

# 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=14486>

Codice Teams del corso: d2cmkh8

# 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 confirmation).
[C 2023-09-22 15:16:10.140 ServerApp]
```

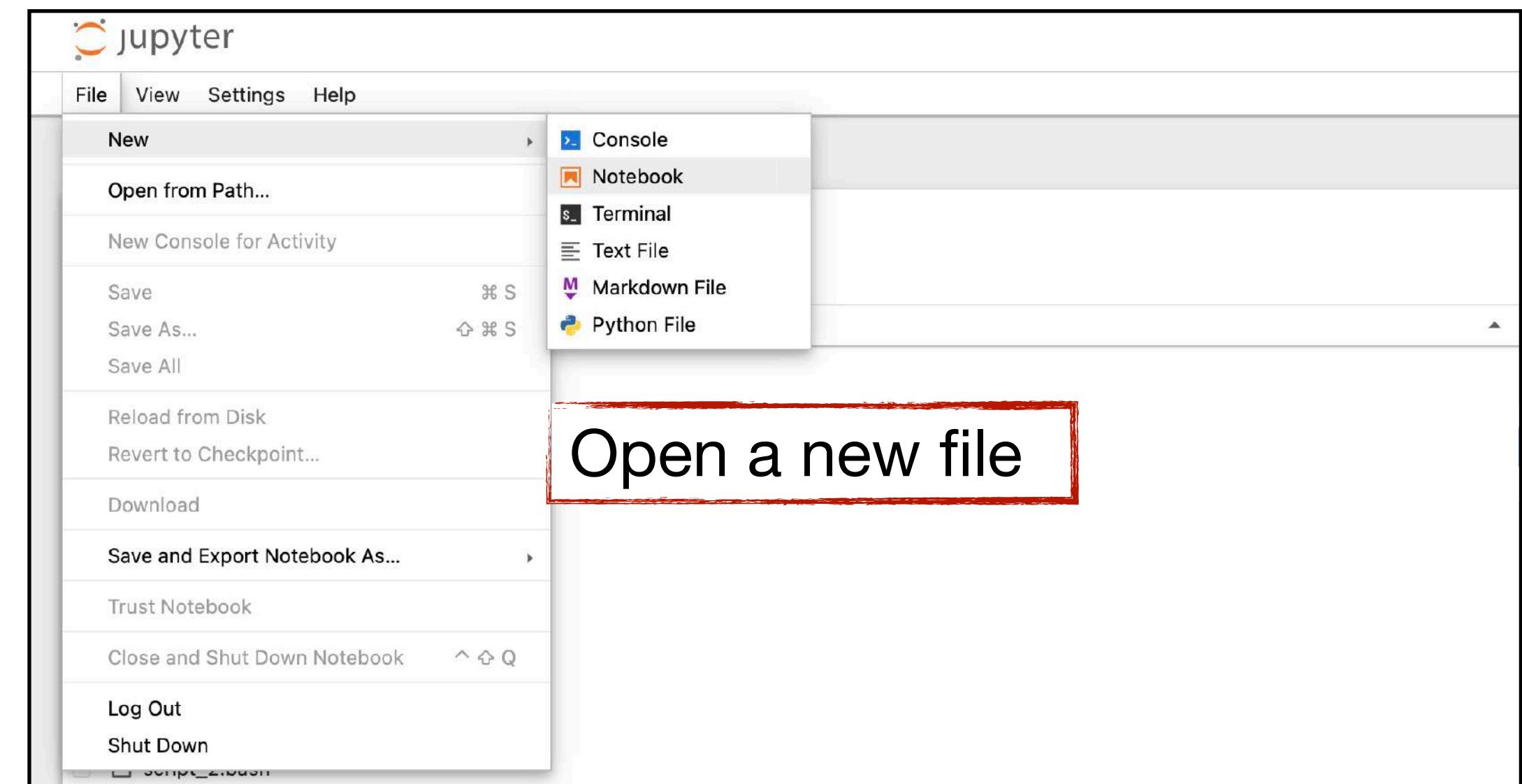
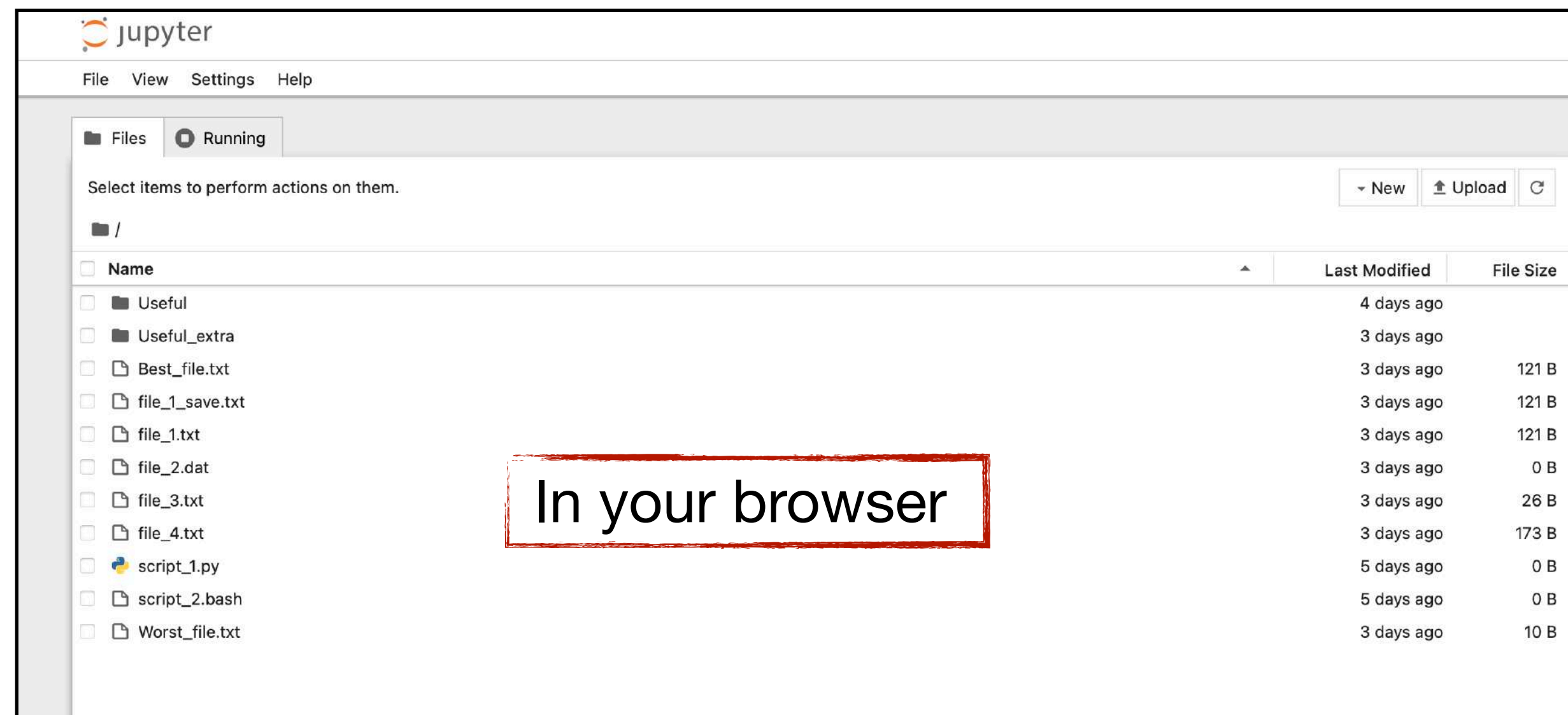
To access the server, open this file in a browser:

`file:///Users/milenavalentini/Library/Jupyter/runtime/jpserver-70912-open.html`

Or copy and paste one of these URLs:

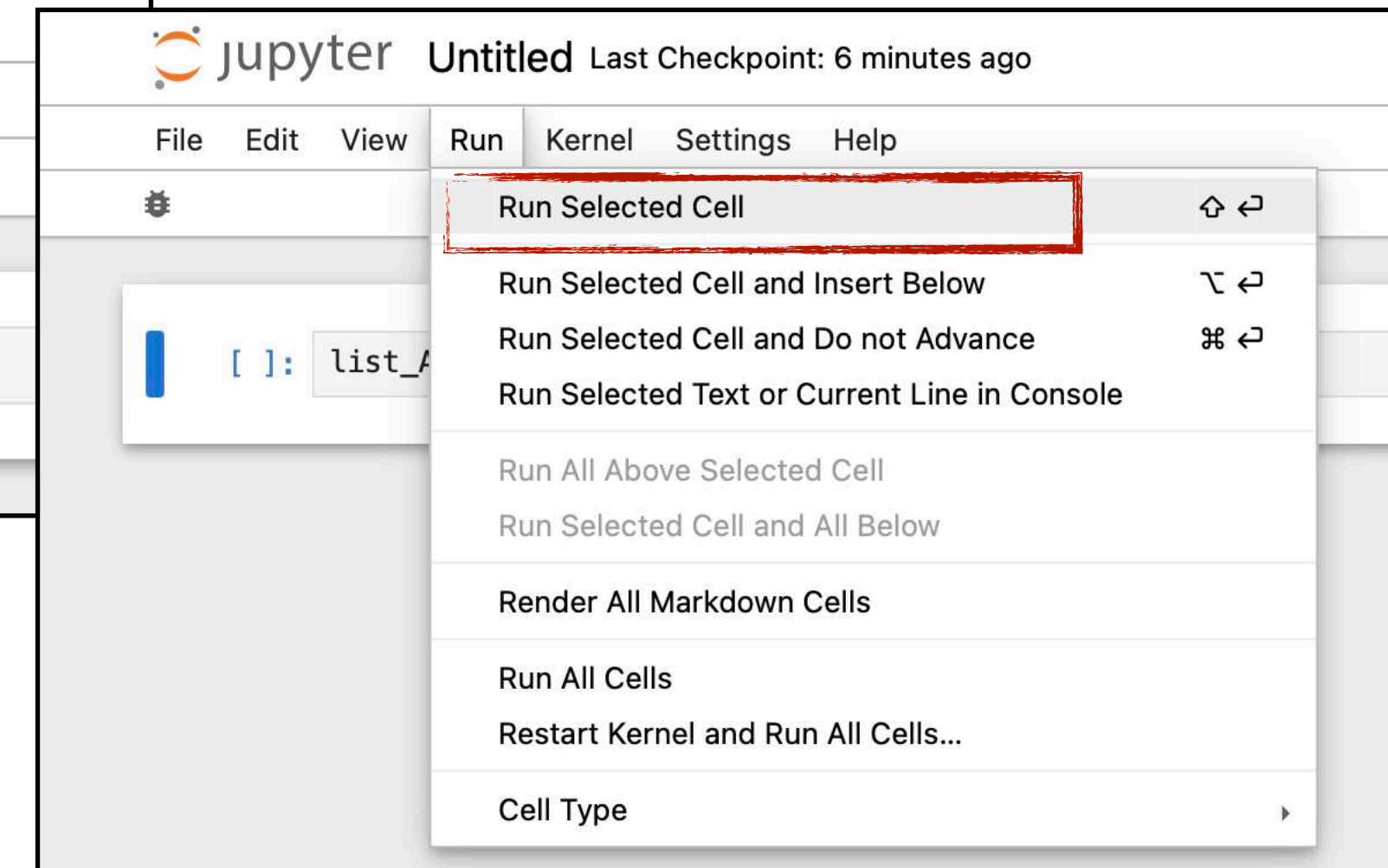
`http://localhost:8888/tree?token=84fd8e0bc4e833913be7f0e14d7bbc6a8650cf79f8d4ae03`

`http://127.0.0.1:8888/tree?token=84fd8e0bc4e833913be7f0e14d7bbc6a8650cf79f8d4ae03`



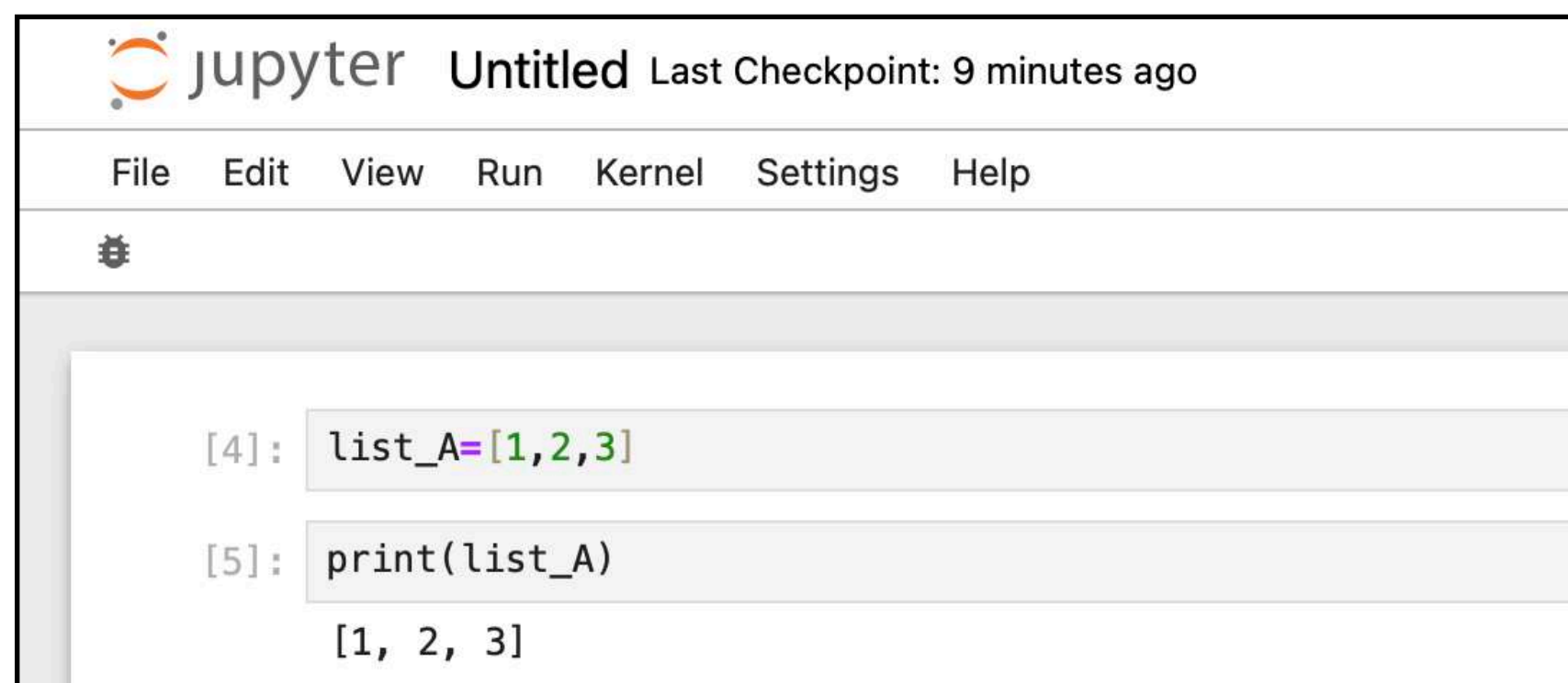
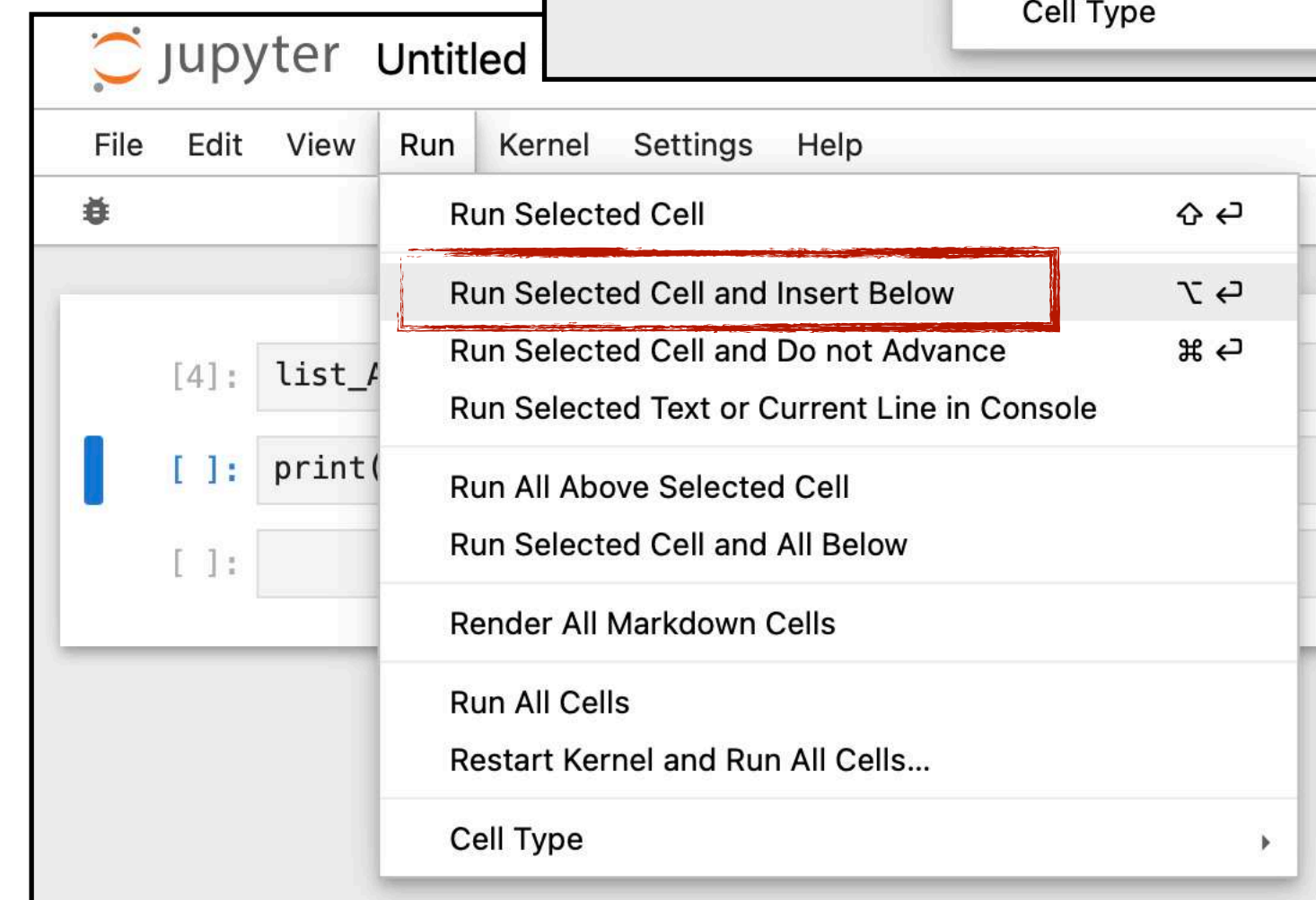
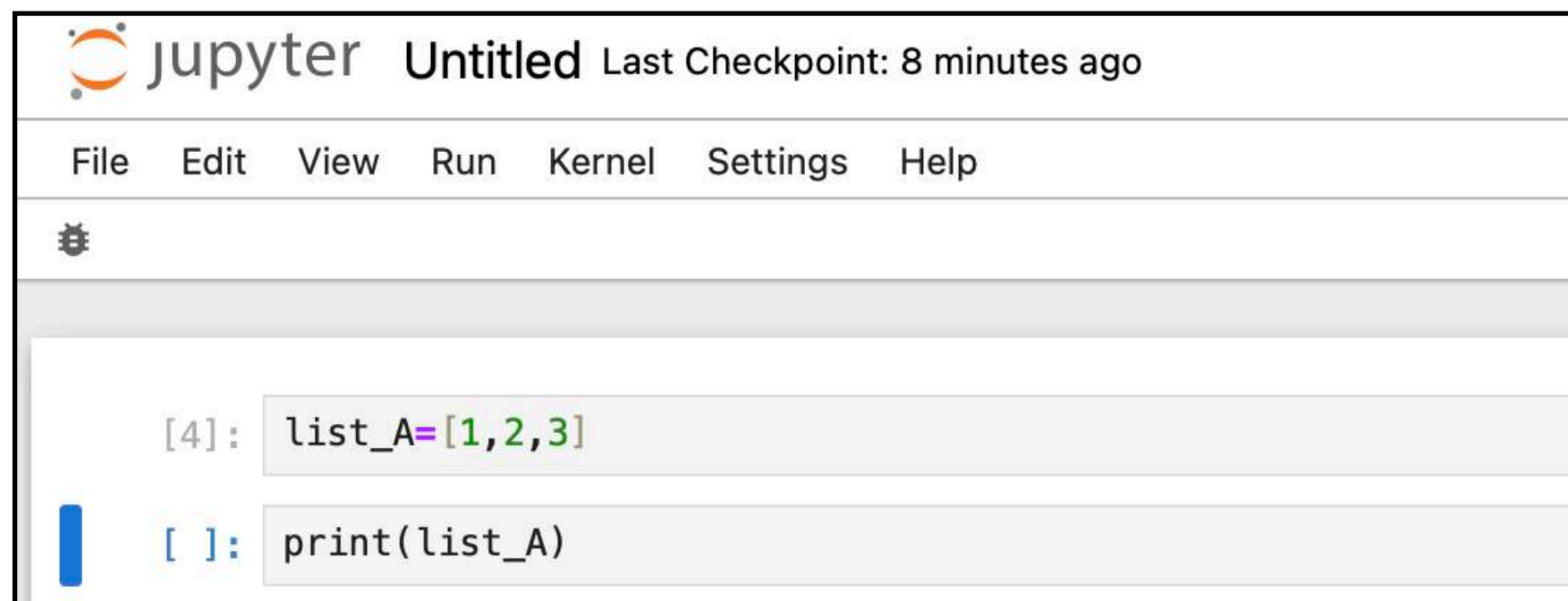
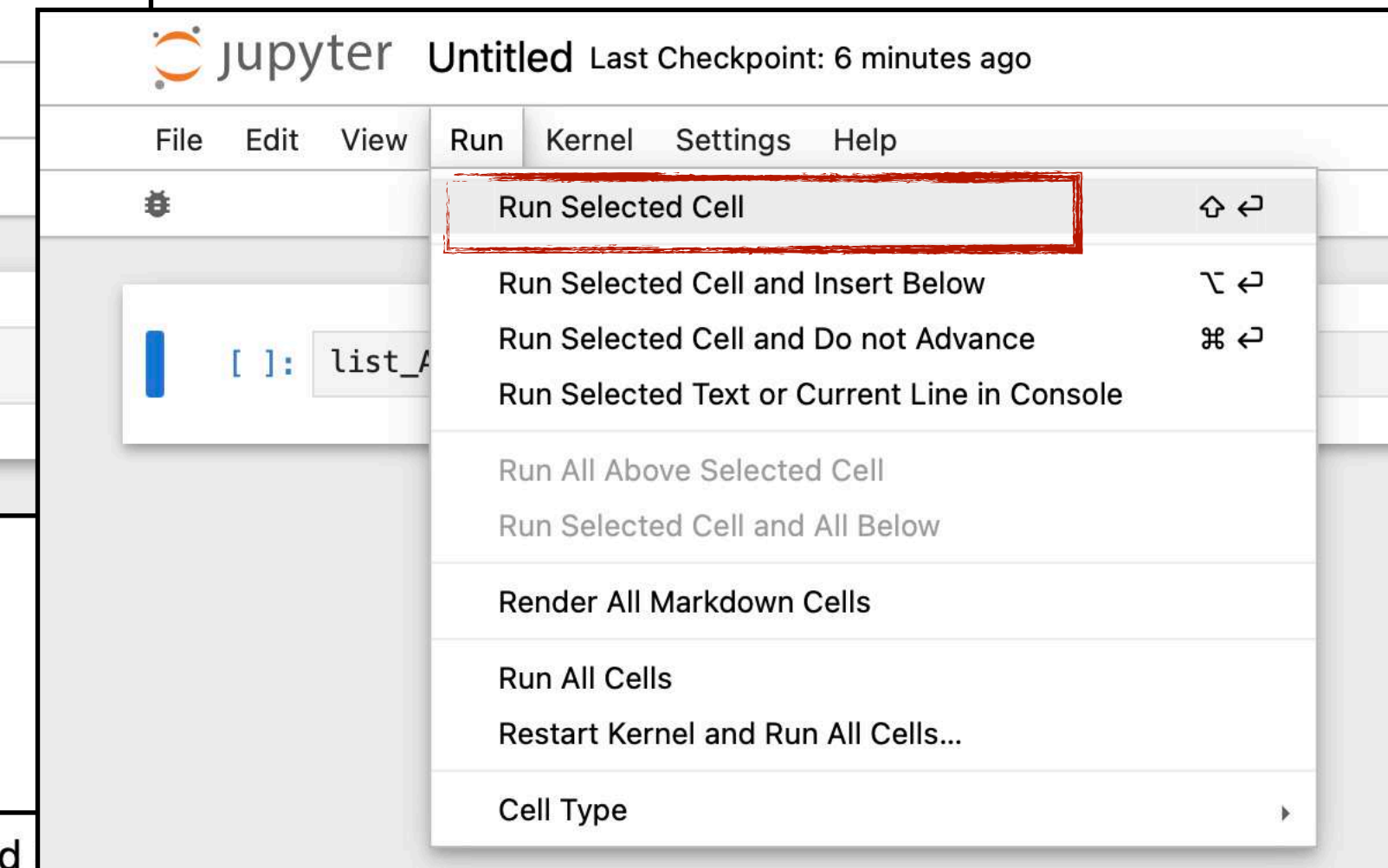
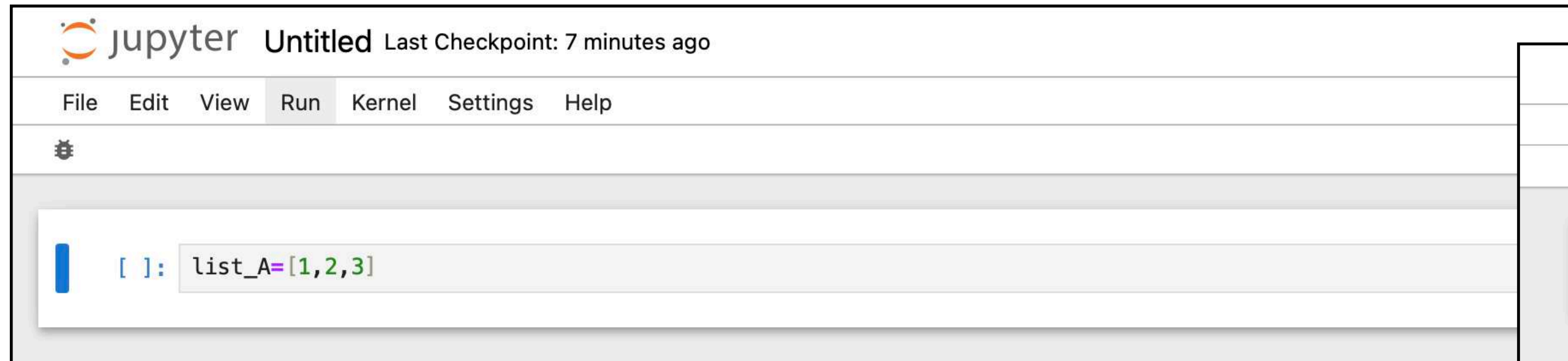
# Jupyter Notebook

Let's use Anaconda via shell



# Jupyter Notebook

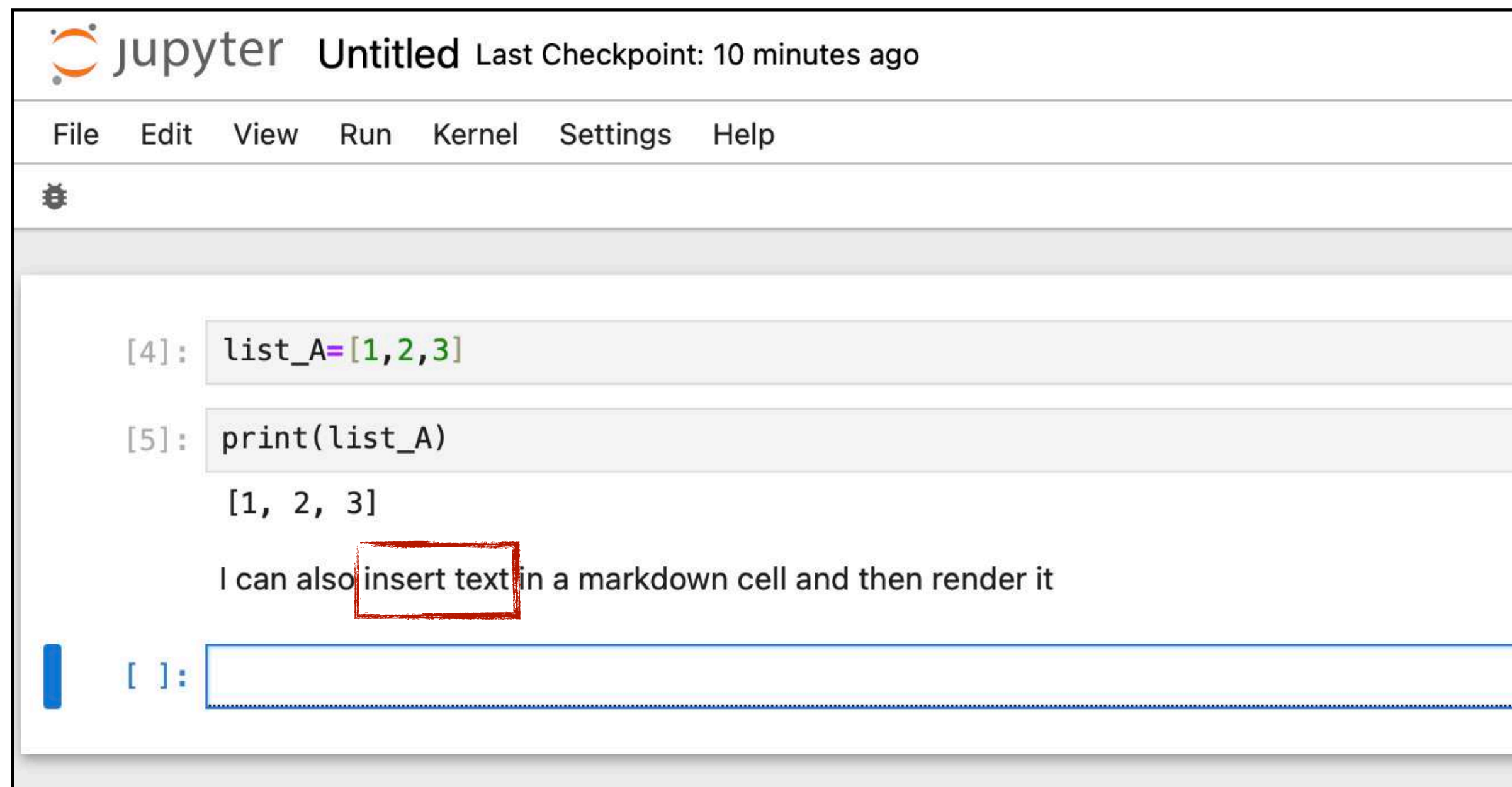
Let's use Anaconda via shell



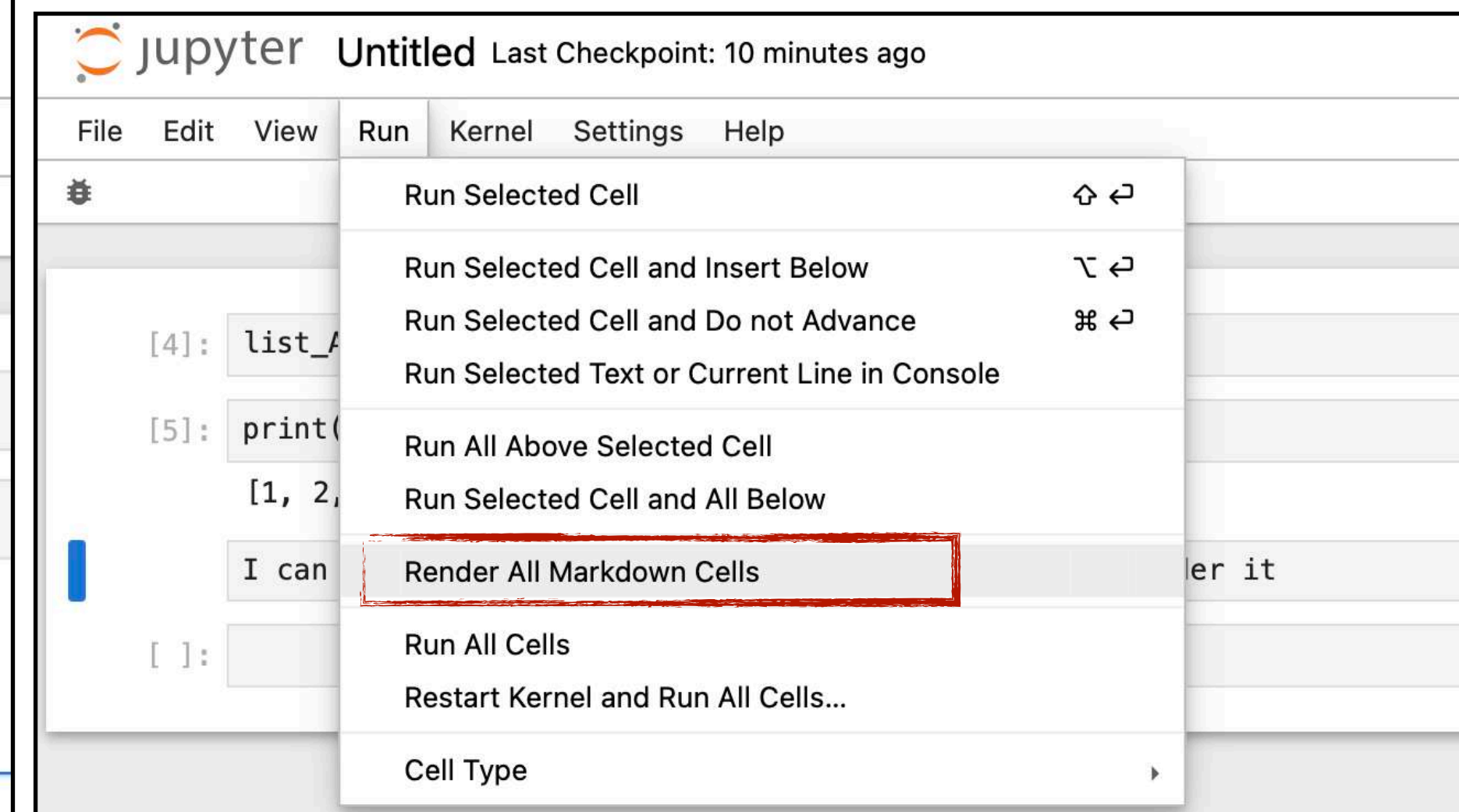
# Jupyter Notebook

Let's use Anaconda via shell

Launch it:



The screenshot shows a Jupyter Notebook window titled "jupyter Untitled" with a last checkpoint of 10 minutes ago. The menu bar includes File, Edit, View, Run, Kernel, Settings, and Help. The notebook contains three cells: a code cell with `list_A=[1,2,3]`, a code cell with `print(list_A)` that has executed and shown the output `[1, 2, 3]`, and a markdown cell containing the text "I can also insert text in a markdown cell and then render it". The words "insert text" in the markdown cell are highlighted with a red box. Below the markdown cell is an empty code cell with the prompt `[ ]:`.



This screenshot shows the same Jupyter Notebook window, but with the "Run" menu open. The menu items are: 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 (highlighted with a red box), Run All Cells, Restart Kernel and Run All Cells..., and Cell Type (with a right-pointing arrow). The background shows the notebook content from the previous screenshot, including the code cells and the markdown cell with "I can also insert text in a markdown cell and then render it".

# Jupyter Notebook

The screenshot shows a Jupyter Notebook window titled "Untitled2" with a last checkpoint of 3 minutes ago. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar. The notebook content consists of a title "Esempio di relazione", a subtitle "Titolo", a section "Testo da sottolineare", and another section "Altro testo". Below the text is a code cell with the following Python code:

```
[1]: from matplotlib import pyplot as plt
plt.plot([1,2,3], [3,6,9], c='green')
plt.show()
```

The code cell is followed by a plot showing a green line with three data points at (1, 3), (2, 6), and (3, 9). The x-axis ranges from 1.00 to 3.00, and the y-axis ranges from 3 to 9.

The screenshot shows a Jupyter Notebook window titled "Untitled" with a last checkpoint of 10 minutes ago. The interface includes a menu bar (Edit, View, Run, Kernel, Settings, Help) and a toolbar. A context menu is open over a code cell, listing various actions:

- 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

The "Render All Markdown Cells" option is highlighted with a red box.

The screenshot shows a Jupyter Notebook window with a text cell containing the text "I can also insert text in a markdown cell and then render it". The text "insert text" is highlighted with a red box. Below the text cell is an empty code cell.

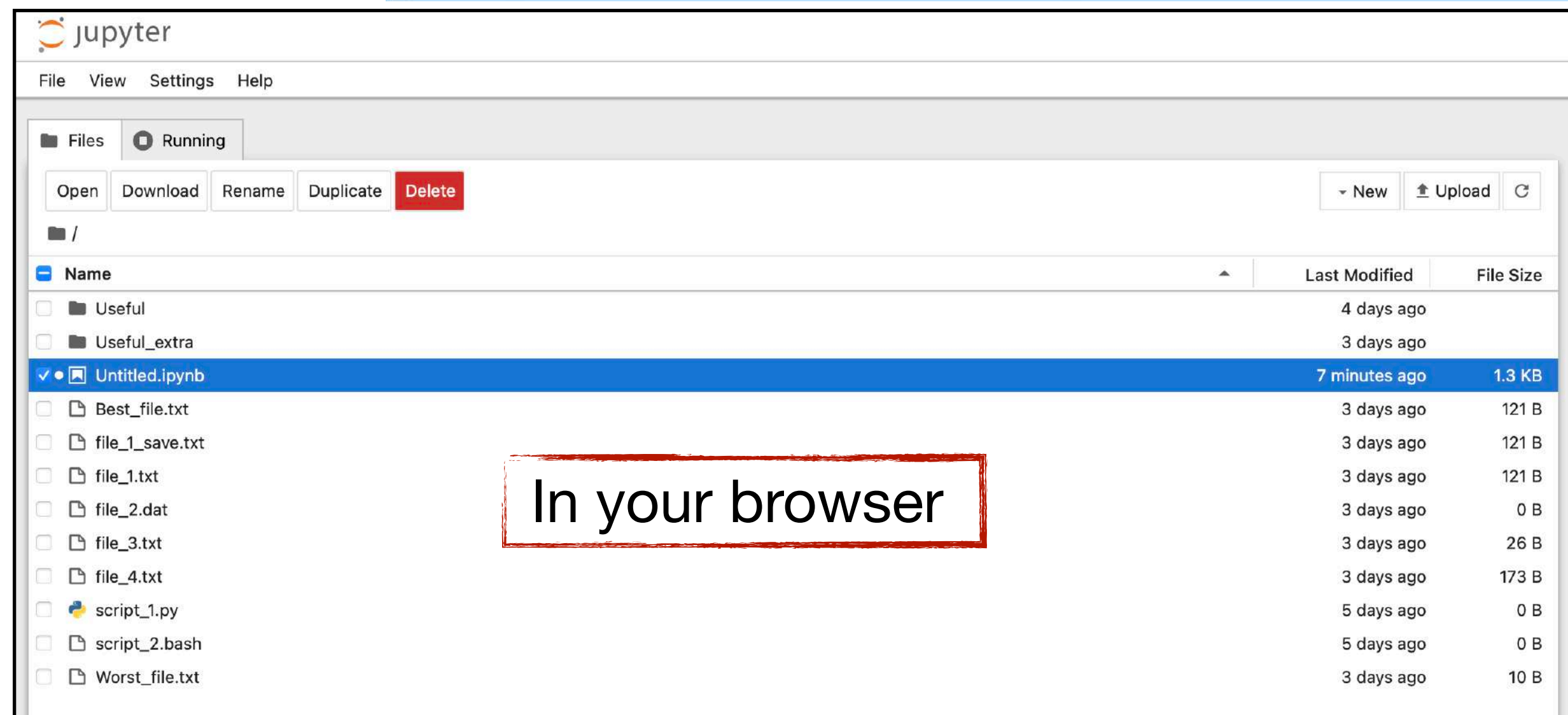
# Setting up the working environment with Anaconda

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
```



