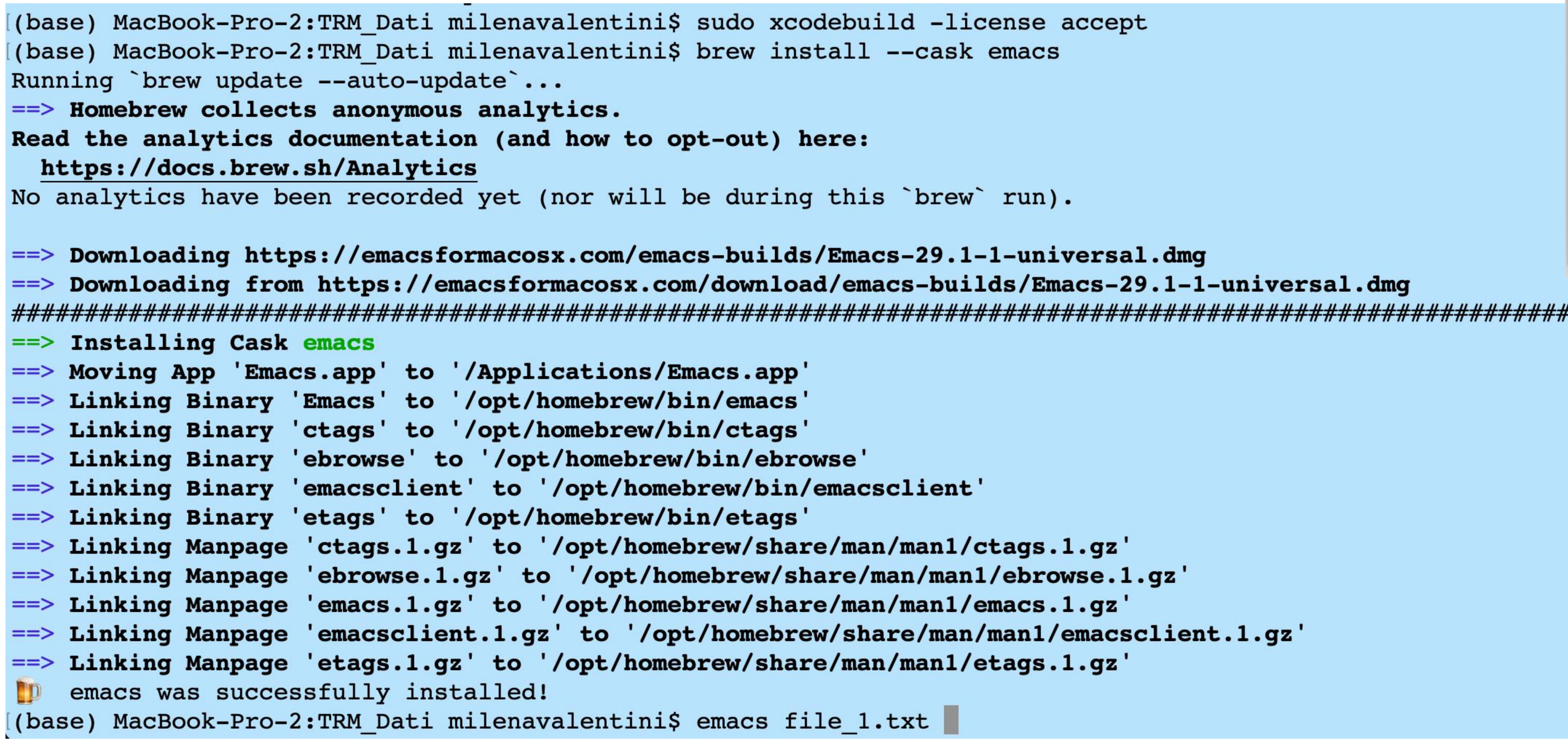
```
# numbers
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
"file_1.txt" 41L, 121B
```

The text editor

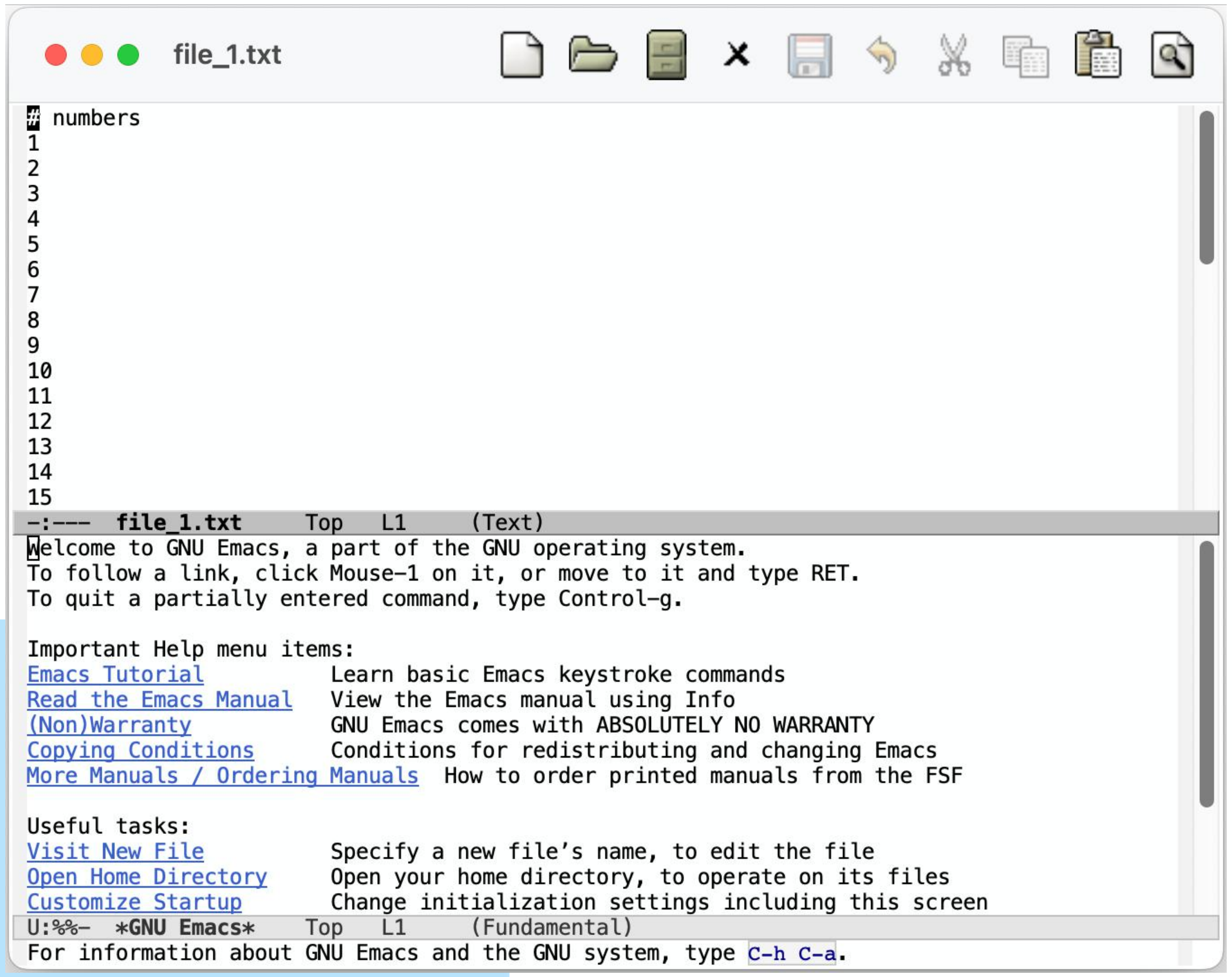
Emacs: it is the advanced, extensible, customizable, self-documenting editor by GNU.

You can follow the instructions to download and install it here: https://www.gnu.org/software/emacs/download.html

For instance, for users with a Mac OS:



```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ sudo xcodebuild -license accept
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ brew install --cask emacs
Running `brew update --auto-update`...
==> Homebrew collects anonymous analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics have been recorded yet (nor will be during this `brew` run).

==> Downloading https://emacsformacosx.com/emacs-builds/Emacs-29.1-1-universal.dmg
==> Downloading from https://emacsformacosx.com/download/emacs-builds/Emacs-29.1-1-universal.dmg
############################################################################################################################ 100.0%
==> Installing Cask emacs
==> Moving App 'Emacs.app' to '/Applications/Emacs.app'
==> Linking Binary 'Emacs' to '/opt/homebrew/bin/emacs'
==> Linking Binary 'ctags' to '/opt/homebrew/bin/ctags'
==> Linking Binary 'ebrowse' to '/opt/homebrew/bin/ebrowse'
==> Linking Binary 'emacsclient' to '/opt/homebrew/bin/emacsclient'
==> Linking Binary 'etags' to '/opt/homebrew/bin/etags'
==> Linking Manpage 'ctags.1.gz' to '/opt/homebrew/share/man/man1/ctags.1.gz'
==> Linking Manpage 'ebrowse.1.gz' to '/opt/homebrew/share/man/man1/ebrowse.1.gz'
==> Linking Manpage 'emacs.1.gz' to '/opt/homebrew/share/man/man1/emacs.1.gz'
==> Linking Manpage 'emacsclient.1.gz' to '/opt/homebrew/share/man/man1/emacsclient.1.gz'
==> Linking Manpage 'etags.1.gz' to '/opt/homebrew/share/man/man1/etags.1.gz'
🍺  emacs was successfully installed!
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ emacs file_1.txt
```



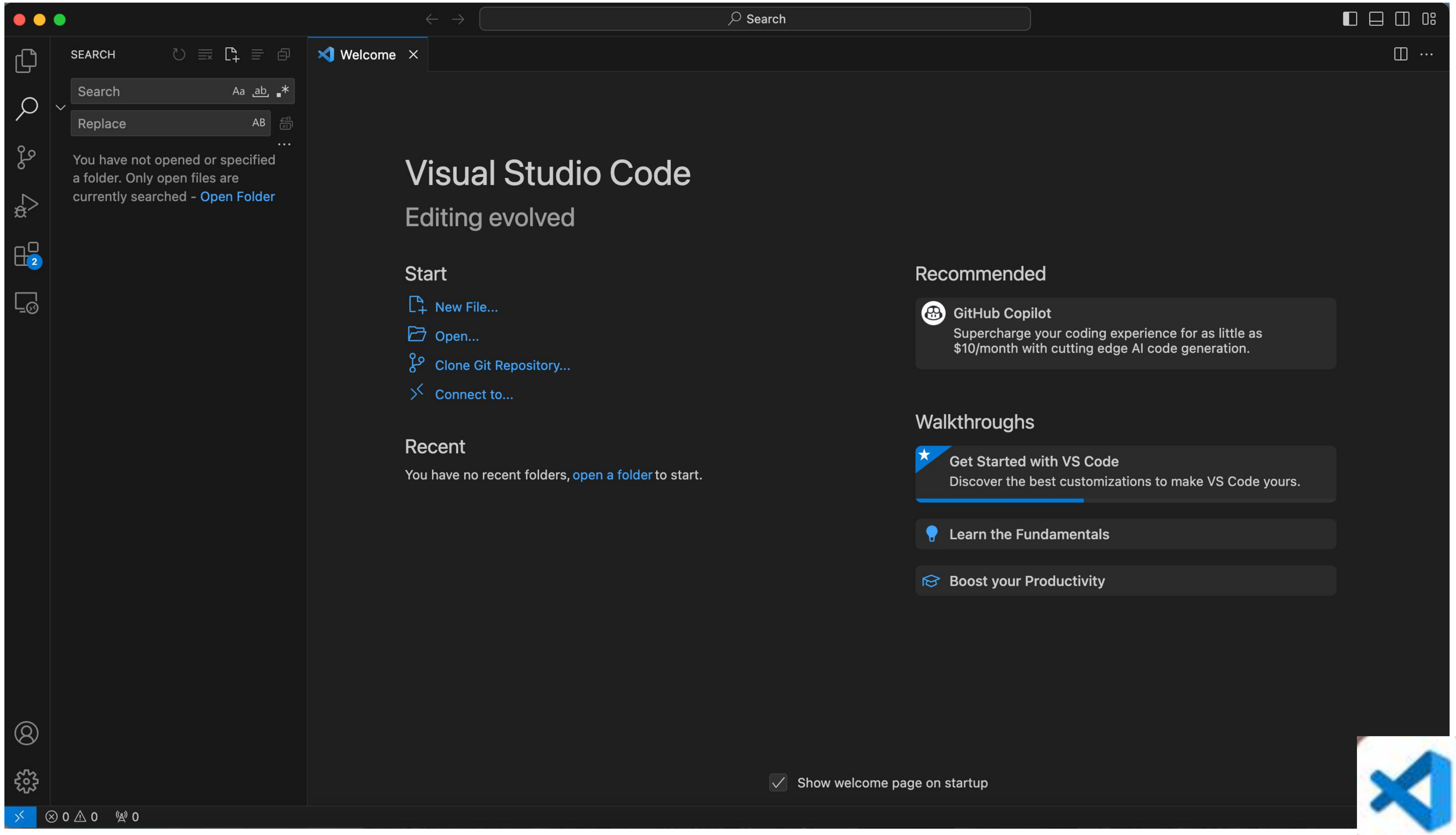
```
# numbers
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
-:---  file_1.txt     Top   L1     (Text)
Welcome to GNU Emacs, a part of the GNU operating system.
To follow a link, click Mouse-1 on it, or move to it and type RET.
To quit a partially entered command, type Control-g.

Important Help menu items:
Emacs Tutorial          Learn basic Emacs keystroke commands
Read the Emacs Manual   View the Emacs manual using Info
(Non)Warranty           GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions      Conditions for redistributing and changing Emacs
More Manuals / Ordering Manuals  How to order printed manuals from the FSF

Useful tasks:
Visit New File          Specify a new file's name, to edit the file
Open Home Directory     Open your home directory, to operate on its files
Customize Startup       Change initialization settings including this screen
U:%%-  *GNU Emacs*    Top   L1     (Fundamental)
For information about GNU Emacs and the GNU system, type C-h C-a.
```

How to use emacs: https://www.gnu.org/software/emacs/manual/html_node/emacs/index.html

The text editor

Visual Studio Code is a powerful source code editor which runs on your desktop.
It is available for Windows, macOS and Linux   https://code.visualstudio.com/docs/?dv=osx

It comes with built-in support for e.g. JavaScript and has several extensions for other languages and runtimes (such as C++, Java, Python…)



You have not opened or specified a folder. Only open files are currently searched - Open Folder

Visual Studio Code

Editing evolved

You have no recent folders, open a folder to start.

GitHub Copilot
Supercharge your coding experience for as little as $10/month with cutting edge AI code generation.

Get Started with VS Code
Discover the best customizations to make VS Code yours.

Learn the Fundamentals

Boost your Productivity

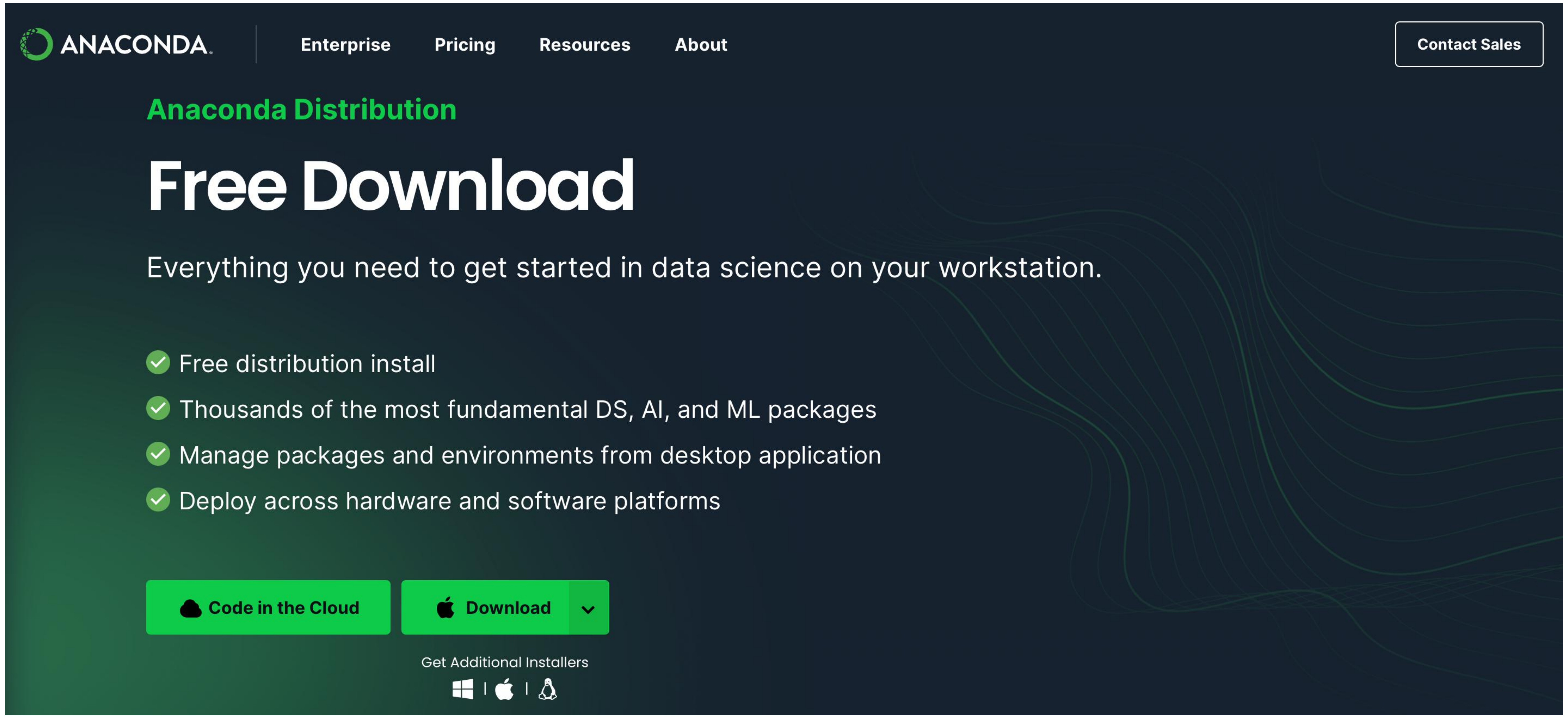
☑ Show welcome page on startup

Setting up the working environment

Anaconda is an open-source package and environment management system that runs on Windows, macOS, and Linux.

Conda quickly installs, runs, and updates packages and their dependencies.
It also easily creates, saves, loads, and switches between environments on your local computer.
It was created for Python programs, but it can package and distribute software for any language.

https://www.anaconda.com/download



Anaconda Distribution

Free Download

Everything you need to get started in data science on your workstation.

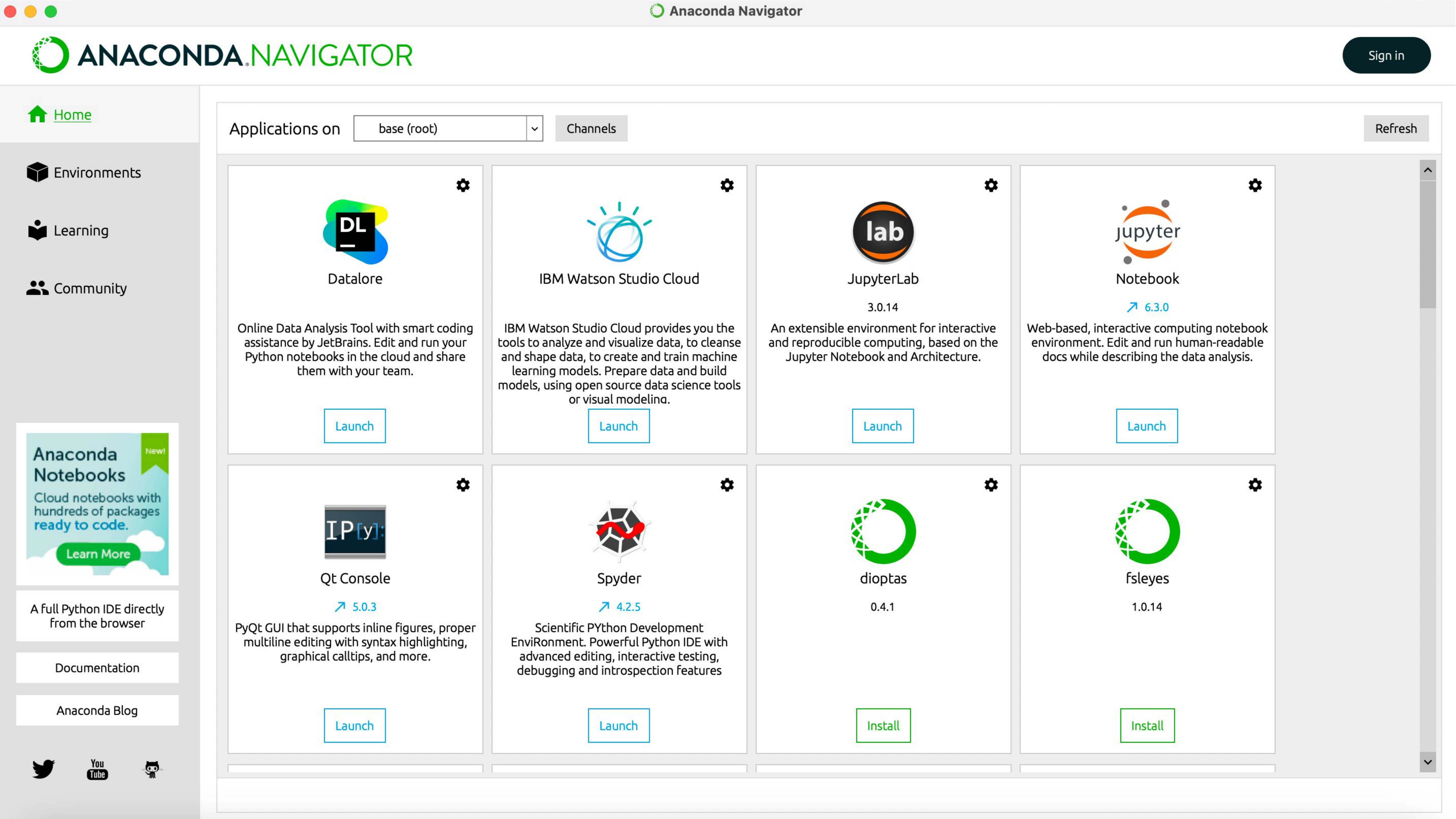
- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

Download, install it and make sure your $PATH environment variable is updated to include Anaconda

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:



A full Python IDE directly from the browser

Datalore: Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud: IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab 3.0.14: An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook 6.3.0: Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console 5.0.3: PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

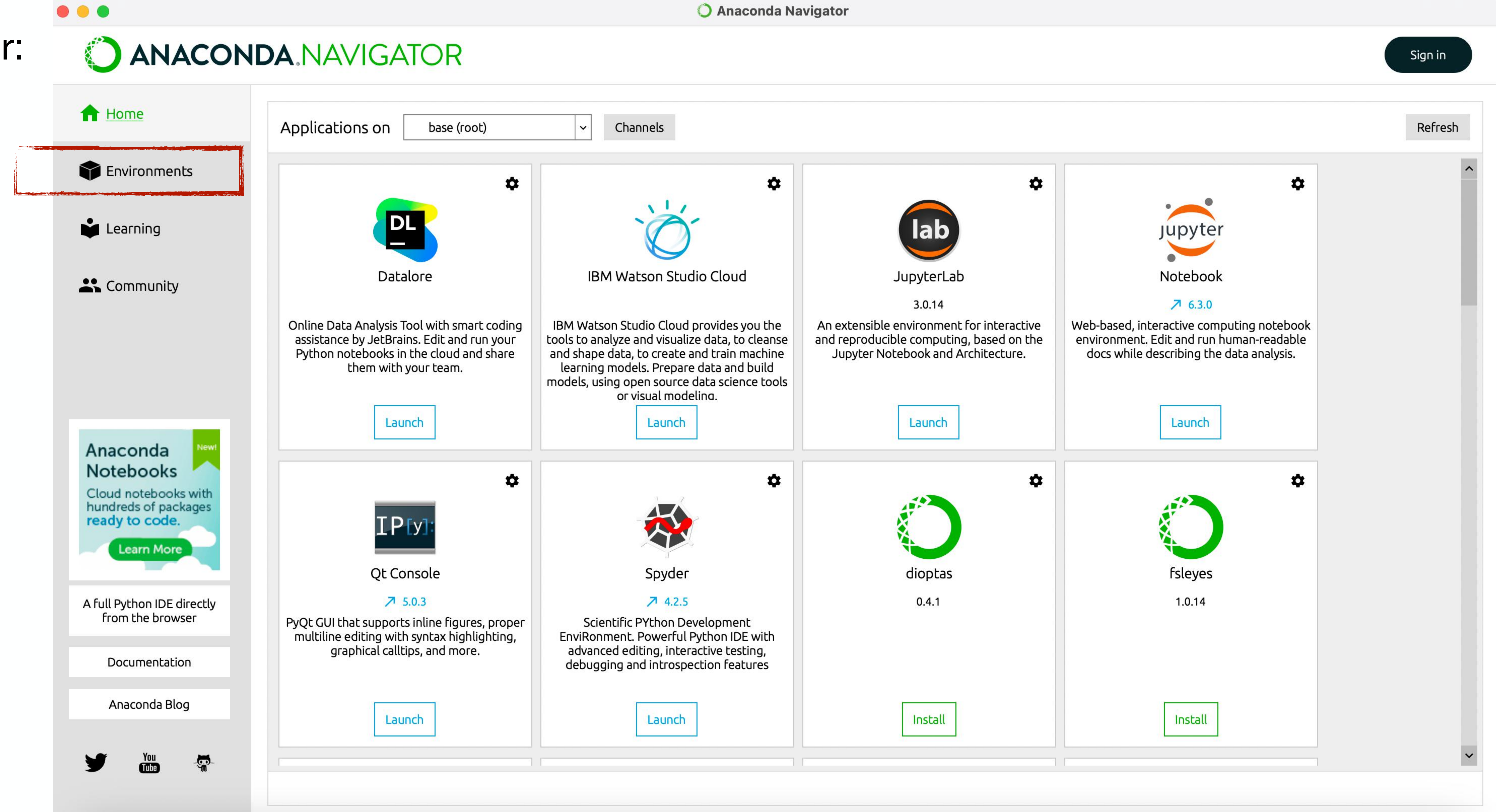
Spyder 4.2.5: Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

Select Environments:



Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Cloud notebooks with hundreds of packages ready to code.

