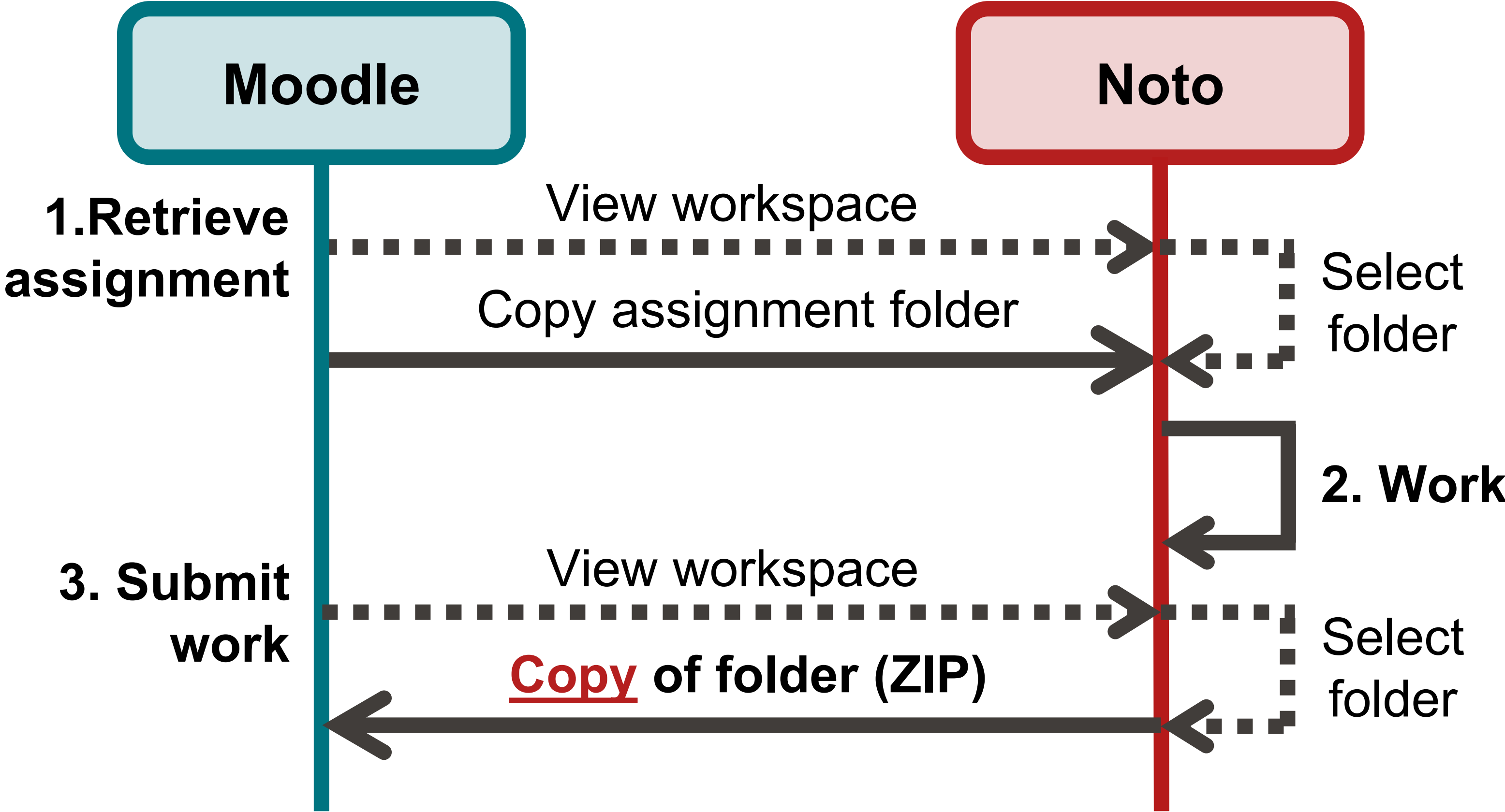


**Submission process
to test!**

Assignment: process on moodle & noto



Submission process to test!