# 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	size	source
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:

cyclor          conda-forge/noarch::cyclor-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$
```

matplotlib  
*scientific library  
for publication quality  
figures in Python*

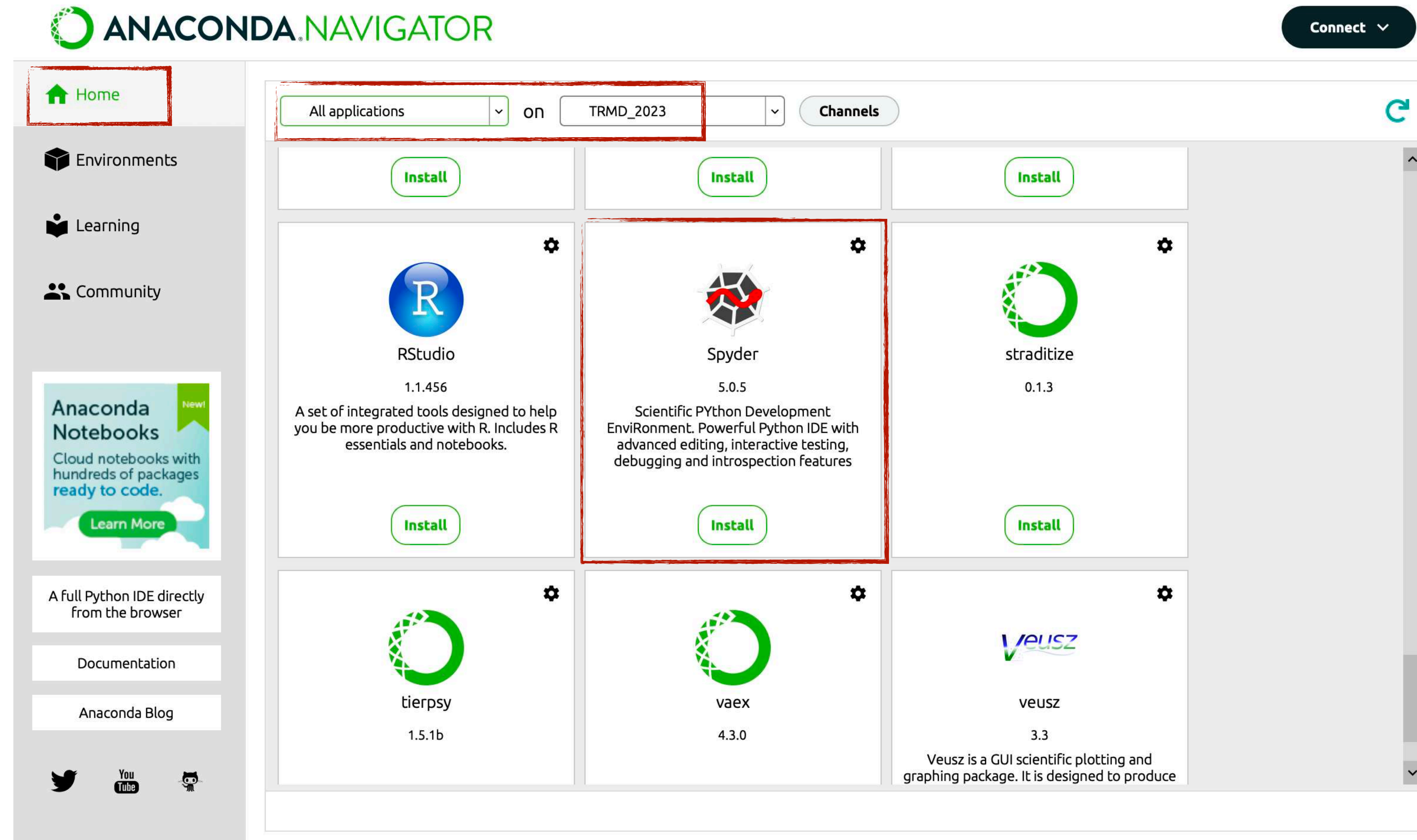


# Setting up the working environment with Anaconda

To exploit it via its graphical user interface:

Launch Anaconda-Navigator:

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



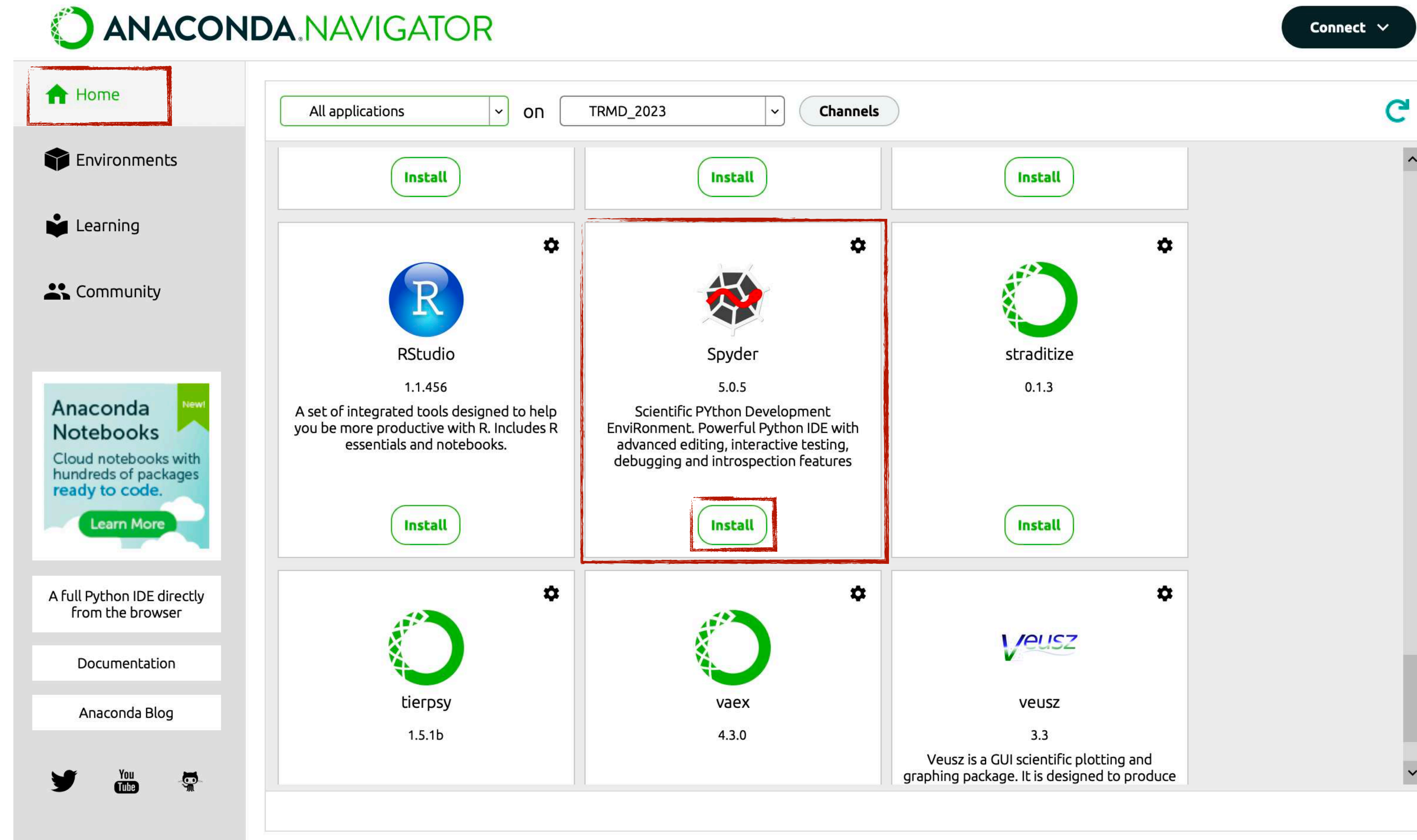
# Setting up the working environment with Anaconda

To exploit it via its graphical user interface:

Launch Anaconda-Navigator:

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

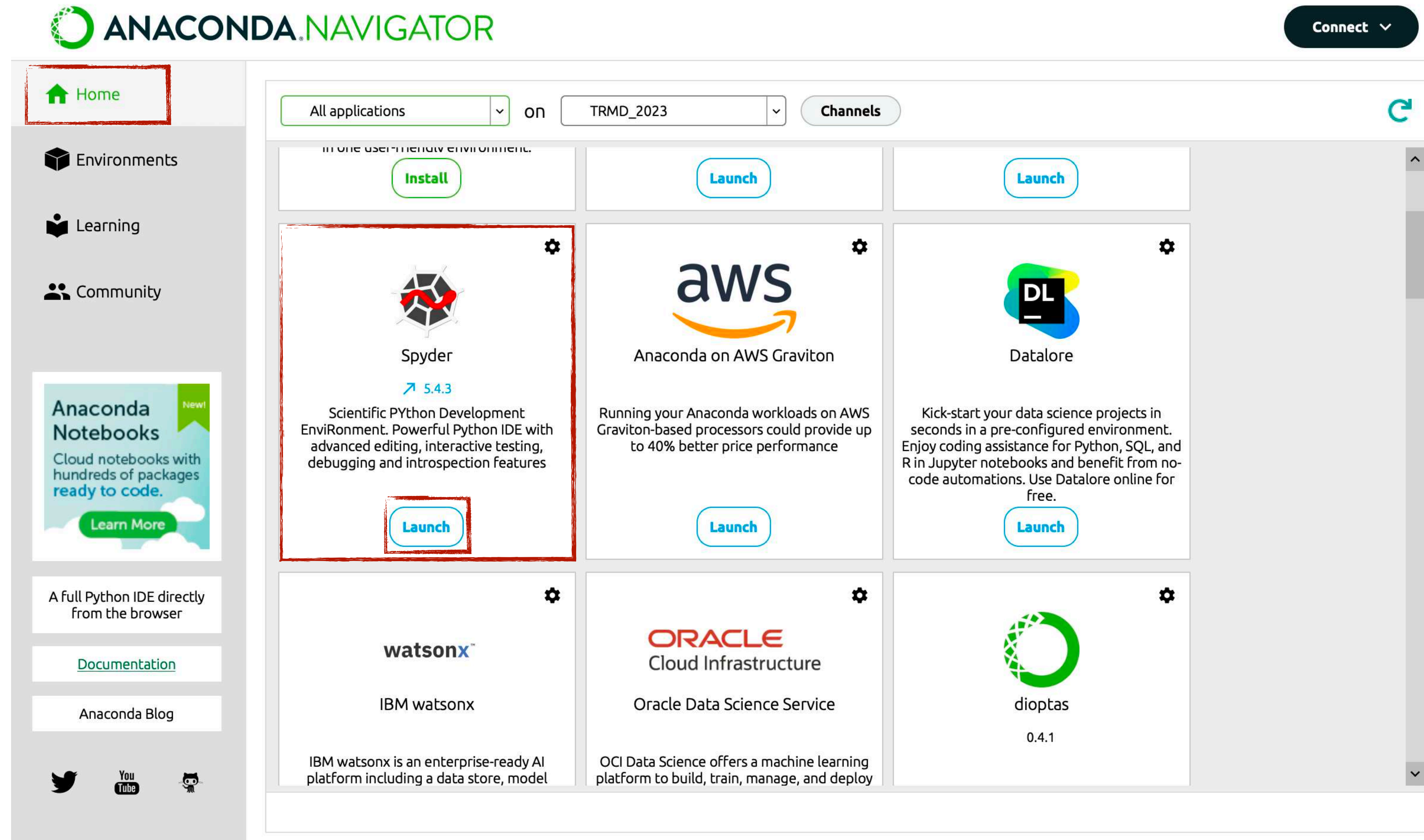
Select the applications to be installed in the environment among available ones



# Setting up the working environment with Anaconda

To exploit it via its graphical user 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

# Spyder

The image shows the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.8)". The top toolbar contains various icons for file operations and execution. The main editor displays a Python script named "temp.py" with the following code:

```
1  # -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4  This is a temporary script file.
5  """
6
7  import numpy as np
8  import matplotlib.pyplot as plt
9  import healpy as hp
10 import math
11
12
13 NSIDE = 64
14 print(
15     "Approximate resolution at NSIDE {} is {:.2} deg".format(
16         NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
17     )
18 )
19
20 NPIX = hp.nside2npix(NSIDE)
21 print(NPIX)
22
23 my_map_I = hp.read_map("test.h.fits")
24
25 hp.mollview(
26     my_map_I,
27     title="Fiducial model sky map",
28     unit="cm$^{-2}$",
29     # norm="hist",
30     norm="log",
31     min=math.modf(np.amin(my_map_I))[1],
32     max=math.modf(np.amax(my_map_I))[1]
33     # min=605,
34     # max=11525
35 )
36 hp.graticule()
37
38 plt.savefig("My_healpix_map.png", dpi=300)
39
40
```

The right-hand side of the interface features a "Console" tab, which is currently displaying a "Usage" help message:

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](#)

Below the help message are tabs for "Help", "Variable Explorer", "Plots", and "Files". The "Console" tab is active, showing the following output:

```
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]:
```

At the bottom of the window, the status bar shows: "conda: TRMD\_2023 (Python 3.8.17) | Completions: conda(TRMD\_2023) | LSP: Python | Line 40, Col 1 | UTF-8 | LF | RW | Mem 94%"

# Spyder

The image shows the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.8)". The top toolbar contains various icons for file operations and execution. The left sidebar shows a file explorer with a folder icon highlighted in red. The main editor displays a Python script named "temp.py" with the following code:

```
1  # -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4  This is a temporary script file.
5  """
6
7  import numpy as np
8  import matplotlib.pyplot as plt
9  import healpy as hp
10 import math
11
12
13 NSIDE = 64
14 print(
15     "Approximate resolution at NSIDE {} is {:.2} deg".format(
16         NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
17     )
18 )
19
20 NPIX = hp.nside2npix(NSIDE)
21 print(NPIX)
22
23 my_map_I = hp.read_map("test.h.fits")
24
25 hp.mollview(
26     my_map_I,
27     title="Fiducial model sky map",
28     unit="cm$^{-2}$",
29     # norm="hist",
30     norm="log",
31     min=math.modf(np.amin(my_map_I))[1],
32     max=math.modf(np.amax(my_map_I))[1]
33     # min=605,
34     # max=11525
35 )
36 hp.graticule()
37
38 plt.savefig("My_healpix_map.png", dpi=300)
39
40
```

The right sidebar contains a help panel titled "Usage" with the following text:

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](#)

Below the help panel are tabs for "Help", "Variable Explorer", "Plots", and "Files". The bottom right corner shows the "IPython Console" with the following text:

```
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]:
```

The status bar at the bottom displays: conda: TRMD\_2023 (Python 3.8.17) | Completions: conda(TRMD\_2023) | LSP: Python | Line 40, Col 1 | UTF-8 | LF | RW | Mem 94%

# Spyder

The image shows the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.8)" and shows a file path of "/Users/milenaValentini". The code editor displays a Python script named "temp.py" with the following content:

```
1  # -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4  This is a temporary script file.
5  """
6
7  import numpy as np
8  import matplotlib.pyplot as plt
9  import healpy as hp
10 import math
11
12
13 NSIDE = 64
14 print(
15     "Approximate resolution at NSIDE {} is {:.2} deg".format(
16         NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
17     )
18 )
19
20 NPIX = hp.nside2npix(NSIDE)
21 print(NPIX)
22
23 my_map_I = hp.read_map("test.h.fits")
24
25 hp.mollview(
26     my_map_I,
27     title="Fiducial model sky map",
28     unit="cm$^{-2}$",
29     # norm="hist",
30     norm="log",
31     min=math.modf(np.amin(my_map_I))[1],
32     max=math.modf(np.amax(my_map_I))[1]
33     # min=605,
34     # max=11525
35 )
36 hp.graticule()
37
38 plt.savefig("My_healpix_map.png", dpi=300)
39
40
```

The right-hand side of the interface is divided into two panels. The top panel, titled "Source Console Object", shows a "Usage" message:

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](#)

The bottom panel, titled "Console 1/A", shows the IPython console output:

```
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]:
```

The status bar at the bottom of the window displays: conda: TRMD\_2023 (Python 3.8.17) Completions: conda(TRMD\_2023) LSP: Python Line 40, Col 1 UTF-8 LF RW Mem 94%

# Spyder

The image shows the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.8)". The top toolbar contains various icons, with a red box highlighting the "Run" button (a green play icon). The left pane shows a code editor with a file named "temp.py" open. The code in the editor is as follows:

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4 This is a temporary script file.
5 """
6
7 import numpy as np
8 import matplotlib.pyplot as plt
9 import healpy as hp
10 import math
11
12
13 NSIDE = 64
14 print(
15     "Approximate resolution at NSIDE {} is {:.2} deg".format(
16         NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
17     )
18 )
19
20 NPIX = hp.nside2npix(NSIDE)
21 print(NPIX)
22
23 my_map_I = hp.read_map("test.h.fits")
24
25 hp.mollview(
26     my_map_I,
27     title="Fiducial model sky map",
28     unit="cm$^{-2}$",
29     # norm="hist",
30     norm="log",
31     min=math.modf(np.amin(my_map_I))[1],
32     max=math.modf(np.amax(my_map_I))[1]
33     # min=605,
34     # max=11525
35 )
36 hp.graticule()
37
38 plt.savefig("My_healpix_map.png", dpi=300)
39
40
```

The right pane shows a plot area with a zoom level of 135%. Below the plot area are tabs for "Help", "Variable Explorer", "Plots", and "Files". The bottom pane shows the IPython Console with the following output:

```
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/milena-valentini/.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]:
```

The status bar at the bottom of the window displays: "conda: TRMD\_2023 (Python 3.8.17) Completions: conda(TRMD\_2023) LSP: Python Line 40, Col 1 UTF-8 LF RW Mem 94%"