A full Python IDE directly from the browser

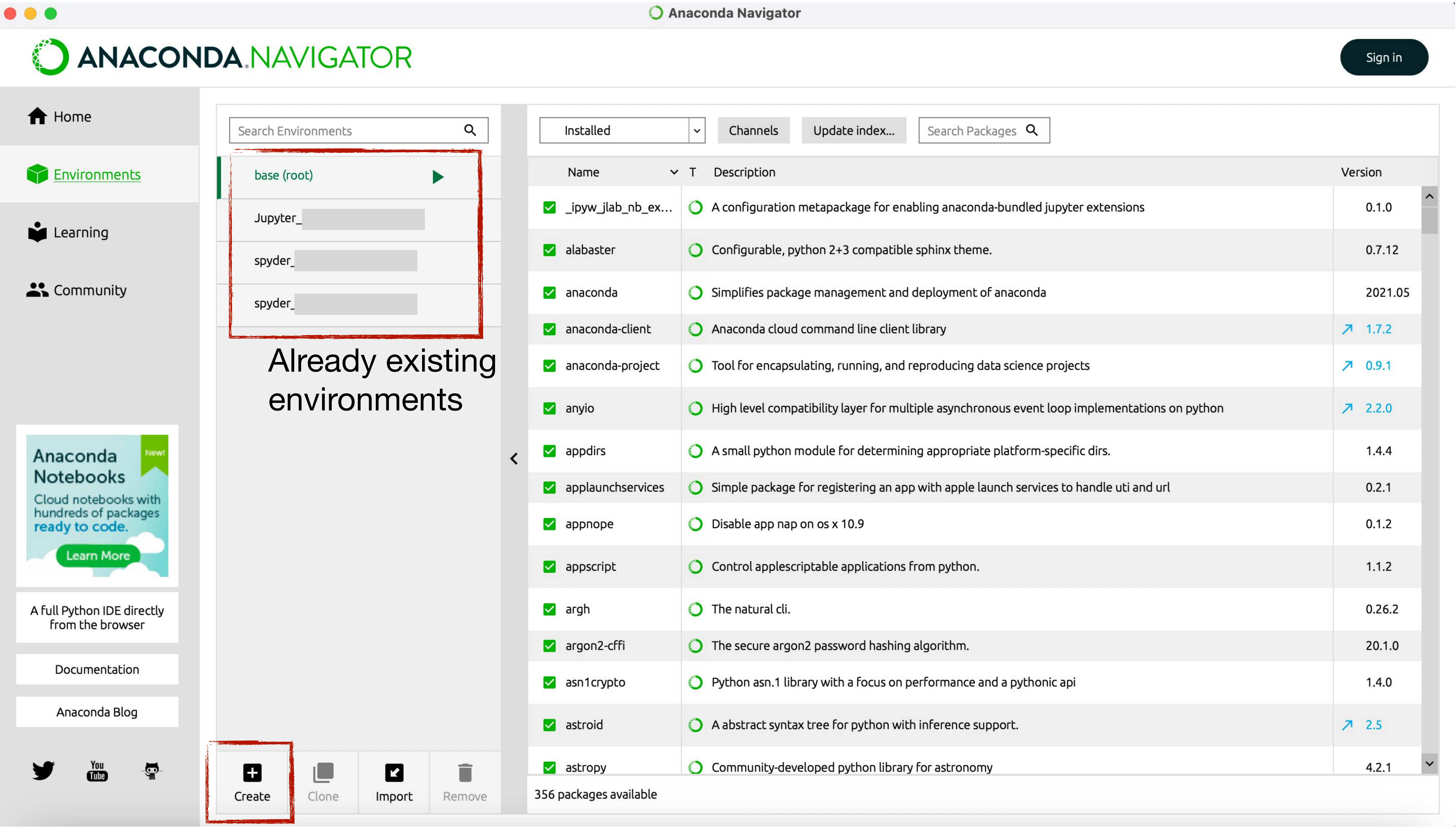
Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

Select Environments:

Create a new environment:



Already existing environments

Name	T	Description	Version
☑ _ipyw_jlab_nb_ex…	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
☑ alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
☑ anaconda	○	Simplifies package management and deployment of anaconda	2021.05
☑ anaconda-client	○	Anaconda cloud command line client library	↗ 1.7.2
☑ anaconda-project	○	Tool for encapsulating, running, and reproducing data science projects	↗ 0.9.1
☑ anyio	○	High level compatibility layer for multiple asynchronous event loop implementations on python	↗ 2.2.0
☑ appdirs	○	A small python module for determining appropriate platform-specific dirs.	1.4.4
☑ applaunchservices	○	Simple package for registering an app with apple launch services to handle uti and url	0.2.1
☑ appnope	○	Disable app nap on os x 10.9	0.1.2
☑ appscript	○	Control applescriptable applications from python.	1.1.2
☑ argh	○	The natural cli.	0.26.2
☑ argon2-cffi	○	The secure argon2 password hashing algorithm.	20.1.0
☑ asn1crypto	○	Python asn.1 library with a focus on performance and a pythonic api	1.4.0
☑ astroid	○	A abstract syntax tree for python with inference support.	↗ 2.5
☑ astropy	○	Community-developed python library for astronomy	4.2.1

356 packages available

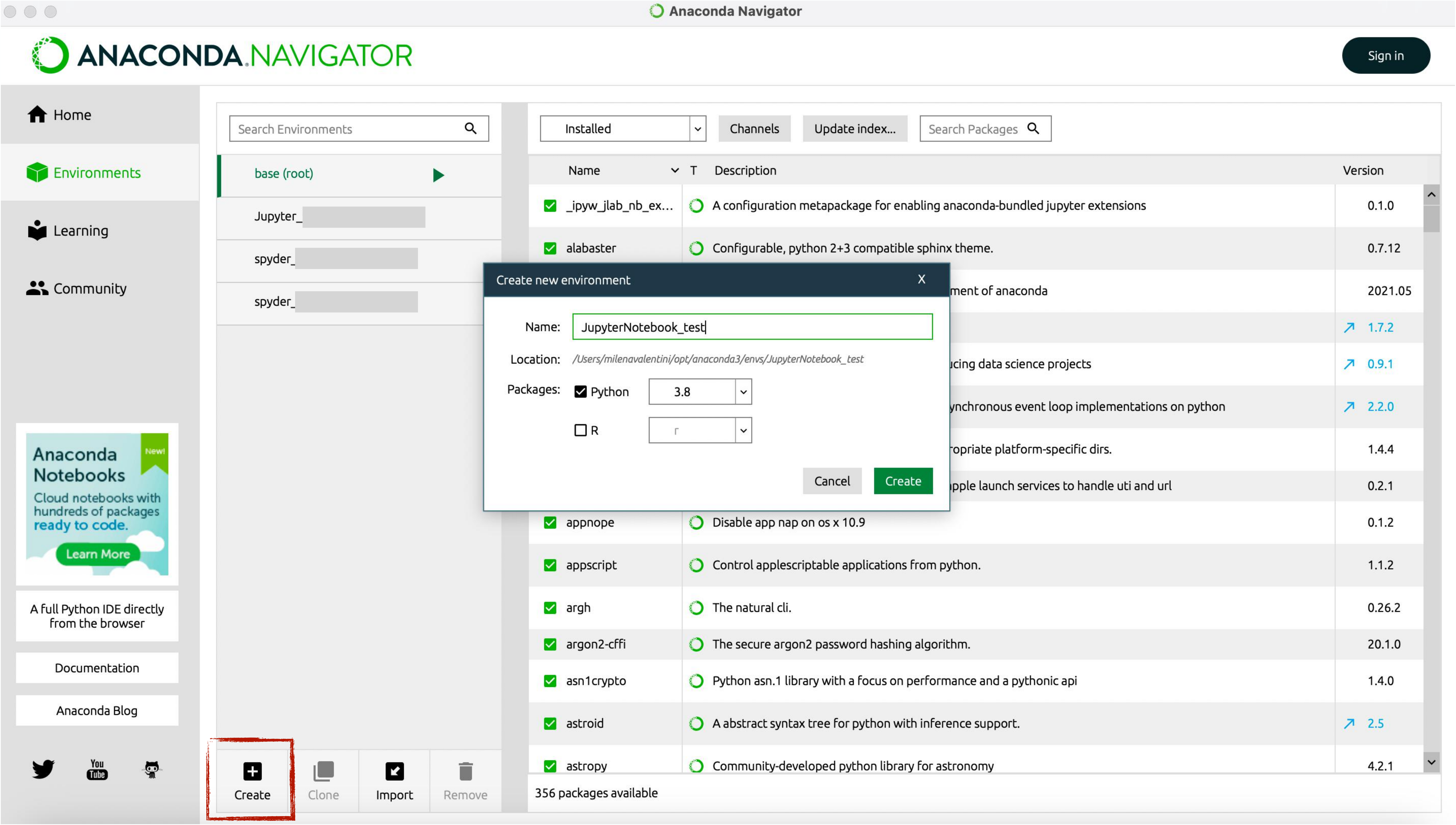
Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

Select Environments:

Create a new environment:



Create new environment X

Name: JupyterNotebook_test

Location: /Users/milenavalentini/opt/anaconda3/envs/JupyterNotebook_test

Packages: ☑ Python 3.8

□ R r

Cancel Create

	Name	T	Description	Version
☑	_ipyw_jlab_nb_ex...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
☑	alabaster	○	Configurable, python 2+3 compatible sphinx theme.	0.7.12
			ment of anaconda	2021.05
				↗ 1.7.2
			cing data science projects	↗ 0.9.1
			ynchronous event loop implementations on python	↗ 2.2.0
			ropriate platform-specific dirs.	1.4.4
			pple launch services to handle uti and url	0.2.1
☑	appnope	○	Disable app nap on os x 10.9	0.1.2
☑	appscript	○	Control applescriptable applications from python.	1.1.2
☑	argh	○	The natural cli.	0.26.2
☑	argon2-cffi	○	The secure argon2 password hashing algorithm.	20.1.0
☑	asn1crypto	○	Python asn.1 library with a focus on performance and a pythonic api	1.4.0
☑	astroid	○	A abstract syntax tree for python with inference support.	↗ 2.5
☑	astropy	○	Community-developed python library for astronomy	4.2.1

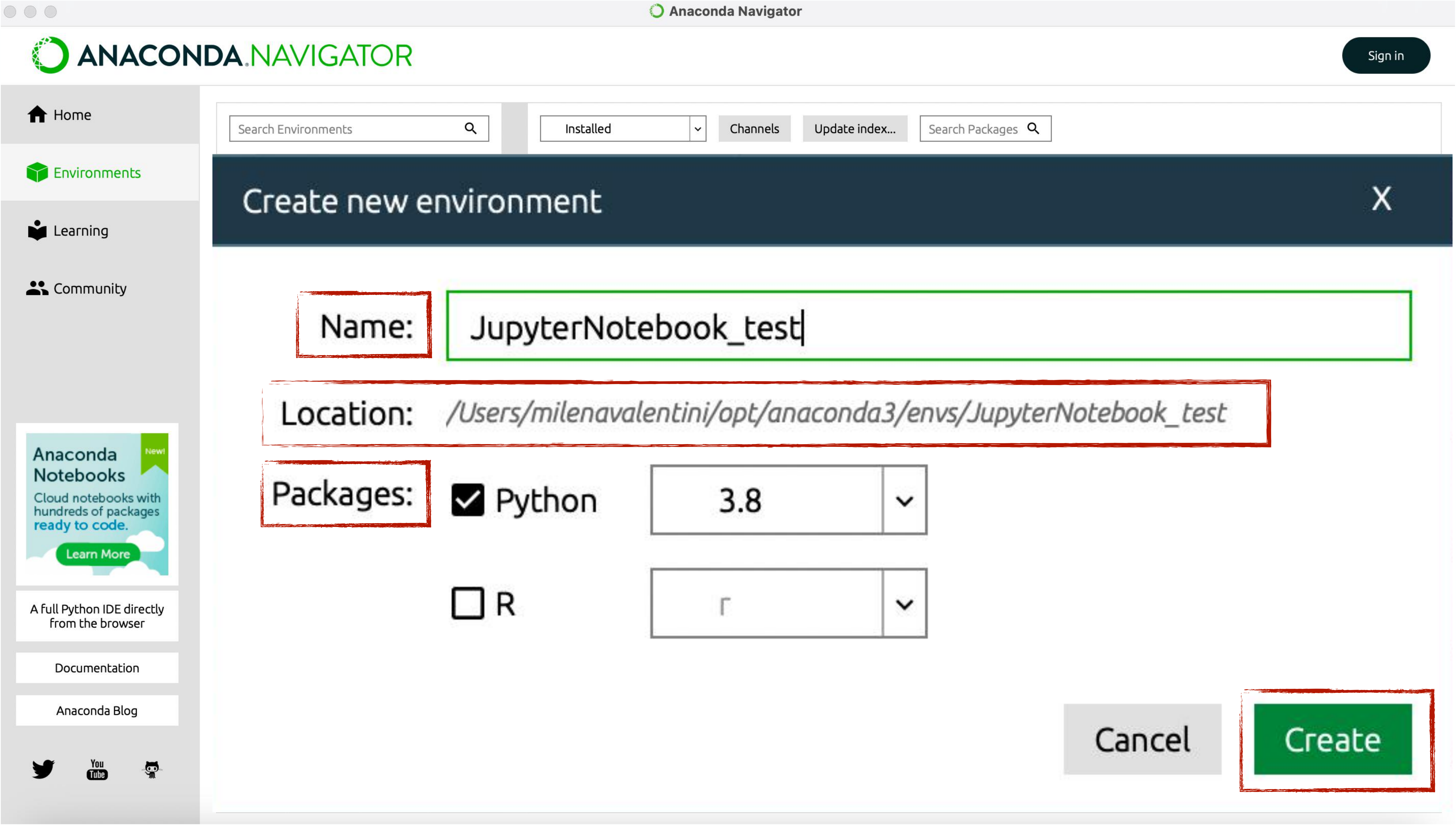
Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

Select Environments:

Create a new environment:



Create new environment

Name: JupyterNotebook_test

Location: /Users/milenavalentini/opt/anaconda3/envs/JupyterNotebook_test

Packages: ☑ Python 3.8

□ R r

Setting up the working environment with Anaconda

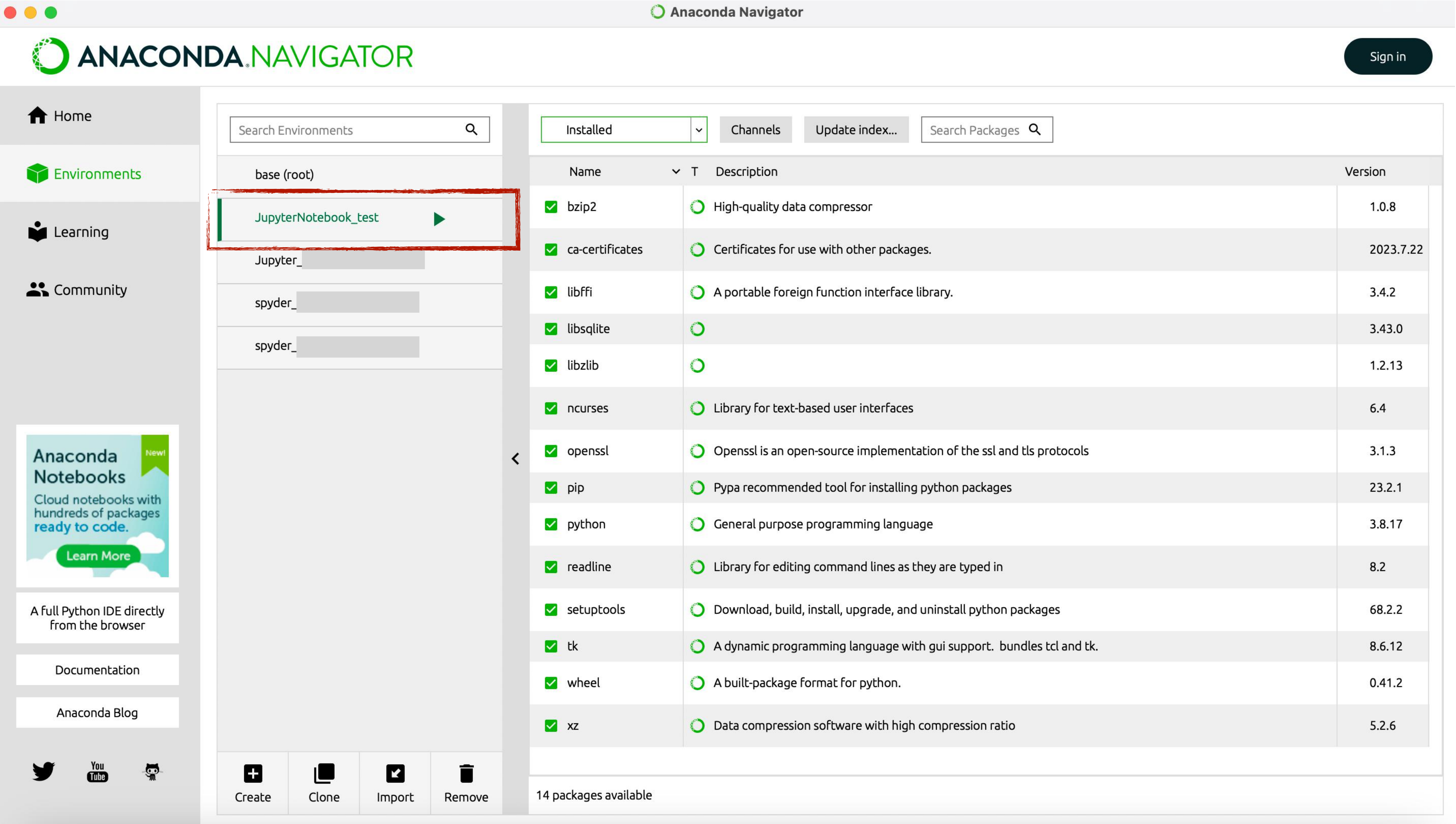
To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

Select Environments:

Create a new environment:

Here is the new environment:



Name	T	Description	Version
☑ bzip2		High-quality data compressor	1.0.8
☑ ca-certificates		Certificates for use with other packages.	2023.7.22
☑ libffi		A portable foreign function interface library.	3.4.2
☑ libsqlite			3.43.0
☑ libzlib			1.2.13
☑ ncurses		Library for text-based user interfaces	6.4
☑ openssl		Openssl is an open-source implementation of the ssl and tls protocols	3.1.3
☑ pip		Pypa recommended tool for installing python packages	23.2.1
☑ python		General purpose programming language	3.8.17
☑ readline		Library for editing command lines as they are typed in	8.2
☑ setuptools		Download, build, install, upgrade, and uninstall python packages	68.2.2
☑ tk		A dynamic programming language with gui support. bundles tcl and tk.	8.6.12
☑ wheel		A built-package format for python.	0.41.2
☑ xz		Data compression software with high compression ratio	5.2.6

14 packages available

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

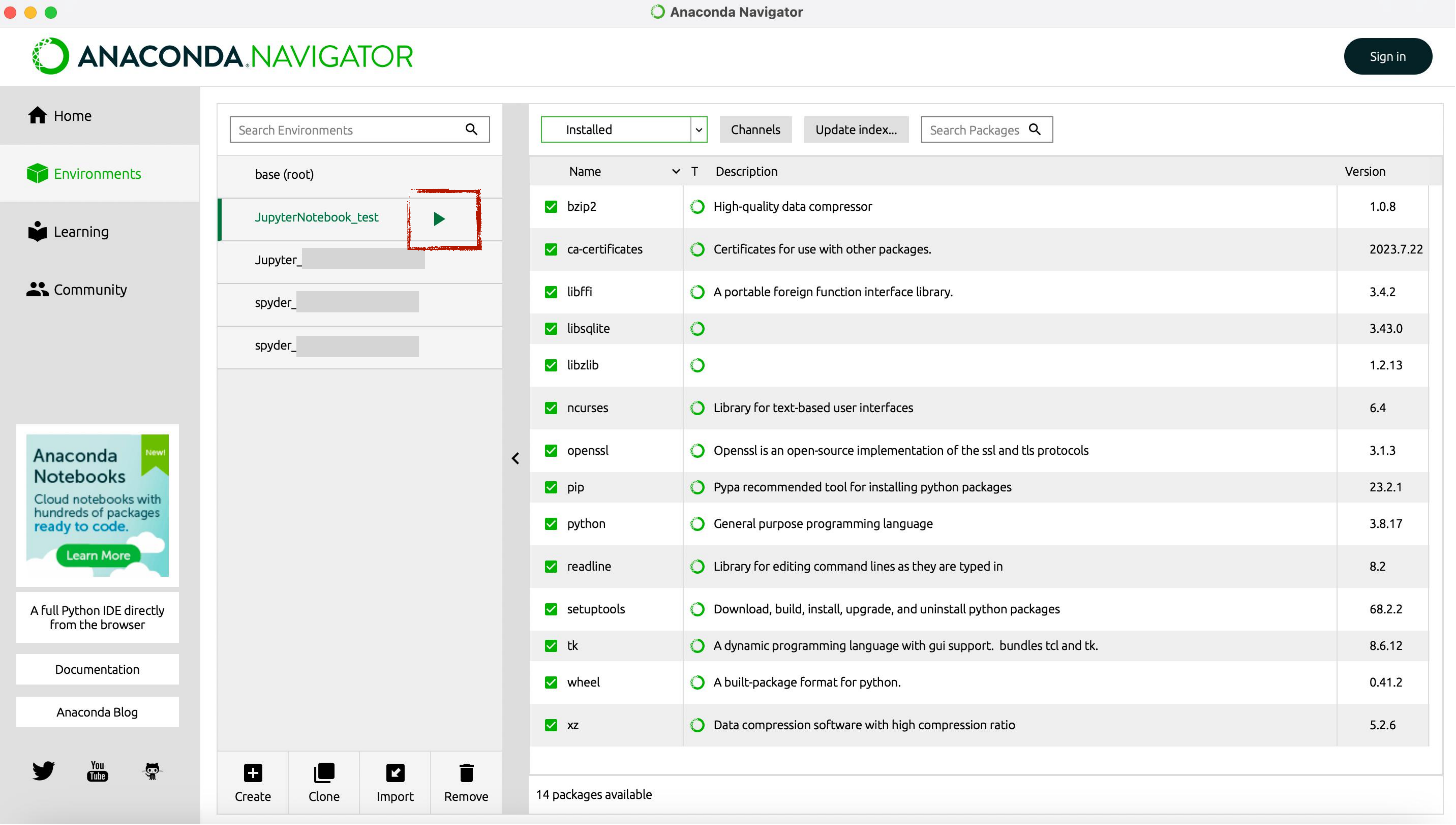
Launch Anaconda-Navigator:

Select Environments:

Create a new environment:

Here is the new environment:

The green arrow tells you that the new environment is active



	Name	T	Description	Version
☑	bzip2	○	High-quality data compressor	1.0.8
☑	ca-certificates	○	Certificates for use with other packages.	2023.7.22
☑	libffi	○	A portable foreign function interface library.	3.4.2
☑	libsqlite	○		3.43.0
☑	libzlib	○		1.2.13
☑	ncurses	○	Library for text-based user interfaces	6.4
☑	openssl	○	Openssl is an open-source implementation of the ssl and tls protocols	3.1.3
☑	pip	○	Pypa recommended tool for installing python packages	23.2.1
☑	python	○	General purpose programming language	3.8.17
☑	readline	○	Library for editing command lines as they are typed in	8.2
☑	setuptools	○	Download, build, install, upgrade, and uninstall python packages	68.2.2
☑	tk	○	A dynamic programming language with gui support. bundles tcl and tk.	8.6.12
☑	wheel	○	A built-package format for python.	0.41.2
☑	xz	○	Data compression software with high compression ratio	5.2.6

14 packages available

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

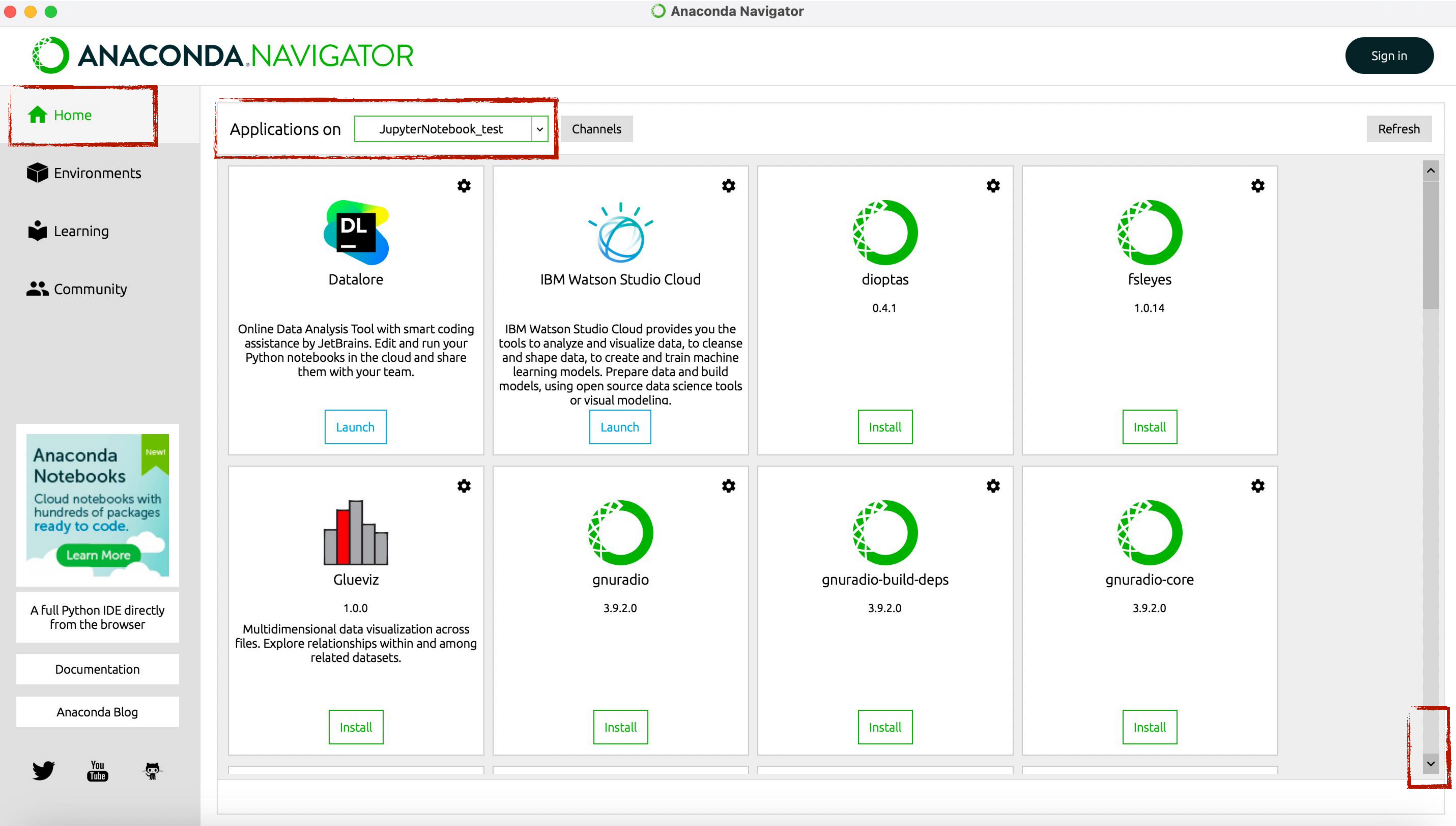
Select Environments:

Create a new environment:

Here is the new environment:

The green arrow tells you that the new environment is active

Select the applications to be installed in the environment among available ones



Datalore

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

dioptas 0.4.1

fsleyes 1.0.14

Glueviz 1.0.0

Multidimensional data visualization across files. Explore relationships within and among related datasets.

gnuradio 3.9.2.0

gnuradio-build-deps 3.9.2.0

gnuradio-core 3.9.2.0

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

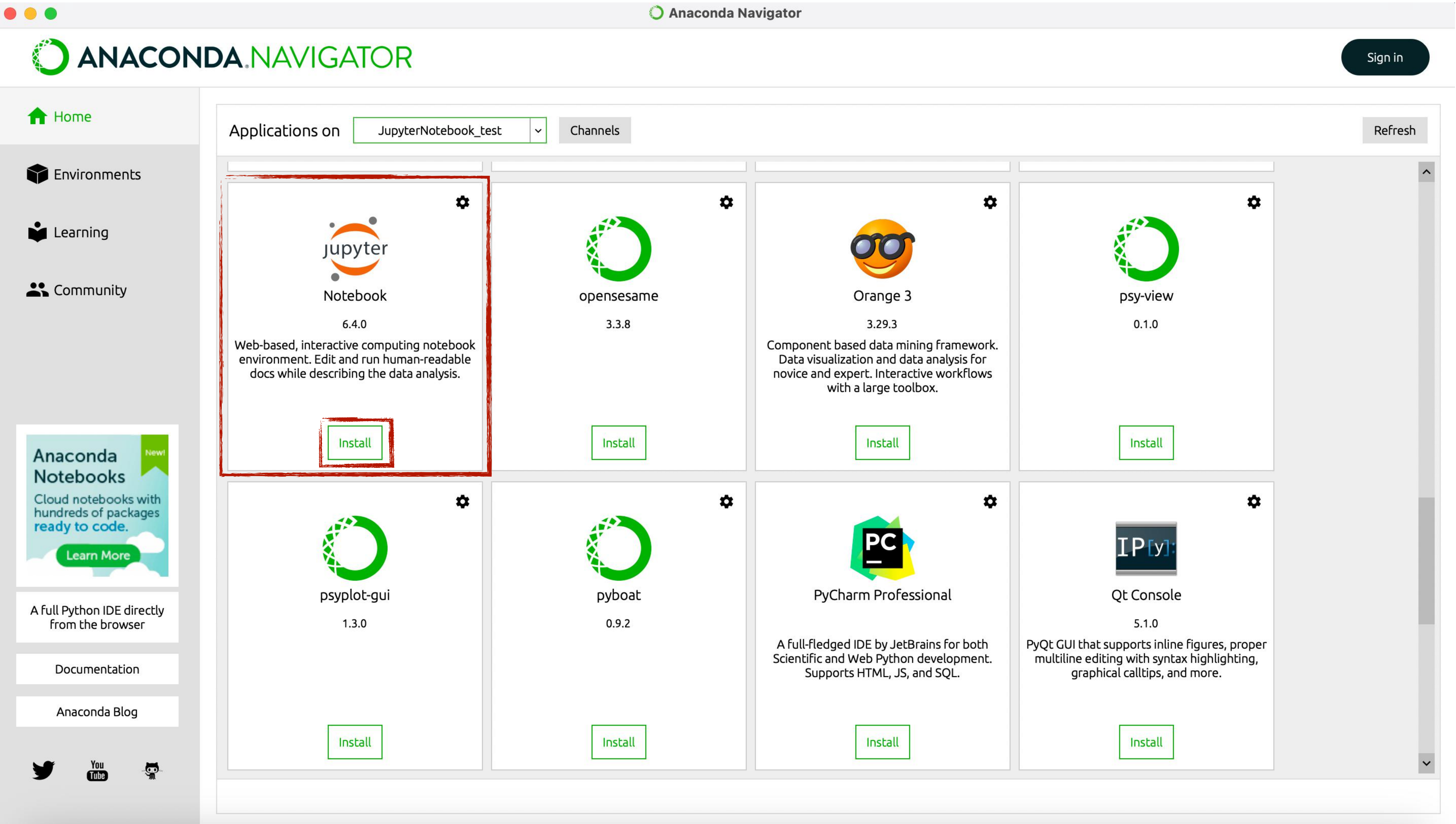
Select Environments:

Create a new environment:

Here is the new environment:

The green arrow tells you that the new environment is active

Select the applications to be installed in the environment among available ones



Anaconda Navigator

ANACONDA.NAVIGATOR

Sign in

Home

Environments

Learning

Community

Anaconda Notebooks New!
Cloud notebooks with hundreds of packages ready to code.
Learn More

A full Python IDE directly from the browser

Documentation

Anaconda Blog

Applications on JupyterNotebook_test Channels Refresh

Notebook
6.4.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Install

opensesame
3.3.8
Install

Orange 3
3.29.3
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

psy-view
0.1.0
Install

psyplot-gui
1.3.0
Install

pyboat
0.9.2
Install

PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.
Install

Qt Console
5.1.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Install

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

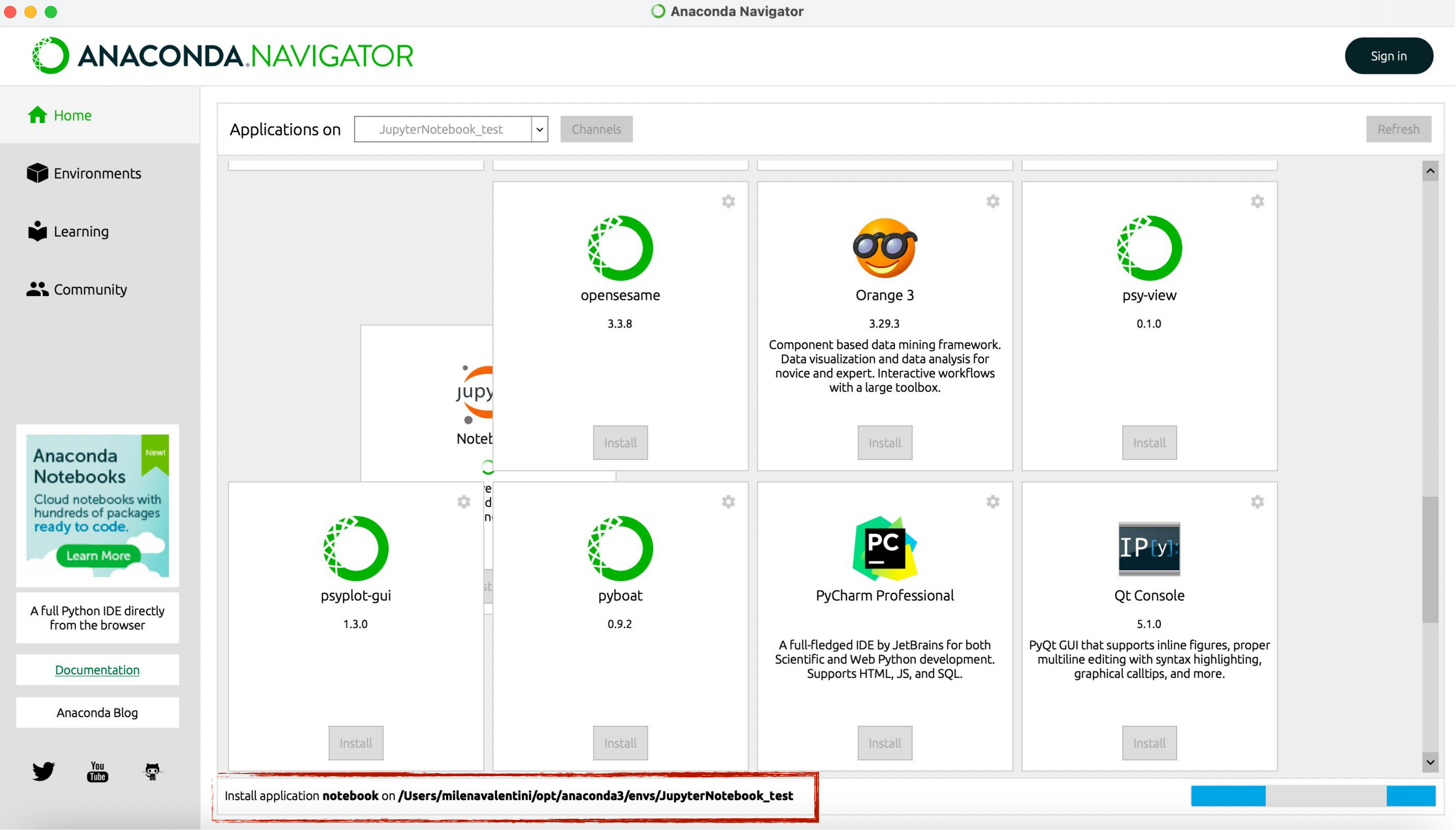
Select Environments:

Create a new environment:

Here is the new environment:

The green arrow tells you that the new environment is active

Select the applications to be installed in the environment among available ones


Anaconda Navigator
ANACONDA.NAVIGATOR
Sign in
Home
Environments
Learning
Community
Anaconda Notebooks
New!
Cloud notebooks with hundreds of packages ready to code.
Learn More
A full Python IDE directly from the browser
Documentation
Anaconda Blog
Applications on
JupyterNotebook_test
Channels
Refresh
opensesame
3.3.8
Install
Orange 3
3.29.3
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install
psy-view
0.1.0
Install
psyplot-gui
1.3.0
Install
pyboat
0.9.2
Install
PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.
Install
Qt Console
5.1.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Install

Install application **notebook** on **/Users/milenavalentini/opt/anaconda3/envs/JupyterNotebook_test**

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

Select Environments:

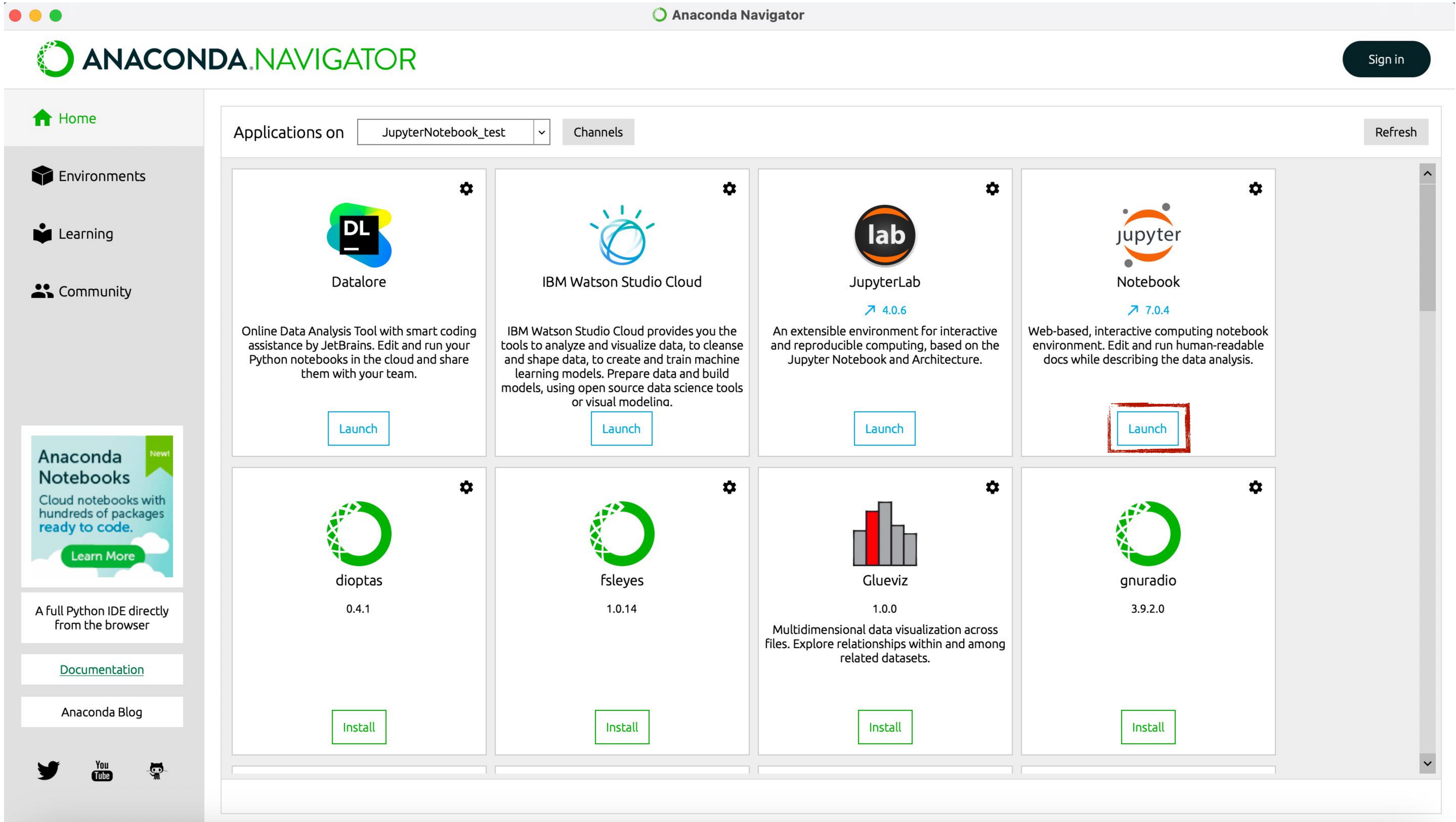
Create a new environment:

Here is the new environment:

The green arrow tells you that the new environment is active

Select the applications to be installed in the environment

The application has just been installed and can be launched



Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.
Launch

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
Launch

JupyterLab
↗ 4.0.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch

Notebook
↗ 7.0.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

dioptas
0.4.1
Install

fsleyes
1.0.14
Install

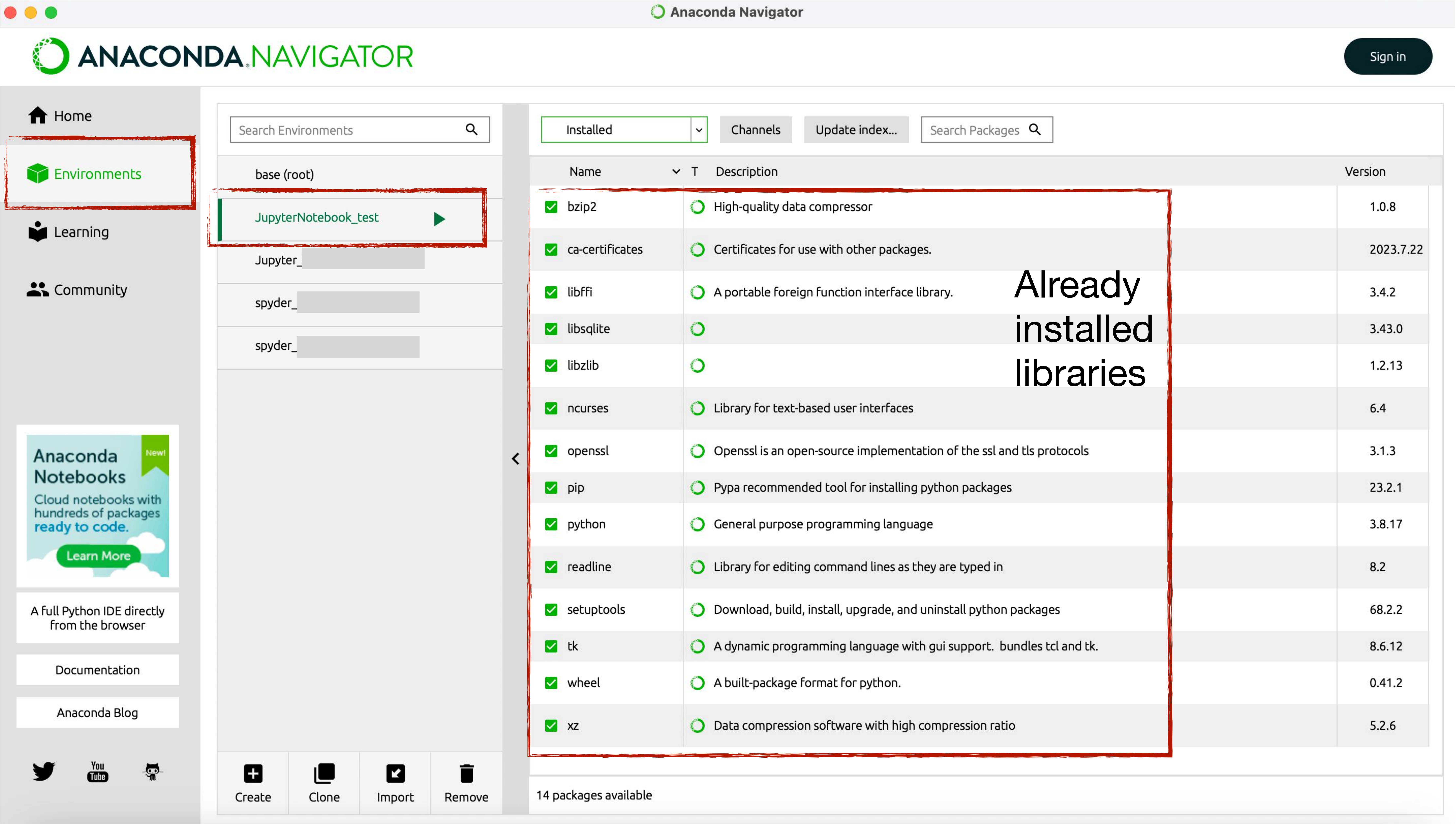
Glueviz
1.0.0
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

gnuradio
3.9.2.0
Install

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

To install libraries
(instead of applications)
within a given environment:



	Name	T	Description	Version
☑	bzip2	○	High-quality data compressor	1.0.8
☑	ca-certificates	○	Certificates for use with other packages.	2023.7.22
☑	libffi	○	A portable foreign function interface library.	3.4.2
☑	libsqlite	○		3.43.0
☑	libzlib	○		1.2.13
☑	ncurses	○	Library for text-based user interfaces	6.4
☑	openssl	○	Openssl is an open-source implementation of the ssl and tls protocols	3.1.3
☑	pip	○	Pypa recommended tool for installing python packages	23.2.1
☑	python	○	General purpose programming language	3.8.17
☑	readline	○	Library for editing command lines as they are typed in	8.2
☑	setuptools	○	Download, build, install, upgrade, and uninstall python packages	68.2.2
☑	tk	○	A dynamic programming language with gui support. bundles tcl and tk.	8.6.12
☑	wheel	○	A built-package format for python.	0.41.2
☑	xz	○	Data compression software with high compression ratio	5.2.6

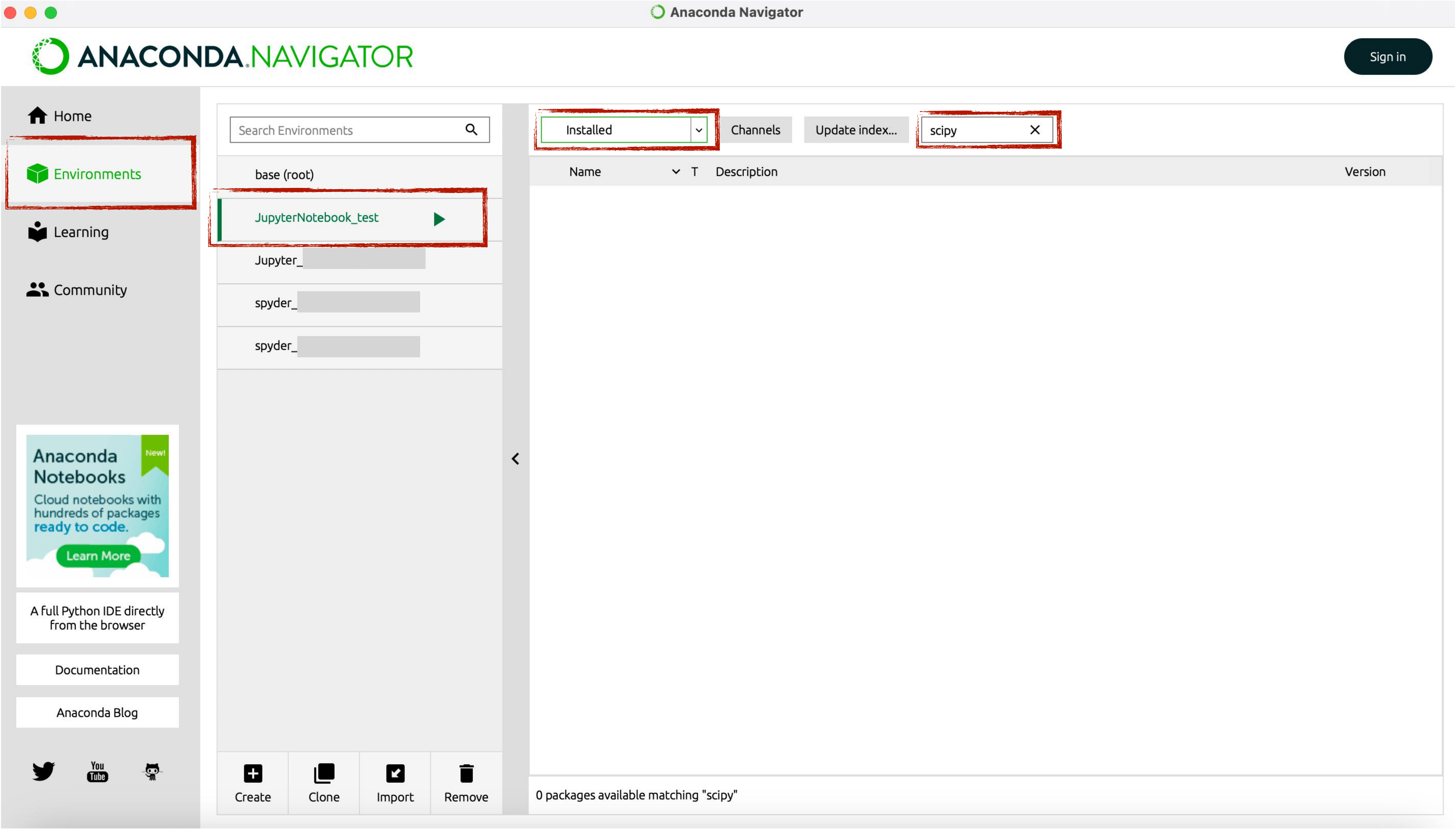
14 packages available

Already installed libraries

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

To install libraries
(instead of applications)
within a given environment:



Environments:

- base (root)
- JupyterNotebook_test
- Jupyter_[redacted]
- spyder_[redacted]
- spyder_[redacted]

Installed

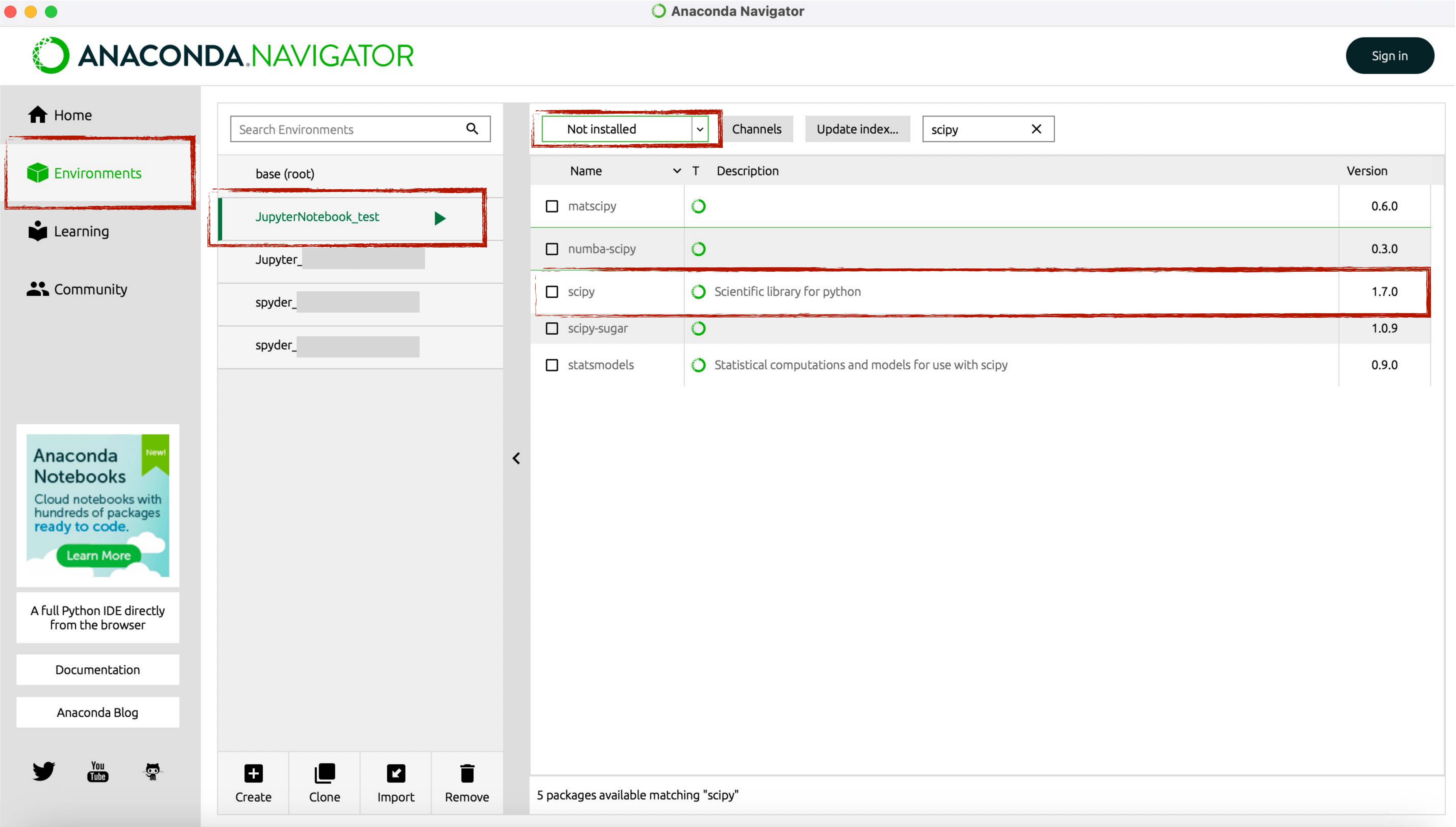
scipy

0 packages available matching "scipy"

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

To install libraries
(instead of applications)
within a given environment:



Not installed

scipy

Name	T	Description	Version
□ matscipy	○		0.6.0
□ numba-scipy	○		0.3.0
□ scipy	○	Scientific library for python	1.7.0
□ scipy-sugar	○		1.0.9
□ statsmodels	○	Statistical computations and models for use with scipy	0.9.0

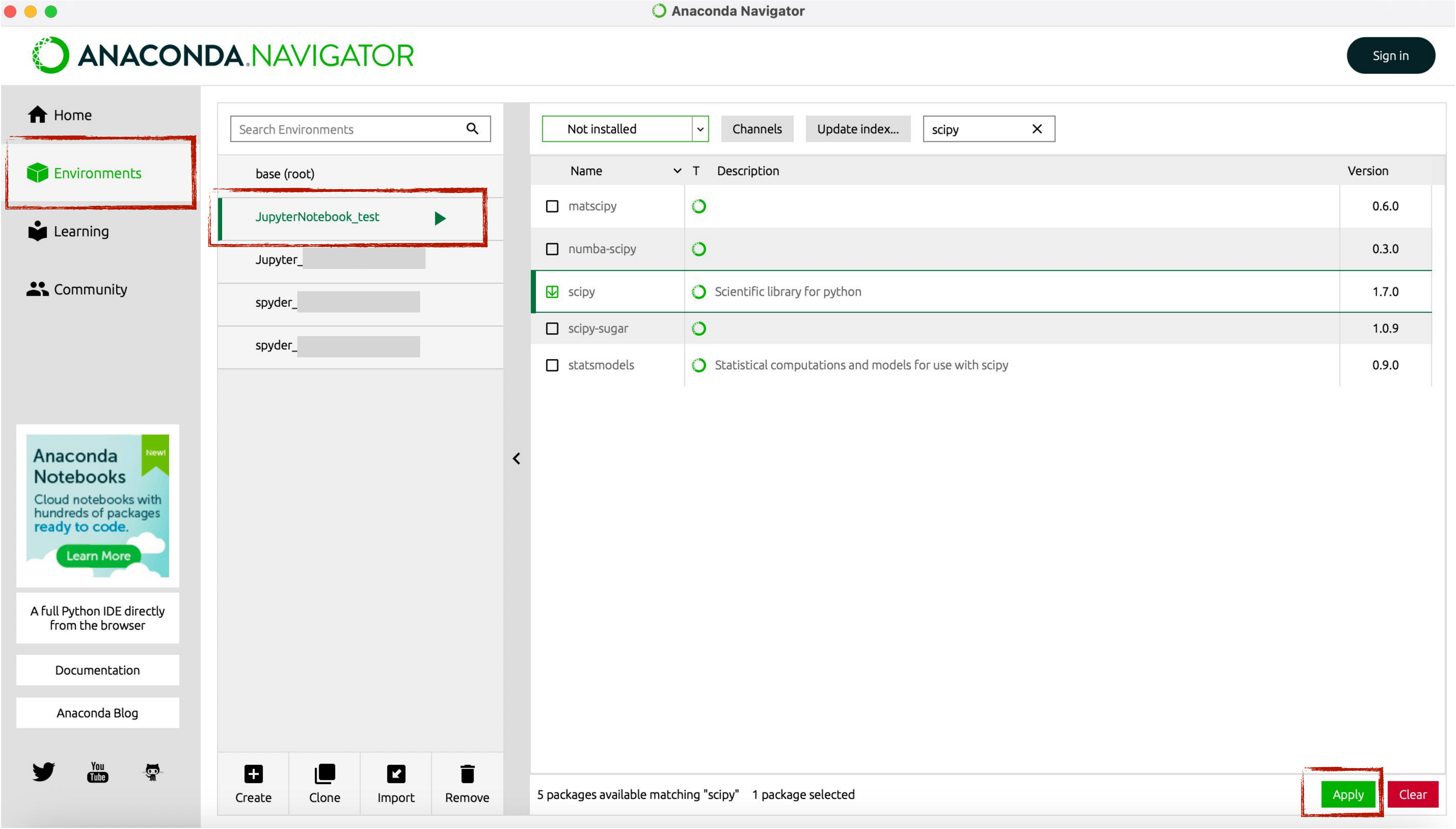
5 packages available matching "scipy"

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

To install libraries (instead of applications) within a given environment:

Select the library to be installed in the environment



Anaconda Navigator

ANACONDA.NAVIGATOR

Sign in

Home

Environments

Learning

Community

Anaconda Notebooks New!

Cloud notebooks with hundreds of packages ready to code.