1. Get the assignment folder

Go to moodle:

- Find the assignment “Test the submission process” (Fairness 1 section)
- Retrieve the assignment folder -> to your noto workspace

2. Work on the assignment

Go to noto:

- Modify the notebook

3. Submit your work

Go to moodle:

- Add a submission
- Select the folder test-assignment from noto -> to moodle

1- Retrieve the assignment folder



ASSIGNMENT

Test the moodle assignment [FOR TEST ONLY]

Opened: Saturday, 27 September 2025, 00:00

Due: Thursday, 9 October 2025, 23:59

[Add submission](#)


Submission status

Submission status	No submissions have been made yet
Grading status	Not graded
Time remaining	12 days 11 hours remaining
Last modified	-
Submission comments	▶ Comments (0)
Jupyter notebooks	Get a copy of the assignment

1- Retrieve the assignment folder

Below is a view of your Jupyter workspace. Please select the folder where to copy the assignment.
Feel free to create a folder in Jupyter before copying the assignment.

Destination folder ? /my_notebooks



- Documentation
- cs-290-responsible-software
- cs-290-responsible-software-2025
- git_HPLstudentsProjects
- git_JupyterExamples
- git_JupyterProjectsDRIL
- git_JupyterProjectsOther
- git_Noto
- git_Other
- my_notebooks
- my_venvs
- mitk_data
- noto-poc-notebooks
- phys-231-exercises
- Untitled.ipynb
- jnb.conf

Refresh tree Copy assignment Cancel

1- Retrieve the assignment folder



ASSIGNMENT

Test the moodle assignment [FOR TEST ONLY]

A copy of the assignment has been copied to "/my_notebooks/test-assignment".

Click here to get to your Jupyter workspace.

You can create another copy of the assignment or click "Cancel".

[Back to assignment](#)

The screenshot shows the Jupyter Notebook interface. The top menu bar includes File, Edit, View, Run, Kernel, Git, Nbgrader, Tabs, Settings, and Help. The left sidebar shows a file explorer with the following structure:

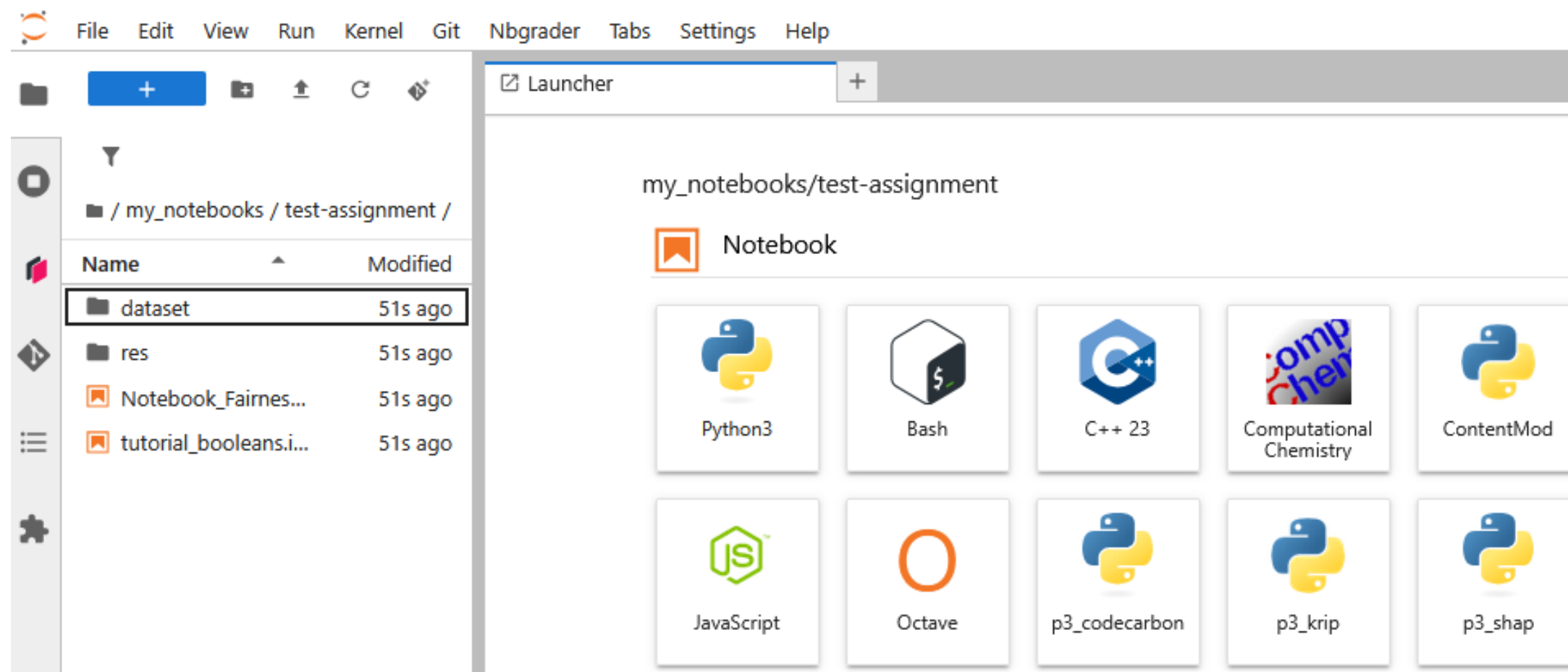
- / my_notebooks / test-assignment /
 - dataset (51s ago)
 - res (51s ago)
 - Notebook_Fairnes... (51s ago)
 - tutorial_booleans.i... (51s ago)