# Spyder

The image shows the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.8)" and shows a file explorer on the left with a file named "temp.py". The code editor displays the following Python code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4 This is a temporary script file.
5 """
6 import numpy as np
7 import matplotlib.pyplot as plt
8 import healpy as hp
9 import math
10
11
12
13 NSIDE = 64
14 print(
15     "Approximate resolution at NSIDE {} is {:.2} deg".format(
16         NSIDE, hp.nside2resol(NSIDE, arcmin=True) / 60
17     )
18 )
19
20 NPIX = hp.nside2npix(NSIDE)
21 print(NPIX)
22
23 my_map_I = hp.read_map("test.h.fits")
24
25 hp.mollview(
26     my_map_I,
27     title="Fiducial model sky map",
28     unit="cm$^{-2}$",
29     # norm="hist",
30     norm="log",
31     min=math.modf(np.amin(my_map_I))[1],
32     max=math.modf(np.amax(my_map_I))[1]
33     # min=605,
34     # max=11525
35 )
36 hp.graticule()
37
38 plt.savefig("My_healpix_map.png", dpi=300)
39
40
```

The console window shows the following output:

```
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'
```

The terminal window shows the command:

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

The status bar at the bottom indicates: conda: TRMD\_2023 (Python 3.8.17) | Completions: conda(TRMD\_2023) | LSP: Python | Line 40, Col 1 | UTF-8 | LF | RW | Mem 94%



# Spyder

The image shows the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.8)". The top toolbar contains various icons for file operations and execution. The left pane shows the file explorer with a single file named "temp.py". The central pane is the code editor, displaying the following Python code:

```
1  # -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4  This is a temporary script file.
5  """
6
7  import numpy as np
8  import matplotlib.pyplot as plt
9  import healpy as hp
10 import math
11
12
13 NSIDE = 64
14 print(
15     "Approximate resolution at NSIDE {} is {:.2} deg".format(
16         NSIDE, hp.nside2resol(NSIDE, arcmim=True) / 60
17     )
18 )
19
20 NPIX = hp.nside2npix(NSIDE)
21 print(NPIX)
22
23 my_map_I = hp.read_map("test.h.fits")
24
25 hp.mollview(
26     my_map_I,
27     title="Fiducial model sky map",
28     unit="cm-2",
29     # norm="hist",
30     norm="log",
31     min=math.modf(np.amin(my_map_I))[1],
32     max=math.modf(np.amax(my_map_I))[1]
33     # min=605,
34     # max=11525
35 )
36 hp.graticule()
37
38 plt.savefig("My_healpix_map.png", dpi=300)
39
40
```

The right pane displays a plot titled "Fiducial model sky map". The plot shows a Mollweide projection of a sky map with a color scale ranging from 613 to 11687 cm<sup>-2</sup>. The plot is zoomed in to 72%. Below the plot are tabs for "Help", "Variable Explorer", "Plots", and "Files".

The bottom pane is the IPython Console, showing the following output:

```
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]:
```

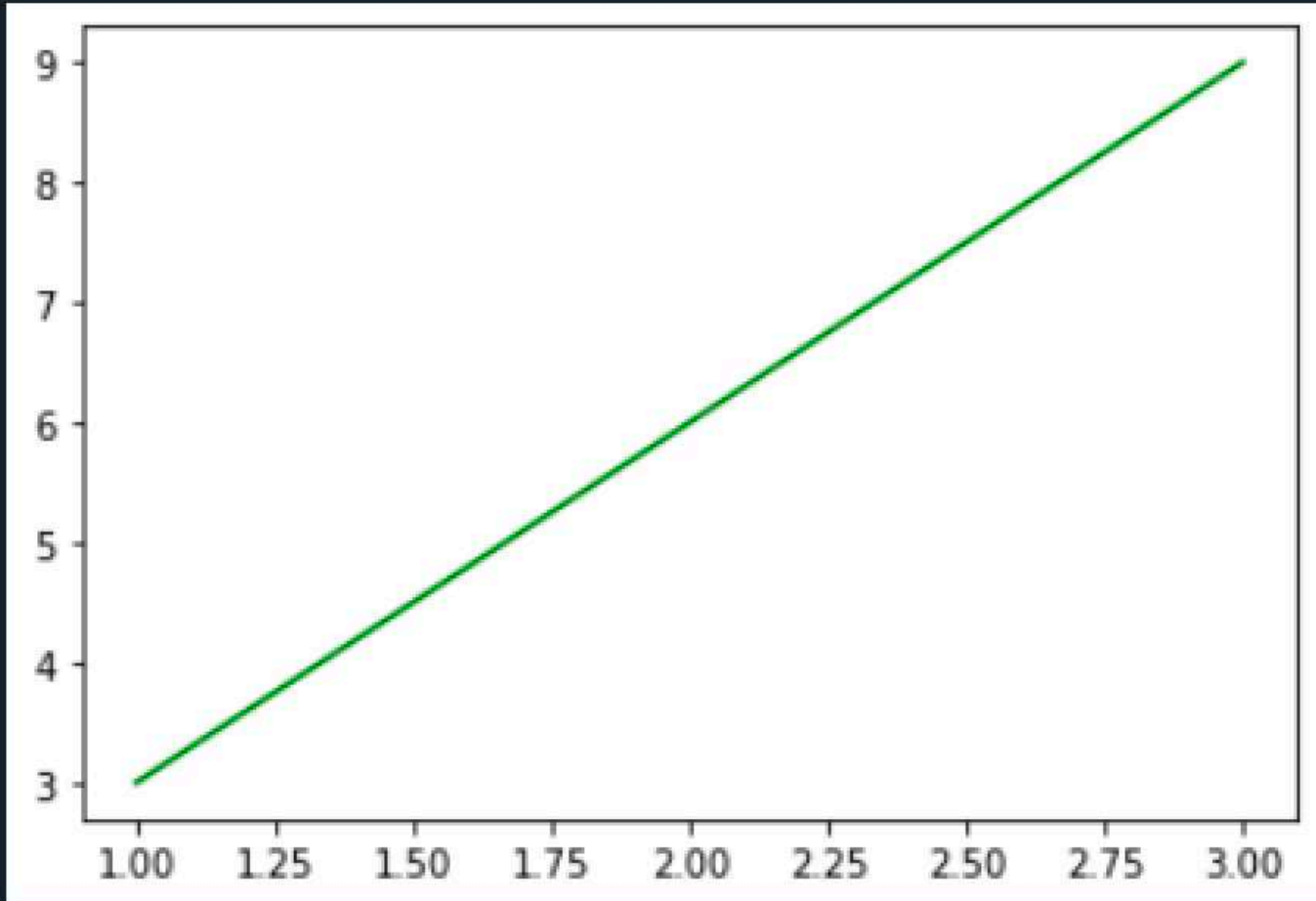
The status bar at the bottom indicates the environment is "conda: TRMD\_2023 (Python 3.8.17)", with "Completions: conda(TRMD\_2023)", "LSP: Python", "Line 23, Col 25", "UTF-8", "LF", "RW", and "Mem 94%".

# Spyder

Spyder (Python 3.8)

/Users/milenaValentini

116 %



x	y
1.00	3
1.25	4
1.50	5
1.75	6
2.00	7
2.25	8
2.50	9
2.75	10
3.00	11

Help Variable Explorer Plots Files

Console 1/A

```
In [4]: import matplotlib
In [5]: from matplotlib import pyplot as plt
In [6]: plt.plot([1,2,3], [3,6,9], c='green')
Out[6]: [<matplotlib.lines.Line2D at 0x11c940dc0>]
```

**Important**

Figures are displayed in the Plots pane by default. To make them also appear inline in the console, you need to uncheck "Mute inline plotting" under the options menu of Plots.