Learn More

A full Python IDE directly from the browser

Documentation

Anaconda Blog

Search Environments

base (root)

JupyterNotebook_test ▶

Jupyter_[redacted]

spyder_[redacted]

spyder_[redacted]

Create | Clone | Import | Remove

Not installed | Channels | Update index... | scipy ✕

	Name	T	Description	Version
□	matscipy	○		0.6.0
□	numba-scipy	○		0.3.0
☑	scipy	○	Scientific library for python	1.7.0
□	scipy-sugar	○		1.0.9
□	statsmodels	○	Statistical computations and models for use with scipy	0.9.0

5 packages available matching "scipy"   1 package selected

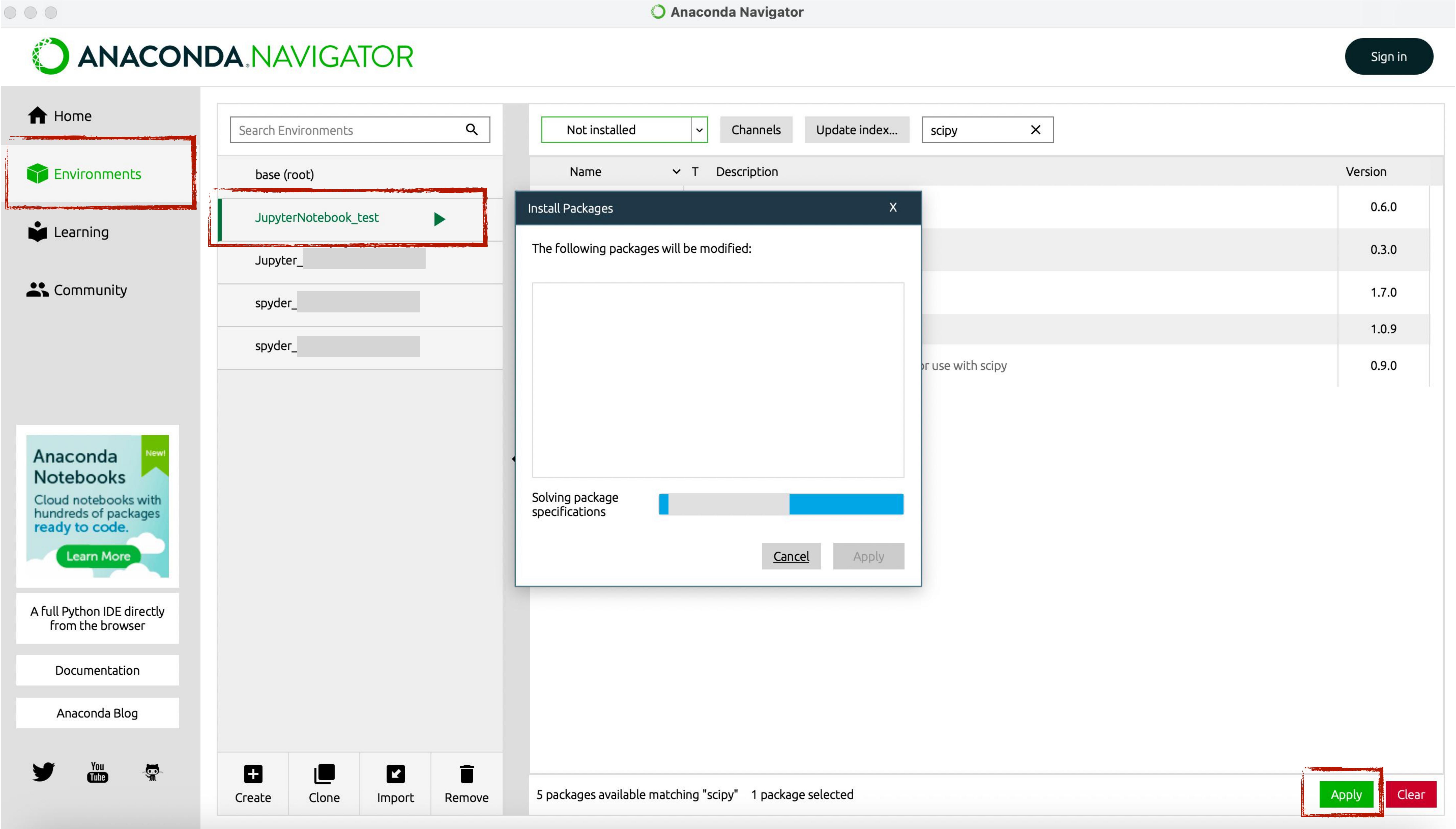
Apply | Clear

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

To install libraries (instead of applications) within a given environment:

Select the library to be installed in the environment



Anaconda Navigator

Install Packages

The following packages will be modified:

Solving package specifications

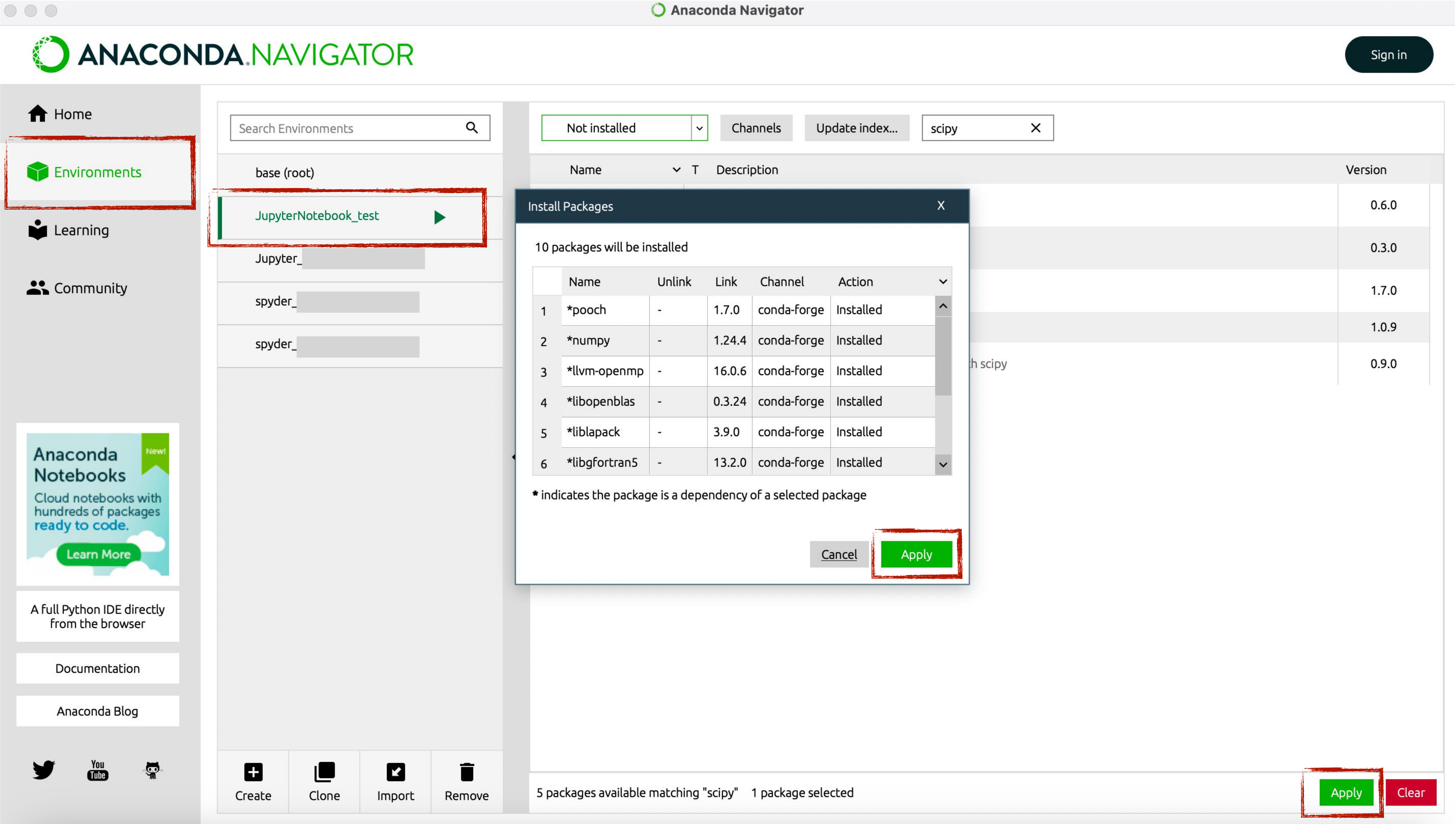
5 packages available matching "scipy"   1 package selected

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

To install libraries (instead of applications) within a given environment:

Select the library to be installed in the environment



10 packages will be installed

	Name	Unlink	Link	Channel	Action
1	*pooch	-	1.7.0	conda-forge	Installed
2	*numpy	-	1.24.4	conda-forge	Installed
3	*llvm-openmp	-	16.0.6	conda-forge	Installed
4	*libopenblas	-	0.3.24	conda-forge	Installed
5	*liblapack	-	3.9.0	conda-forge	Installed
6	*libgfortran5	-	13.2.0	conda-forge	Installed

* indicates the package is a dependency of a selected package

5 packages available matching "scipy"   1 package selected

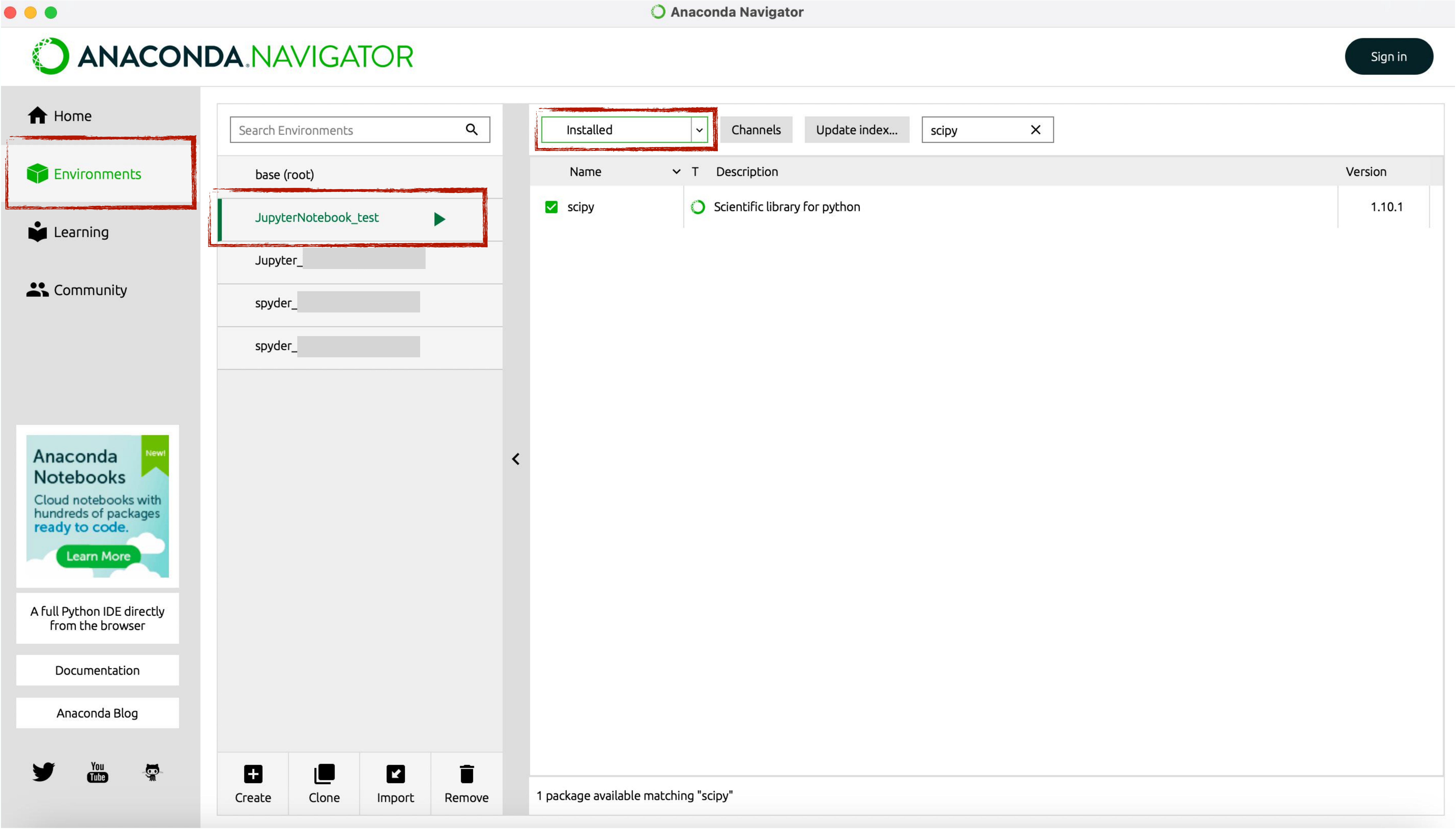
Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

To install libraries (instead of applications) within a given environment:

Select the library to be installed in the environment

The library has just been installed and can be launched

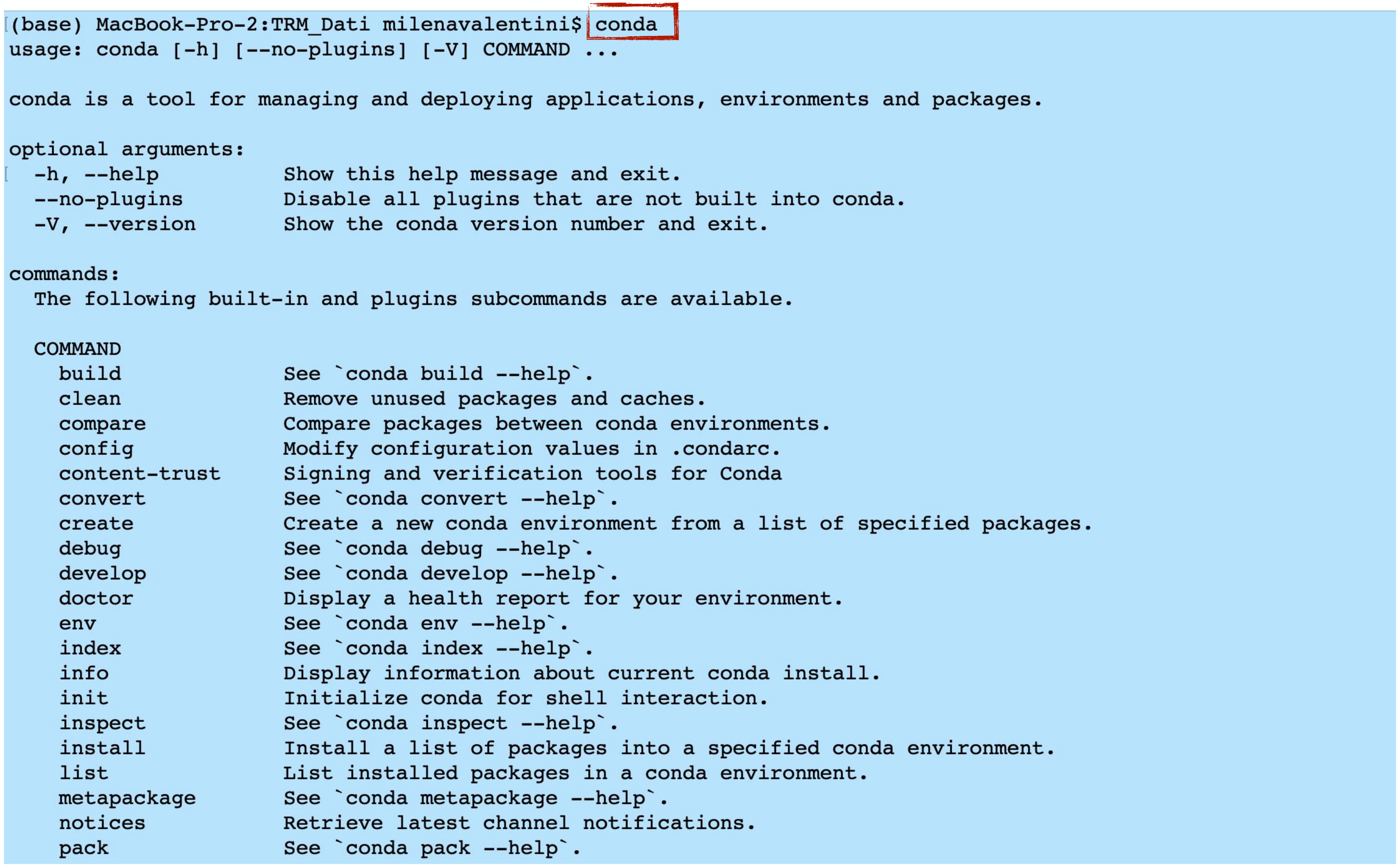


☐	Name	T	Description	Version
☑	scipy	○	Scientific library for python	1.10.1

1 package available matching "scipy"

Setting up the working environment with Anaconda

Let’s use Anaconda via shell (i.e. without its graphical unit interface):



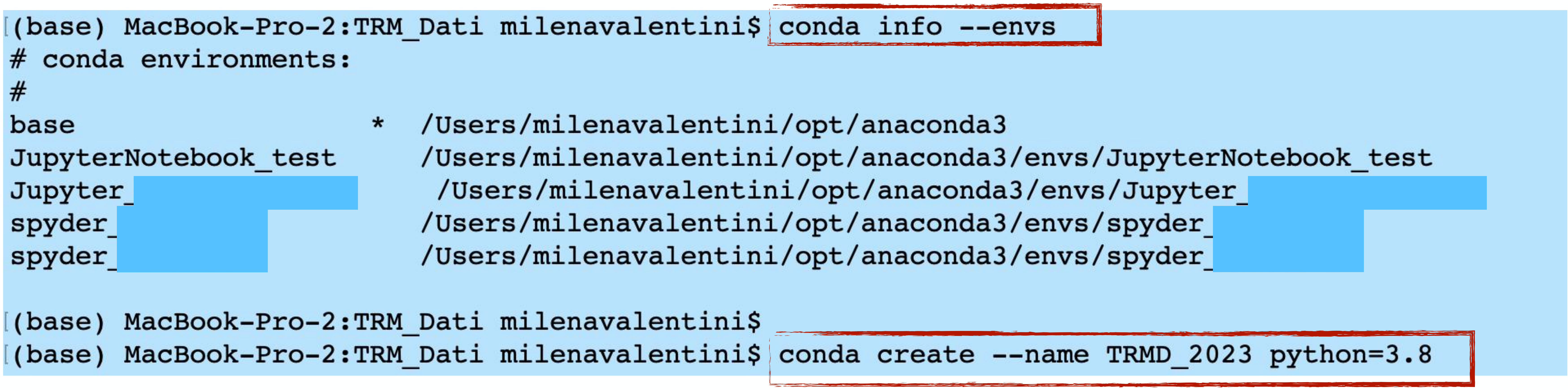
```
[(base) MacBook-Pro-2:TRM_Dati milenavalentini$ conda
usage: conda [-h] [--no-plugins] [-V] COMMAND ...

conda is a tool for managing and deploying applications, environments and packages.

optional arguments:
[  -h, --help            Show this help message and exit.
   --no-plugins          Disable all plugins that are not built into conda.
   -V, --version         Show the conda version number and exit.

commands:
  The following built-in and plugins subcommands are available.

  COMMAND
    build                See `conda build --help`.
    clean                Remove unused packages and caches.
    compare              Compare packages between conda environments.
    config               Modify configuration values in .condarc.
    content-trust        Signing and verification tools for Conda
    convert              See `conda convert --help`.
    create               Create a new conda environment from a list of specified packages.
    debug                See `conda debug --help`.
    develop              See `conda develop --help`.
    doctor               Display a health report for your environment.
    env                  See `conda env --help`.
    index                See `conda index --help`.
    info                 Display information about current conda install.
    init                 Initialize conda for shell interaction.
    inspect              See `conda inspect --help`.
    install              Install a list of packages into a specified conda environment.
    list                 List installed packages in a conda environment.
    metapackage          See `conda metapackage --help`.
    notices              Retrieve latest channel notifications.
    pack                 See `conda pack --help`.
```

Setting up the working environment with Anaconda

Let’s use Anaconda via shell

Already available environments:

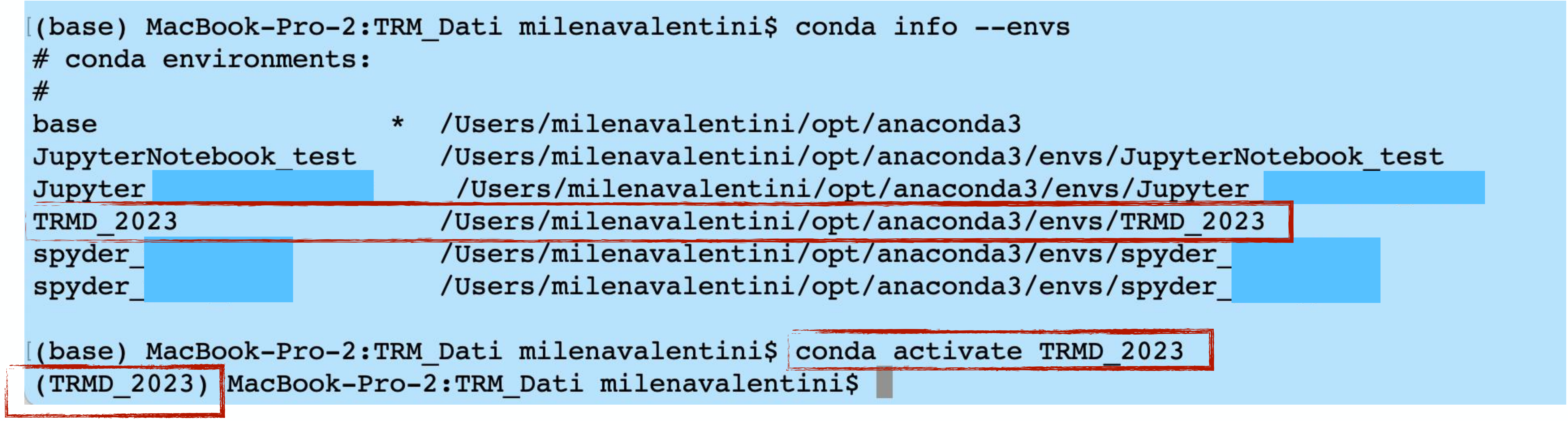
Create a new environment (you can also specify which version of Python you want to use by including the version number after the environment name):



```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ conda info --envs
# conda environments:
#
base                  *  /Users/milenavalentini/opt/anaconda3
JupyterNotebook_test     /Users/milenavalentini/opt/anaconda3/envs/JupyterNotebook_test
Jupyter_[redacted]        /Users/milenavalentini/opt/anaconda3/envs/Jupyter_[redacted]
spyder_[redacted]         /Users/milenavalentini/opt/anaconda3/envs/spyder_[redacted]
spyder_[redacted]         /Users/milenavalentini/opt/anaconda3/envs/spyder_[redacted]

(base) MacBook-Pro-2:TRM_Dati milenavalentini$
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ conda create --name TRMD_2023 python=3.8
```

The new environment has been create

Activate it:

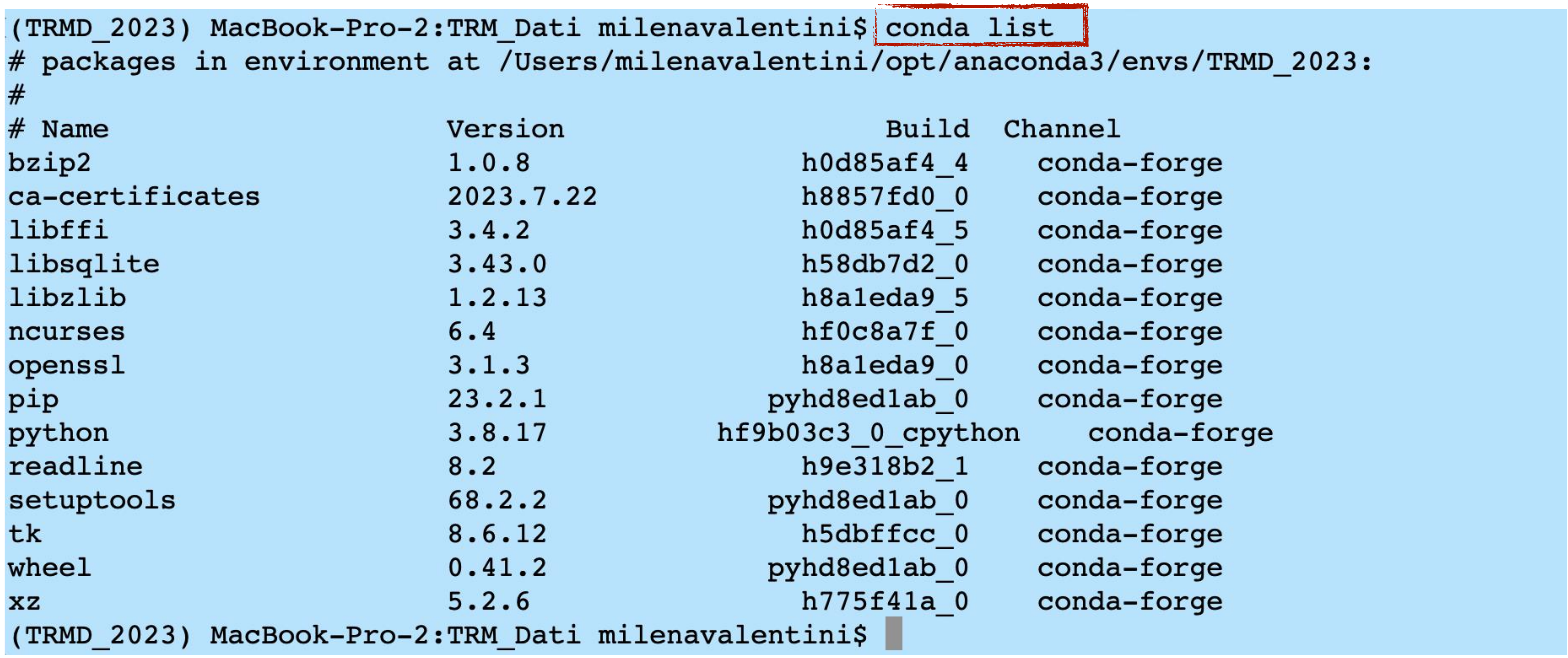


```
(base) MacBook-Pro-2:TRM_Dati milenavalentini$ conda info --envs
# conda environments:
#
base                  *  /Users/milenavalentini/opt/anaconda3
JupyterNotebook_test     /Users/milenavalentini/opt/anaconda3/envs/JupyterNotebook_test
Jupyter [redacted]        /Users/milenavalentini/opt/anaconda3/envs/Jupyter [redacted]
TRMD_2023                /Users/milenavalentini/opt/anaconda3/envs/TRMD_2023
spyder_[redacted]         /Users/milenavalentini/opt/anaconda3/envs/spyder_[redacted]
spyder_[redacted]         /Users/milenavalentini/opt/anaconda3/envs/spyder_[redacted]

(base) MacBook-Pro-2:TRM_Dati milenavalentini$ conda activate TRMD_2023
(TRMD_2023) MacBook-Pro-2:TRM_Dati milenavalentini$
```

Setting up the working environment with Anaconda

Let’s use Anaconda via shell

Packages already available within the active environment:



```
(TRMD_2023) MacBook-Pro-2:TRM_Dati milenavalentini$ conda list
# packages in environment at /Users/milenavalentini/opt/anaconda3/envs/TRMD_2023:
#
# Name                    Version                   Build  Channel
bzip2                     1.0.8                h0d85af4_4    conda-forge
ca-certificates           2023.7.22            h8857fd0_0    conda-forge
libffi                    3.4.2                h0d85af4_5    conda-forge
libsqlite                 3.43.0               h58db7d2_0    conda-forge
libzlib                   1.2.13               h8a1eda9_5    conda-forge
ncurses                   6.4                  hf0c8a7f_0    conda-forge
openssl                   3.1.3                h8a1eda9_0    conda-forge
pip                       23.2.1             pyhd8ed1ab_0    conda-forge
python                    3.8.17          hf9b03c3_0_cpython    conda-forge
readline                  8.2                  h9e318b2_1    conda-forge
setuptools                68.2.2             pyhd8ed1ab_0    conda-forge
tk                        8.6.12               h5dbffcc_0    conda-forge
wheel                     0.41.2             pyhd8ed1ab_0    conda-forge
xz                        5.2.6                h775f41a_0    conda-forge
(TRMD_2023) MacBook-Pro-2:TRM_Dati milenavalentini$ 
```