The main area is titled "my_notebooks/test-assignment" and shows a "Launcher" tab. Below the title, there is a "Notebook" section with a grid of 10 options:

- Python3
- Bash
- C++ 23
- Computational Chemistry
- ContentMod
- JavaScript
- Octave
- p3_codecarbon
- p3_krip
- p3_shap

2- Work on the assignment

- Make sure to save your work regularly
- Use print for debugging but make sure to remove the prints at the end



3- Submit your work (whole folder)



ASSIGNMENT

Test the moodle assignment [FOR TEST ONLY]

Opened: Saturday, 27 September 2025, 00:00

Due: Thursday, 9 October 2025, 23:59

[Add submission](#)

Submission status

Submission status	No submissions have been made yet
Grading status	Not graded
Time remaining	12 days 11 hours remaining
Last modified	-
Submission comments	▶ Comments (0)
Jupyter notebooks	Get a copy of the assignment

3- Submit your work (whole folder)

▼ Add submission

This is your Jupyter workspace. Please select the folder you want to submit.

Folder to submit



/my_notebooks/test-assignment

- Documentation
- cs-290-responsible-software
- cs-290-responsible-software-2025
- git_HPLstudentsProjects
- git_JupyterExamples
- git_JupyterProjectsDRIL
- git_JupyterProjectsOther
- git_Noto
- git_Other
- my_notebooks
 - test-assignment
 - dataset
 - res
 - Notebook_Fairness_1.ipynb
 - tutorial_booleans.ipynb
- my_vervs
- nltk_data
- notebook_tokeep
- jnb.conf

Refresh tree

Save changes

Cancel

3- Submit your work (whole folder)

Edit submission

Remove submission

Submission status

Submission status	Submitted for grading
Grading status	Not graded
Time remaining	Assignment was submitted 12 days 10 hours early
Last modified	Saturday, 27 September 2025, 13:16
Submission comments	▶ Comments (0)
Jupyter notebooks	Get a copy of the assignment View your submission

- After submission, provided the deadline is not passed, you can:
 - **Edit submission:** upload a **new** submission (erase and replace)
 - **Remove submission:** delete the submission (you can submit a new one)

4- View a copy of your submission

Edit submission

Remove submission

Submission status


Submission status	Submitted for grading
Grading status	Not graded
Time remaining	Assignment was submitted 12 days 10 hours early
Last modified	Saturday, 27 September 2025, 13:16
Submission comments	▶ Comments (0)
Jupyter notebooks	Get a copy of the assignment View your submission




4- View a copy of your submission

You have already made a Jupyter notebook submission.
Below you can retrieve a copy of your current submission.