```
In [7]:
```

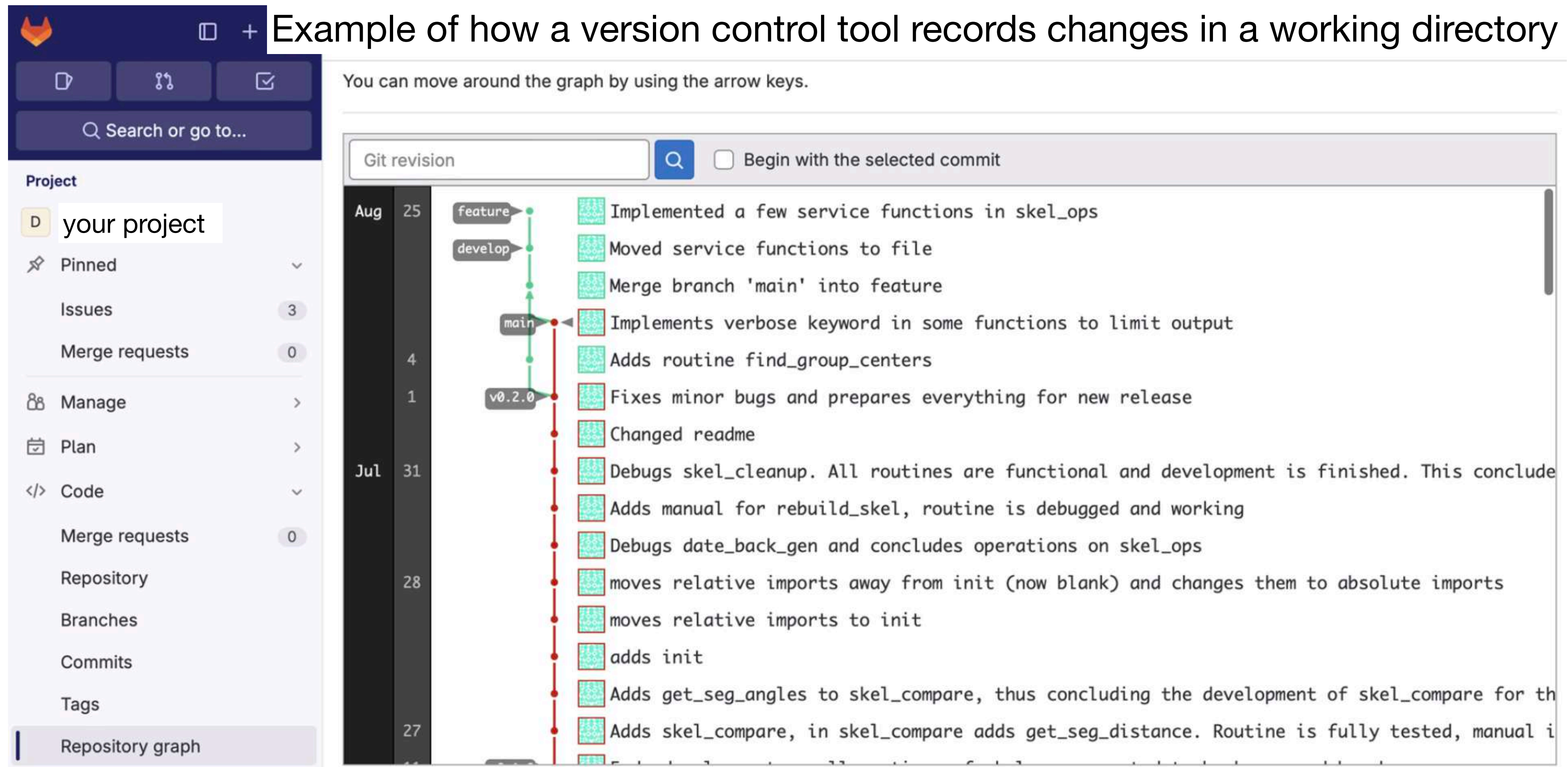
IPython Console History

# 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.

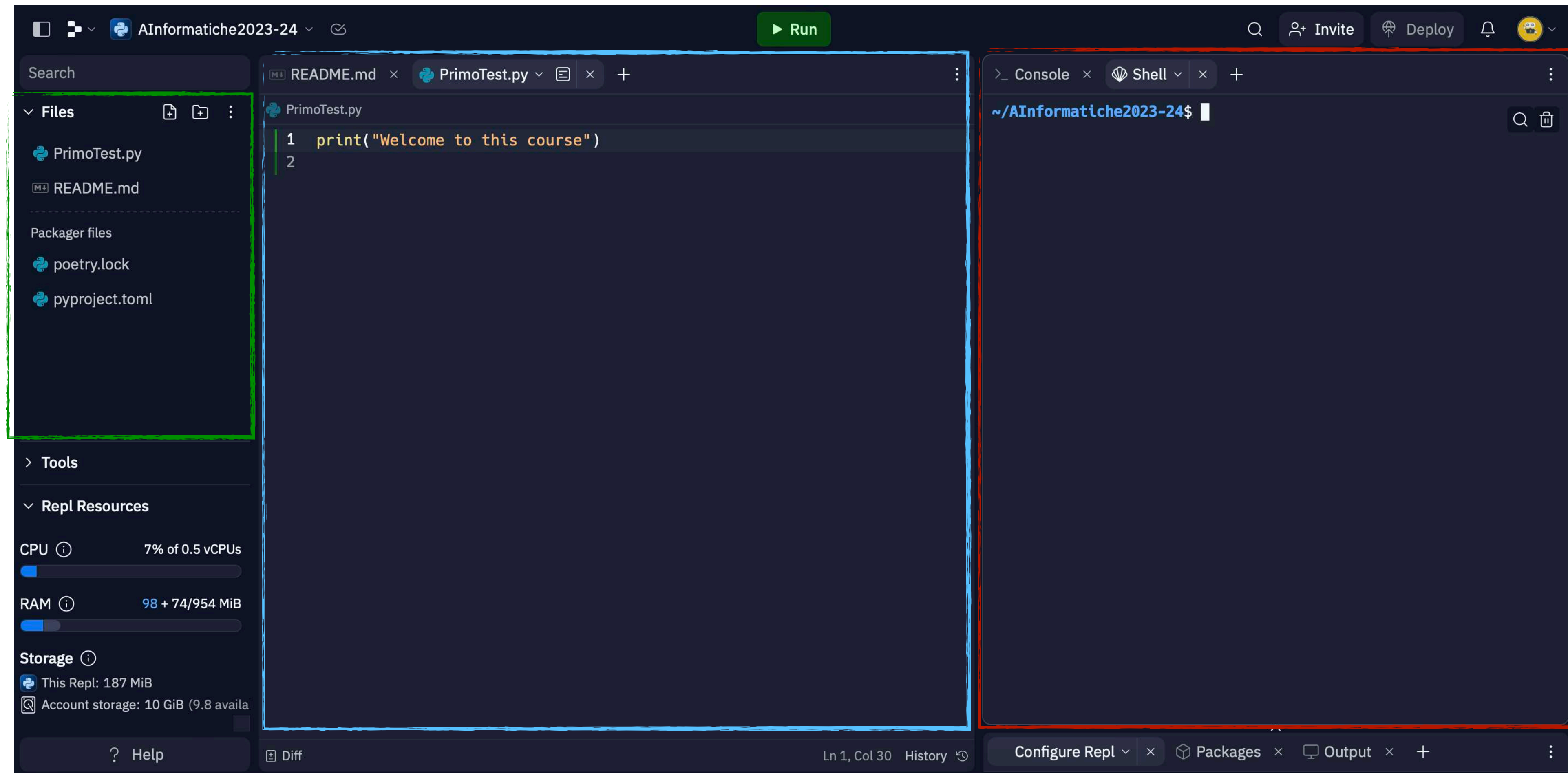
Example of how a version control tool records changes in a working directory



The screenshot displays a Git repository interface. On the left, a sidebar shows the project name 'your project' and various navigation options like 'Pinned', 'Issues', 'Merge requests', 'Manage', 'Plan', 'Code', 'Merge requests', 'Repository', 'Branches', 'Commits', 'Tags', and 'Repository graph'. The main area shows a commit history graph with branches 'feature', 'develop', 'main', and 'v0.2.0'. The graph shows a sequence of commits starting from 'v0.2.0' and moving through 'main' to 'develop' and 'feature'. The commit messages are listed on the right, including: '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', 'Adds routine find\_group\_centers', 'Fixes minor bugs and prepares everything for new release', 'Changed readme', 'Debugs skel\_cleanup. All routines are functional and development is finished. This concludes', 'Adds manual for rebuild\_skel, routine is debugged and working', 'Debugs date\_back\_gen and concludes operations on skel\_ops', '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', and '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.



# Other tools: Integrated Development Environment

---

Setup a working environment:

Register on Git ([github.com](https://github.com))



# Other tools: Integrated Development Environment

Setup a working environment:

Register on Git

Create a public repository on Git

New repository

Type to search

## 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.

**Create repository**

# Other tools: Integrated Development Environment

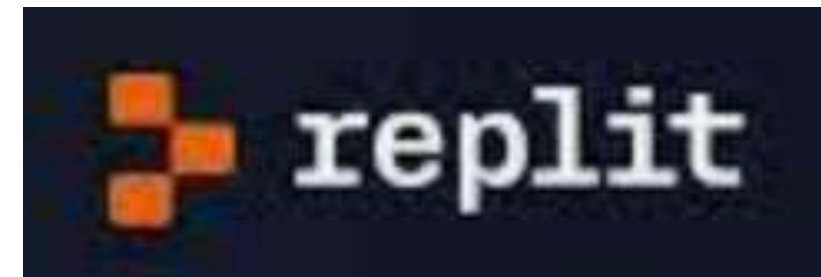
---

Setup a working environment:

Register on Git

Create a public repository on Git

Register on repl.it ([replit.com](https://replit.com))



# Other tools: Integrated Development Environment

Setup a working environment:

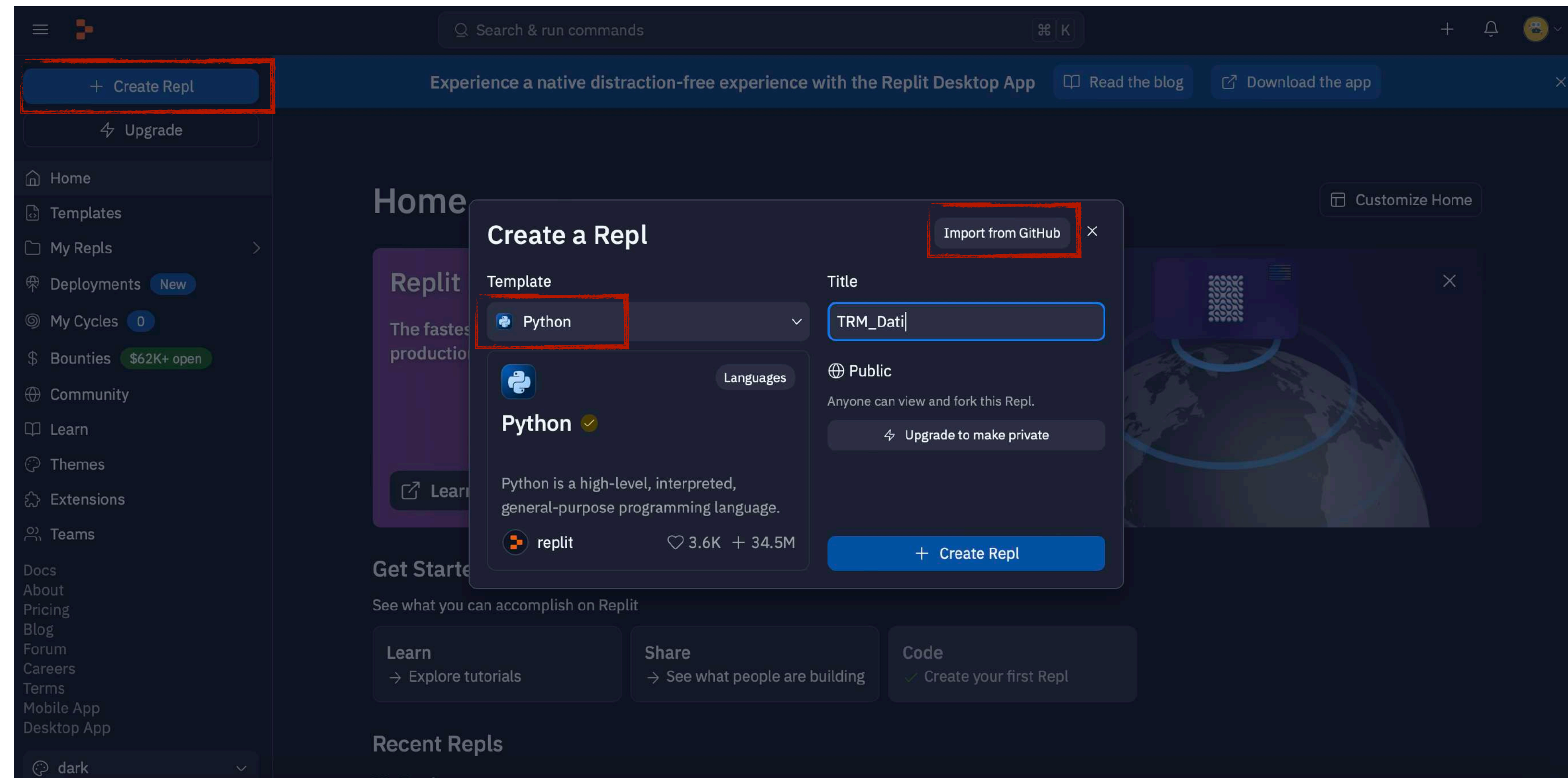
Register on Git

Create a public repository on Git

Register on repl.it ([replit.com](https://replit.com))

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

Set Python as default language





# The Integrated Development Environment

Setup a working environment:

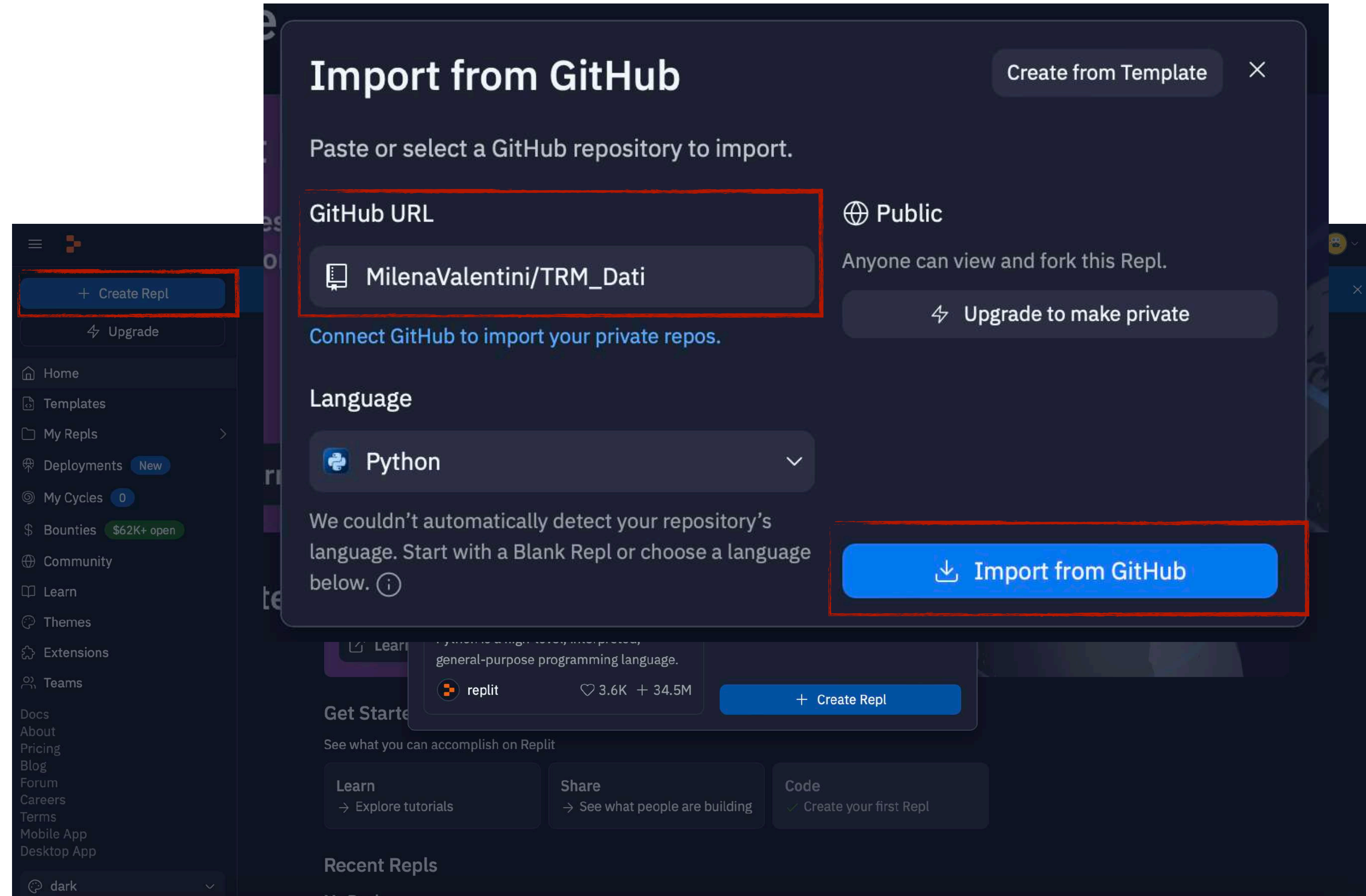
Register on Git

Create a public repository on Git

Register on repl.it ([replit.com](https://replit.com))

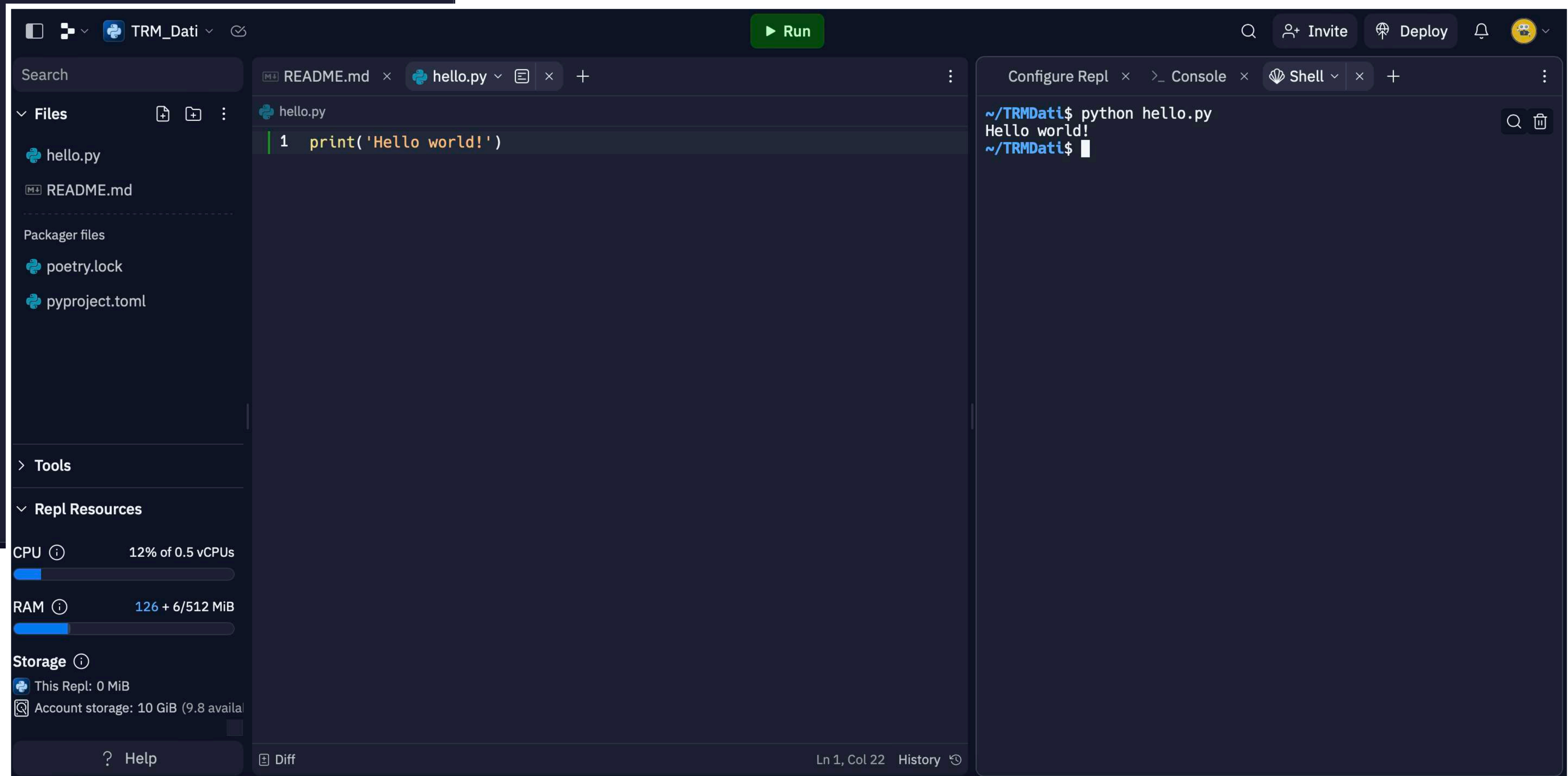
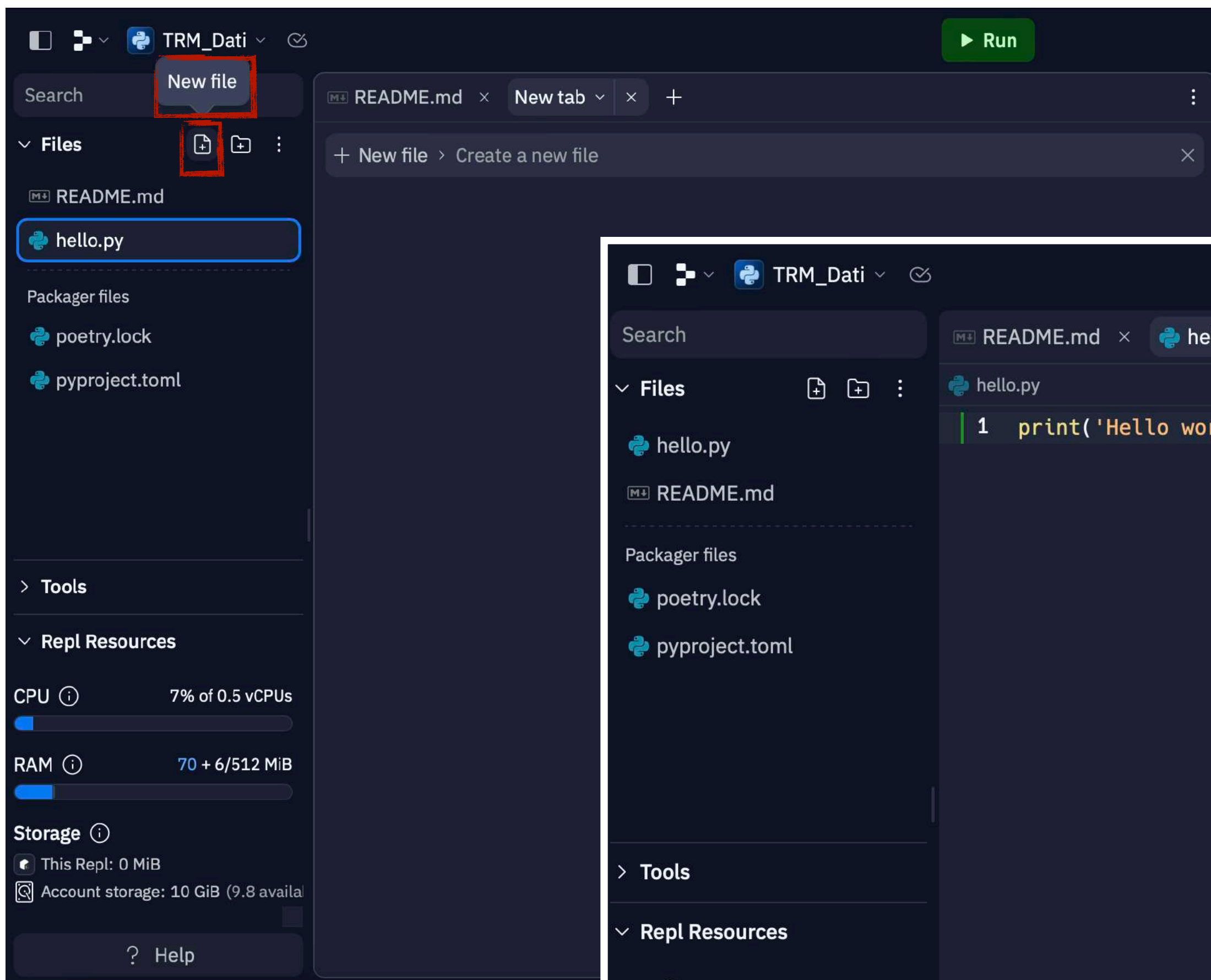
Create a new repl project by importing from Git the repository you have just created there

Set Python as default language



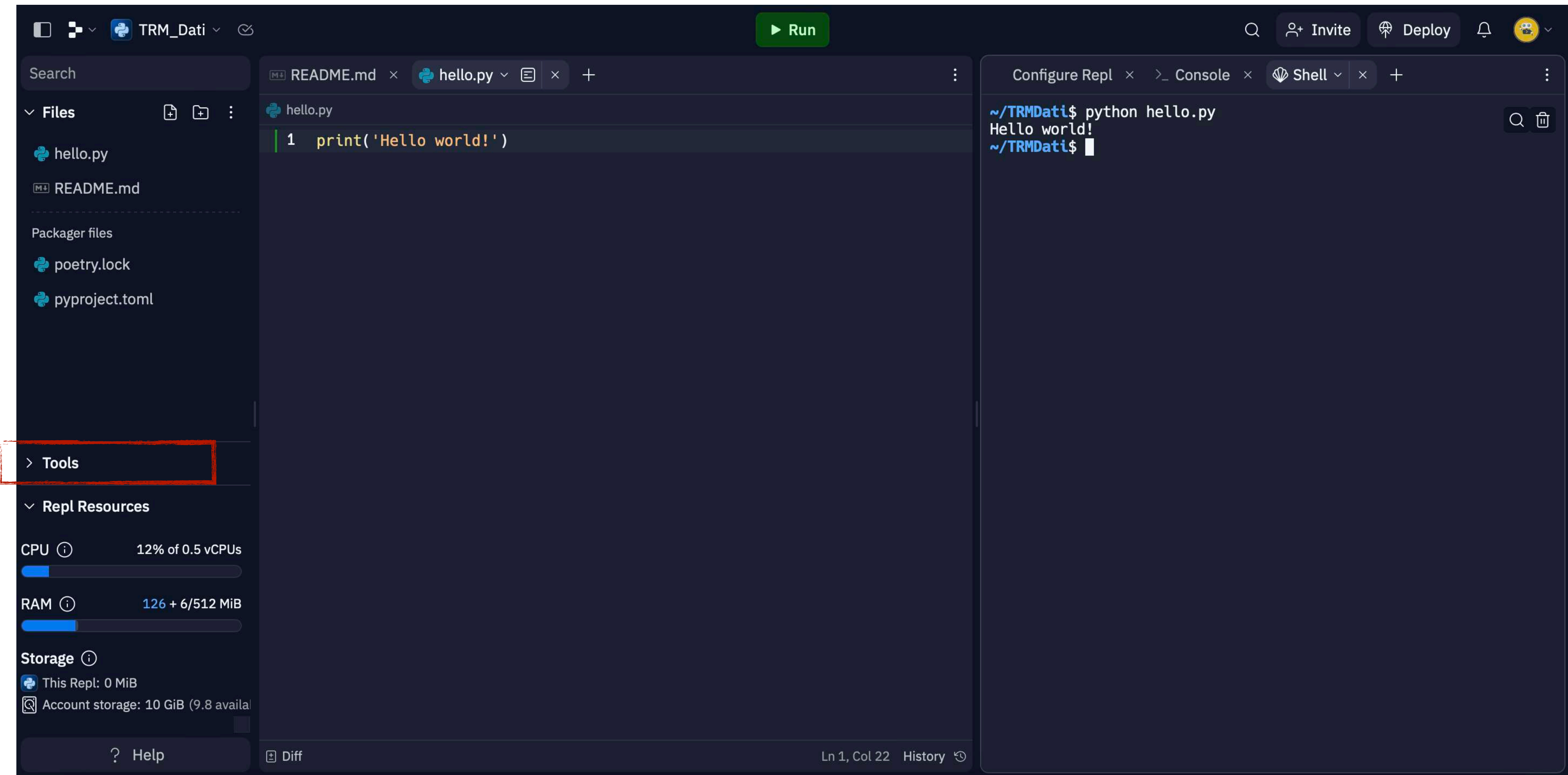
# The Integrated Development Environment

Create a new file to produce the first working script



# The Integrated Development Environment

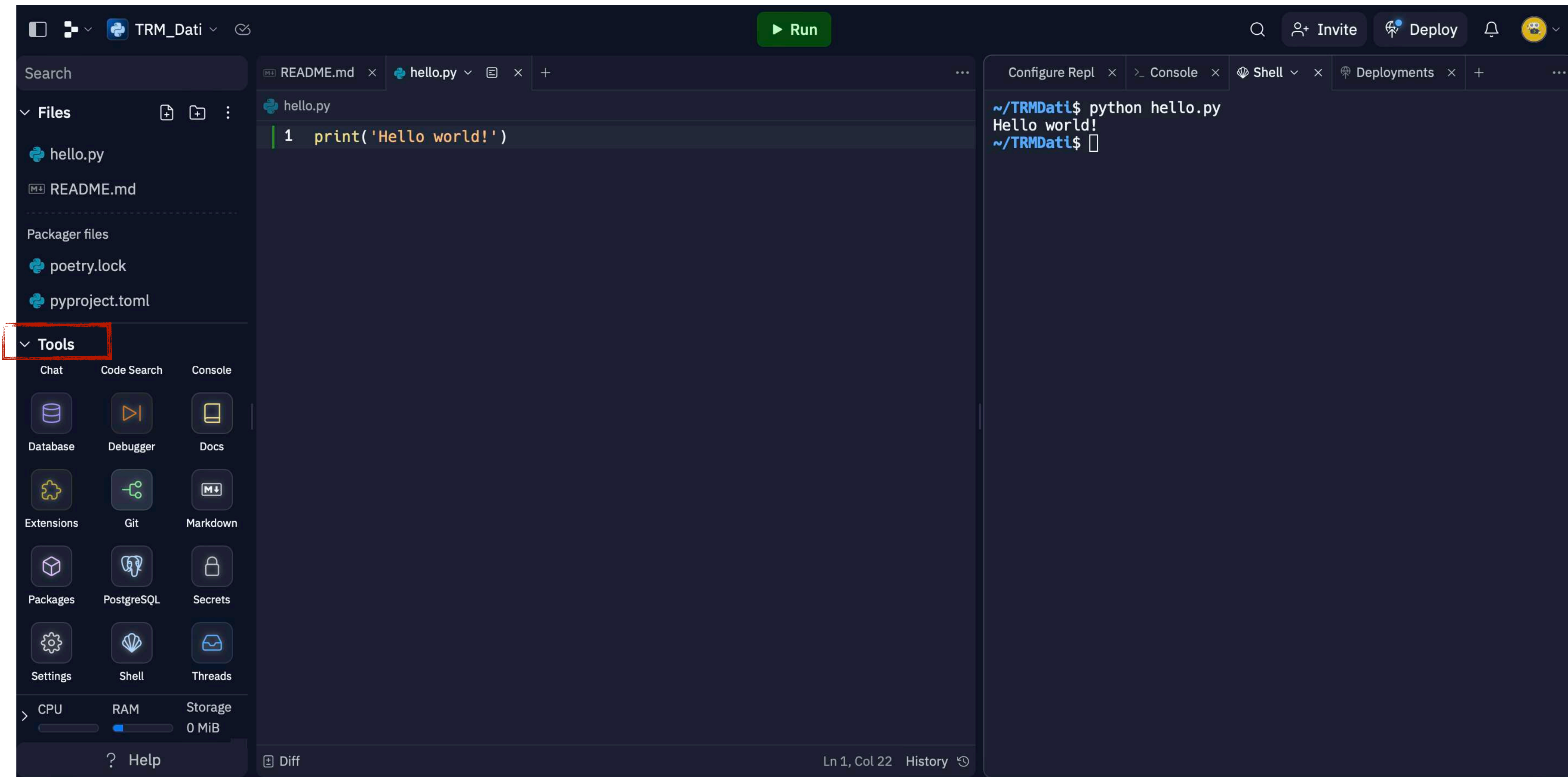
## Connecting replit to Git



Other useful material: <https://replit.com/talk/learn/Replit-Git-Tutorial/23331>

# The Integrated Development Environment

## Connecting replit to Git




# The Integrated Development Environment

## Connecting replit to Git

The screenshot displays the Replit IDE interface. On the left sidebar, the 'Tools' section is expanded, and the 'Git' icon is highlighted with a red box. The main editor area shows a Python file named 'hello.py' with the code `print('Hello world!')`. On the right-hand side, the 'main' branch view is active, and a notification box is highlighted with a red border. The notification reads: 'Can't push or pull from GitHub. This Repl has a GitHub repository as a remote, but you are not connected to GitHub.' Below the notification is a blue button labeled 'Connect to GitHub'. The bottom right panel shows the 'Commit' section with a list of 5 changed files: `.replit`, `hello.py`, `poetry.lock`, `pyproject.toml`, and `replit.nix`, all marked as 'Added'.

# The Integrated Development Environment



## Install & Authorize Replit

Install & Authorize on your personal account Milena Valentini

**All repositories**  
This applies to all current *and* future repositories owned by the resource owner.  
Also includes public repositories (read-only).

**Only select repositories**  
Select at least one repository.  
Also includes public repositories (read-only).

with these permissions:

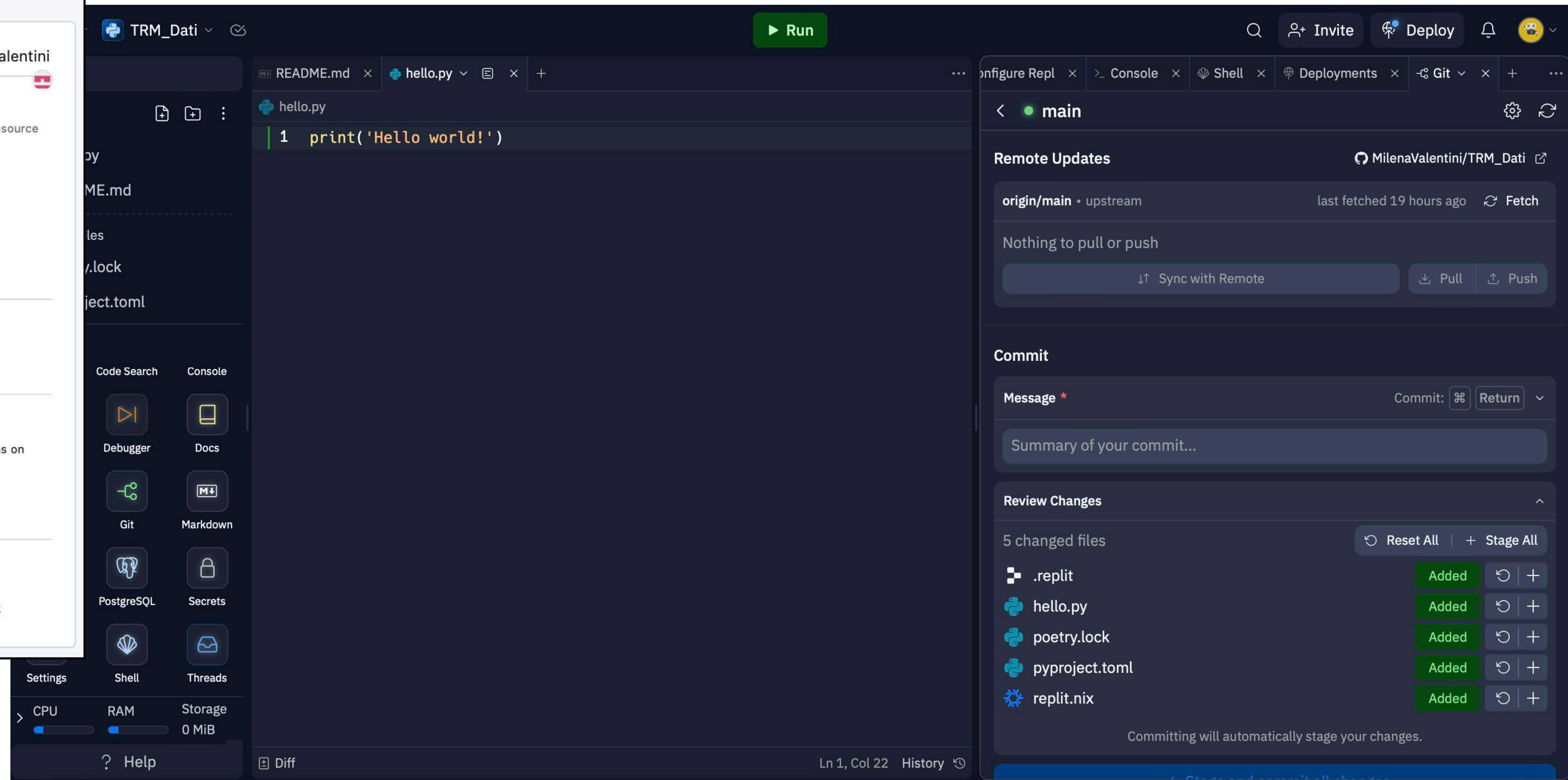
- ✓ Read access to metadata
- ✓ Read and write access to administration and code

User permissions  
Installing and authorizing Replit immediately grants these permissions on your account: MilenaValentini.

- ✓ Read access to email addresses

**Install & Authorize** Cancel

Next: you'll be redirected to <https://replit.com/auth/github/callback>