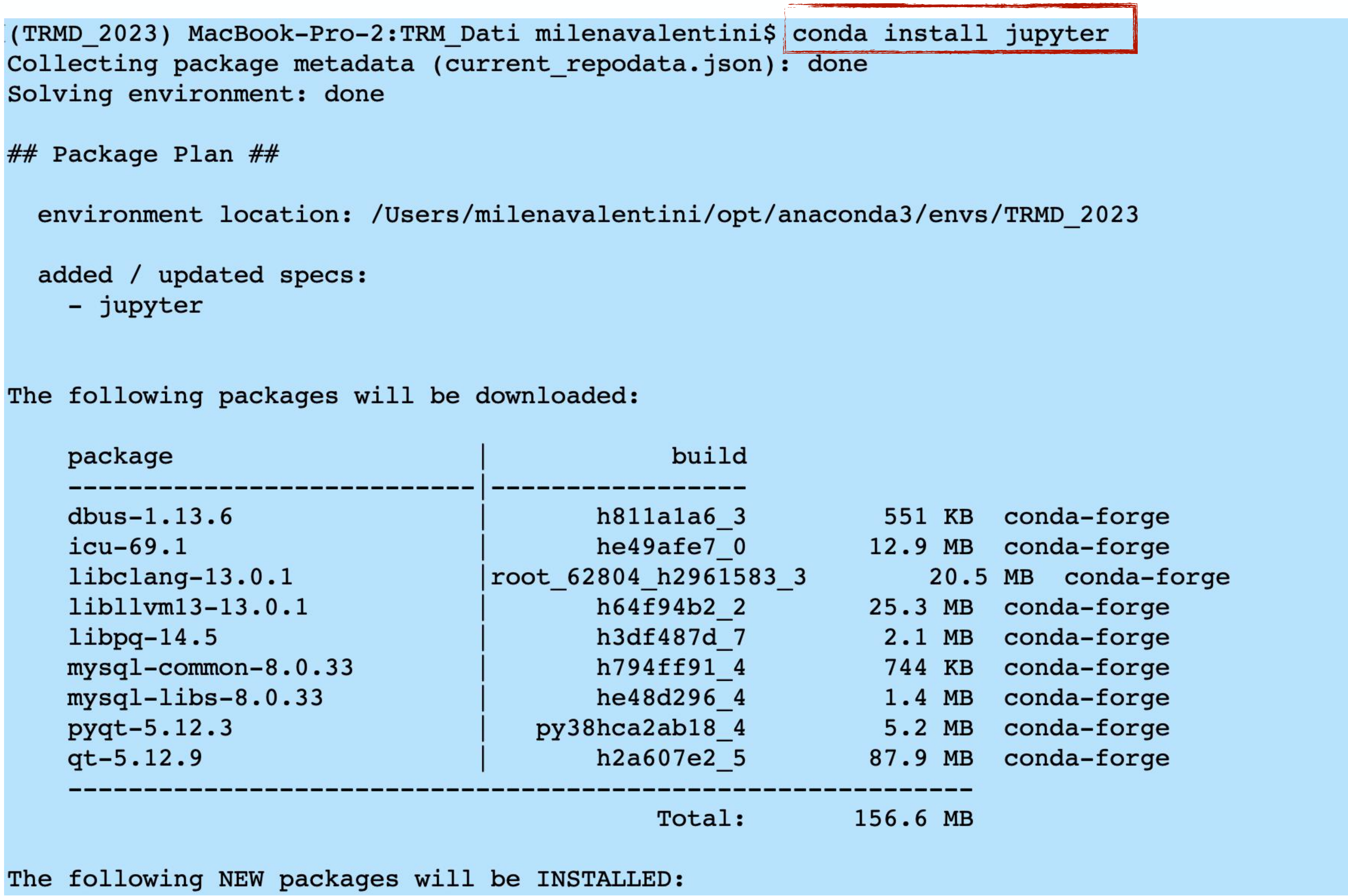
Setting up the working environment with Anaconda

Let’s use Anaconda via shell

Packages already available within the active environment:

As an example of how to install an application:

Install the Jupyter Notebook

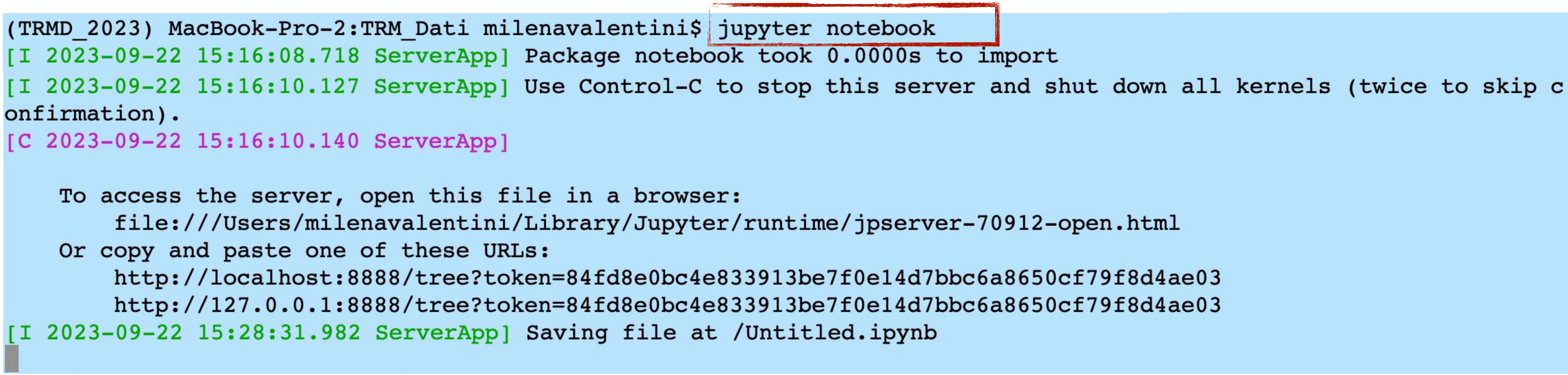


```
(TRMD_2023) MacBook-Pro-2:TRM_Dati milenavalentini$ conda install jupyter
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /Users/milenavalentini/opt/anaconda3/envs/TRMD_2023

  added / updated specs:
    - jupyter


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    dbus-1.13.6                |       h811a1a6_3         551 KB  conda-forge
    icu-69.1                   |       he49afe7_0        12.9 MB  conda-forge
    libclang-13.0.1            |root_62804_h2961583_3        20.5 MB  conda-forge
    libllvm13-13.0.1           |       h64f94b2_2        25.3 MB  conda-forge
    libpq-14.5                 |       h3df487d_7         2.1 MB  conda-forge
    mysql-common-8.0.33        |       h794ff91_4         744 KB  conda-forge
    mysql-libs-8.0.33          |       he48d296_4         1.4 MB  conda-forge
    pyqt-5.12.3                |   py38hca2ab18_4         5.2 MB  conda-forge
    qt-5.12.9                  |       h2a607e2_5        87.9 MB  conda-forge
    ------------------------------------------------------------
                                           Total:       156.6 MB

The following NEW packages will be INSTALLED:
```

Jupyter Notebook

Let’s use Anaconda via shell

Launch it:

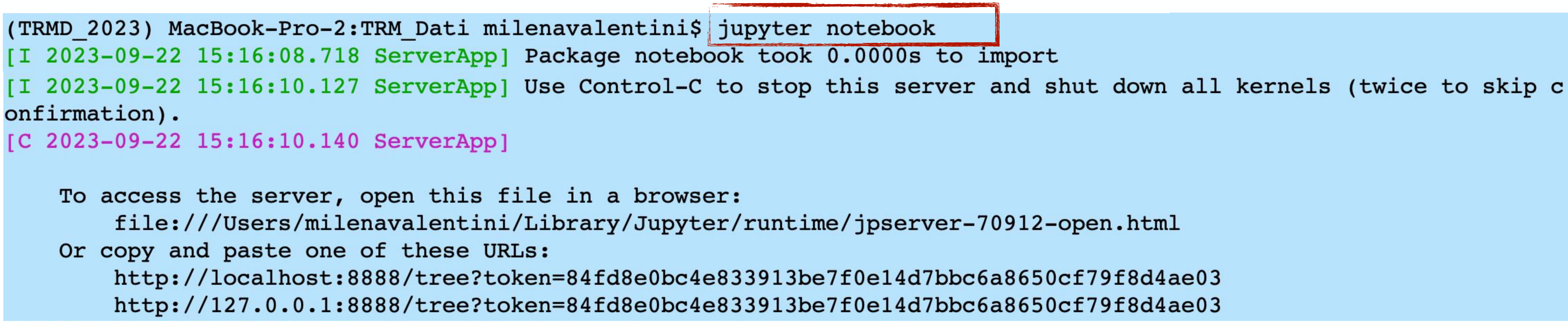


```
(TRMD_2023) MacBook-Pro-2:TRM_Dati milenavalentini$ jupyter notebook
[I 2023-09-22 15:16:08.718 ServerApp] Package notebook took 0.0000s to import
[I 2023-09-22 15:16:10.127 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip c
onfirmation).
[C 2023-09-22 15:16:10.140 ServerApp]

    To access the server, open this file in a browser:
        file:///Users/milenavalentini/Library/Jupyter/runtime/jpserver-70912-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/tree?token=84fd8e0bc4e833913be7f0e14d7bbc6a8650cf79f8d4ae03
        http://127.0.0.1:8888/tree?token=84fd8e0bc4e833913be7f0e14d7bbc6a8650cf79f8d4ae03
[I 2023-09-22 15:28:31.982 ServerApp] Saving file at /Untitled.ipynb
```

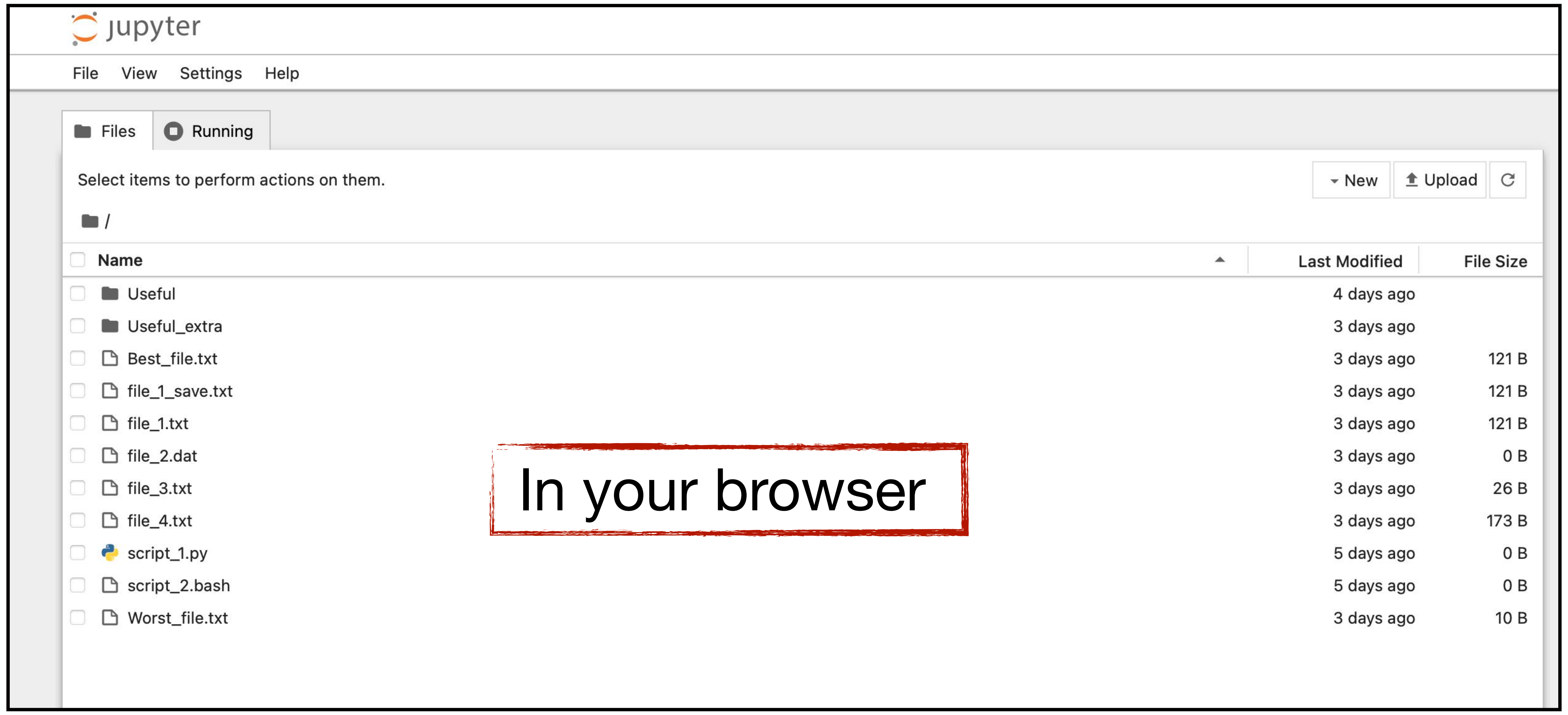
Jupyter Notebook

Let’s use Anaconda via shell

Launch it:



```
(TRMD_2023) MacBook-Pro-2:TRM_Dati milenavalentini$ jupyter notebook
[I 2023-09-22 15:16:08.718 ServerApp] Package notebook took 0.0000s to import
[I 2023-09-22 15:16:10.127 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip c
onfirmation).
[C 2023-09-22 15:16:10.140 ServerApp]

    To access the server, open this file in a browser:
        file:///Users/milenavalentini/Library/Jupyter/runtime/jpserver-70912-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/tree?token=84fd8e0bc4e833913be7f0e14d7bbc6a8650cf79f8d4ae03
        http://127.0.0.1:8888/tree?token=84fd8e0bc4e833913be7f0e14d7bbc6a8650cf79f8d4ae03
```

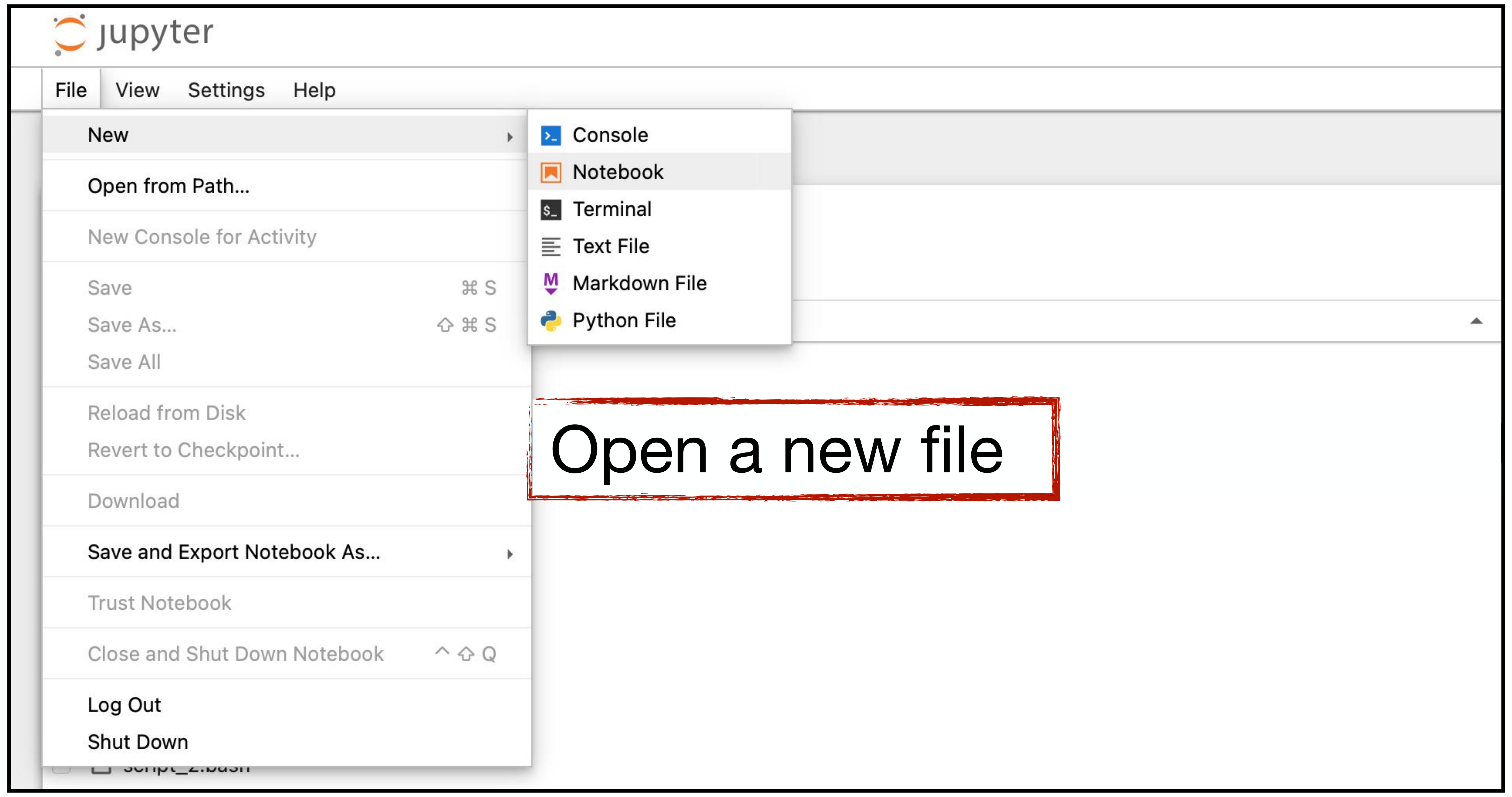


Select items to perform actions on them.

/

Name	Last Modified	File Size
Useful	4 days ago	
Useful_extra	3 days ago	
Best_file.txt	3 days ago	121 B
file_1_save.txt	3 days ago	121 B
file_1.txt	3 days ago	121 B
file_2.dat	3 days ago	0 B
file_3.txt	3 days ago	26 B
file_4.txt	3 days ago	173 B
script_1.py	5 days ago	0 B
script_2.bash	5 days ago	0 B
Worst_file.txt	3 days ago	10 B

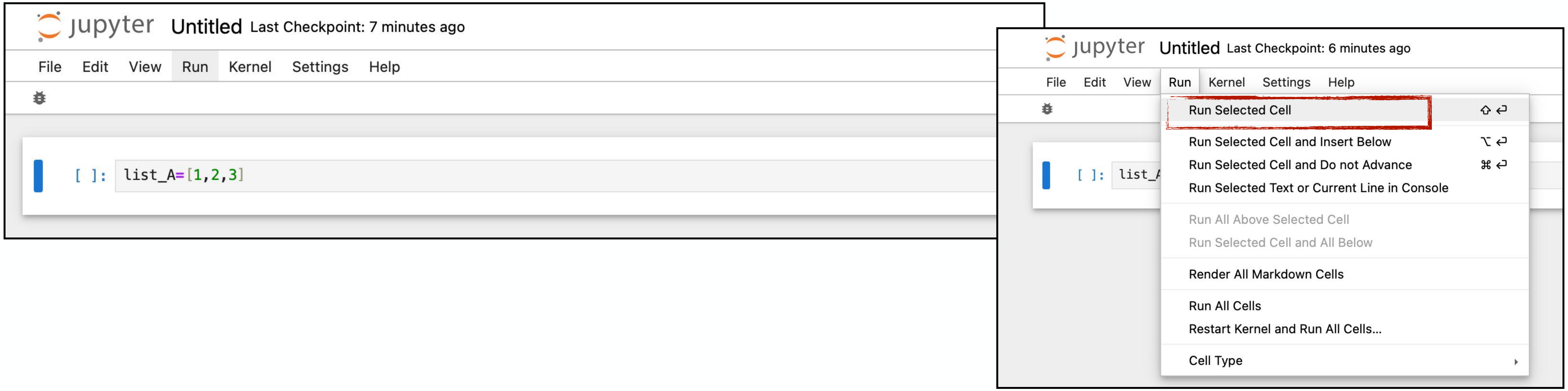
In your browser



Open a new file

Jupyter Notebook

Let’s use Anaconda via shell

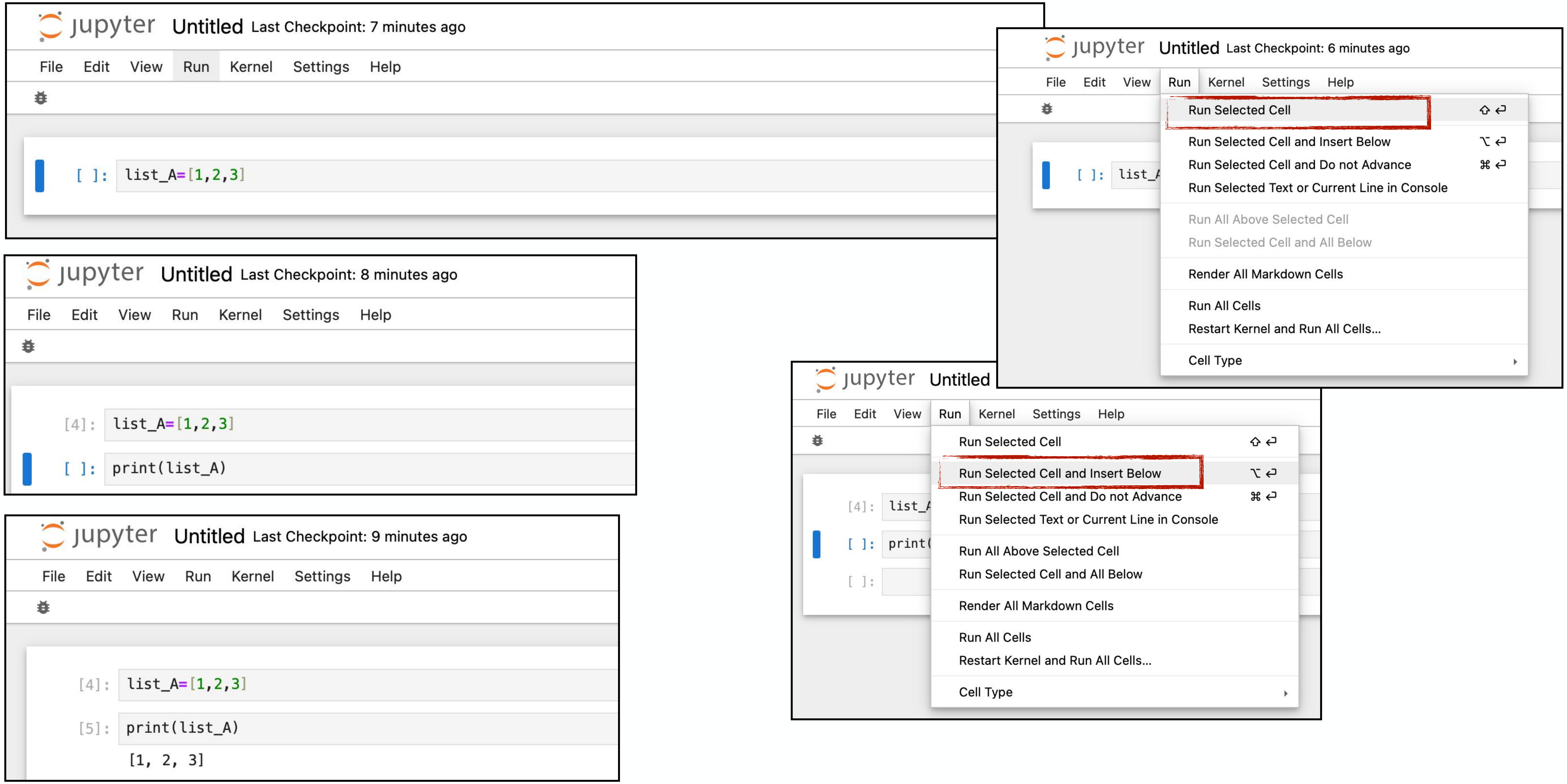


```
[ ]: list_A=[1,2,3]
```

jupyter
Untitled Last Checkpoint: 7 minutes ago
Untitled Last Checkpoint: 6 minutes ago
File
Edit
View
Run
Kernel
Settings
Help
Run Selected Cell ⇧ ↩
Run Selected Cell and Insert Below ⌥ ↩
Run Selected Cell and Do not Advance ⌘ ↩
Run Selected Text or Current Line in Console
Run All Above Selected Cell
Run Selected Cell and All Below
Render All Markdown Cells
Run All Cells
Restart Kernel and Run All Cells...
Cell Type

Jupyter Notebook

Let’s use Anaconda via shell



```
[ ]: list_A=[1,2,3]
```

```
[4]: list_A=[1,2,3]

[ ]: print(list_A)
```

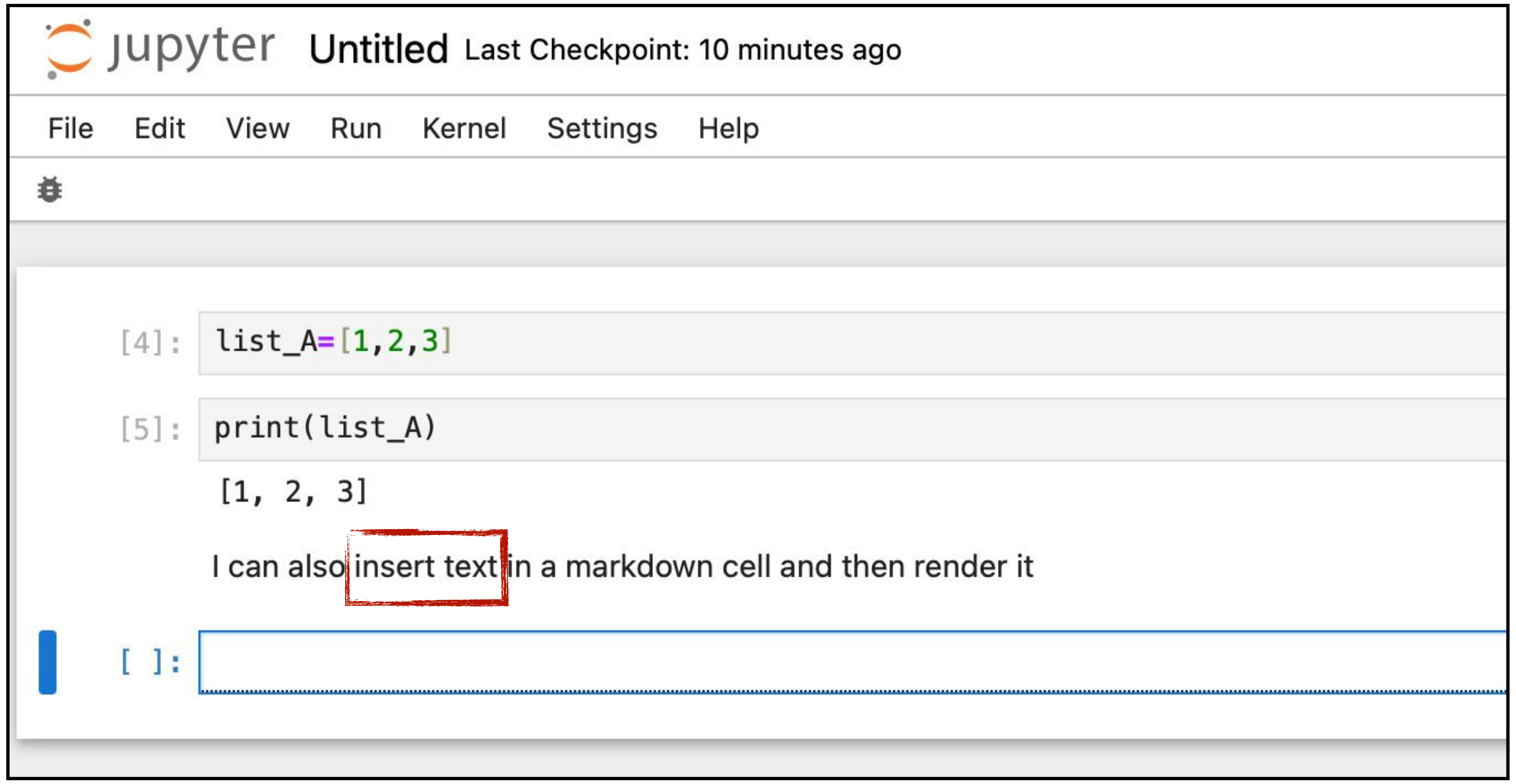
```
[4]: list_A=[1,2,3]

[5]: print(list_A)

     [1, 2, 3]
```

jupyter
Untitled Last Checkpoint: 7 minutes ago
Untitled Last Checkpoint: 6 minutes ago
Untitled Last Checkpoint: 8 minutes ago
Untitled Last Checkpoint: 9 minutes ago
File
Edit
View
Run
Kernel
Settings
Help
Run Selected Cell ⇧↩
Run Selected Cell and Insert Below ⌥↩
Run Selected Cell and Do not Advance ⌘↩
Run Selected Text or Current Line in Console
Run All Above Selected Cell
Run Selected Cell and All Below
Render All Markdown Cells
Run All Cells
Restart Kernel and Run All Cells...
Cell Type