Submission date: Sat Sep 27 13:16:47 CEST 2025

Below is a view of your Jupyter workspace. Please select the folder where to copy your current submission.
Feel free to create a folder in Jupyter before copying the submission.

Destination folder 

- Documentation
- cs-290-responsible-software
- cs-290-responsible-software-2025
- git_HPLstudentsProjects
- git_JupyterExamples
- git_JupyterProjectsDRIL
- git_JupyterProjectsOther
- git_Noto
- git_Other
- my_notebooks
- my_venvs
- nltk_data
- notebook_tokeep
- jnb.conf

-   After this step, you will have **2 versions of your work** on noto:
1. The original assignment folder in which you have worked
 2. A copy of this assignment folder, i.e. a **duplicate**

Important advice – Submitting your work

- The content available on moodle is the **reference for grading**
- After submission, **changes on noto** are NOT reflected on moodle **unless you resubmit**
- **Do NOT wait until the last moment to submit!!!!**
 - You can re-submit as many times as you want until the deadline
 - Each re-submission will erase and replace the previous

Important advice – Working in the notebook

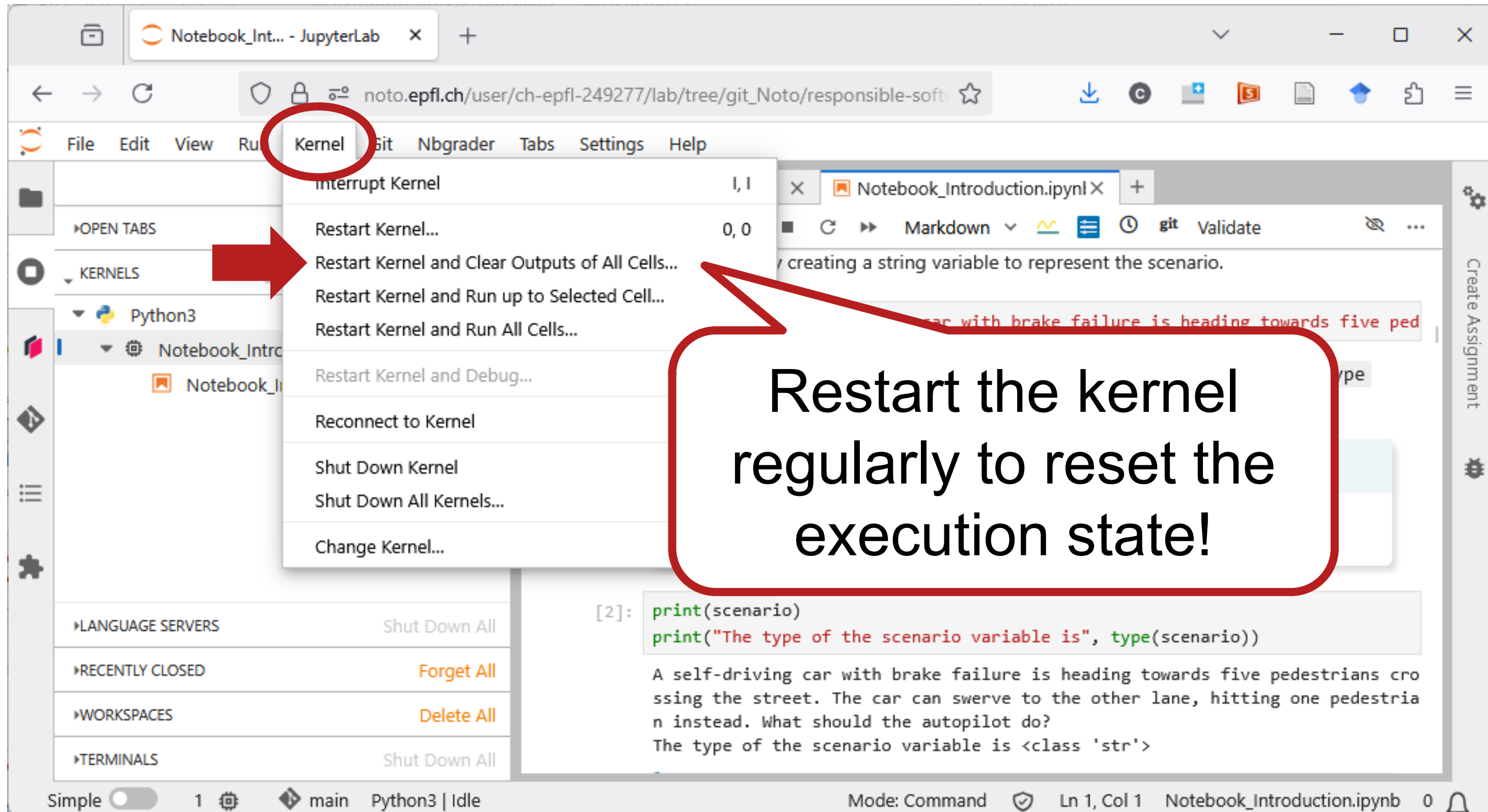
■ For your notebook to be correctly graded:

- Do not **delete** any cell!
- Remove all calls to **print()** from your code/functions (e.g., for debug)
- Regularly **restart** the kernel and run all cells
- Cite your **numerical results** from your code in your answers to open questions (we don't have code outputs when we grade open questions)

■ To obtain the best performance on noto:

- Do not open multiple notebooks in browser tabs!
- Open maximum 3 notebooks at the same time in noto (in panels)
- Regularly kill the kernels you do not use

Actions on the kernel



The image shows a screenshot of the JupyterLab interface. The 'Kernel' menu is open, and the 'Restart Kernel...' option is highlighted with a red arrow. A red callout box contains the text: 'Restart the kernel regularly to reset the execution state!'. The background shows a notebook with Python code and its output.

Kernel menu options:

- Interrupt Kernel
- Restart Kernel...
- Restart Kernel and Clear Outputs of All Cells...
- Restart Kernel and Run up to Selected Cell...
- Restart Kernel and Run All Cells...
- Restart Kernel and Debug...
- Reconnect to Kernel
- Shut Down Kernel
- Shut Down All Kernels...
- Change Kernel...

Callout text:

Restart the kernel regularly to reset the execution state!

Code in the notebook:

```
[2]: print(scenario)
print("The type of the scenario variable is", type(scenario))
```

Output of the code:

```
A self-driving car with brake failure is heading towards five pedestrians crossing the street. The car can swerve to the other lane, hitting one pedestrian instead. What should the autopilot do?
The type of the scenario variable is <class 'str'>
```

Saving your work

The image shows a JupyterLab notebook interface. A red callout box with the text "Ctrl + S" points to the save icon (a floppy disk) in the notebook's toolbar. The notebook contains two code cells. The first cell, labeled [2], contains the following Python code:

```
scenario = "A self-driving car with brake failure is heading towards five ped |
```

The second cell, labeled [3], contains the following Python code:

```
print(scenario)
print("The type of the scenario variable is", type(scenario))
```

The output of the second cell is:

```
A self-driving car with brake failure is heading towards five pedestrians cro
ssing the street. The car can swerve to the other lane, hitting one pedestria
n instead. What should the autopilot do?
The type of the scenario variable is <class 'str'>
```

The interface also shows a file browser on the left with a file named "Notebook_Introduction.ipynb" selected. The status bar at the bottom indicates "Mode: Command" and "Ln 1, Col 1".