The screenshot shows the Replit IDE interface. The main editor displays a Python file named `hello.py` with the following code:

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

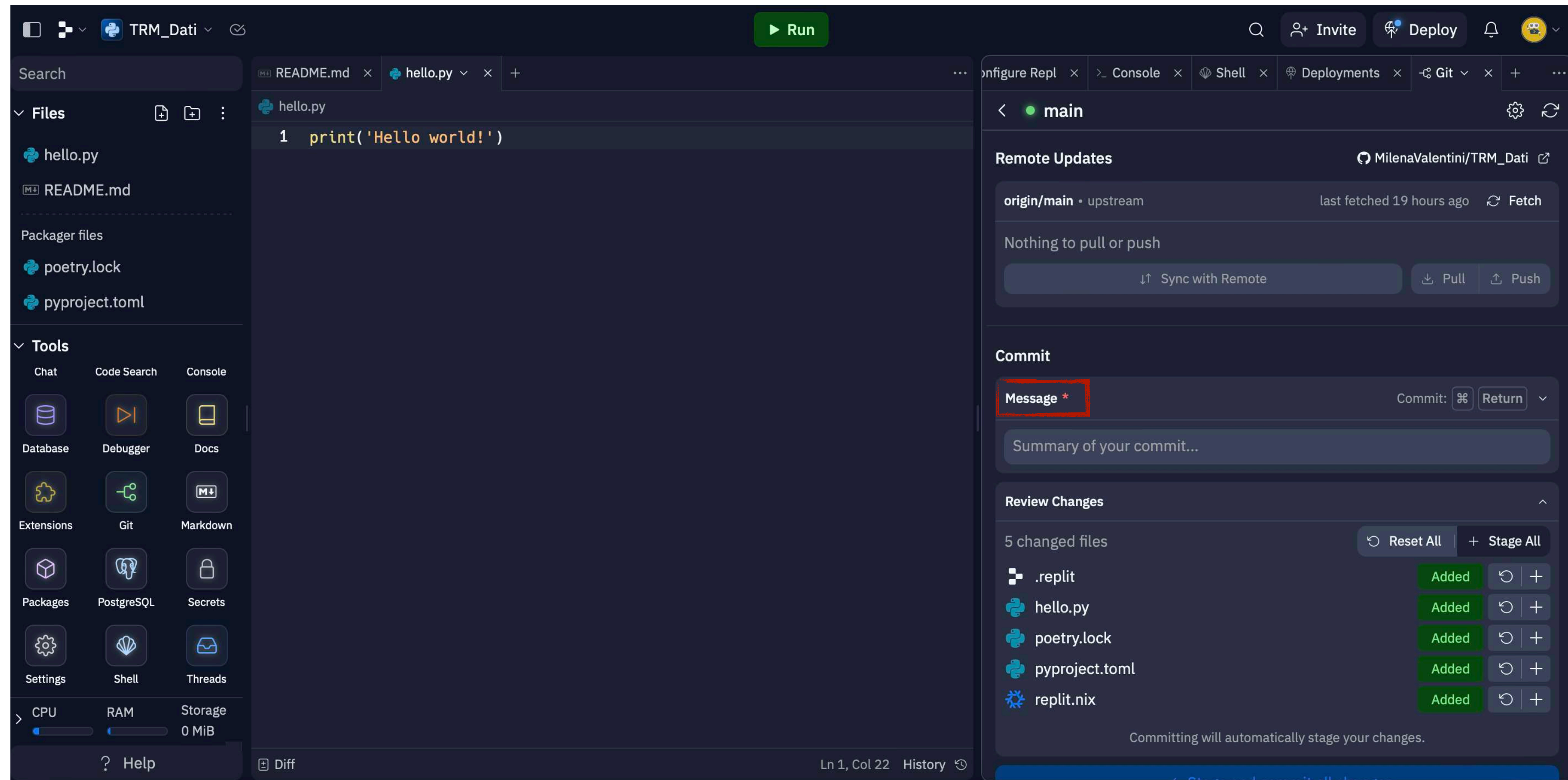
The interface includes a top navigation bar with a `Run` button, search, and user options. The left sidebar shows a file explorer with `README.md`, `poetry.lock`, and `pyproject.toml`. The bottom left shows system resources: CPU, RAM, and Storage (0 MiB). The right sidebar shows the Git interface for the `main` branch, indicating no updates to pull or push. Below this is a commit dialog with a message field and a list of 5 changed files:

File	Status	Actions
<code>.replit</code>	Added	Undo, Add
<code>hello.py</code>	Added	Undo, Add
<code>poetry.lock</code>	Added	Undo, Add
<code>pyproject.toml</code>	Added	Undo, Add
<code>replit.nix</code>	Added	Undo, Add

Authorize Replit on Git

# The Integrated Development Environment

Add message to describe your commit

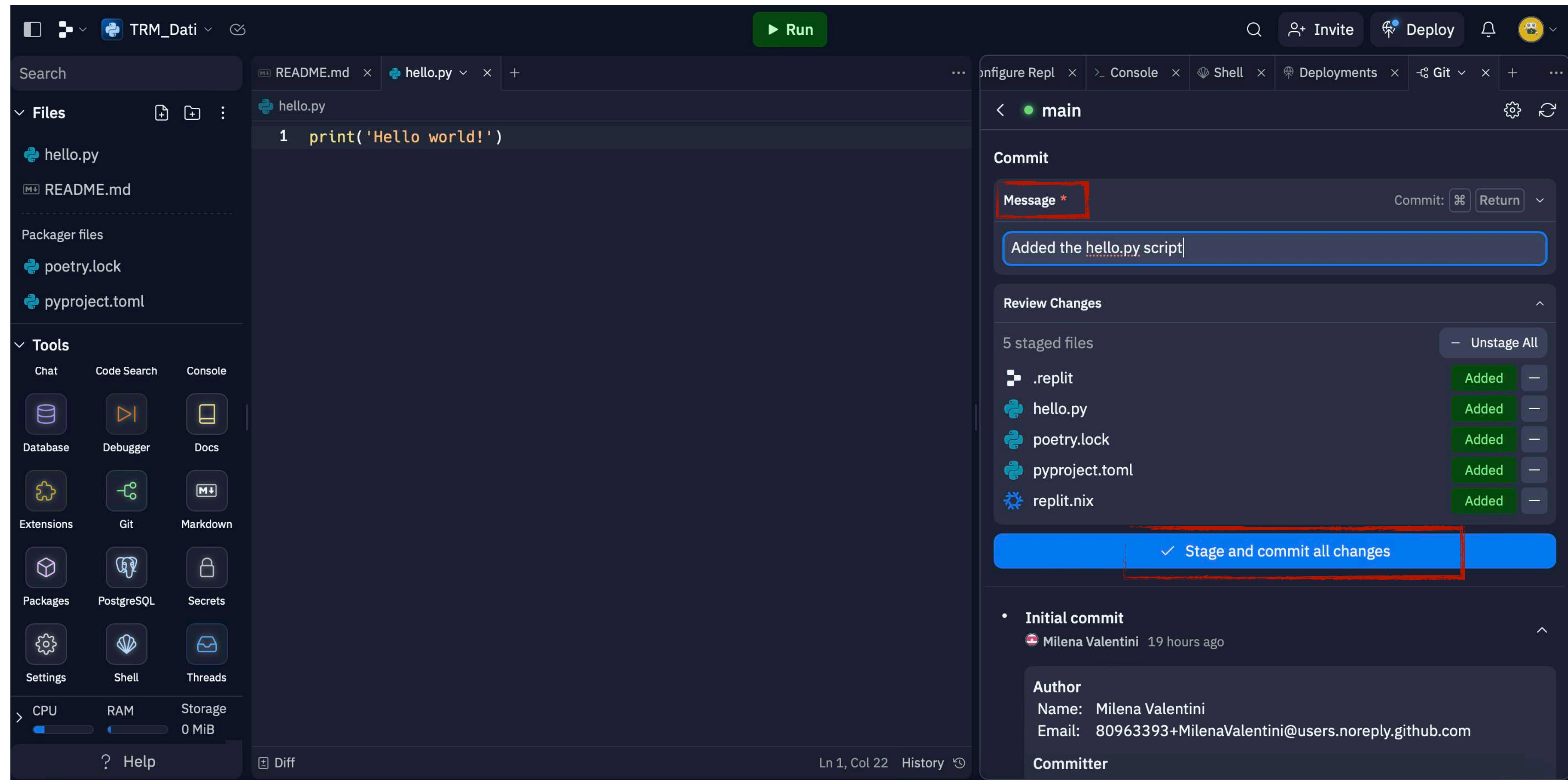


# The Integrated Development Environment

Add message to describe your commit

Stage files to commit (the hello.py script, and additional files internal to replit plus related to libraries)

Commit





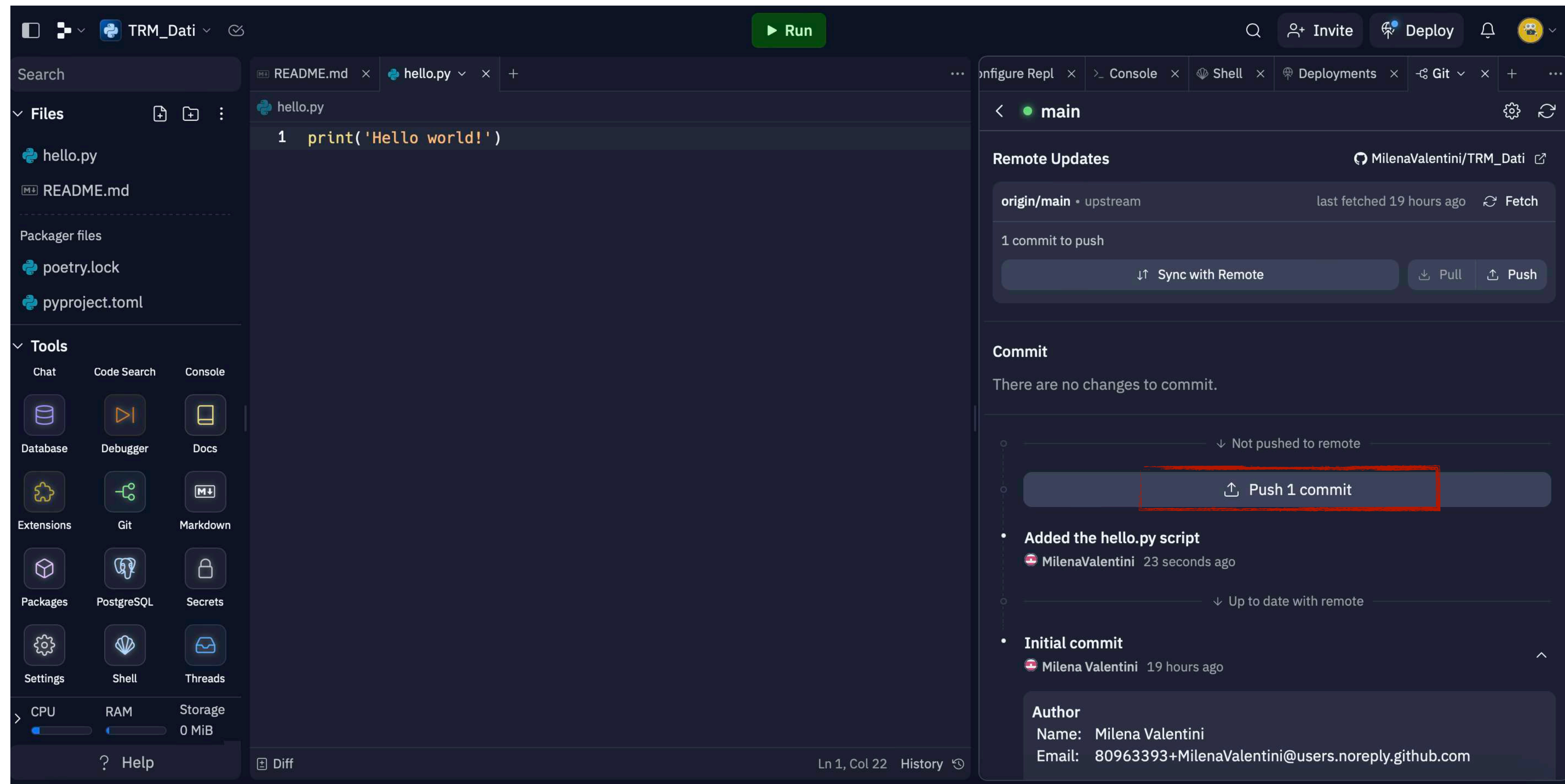
# The Integrated Development Environment

Add message to describe your commit

Stage files to commit (the hello.py script, and additional files internal to replit plus related to libraries)

Commit

Push to Git



# The Integrated Development Environment

This screenshot shows the GitHub repository page for 'MilenaValentini / TRM\_Dati'. The repository is public and has one branch, 'main'. The commit history shows an initial commit by MilenaValentini 19 hours ago. The README file content is visible, displaying the repository name and a description: 'Public repository of the course: Tecniche di Rappresentazione e Modellizzazione Dati'.

This screenshot shows the same GitHub repository page, but with a recent commit by MilenaValentini 1 minute ago. The commit message is 'Added the hello.py script'. The file list shows several files added in this commit: .replit, README.md, hello.py, poetry.lock, pyproject.toml, and replit.nix. The README content is also visible, showing the repository name and description.

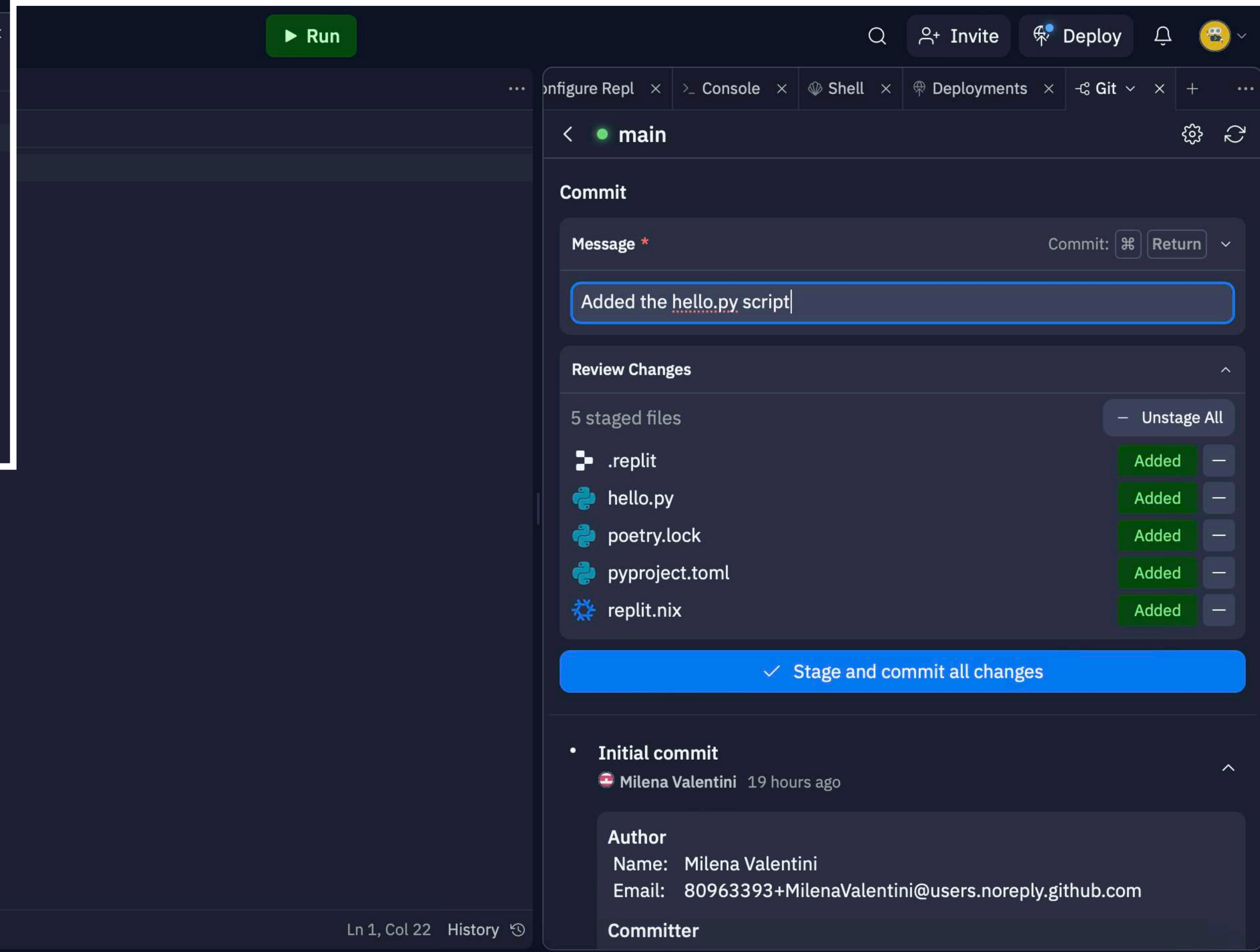
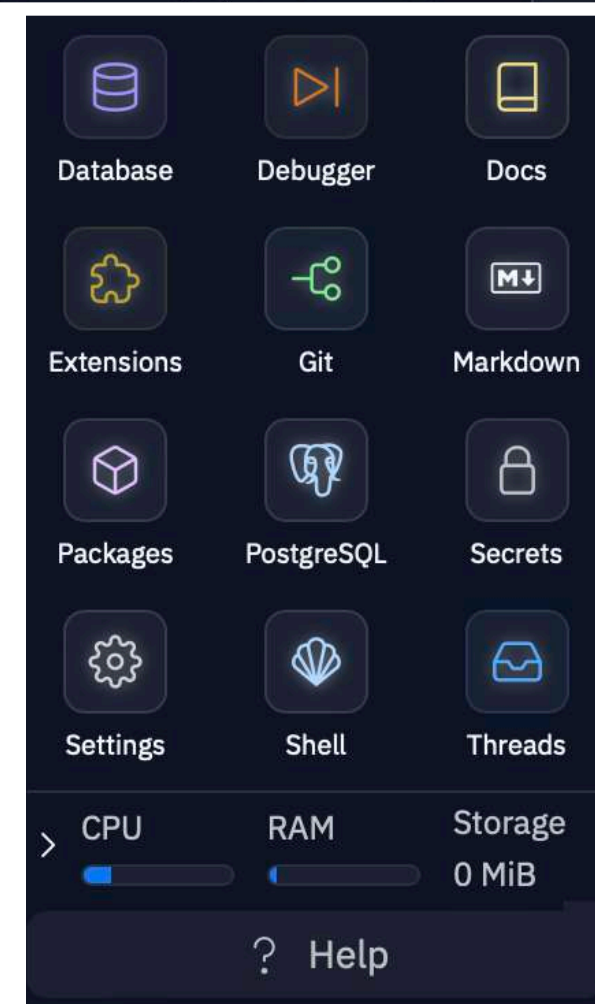
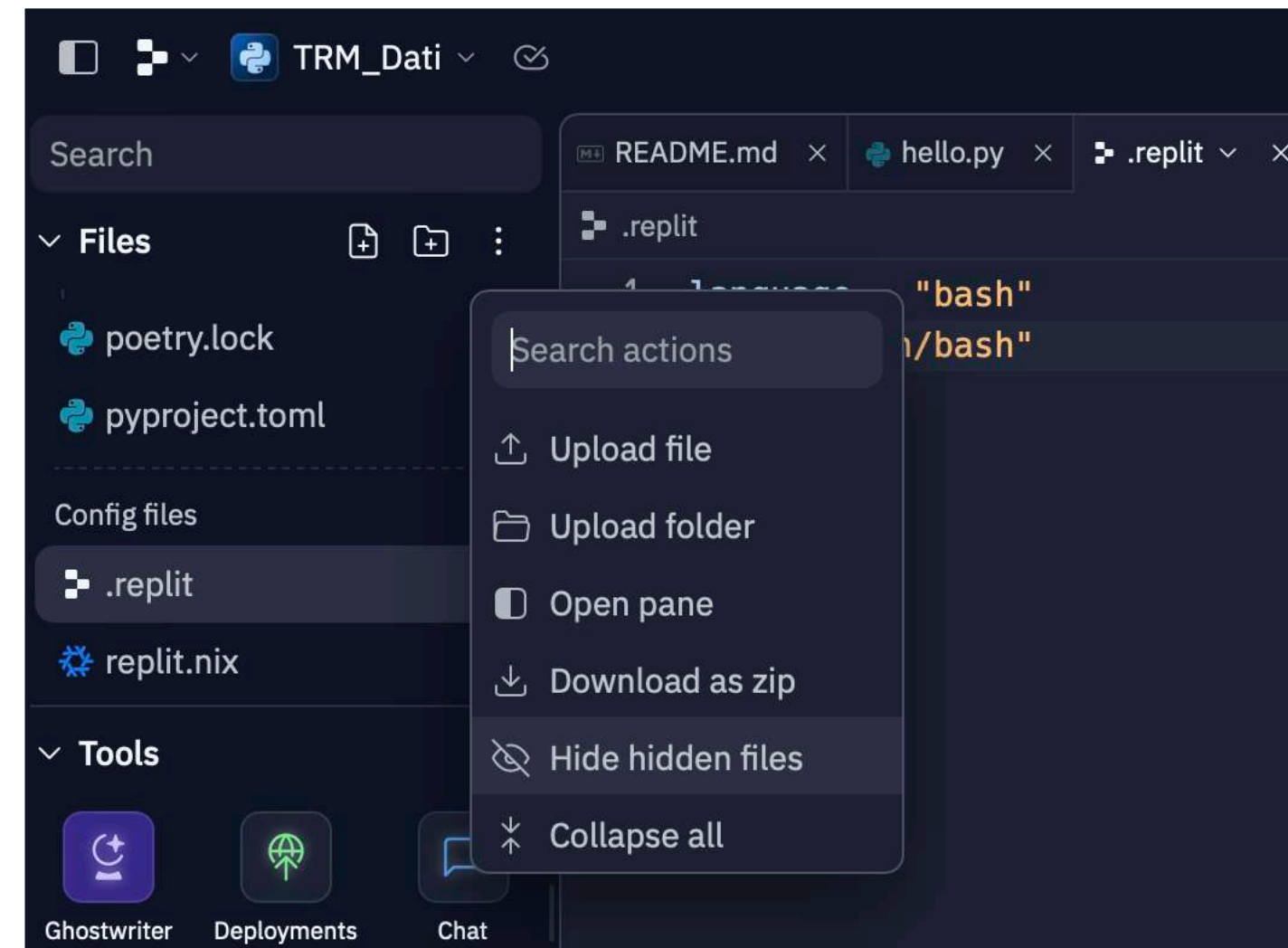
The Git repository has been successfully updated

# The Integrated Development Environment

Make sure to configure the hidden file `.replit` as follows:

language = "bash"  
run = "/bin/bash"

Stage, commit and push it to Git



# Git: Intro

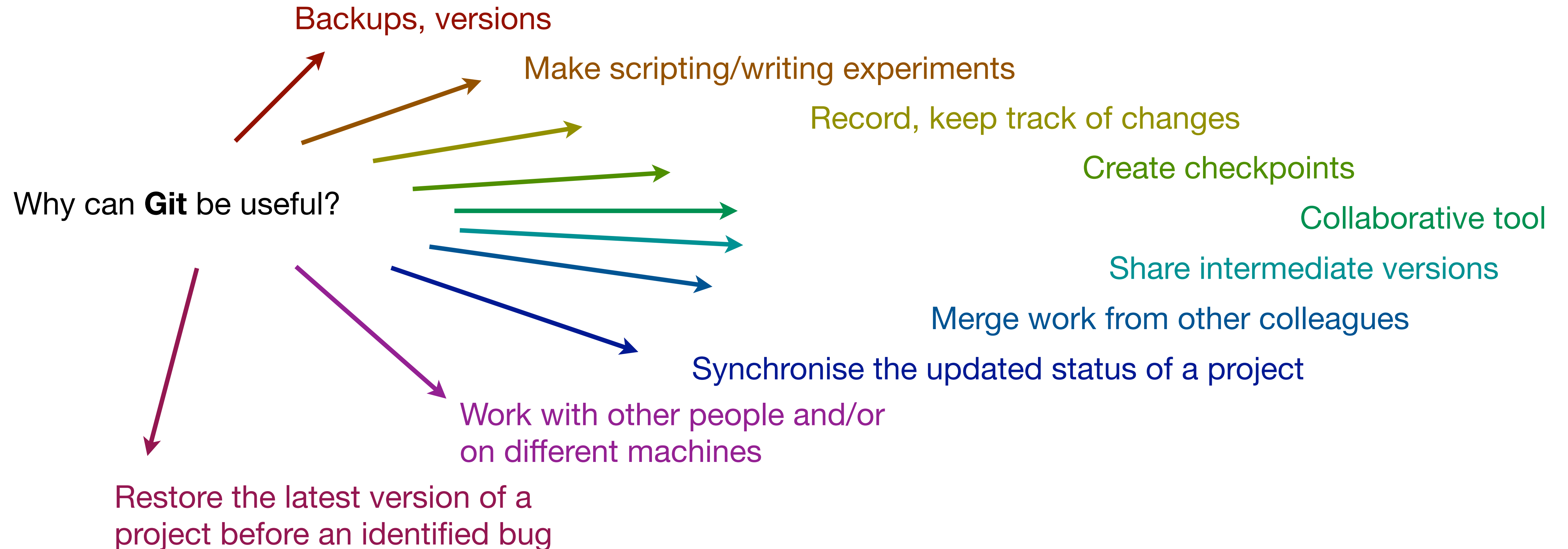
Git is the most used version control system and it's under continuous development

A screenshot of the Git website homepage. At the top left is the Git logo (a red diamond with a white branching diagram) followed by the word "git" in a bold, lowercase font. To the right of the logo is the tagline "--distributed-even-if-your-workflow-isnt". In the top right corner, there is a search bar with a magnifying glass icon and the text "Search entire site...". Below the header, the main content area has a light gray background with a subtle grid pattern. On the left side, there are two paragraphs of text. The first paragraph describes Git as a free and open source distributed version control system. The second paragraph describes Git as easy to learn and having a tiny footprint with lightning fast performance. On the right side, there is a 3D diagram showing several stacks of white papers representing code repositories. These stacks are connected by colored lines (red, blue, yellow) that represent branching and merging operations. The diagram shows a central stack with branches extending outwards and connecting to other stacks, illustrating the distributed nature of the system.

Git clients: <https://git-scm.com/downloads/guis/>

You can work from the terminal

# Git: essentials and how-to

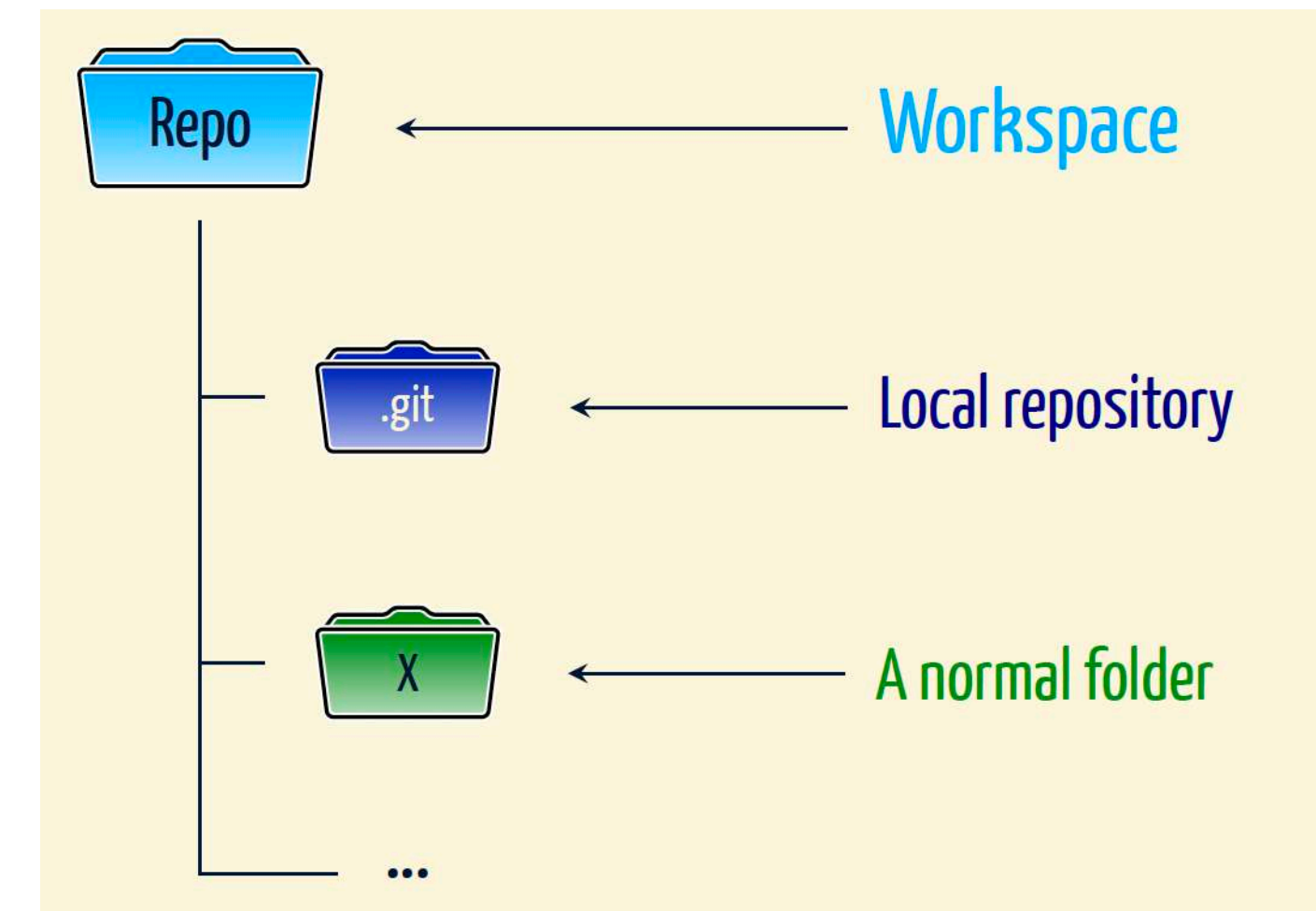


How would that be without Git?

Useful resources:  
<https://git-scm.com/book/en/v2>  
<https://github.com/AxelKrypton/Git-crash-course>

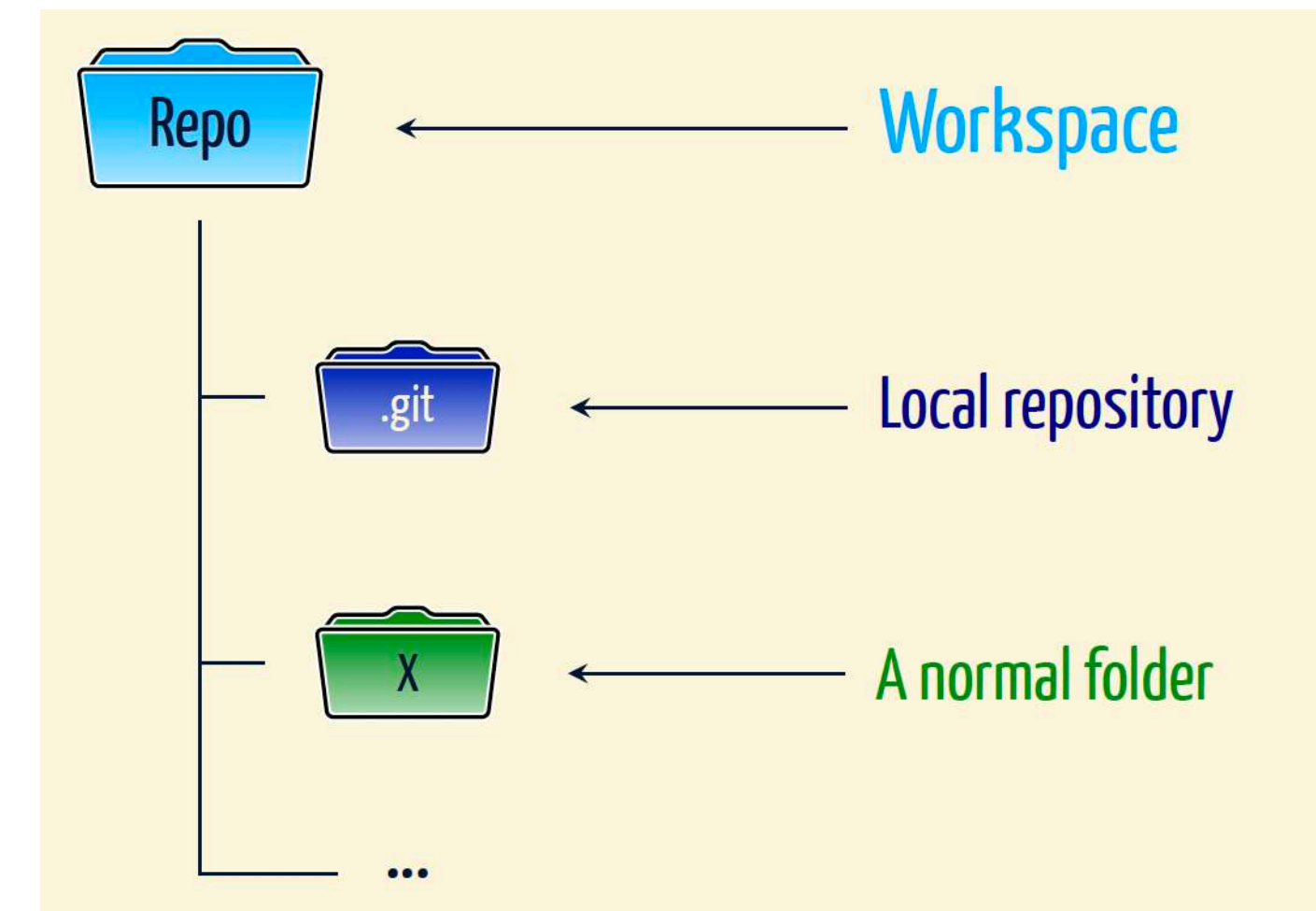
# Git: Intro

A repository is a directory containing all the versions of the files

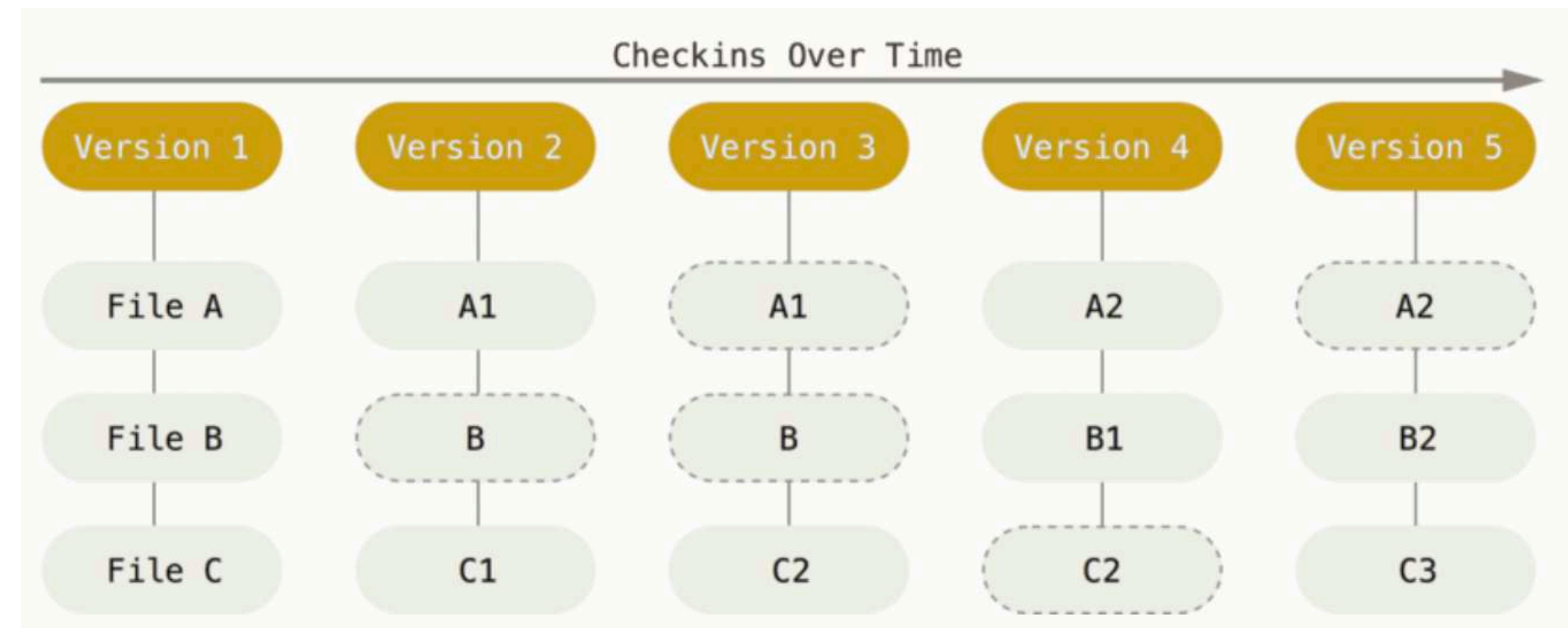


# Git: Intro

A repository is a directory containing all the versions of the files



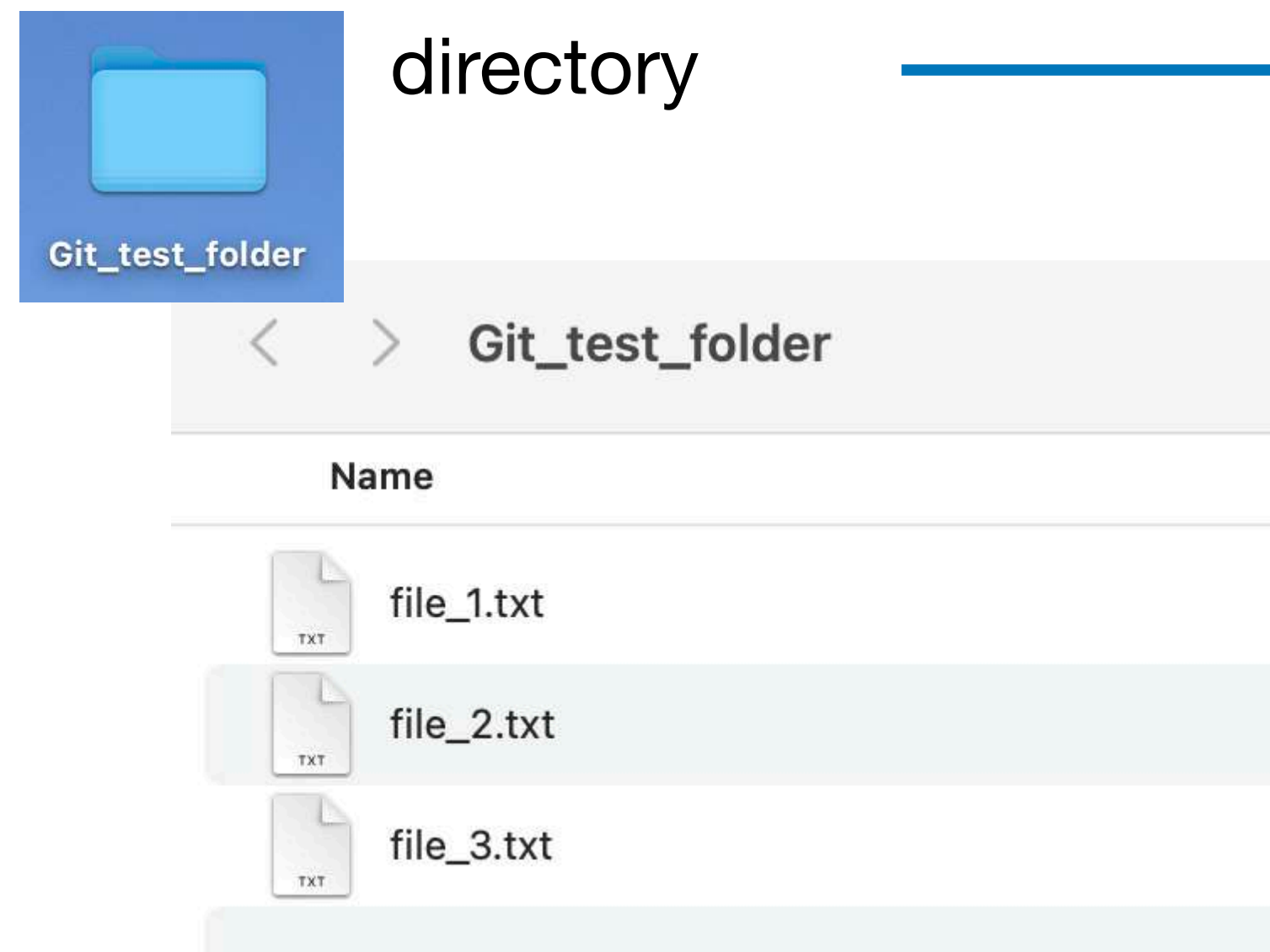
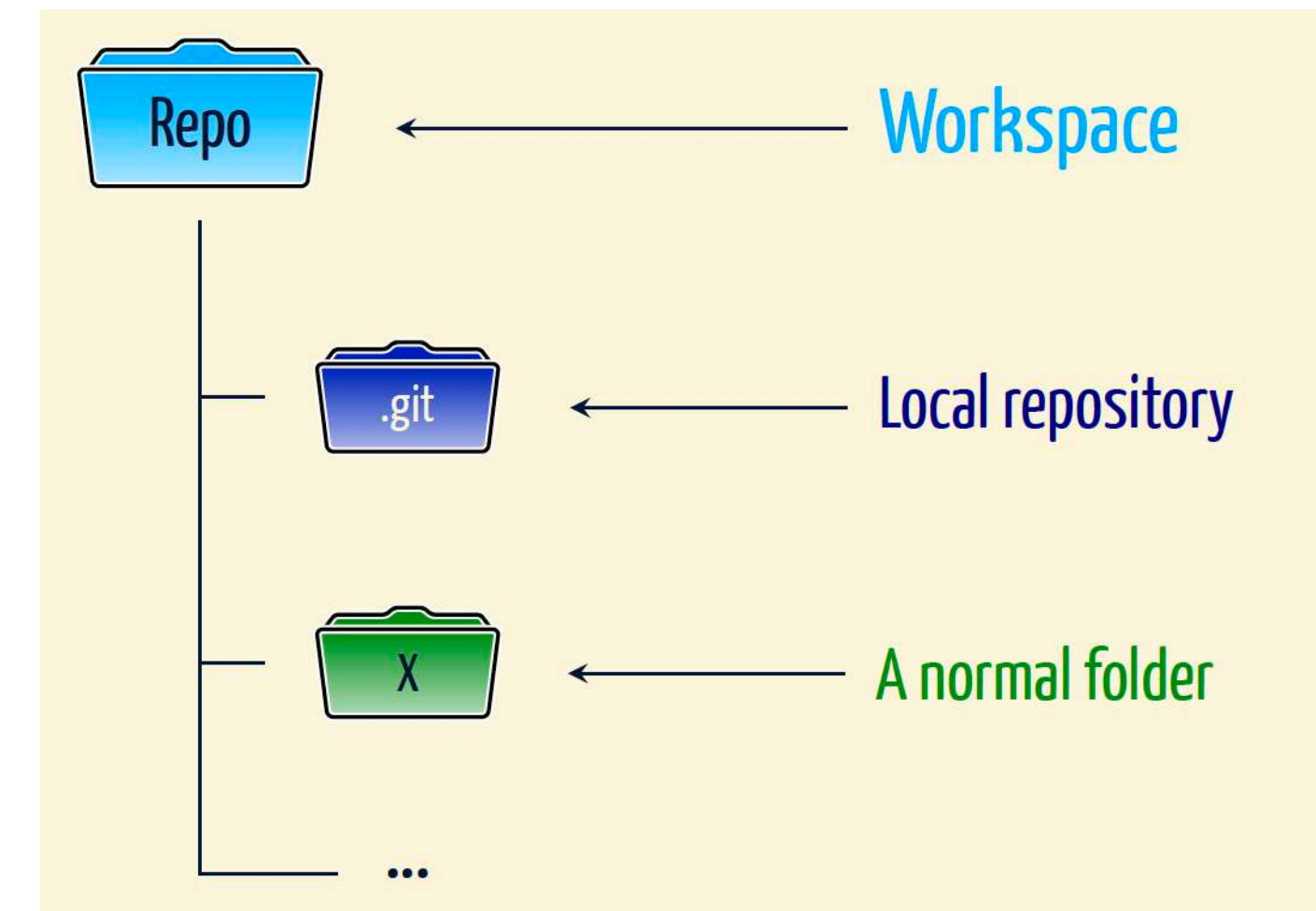
Records are based on snapshots taken over time



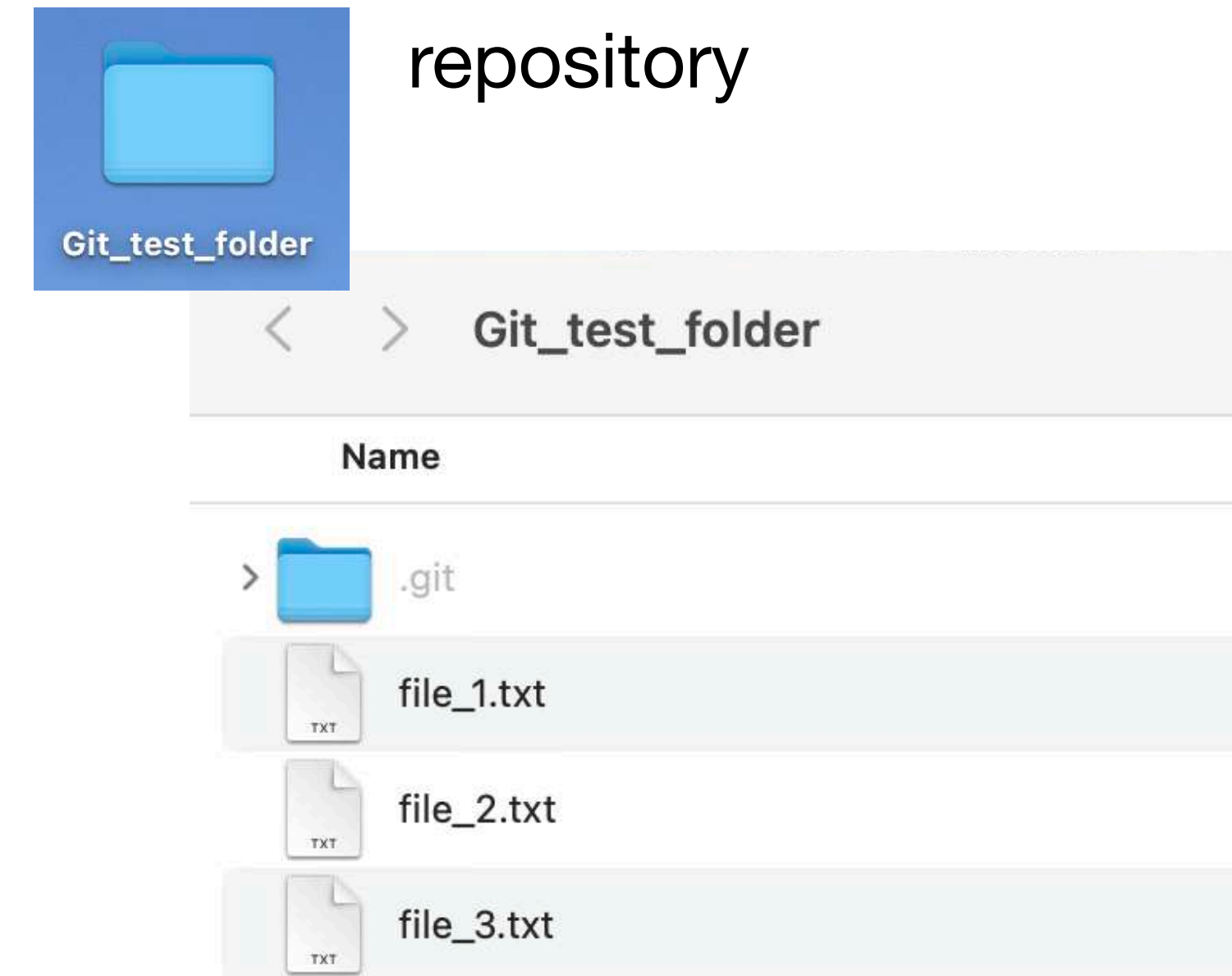
Snapshots are called commits and are referenced by a checksum

# Git: Intro

A repository is a directory containing all the versions of the files



**git init** →





# Git: essentials and how-to