Jupyter Notebook

Let’s use Anaconda via shell

Launch it:



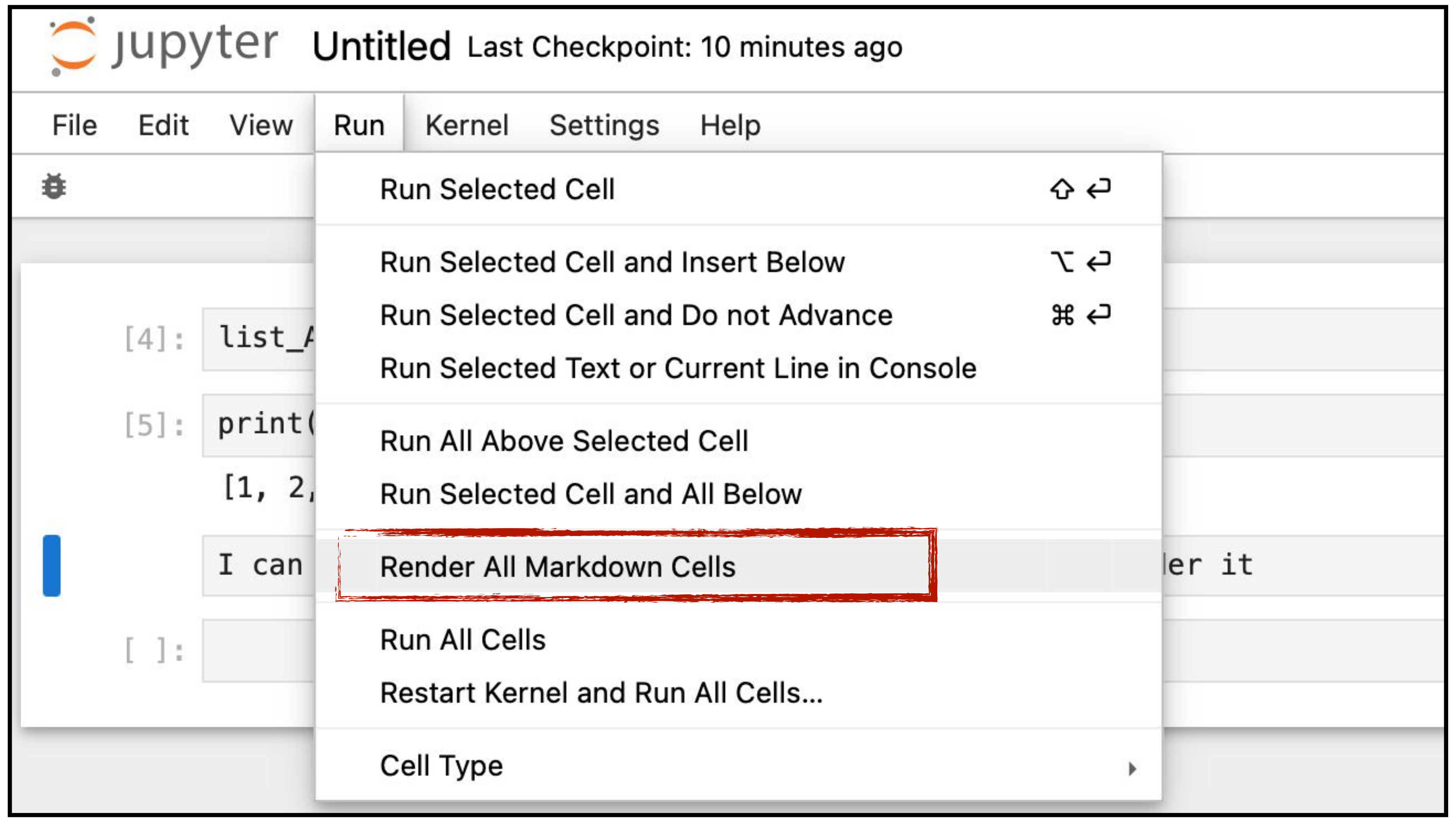
jupyter Untitled Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help

```
[4]: list_A=[1,2,3]

[5]: print(list_A)
     [1, 2, 3]
```

I can also insert text in a markdown cell and then render it

```
[ ]:
```



jupyter Untitled Last Checkpoint: 10 minutes ago

File Edit View Run Kernel Settings Help

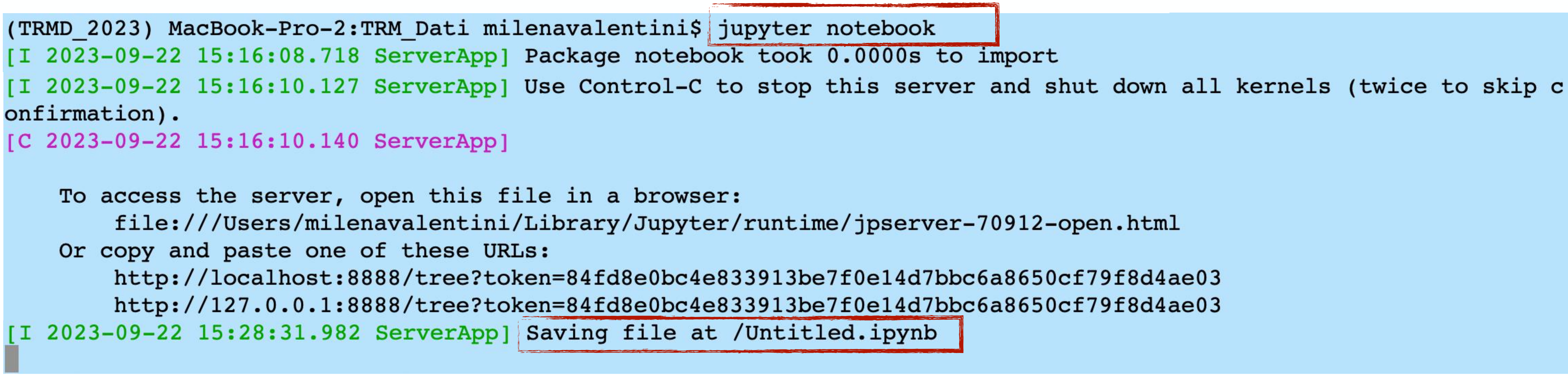
```
[4]: list_A
[5]: print(
     [1, 2,
     I can                er it
[ ]:
```

Run Selected Cell ⇧ ↩
Run Selected Cell and Insert Below ⌥ ↩
Run Selected Cell and Do not Advance ⌘ ↩
Run Selected Text or Current Line in Console
Run All Above Selected Cell
Run Selected Cell and All Below
Render All Markdown Cells
Run All Cells
Restart Kernel and Run All Cells...
Cell Type

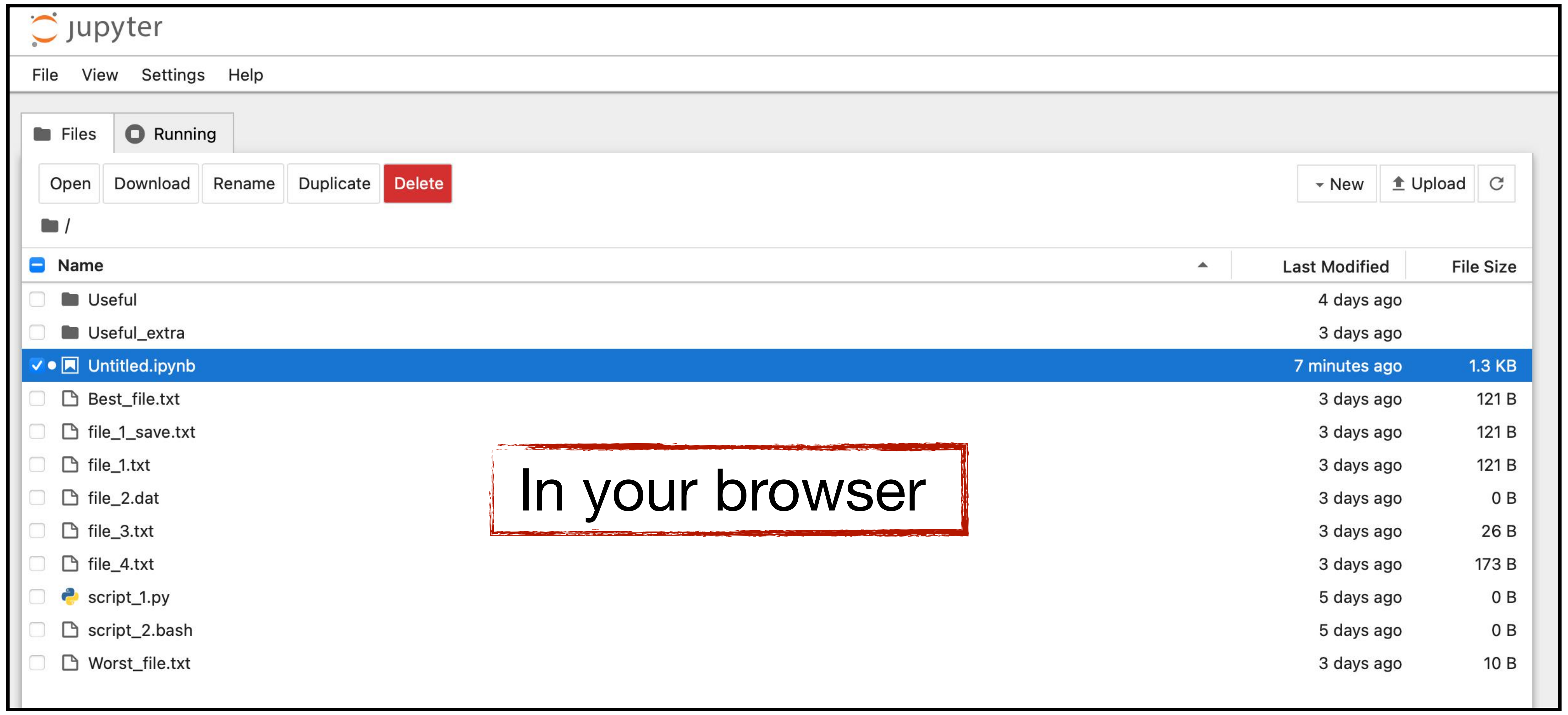
Setting up the working environment with Anaconda

Let’s use Anaconda via shell

Launch it:



```
(TRMD_2023) MacBook-Pro-2:TRM_Dati milenavalentini$ jupyter notebook
[I 2023-09-22 15:16:08.718 ServerApp] Package notebook took 0.0000s to import
[I 2023-09-22 15:16:10.127 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip c
onfirmation).
[C 2023-09-22 15:16:10.140 ServerApp]

    To access the server, open this file in a browser:
        file:///Users/milenavalentini/Library/Jupyter/runtime/jpserver-70912-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/tree?token=84fd8e0bc4e833913be7f0e14d7bbc6a8650cf79f8d4ae03
        http://127.0.0.1:8888/tree?token=84fd8e0bc4e833913be7f0e14d7bbc6a8650cf79f8d4ae03
[I 2023-09-22 15:28:31.982 ServerApp] Saving file at /Untitled.ipynb
```



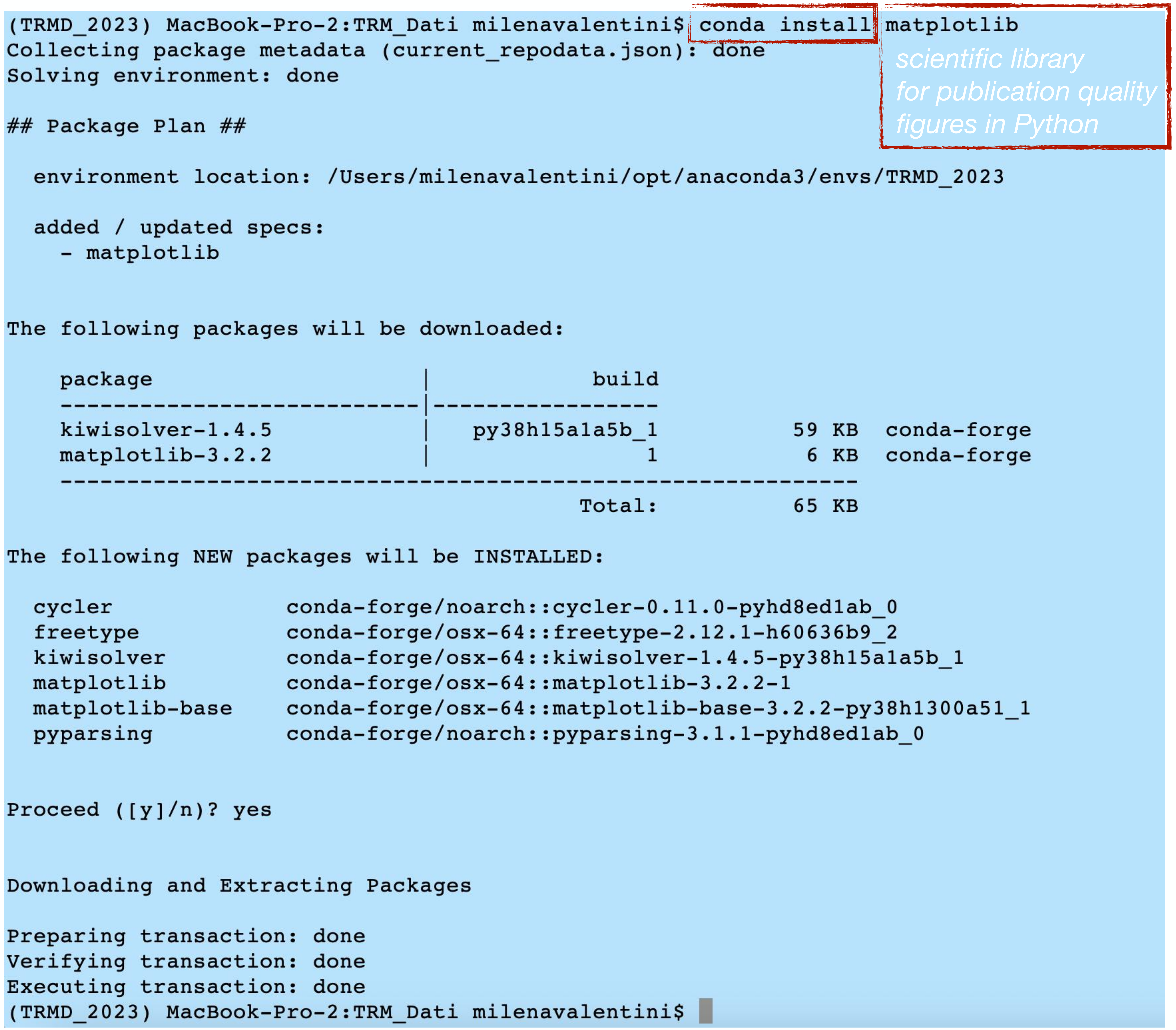
Name	Last Modified	File Size
Useful	4 days ago	
Useful_extra	3 days ago	
Untitled.ipynb	7 minutes ago	1.3 KB
Best_file.txt	3 days ago	121 B
file_1_save.txt	3 days ago	121 B
file_1.txt	3 days ago	121 B
file_2.dat	3 days ago	0 B
file_3.txt	3 days ago	26 B
file_4.txt	3 days ago	173 B
script_1.py	5 days ago	0 B
script_2.bash	5 days ago	0 B
Worst_file.txt	3 days ago	10 B

In your browser

Setting up the working environment with Anaconda

Let's use Anaconda via shell

Install libraries
within an active environment:



```
(TRMD_2023) MacBook-Pro-2:TRM_Dati milenavalentini$ conda install matplotlib
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /Users/milenavalentini/opt/anaconda3/envs/TRMD_2023

  added / updated specs:
    - matplotlib


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    kiwisolver-1.4.5           |   py38h15a1a5b_1          59 KB  conda-forge
    matplotlib-3.2.2           |                1           6 KB  conda-forge
    ------------------------------------------------------------
                                           Total:          65 KB

The following NEW packages will be INSTALLED:

  cycler             conda-forge/noarch::cycler-0.11.0-pyhd8ed1ab_0
  freetype           conda-forge/osx-64::freetype-2.12.1-h60636b9_2
  kiwisolver         conda-forge/osx-64::kiwisolver-1.4.5-py38h15a1a5b_1
  matplotlib         conda-forge/osx-64::matplotlib-3.2.2-1
  matplotlib-base    conda-forge/osx-64::matplotlib-base-3.2.2-py38h1300a51_1
  pyparsing          conda-forge/noarch::pyparsing-3.1.1-pyhd8ed1ab_0


Proceed ([y]/n)? yes


Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(TRMD_2023) MacBook-Pro-2:TRM_Dati milenavalentini$
```

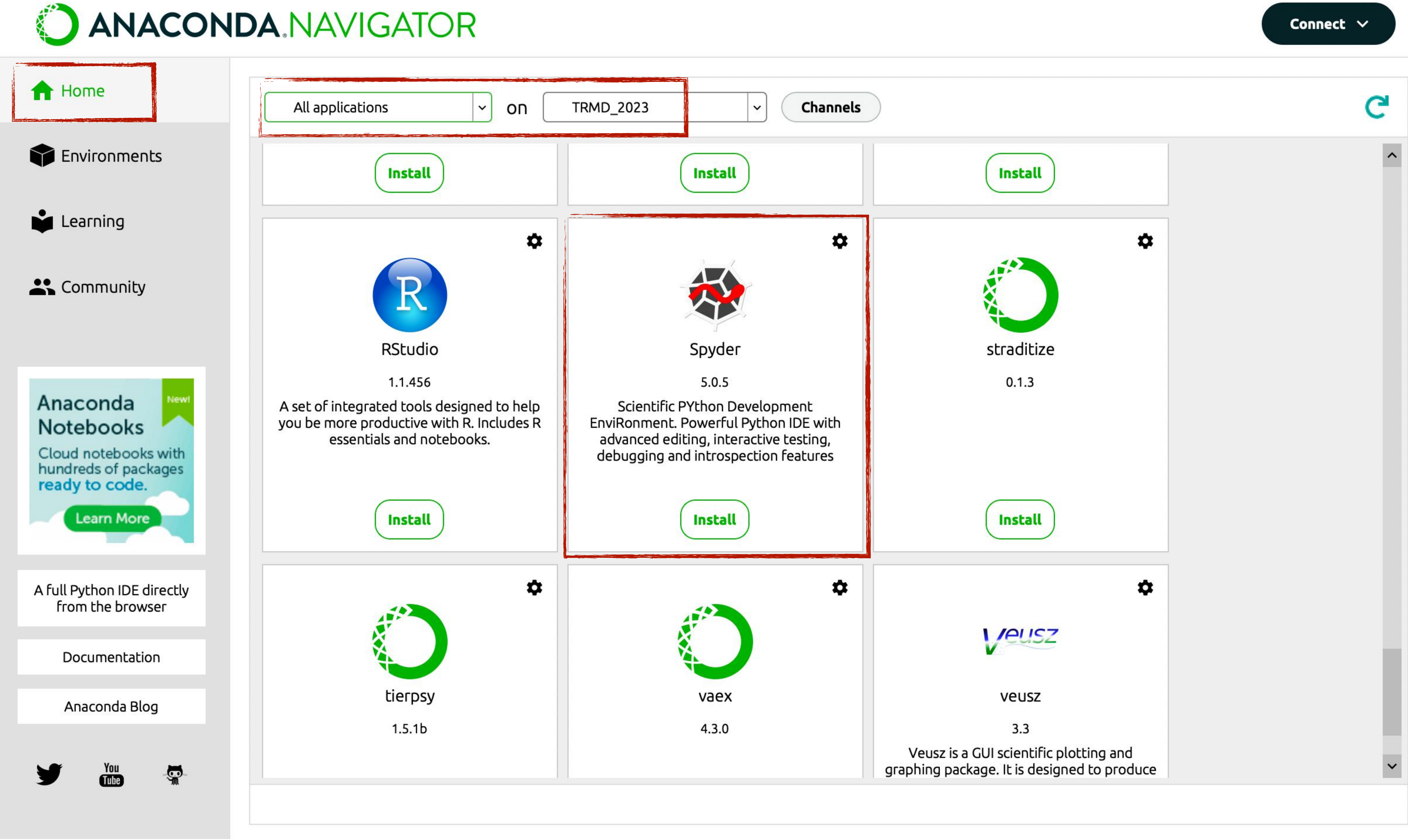
scientific library for publication quality figures in Python

Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

A very useful application which can be used as a code editor:


ANACONDA.NAVIGATOR
Connect
Home
Environments
Learning
Community
Anaconda Notebooks
New!
Cloud notebooks with hundreds of packages ready to code.
Learn More
A full Python IDE directly from the browser
Documentation
Anaconda Blog
All applications
on
TRMD_2023
Channels
Install
Install
Install
RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install
Spyder
5.0.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Install
straditize
0.1.3
Install
tierpsy
1.5.1b
vaex
4.3.0
veusz
3.3
Veusz is a GUI scientific plotting and graphing package. It is designed to produce

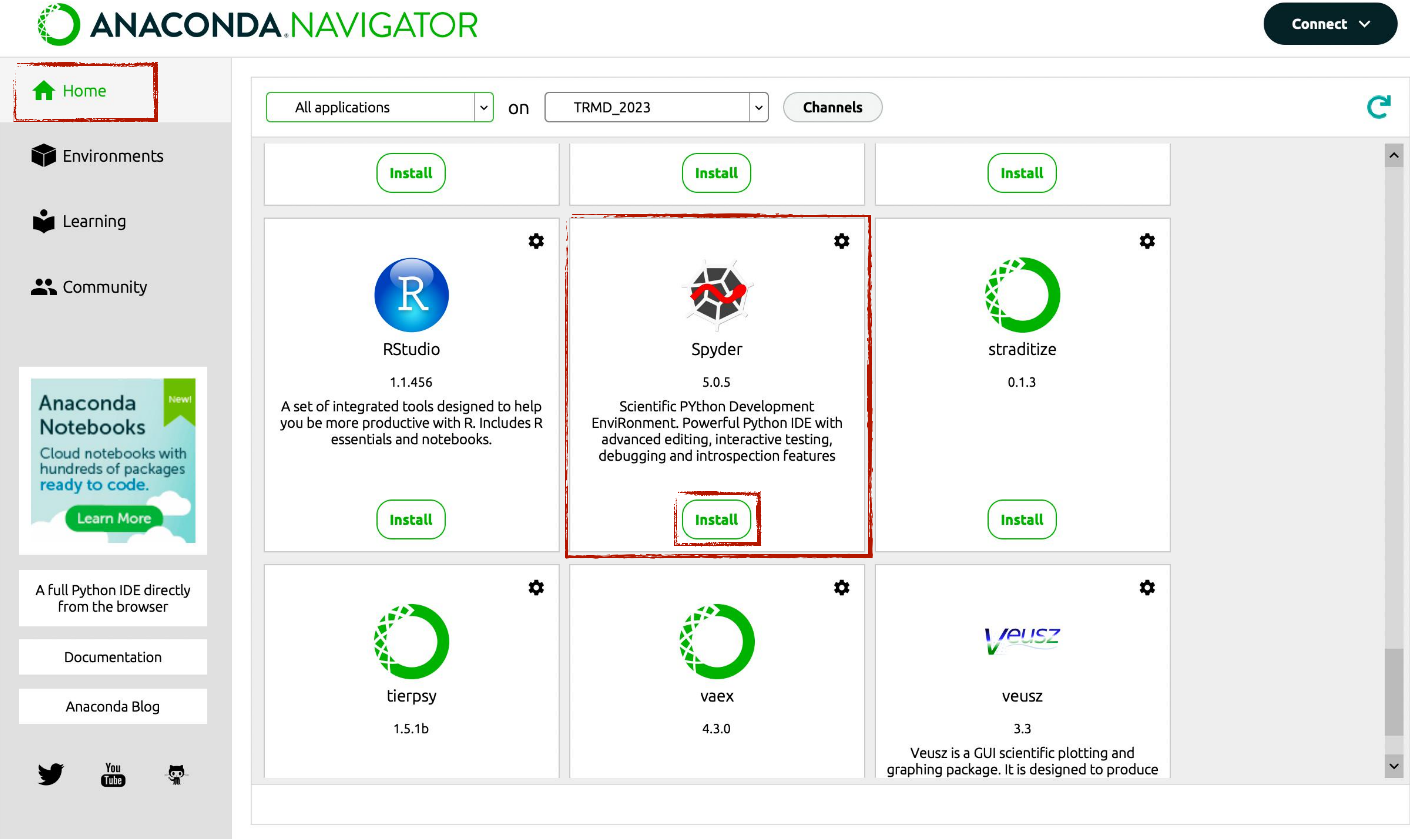
Setting up the working environment with Anaconda

To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

A very useful application which can be used as a code editor:

Select the applications to be installed in the environment among available ones



A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Veusz is a GUI scientific plotting and graphing package. It is designed to produce

Setting up the working environment with Anaconda

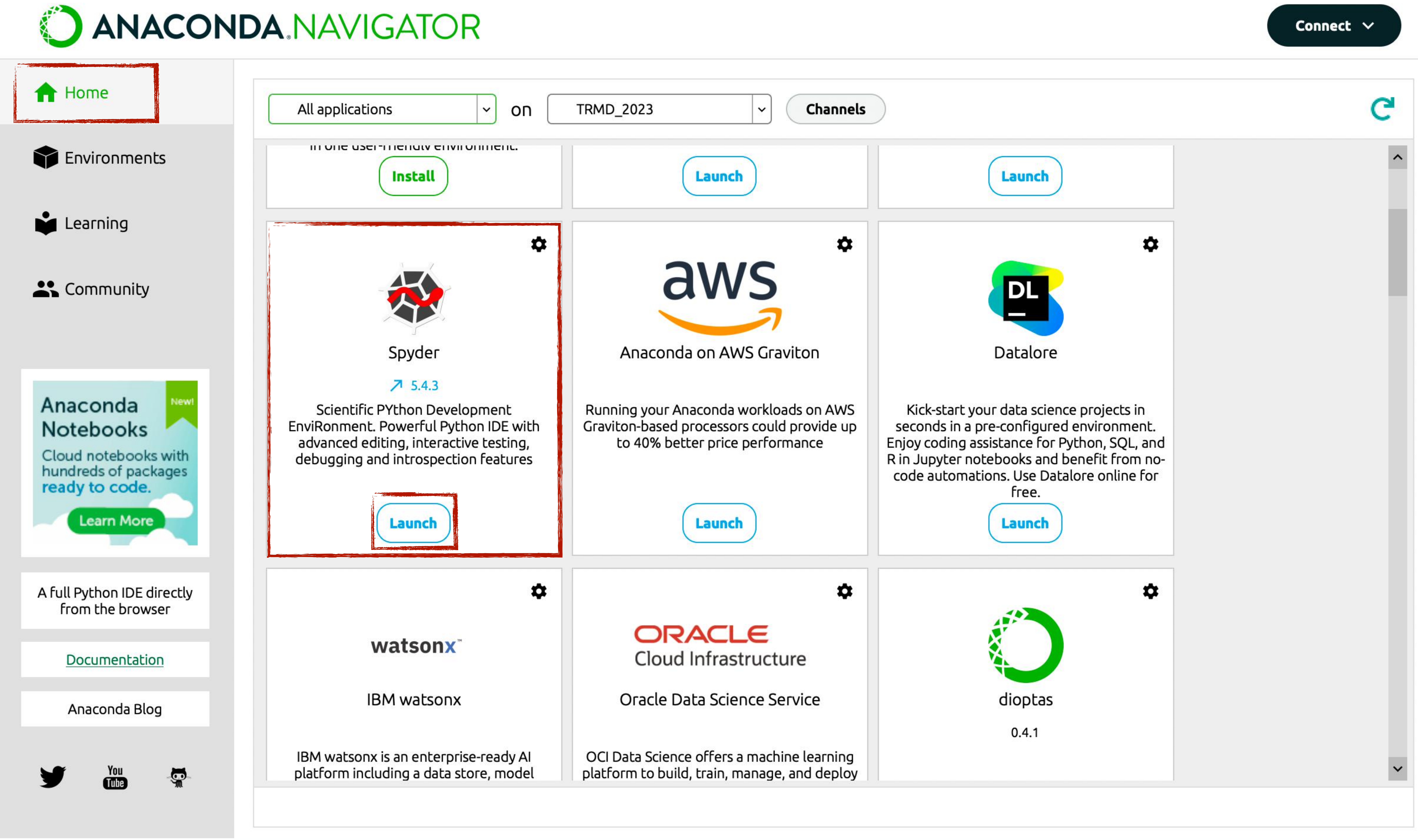
To exploit it via its graphical unit interface:

Launch Anaconda-Navigator:

A very useful application which can be used as a code editor:

Select the applications to be installed in the environment

The application has just been installed and can be launched



All applications on TRMD_2023

in one user-friendly environment.

Spyder 5.4.3

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Anaconda on AWS Graviton

Running your Anaconda workloads on AWS Graviton-based processors could provide up to 40% better price performance

Datalore

Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.

IBM watsonx

IBM watsonx is an enterprise-ready AI platform including a data store, model

Oracle Data Science Service

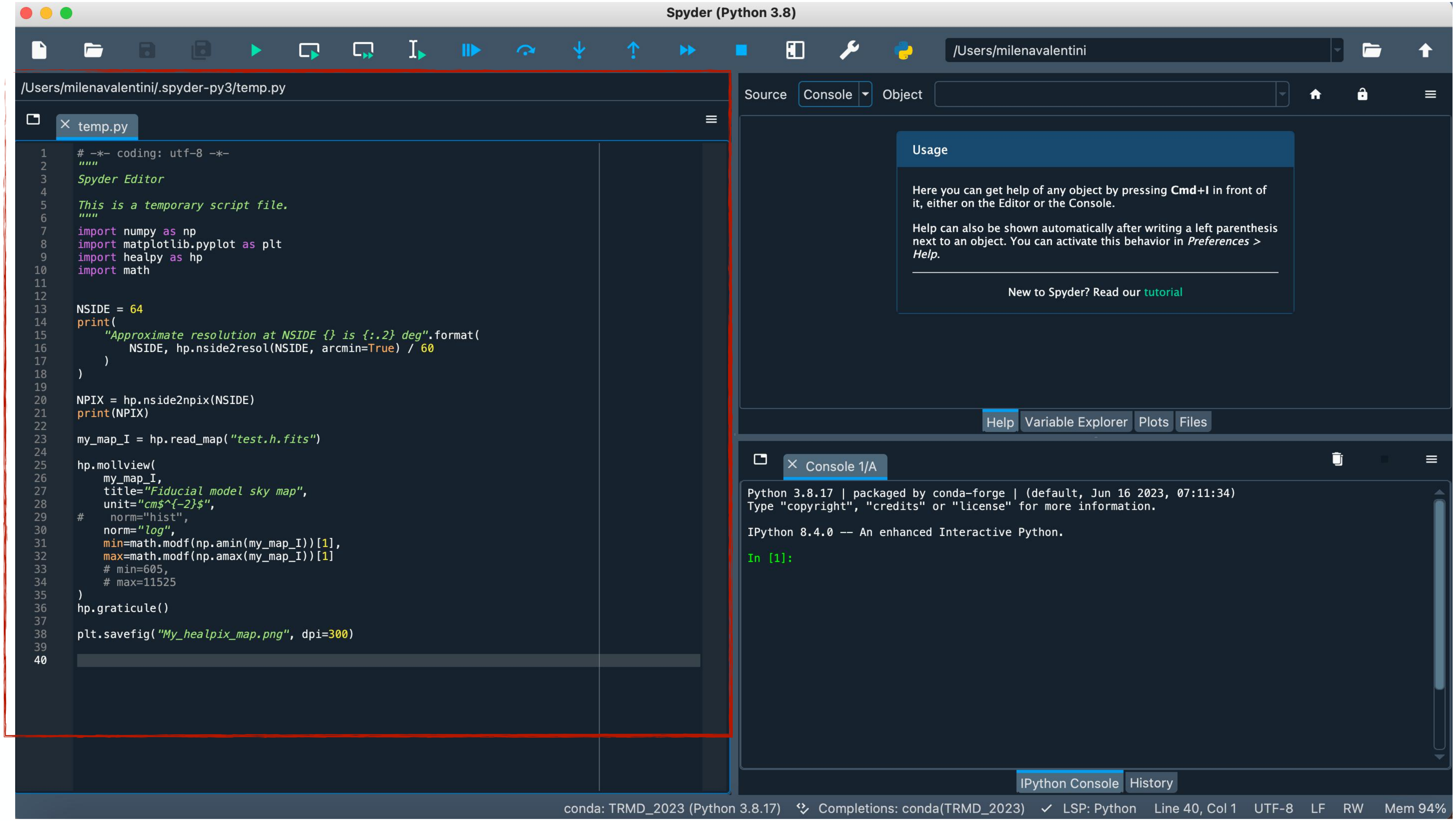
OCI Data Science offers a machine learning platform to build, train, manage, and deploy

dioptas 0.4.1

Anaconda Notebooks: Cloud notebooks with hundreds of packages ready to code.

A full Python IDE directly from the browser

Spyder



```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import numpy as np
import matplotlib.pyplot as plt
import healpy as hp
import math


NSIDE = 64
print(
    "Approximate resolution at NSIDE {} is {:.2} deg".format(
        NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
    )
)

NPIX = hp.nside2npix(NSIDE)
print(NPIX)

my_map_I = hp.read_map("test.h.fits")

hp.mollview(
    my_map_I,
    title="Fiducial model sky map",
    unit="cm$^{-2}$",
#    norm="hist",
    norm="log",
    min=math.modf(np.amin(my_map_I))[1],
    max=math.modf(np.amax(my_map_I))[1]
    # min=605,
    # max=11525
)
hp.graticule()

plt.savefig("My_healpix_map.png", dpi=300)

```

Usage

Here you can get help of any object by pressing **Cmd+I** in front of it, either on the Editor or the Console.

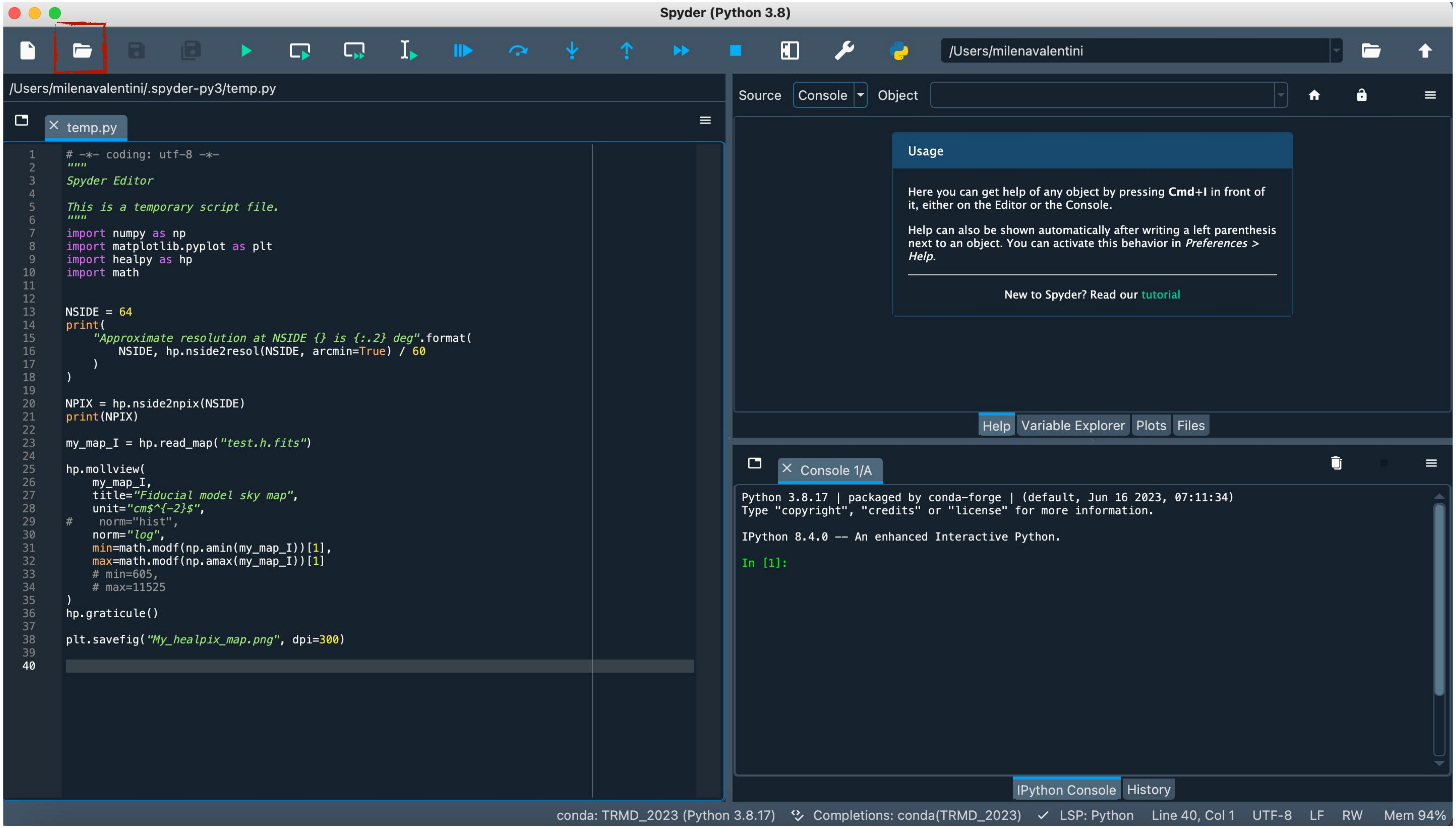
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Python 3.8.17 | packaged by conda-forge | (default, Jun 16 2023, 07:11:34)
Type "copyright", "credits" or "license" for more information.

IPython 8.4.0 -- An enhanced Interactive Python.

In [1]:
```

conda: TRMD_2023 (Python 3.8.17)   Completions: conda(TRMD_2023)   ✓ LSP: Python   Line 40, Col 1   UTF-8   LF   RW   Mem 94%

Spyder



```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import numpy as np
import matplotlib.pyplot as plt
import healpy as hp
import math


NSIDE = 64
print(
    "Approximate resolution at NSIDE {} is {:.2} deg".format(
        NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
    )
)

NPIX = hp.nside2npix(NSIDE)
print(NPIX)

my_map_I = hp.read_map("test.h.fits")

hp.mollview(
    my_map_I,
    title="Fiducial model sky map",
    unit="cm$^{-2}$",
#    norm="hist",
    norm="log",
    min=math.modf(np.amin(my_map_I))[1],
    max=math.modf(np.amax(my_map_I))[1]
    # min=605,
    # max=11525
)
hp.graticule()

plt.savefig("My_healpix_map.png", dpi=300)

```

Usage

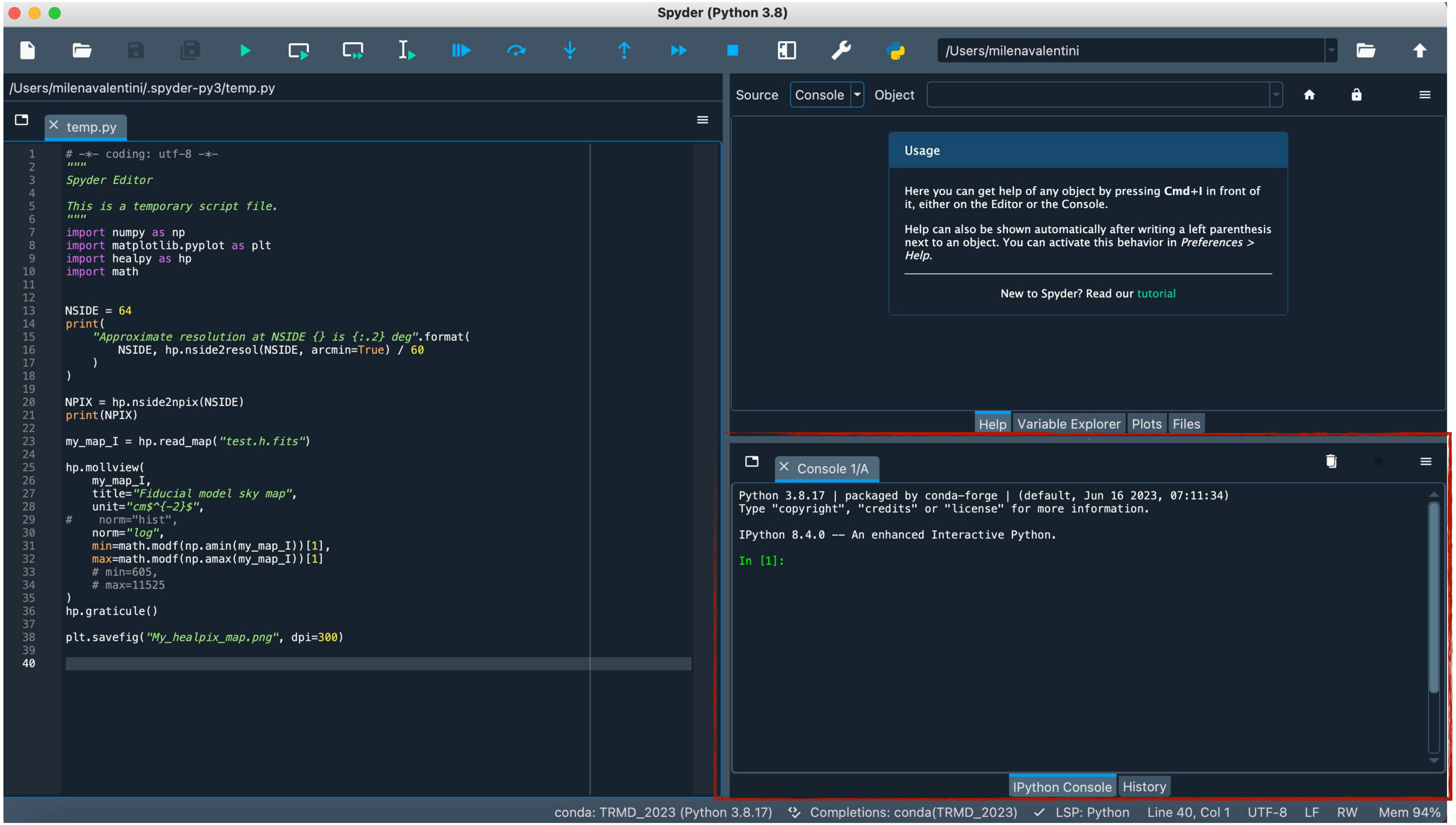
Here you can get help of any object by pressing **Cmd+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Python 3.8.17 | packaged by conda-forge | (default, Jun 16 2023, 07:11:34)
Type "copyright", "credits" or "license" for more information.

IPython 8.4.0 -- An enhanced Interactive Python.

In [1]:
```

Spyder



```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import numpy as np
import matplotlib.pyplot as plt
import healpy as hp
import math


NSIDE = 64
print(
    "Approximate resolution at NSIDE {} is {:.2} deg".format(
        NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
    )
)

NPIX = hp.nside2npix(NSIDE)
print(NPIX)

my_map_I = hp.read_map("test.h.fits")

hp.mollview(
    my_map_I,
    title="Fiducial model sky map",
    unit="cm$^{-2}$",
#    norm="hist",
    norm="log",
    min=math.modf(np.amin(my_map_I))[1],
    max=math.modf(np.amax(my_map_I))[1]
    # min=605,
    # max=11525
)
hp.graticule()

plt.savefig("My_healpix_map.png", dpi=300)

```

Usage

Here you can get help of any object by pressing **Cmd+I** in front of it, either on the Editor or the Console.

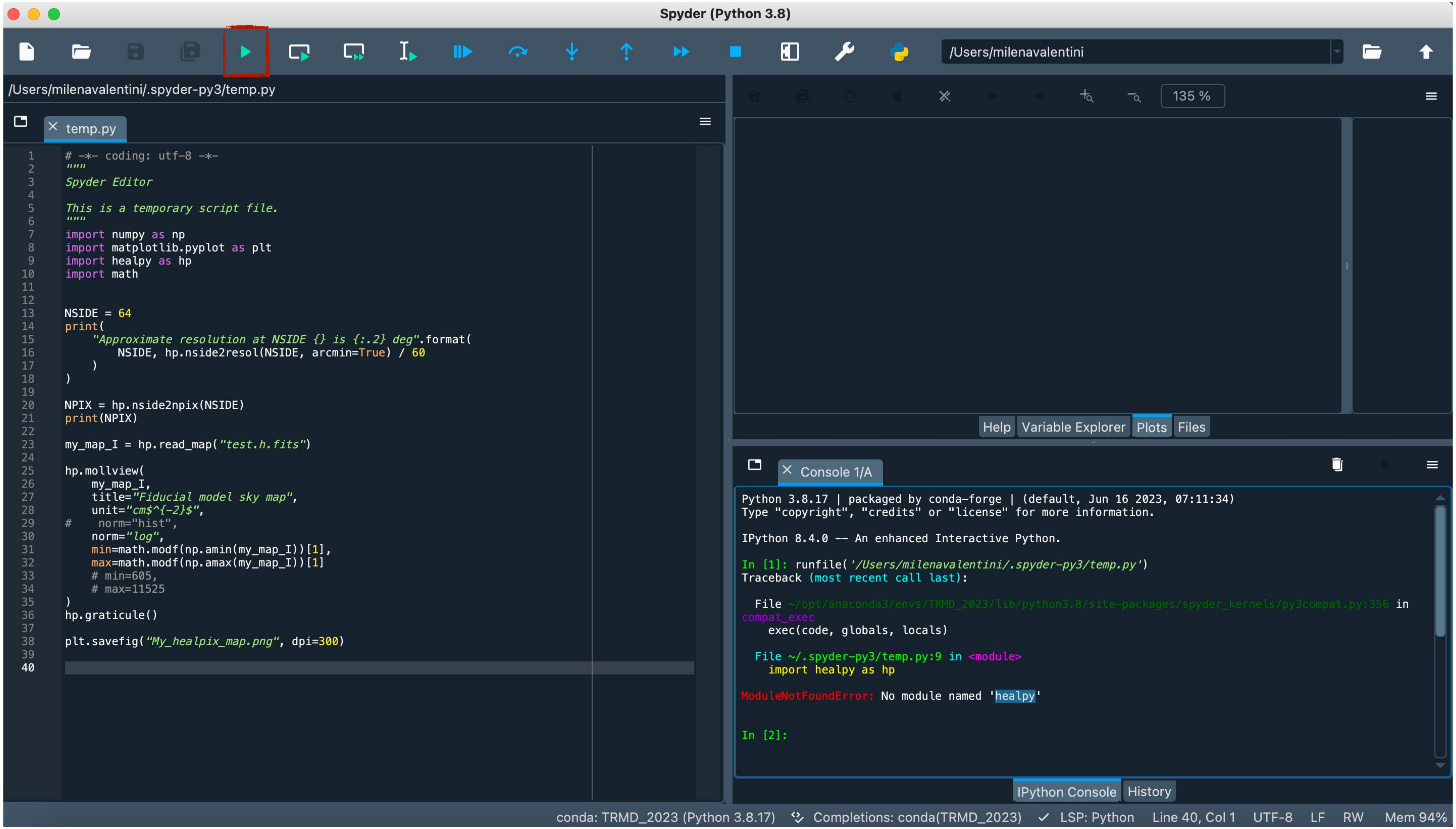
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Python 3.8.17 | packaged by conda-forge | (default, Jun 16 2023, 07:11:34)
Type "copyright", "credits" or "license" for more information.

IPython 8.4.0 -- An enhanced Interactive Python.

In [1]:
```

conda: TRMD_2023 (Python 3.8.17)   Completions: conda(TRMD_2023)   ✓ LSP: Python   Line 40, Col 1   UTF-8   LF   RW   Mem 94%

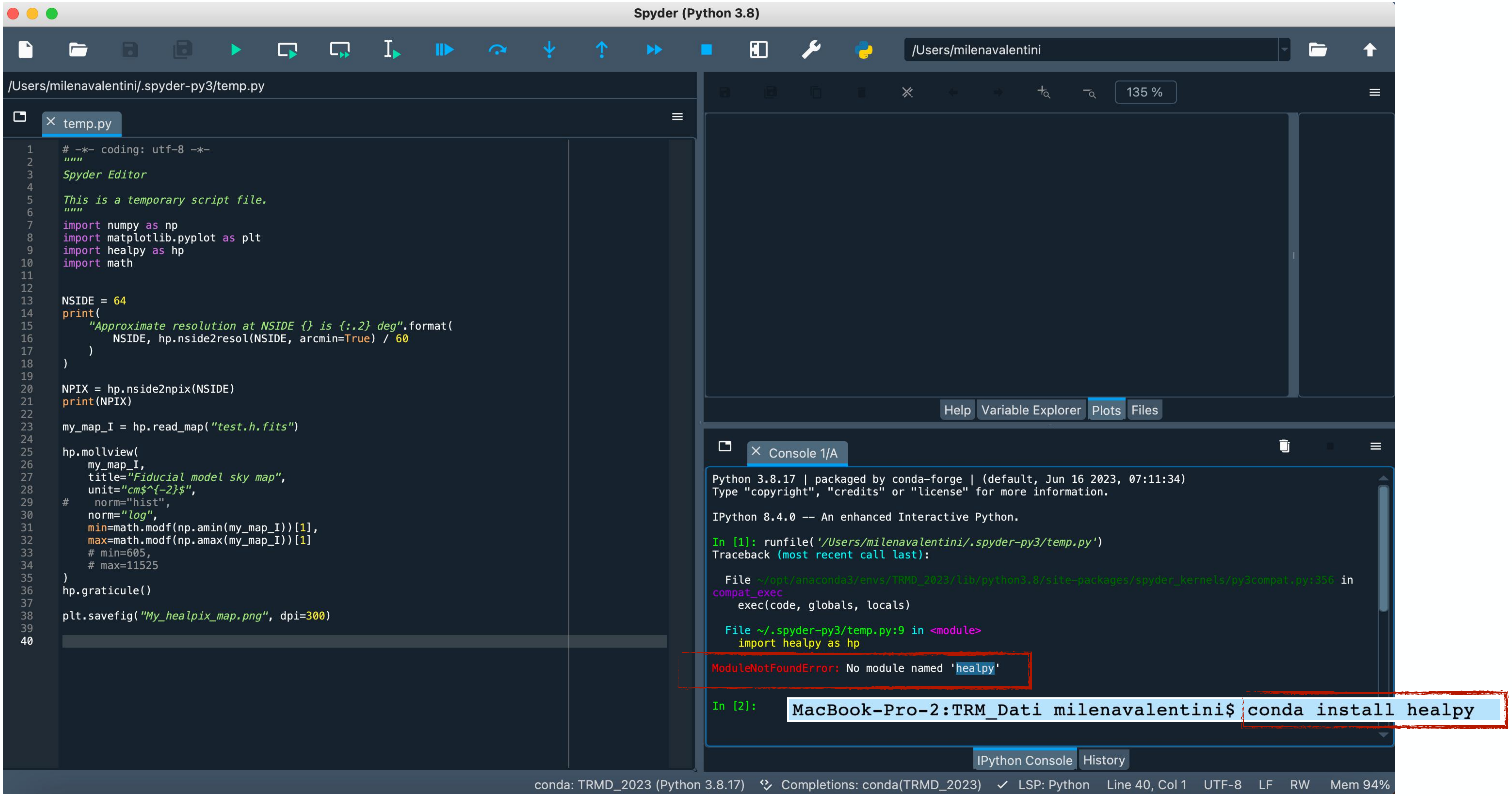
Spyder



```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import numpy as np
import matplotlib.pyplot as plt
import healpy as hp
import math


NSIDE = 64
print(
    "Approximate resolution at NSIDE {} is {:.2} deg".format(
        NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
    )
)

NPIX = hp.nside2npix(NSIDE)
print(NPIX)

my_map_I = hp.read_map("test.h.fits")

hp.mollview(
    my_map_I,
    title="Fiducial model sky map",
    unit="cm$^{-2}$",
#    norm="hist",
    norm="log",
    min=math.modf(np.amin(my_map_I))[1],
    max=math.modf(np.amax(my_map_I))[1]
    # min=605,
    # max=11525
)
hp.graticule()

plt.savefig("My_healpix_map.png", dpi=300)

```

```
Python 3.8.17 | packaged by conda-forge | (default, Jun 16 2023, 07:11:34)
Type "copyright", "credits" or "license" for more information.

IPython 8.4.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/milenavalentini/.spyder-py3/temp.py')
Traceback (most recent call last):

  File ~/opt/anaconda3/envs/TRMD_2023/lib/python3.8/site-packages/spyder_kernels/py3compat.py:356 in
compat_exec
    exec(code, globals, locals)

  File ~/.spyder-py3/temp.py:9 in <module>
    import healpy as hp

ModuleNotFoundError: No module named 'healpy'


In [2]:
```

Spyder

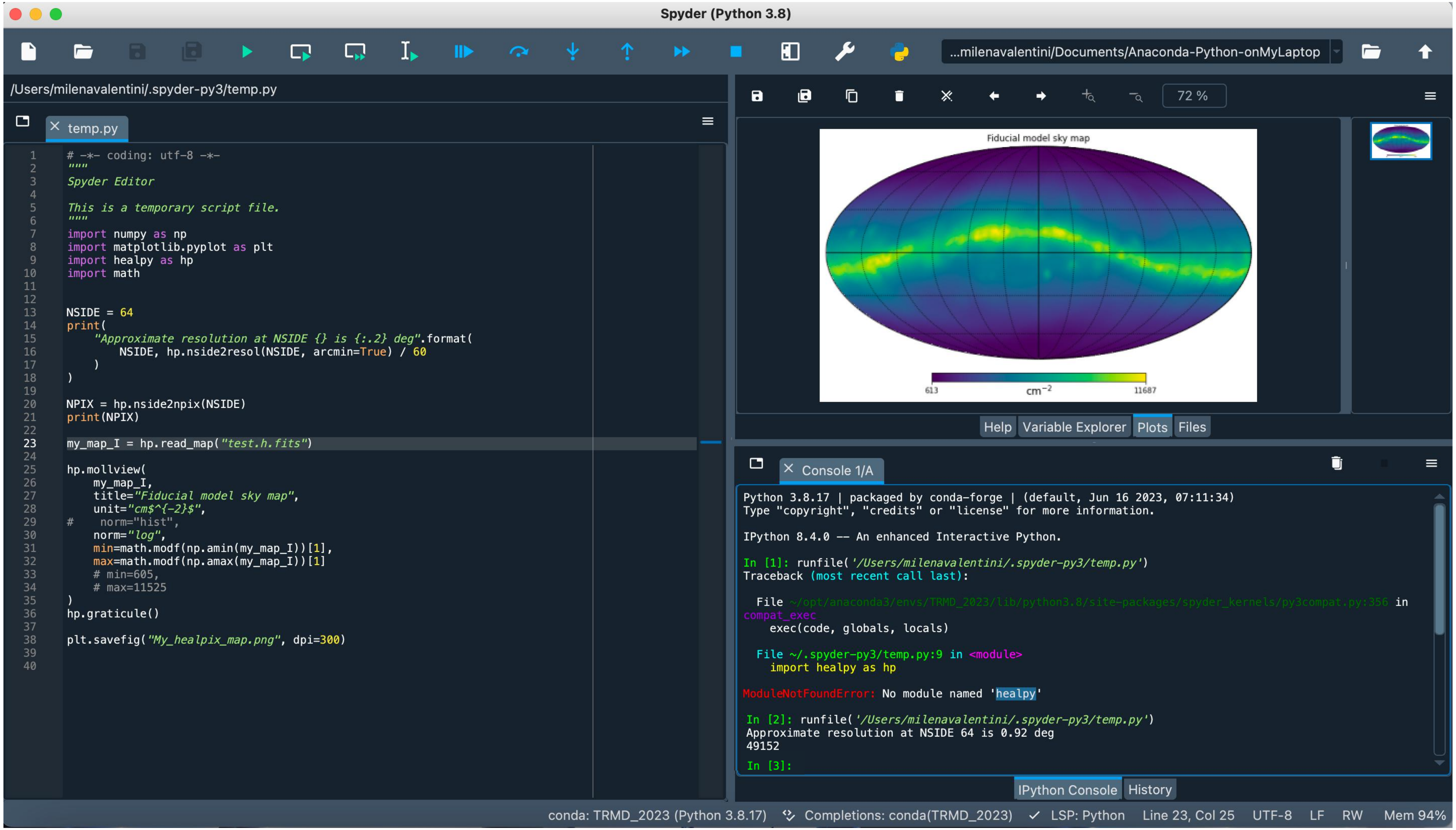


```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import numpy as np
import matplotlib.pyplot as plt
import healpy as hp
import math


NSIDE = 64
print(
    "Approximate resolution at NSIDE {} is {:.2} deg".format(
        NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
    )
)

NPIX = hp.nside2npix(NSIDE)
print(NPIX)

my_map_I = hp.read_map("test.h.fits")

hp.mollview(
    my_map_I,
    title="Fiducial model sky map",
    unit="cm$^{-2}$",
#    norm="hist",
    norm="log",
    min=math.modf(np.amin(my_map_I))[1],
    max=math.modf(np.amax(my_map_I))[1]
    # min=605,
    # max=11525
)
hp.graticule()

plt.savefig("My_healpix_map.png", dpi=300)

```

```
Python 3.8.17 | packaged by conda-forge | (default, Jun 16 2023, 07:11:34)
Type "copyright", "credits" or "license" for more information.