```
[(base) MacBook-Pro-2:TRM_Dati_Lezioni milenavalentini$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        [--super-prefix=<path>] [--config-env=<name>=<envvar>]
        <command> [<args>]
```

These are common Git commands used in various situations:

start a working area (see also: `git help tutorial`)

<code>clone</code>	Clone a repository into a new directory
<code>init</code>	Create an empty Git repository or reinitialize an existing one

work on the current change (see also: `git help everyday`)

<code>add</code>	Add file contents to the index
<code>mv</code>	Move or rename a file, a directory, or a symlink
<code>restore</code>	Restore working tree files
<code>rm</code>	Remove files from the working tree and from the index

examine the history and state (see also: `git help revisions`)

<code>bisect</code>	Use binary search to find the commit that introduced a bug
<code>diff</code>	Show changes between commits, commit and working tree, etc
<code>grep</code>	Print lines matching a pattern
<code>log</code>	Show commit logs
<code>show</code>	Show various types of objects
<code>status</code>	Show the working tree status

grow, mark and tweak your common history

<code>branch</code>	List, create, or delete branches
<code>commit</code>	Record changes to the repository
<code>merge</code>	Join two or more development histories together
<code>rebase</code>	Reapply commits on top of another base tip

```
apt-get update
apt-get install (--user) git(-all)
```

# Git: essentials and how-to