IPython 8.4.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/milenavalentini/.spyder-py3/temp.py')
Traceback (most recent call last):

  File ~/opt/anaconda3/envs/TRMD_2023/lib/python3.8/site-packages/spyder_kernels/py3compat.py:356 in
compat_exec
    exec(code, globals, locals)

  File ~/.spyder-py3/temp.py:9 in <module>
    import healpy as hp

ModuleNotFoundError: No module named 'healpy'

In [2]:
```

```
MacBook-Pro-2:TRM_Dati milenavalentini$ conda install healpy
```

Spyder



```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import numpy as np
import matplotlib.pyplot as plt
import healpy as hp
import math


NSIDE = 64
print(
    "Approximate resolution at NSIDE {} is {:.2} deg".format(
        NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
    )
)

NPIX = hp.nside2npix(NSIDE)
print(NPIX)

my_map_I = hp.read_map("test.h.fits")

hp.mollview(
    my_map_I,
    title="Fiducial model sky map",
    unit="cm$^{-2}$",
#    norm="hist",
    norm="log",
    min=math.modf(np.amin(my_map_I))[1],
    max=math.modf(np.amax(my_map_I))[1]
    # min=605,
    # max=11525
)
hp.graticule()

plt.savefig("My_healpix_map.png", dpi=300)
```

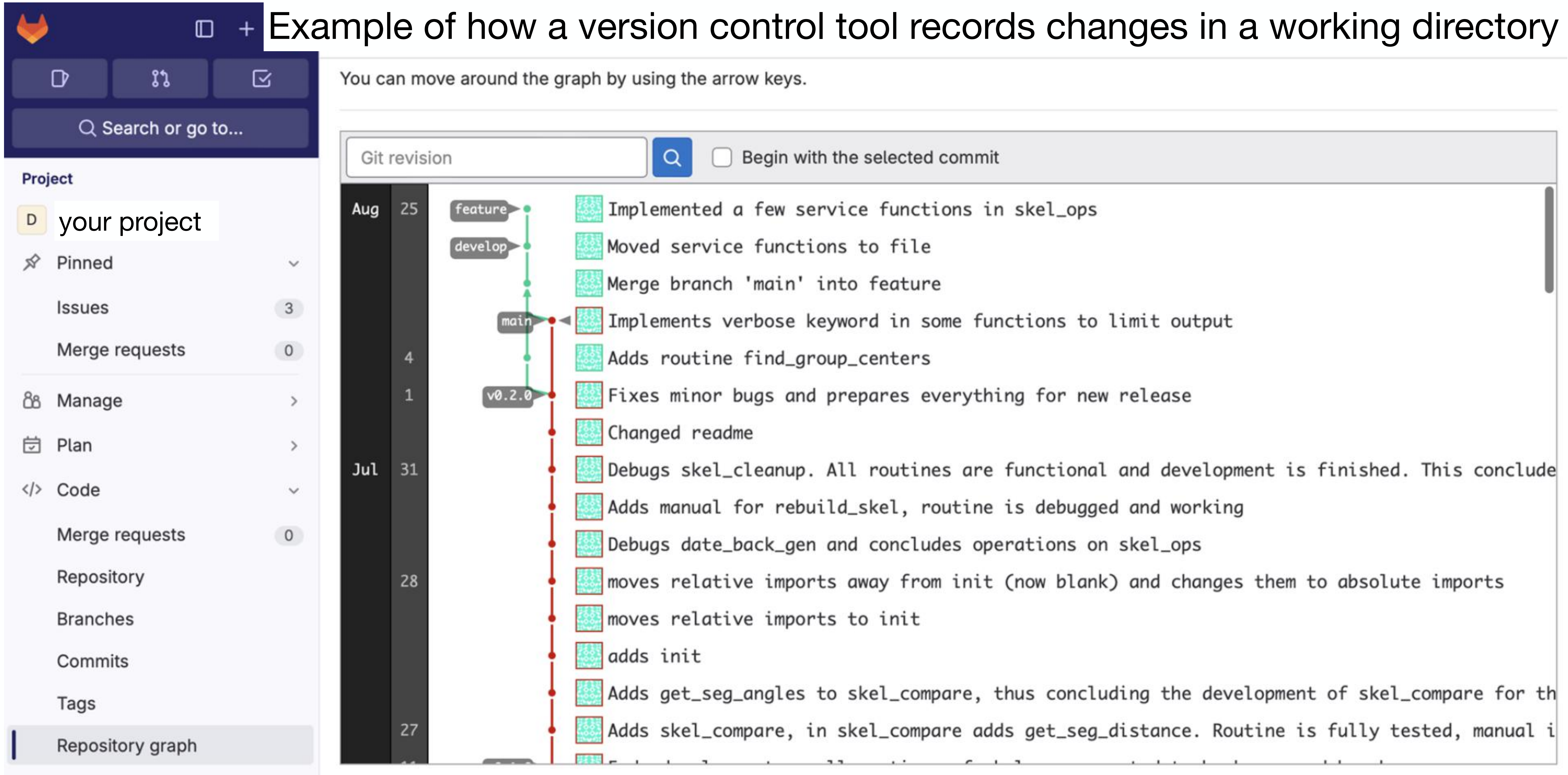
```
Python 3.8.17 | packaged by conda-forge | (default, Jun 16 2023, 07:11:34)
Type "copyright", "credits" or "license" for more information.

IPython 8.4.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/milenavalentini/.spyder-py3/temp.py')
Traceback (most recent call last):

  File ~/opt/anaconda3/envs/TRMD_2023/lib/python3.8/site-packages/spyder_kernels/py3compat.py:356 in
compat_exec
    exec(code, globals, locals)

  File ~/.spyder-py3/temp.py:9 in <module>
    import healpy as hp

ModuleNotFoundError: No module named 'healpy'

In [2]: runfile('/Users/milenavalentini/.spyder-py3/temp.py')
Approximate resolution at NSIDE 64 is 0.92 deg
49152

In [3]:
```

Other tools: version control system

Version control system (e.g., GIT): allows you to keep track of changes, useful as a collaborative tool

Repositories are (working) directories hosting your projects.

Example of how a version control tool records changes in a working directory

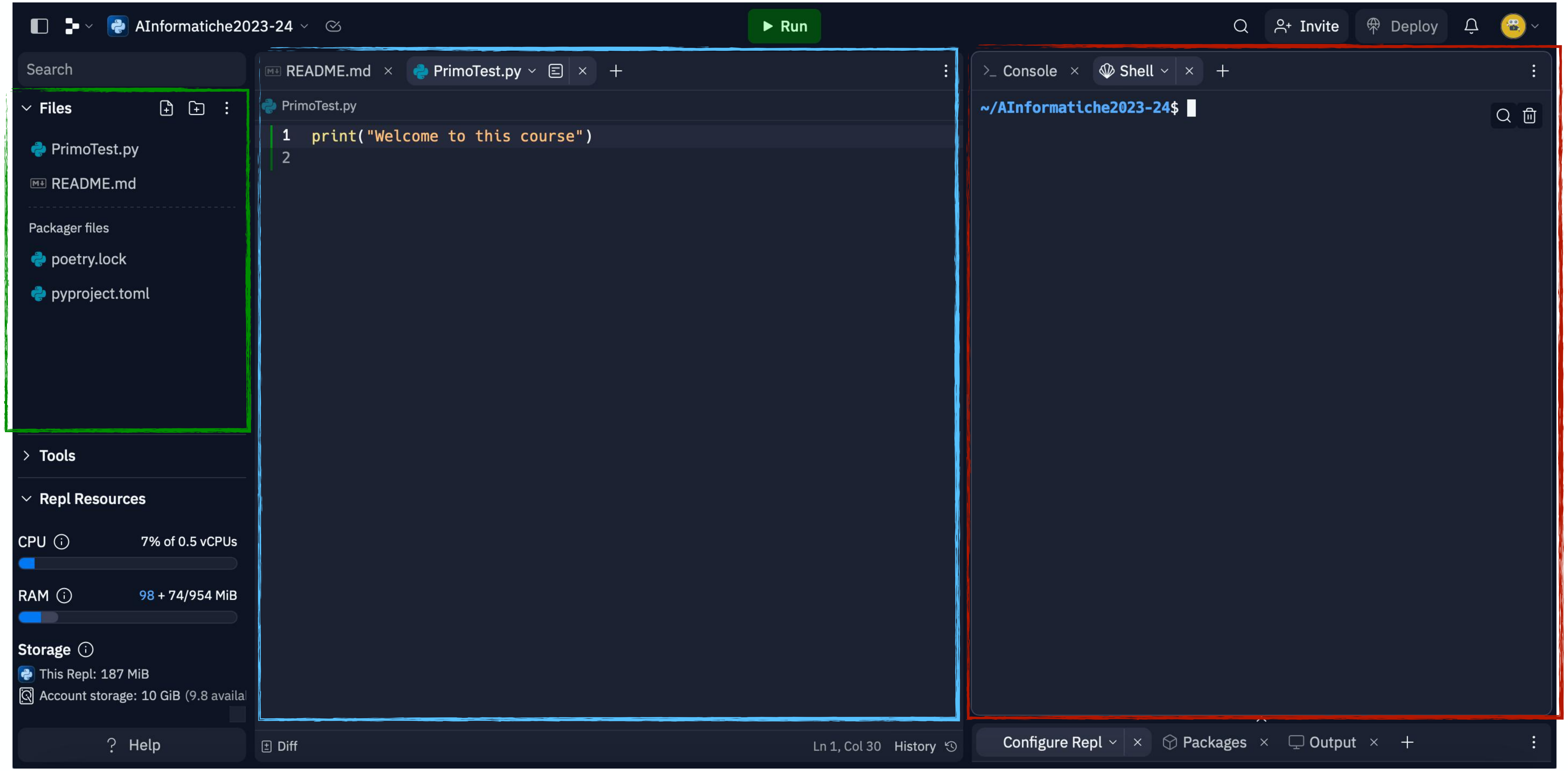


You can move around the graph by using the arrow keys.

```
Aug 25  Implemented a few service functions in skel_ops
        Moved service functions to file
        Merge branch 'main' into feature
        Implements verbose keyword in some functions to limit output
     4  Adds routine find_group_centers
     1  Fixes minor bugs and prepares everything for new release
        Changed readme
Jul 31  Debugs skel_cleanup. All routines are functional and development is finished. This conclude
        Adds manual for rebuild_skel, routine is debugged and working
        Debugs date_back_gen and concludes operations on skel_ops
    28  moves relative imports away from init (now blank) and changes them to absolute imports
        moves relative imports to init
        adds init
        Adds get_seg_angles to skel_compare, thus concluding the development of skel_compare for th
    27  Adds skel_compare, in skel_compare adds get_seg_distance. Routine is fully tested, manual i
```

Other tools: Integrated Development Environment

A tool which displays all together a file manager, a code editor, a version control system, a console and other additional functionalities like for e.g. a debugger.



```
print("Welcome to this course")
```

```
~/AInformatiche2023-24$
```

CPU 7% of 0.5 vCPUs

RAM 98 + 74/954 MiB

Storage

This Repl: 187 MiB

Account storage: 10 GiB (9.8 availal

Other tools: Integrated Development Environment

Setup a working environment:

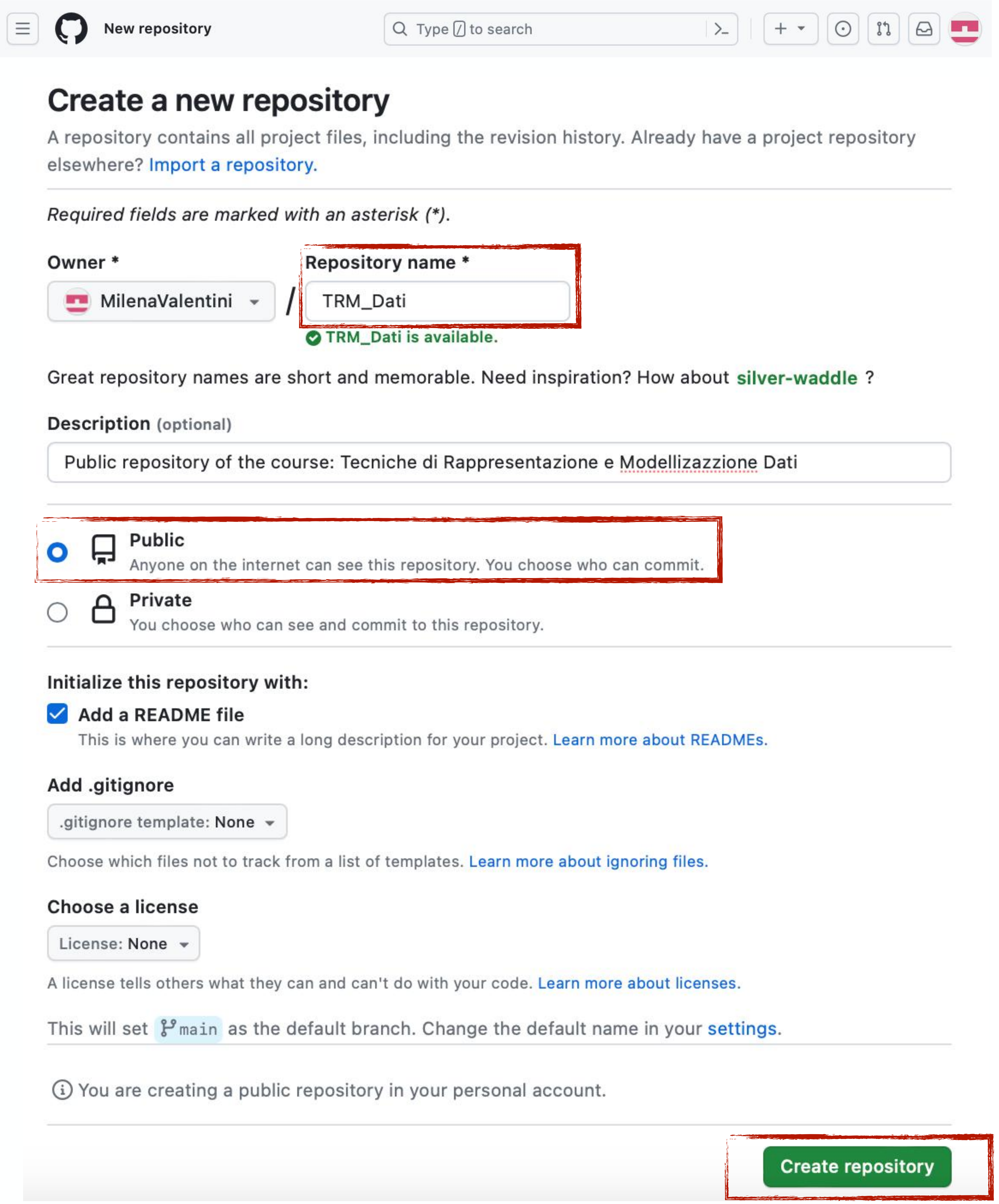
Register on Git (github.com)

Other tools: Integrated Development Environment

Setup a working environment:

Register on Git

Create a public repository on Git



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner *: MilenaValentini / Repository name *: TRM_Dati

TRM_Dati is available.

Great repository names are short and memorable. Need inspiration? How about **silver-waddle** ?

Description (optional)

Public repository of the course: Tecniche di Rappresentazione e Modellizzazione Dati

● **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☑ **Add a README file**
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

This will set `main` as the default branch. Change the default name in your settings.

You are creating a public repository in your personal account.

Other tools: Integrated Development Environment

Setup a working environment:

Register on Git

Create a public repository on Git

Register on repl.it (replit.com)

Other tools: Integrated Development Environment

Setup a working environment:

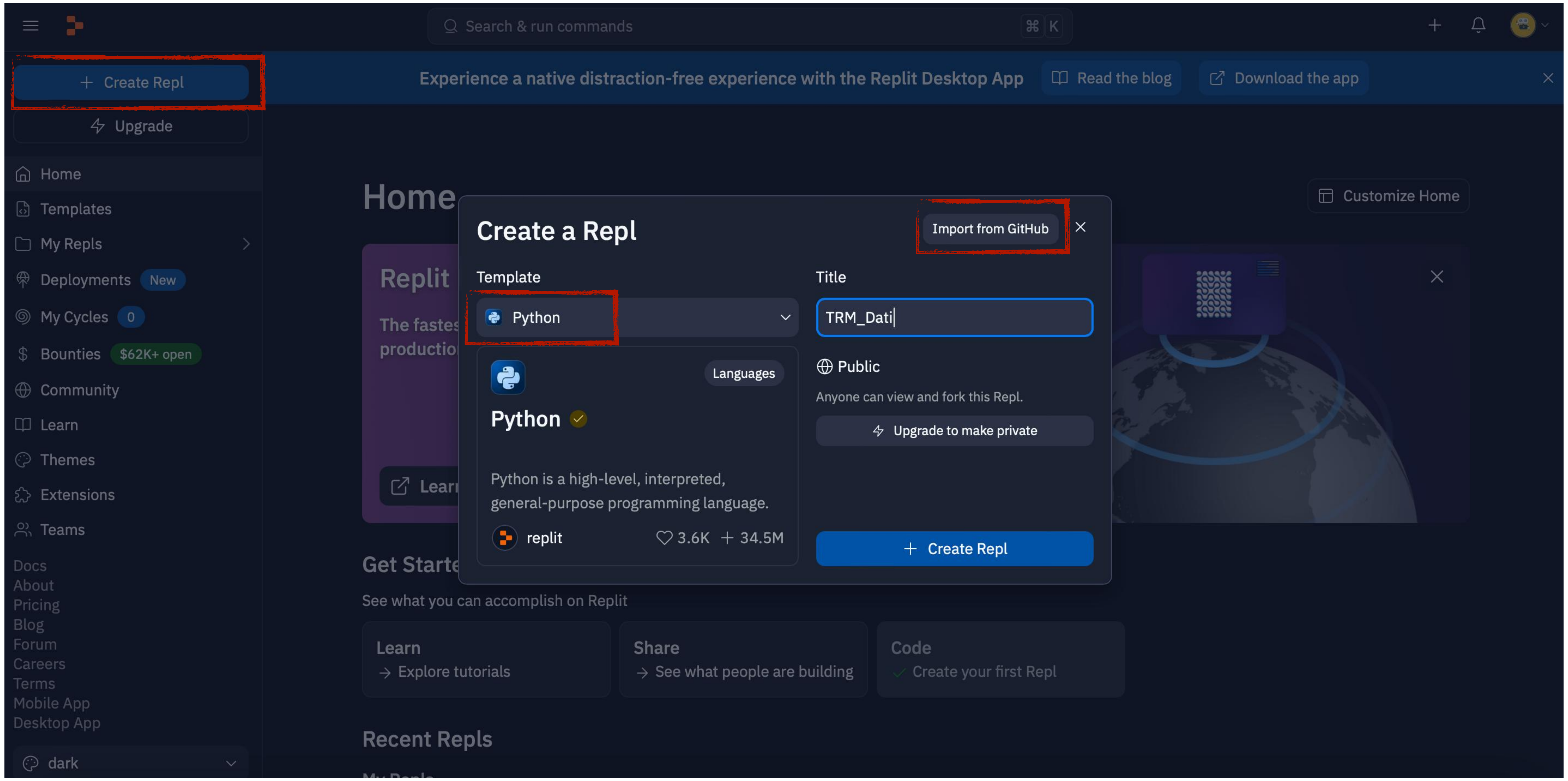
Register on Git

Create a public repository on Git

Register on repl.it (replit.com)

Create a new repl project by importing from Git the repository you have just created there

Set Python as default language



Experience a native distraction-free experience with the Replit Desktop App

Create a Repl

Template: Python

Title: TRM_Dati

Public

Anyone can view and fork this Repl.

Python

Python is a high-level, interpreted, general-purpose programming language.

Get Started

See what you can accomplish on Replit

Other tools: Integrated Development Environment

Setup a working environment:

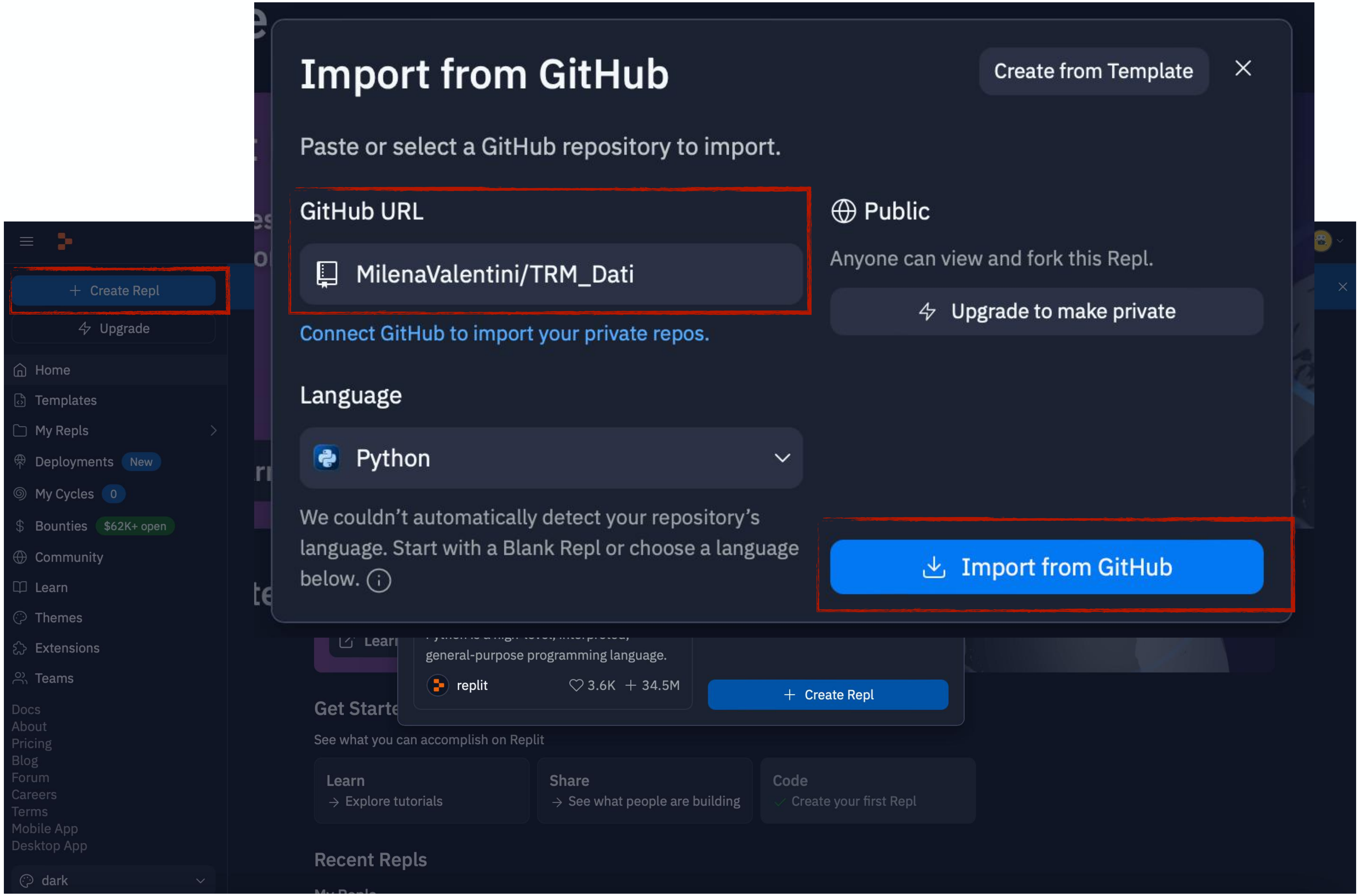
Register on Git

Create a public repository on Git

Register on repl.it (replit.com)

Create a new repl project by importing from Git the repository you have just created there

Set Python as default language



Import from GitHub

Paste or select a GitHub repository to import.

GitHub URL: MilenaValentini/TRM_Dati

Connect GitHub to import your private repos.

Public

Anyone can view and fork this Repl.

Language: Python

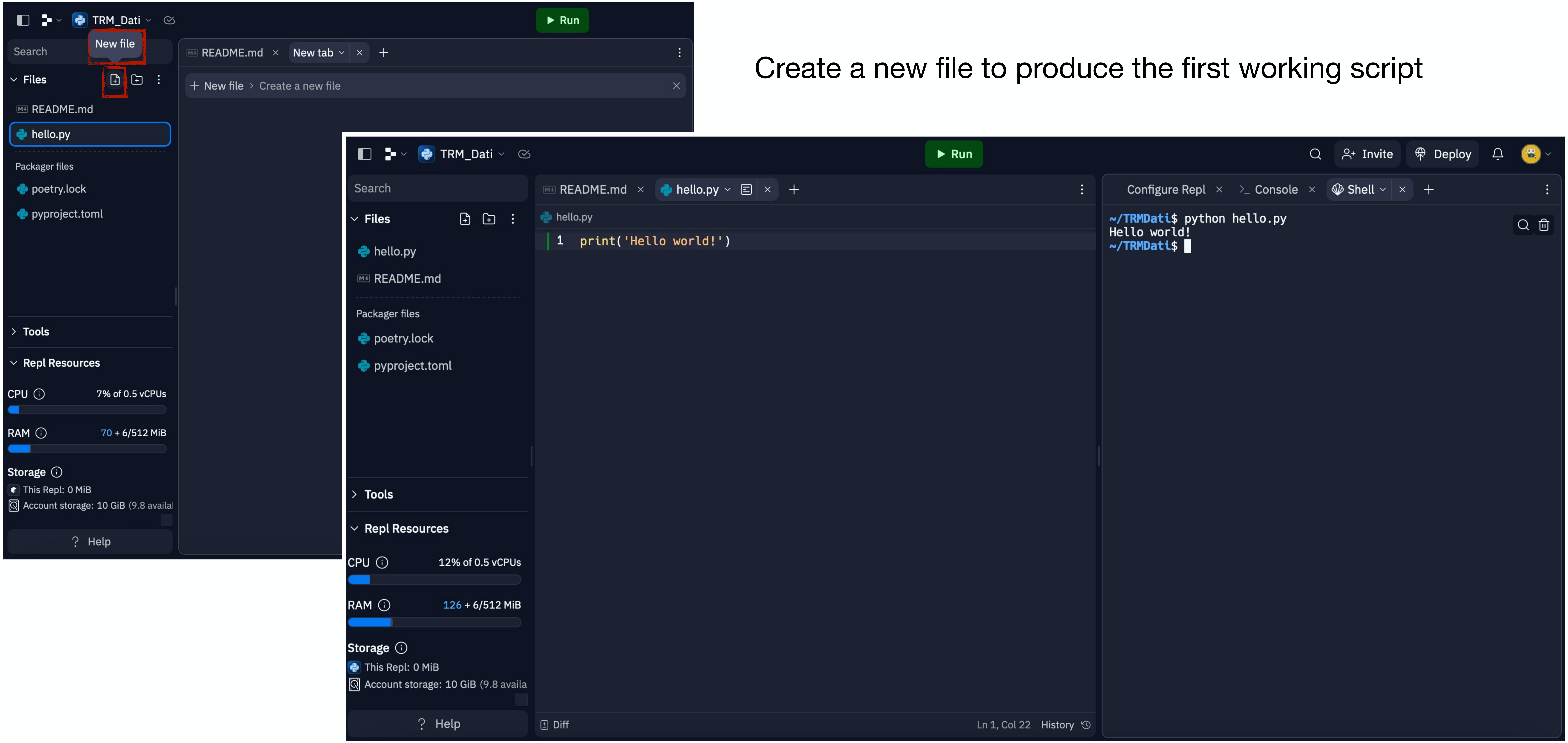
We couldn't automatically detect your repository's language. Start with a Blank Repl or choose a language below.

general-purpose programming language.

See what you can accomplish on Replit

Other tools: Integrated Development Environment

Create a new file to produce the first working script



hello.py

```
print('Hello world!')
```

```
~/TRMDati$ python hello.py
Hello world!
~/TRMDati$ 
```