```
(base) MacBook-Pro-2:~ milenavalentini$
(base) MacBook-Pro-2:~ milenavalentini$ cd Desktop/
(base) MacBook-Pro-2:Desktop milenavalentini$ mkdir Git_test_folder
(base) MacBook-Pro-2:Desktop milenavalentini$ cd Git_test_folder/
(base) MacBook-Pro-2:Git_test_folder milenavalentini$ touch file_1.txt
(base) MacBook-Pro-2:Git_test_folder milenavalentini$ touch file_2.txt
(base) MacBook-Pro-2:Git_test_folder milenavalentini$ touch file_3.txt
(base) MacBook-Pro-2:Git_test_folder milenavalentini$ ls
file_1.txt      file_2.txt      file_3.txt
(base) MacBook-Pro-2:Git_test_folder milenavalentini$ ls -la
total 0
drwxr-xr-x   5 milenavalentini  staff   160 Oct  1 22:00 .
drwx-----@ 50 milenavalentini  staff  1600 Oct  1 21:59 ..
-rw-r--r--   1 milenavalentini  staff    0 Oct  1 21:59 file_1.txt
-rw-r--r--   1 milenavalentini  staff    0 Oct  1 21:59 file_2.txt
-rw-r--r--   1 milenavalentini  staff    0 Oct  1 22:00 file_3.txt
(base) MacBook-Pro-2:Git_test_folder milenavalentini$ cd ..
(base) MacBook-Pro-2:Desktop milenavalentini$ git init Git_test_folder/
Initialized empty Git repository in /Users/milenavalentini/Desktop/Git_test_folder/.git/
(base) MacBook-Pro-2:Desktop milenavalentini$
(base) MacBook-Pro-2:Desktop milenavalentini$ cd Git_test_folder/
(base) MacBook-Pro-2:Git_test_folder milenavalentini$ ls -la
total 0
drwxr-xr-x   6 milenavalentini  staff   192 Oct  1 22:00 .
drwx-----@ 50 milenavalentini  staff  1600 Oct  1 21:59 ..
drwxr-xr-x   9 milenavalentini  staff   288 Oct  1 22:00 .git
-rw-r--r--   1 milenavalentini  staff    0 Oct  1 21:59 file_1.txt
-rw-r--r--   1 milenavalentini  staff    0 Oct  1 21:59 file_2.txt
-rw-r--r--   1 milenavalentini  staff    0 Oct  1 22:00 file_3.txt
```