Opening multiple notebooks

The image shows a screenshot of the JupyterLab web interface. At the top, the browser's tab bar contains two tabs: "Notebook_Int... (2) - JupyterLab" and "Sample_Py3.i... - auto-5". A red oval highlights these tabs. A red callout bubble with a warning icon and the text "Avoid opening noto in multiple browser tabs!" points to the tabs. The main interface shows a file browser on the left with a list of files and folders, including "Sample_HTML-PDF", "Sample_Octave", and several ".ipynb" files. The main content area displays the "Introduction: Python and ethical dilemmas" notebook, which includes a Creative Commons Attribution International License (CC BY 4.0 International) logo and introductory text.

⚠️ Avoid opening noto in multiple browser tabs!

Name	Modified
Sample_HTML-PDF	yesterday
Sample_Octave	yesterday
Sample_Bash.ipynb	yesterday
Sample_C_CPP.ipynb	yesterday
Sample_dask.ipynb	yesterday
Sample_Font.ipynb	yesterday
Sample_panel.ipynb	yesterday
Sample_Py3_cartopy.ipynb	yesterday
Sample_Py3_R.ipynb	yesterday
• Sample_Py3.ipynb	next yr.
Sample_R.ipynb	yesterday
Sample_SoS.ipynb	yesterday

Introduction: Python and ethical dilemmas

Notebook by Athina Papageorgiou Koufidou, Cécile Hardebolle and the Responsible software team (2025).

Except where otherwise noted, the content of this notebook is licensed under a [Creative Commons Attribution International License \(CC BY 4.0 International\)](#).

CC BY

Introduction

Welcome to the first exercise session of Responsible Software!

This tutorial notebook will serve as an introduction to Python basics, as well as some

Opening multiple notebooks

The screenshot shows the JupyterLab interface. The top bar displays three open notebooks: Notebook_Introduction.ipyn, 30_Using_git.ipynb, and Sample_Py3.ipynb. A red circle highlights this top bar. A callout box with a red border points to the text "Instead, open the notebooks inside noto". The main content area shows the "Python3 Notebook" with the following text:

First example

This first example will plot a simple

$$e^{-x} = \sum_{i=0}^{\infty} \frac{(-1)^i x^i}{i!}$$

Well... not really. This was to demo *L^AT_EX* equations, courtesy of `MathJax`.

The equation we'll plot is the following:

$$y = \sin(x) + \cos(x)$$

```
[ ]: import numpy
      from matplotlib import pyplot
```

At the bottom, the status bar shows "Mode: Command", "Ln 1, Col 1", and "Sample_Py3.ipynb 0".

Opening multiple notebooks

The screenshot shows the JupyterLab web interface. At the top, there's a browser window with the URL `noto.epfl.ch/user/ch-epfl-249277/lab/tree/Documentation/Examples/Samp`. Below the browser, the JupyterLab menu bar includes `File`, `Edit`, `View`, `Run`, `Kernel`, `Git`, `Nbgrader`, `Tabs`, `Settings`, and `Help`. Two notebook panels are open side-by-side. The left panel is titled `Notebook_Introduction.ipynl` and shows the title `Introduction: Python and al dilemmas`. The right panel is titled `Sample_Py3.ipynb` and shows the title `Python3 Notebook` and the text `First example`. The interface also shows a sidebar on the left with a file tree and a bottom status bar with `Simple`, `3`, `main`, `Python3 | Idle`, `Mode: Command`, `Ln 1, Col 1`, `Sample_Py3.ipynb`, and `0`.

You can collapse the side bar

You can arrange panels side by side! (or any other config you want)

Where is the kernel?

The image shows a screenshot of the JupyterLab web interface. The browser address bar shows the URL `noto.epfl.ch/user/ch-epfl-249277/lab/tree/git_Noto/responsible-soft`. The JupyterLab menu bar includes File, Edit, View, Run, Kernel, Git, Nbgrader, Tabs, Settings, and Help. On the left sidebar, the 'Kernels' panel is highlighted with a red circle, and a red arrow points to the 'Python3' kernel. Below it, the active notebook 'Notebook_Introduction.ipynb' is listed. A red speech bubble with the text 'List of active kernels' points to the Kernels panel. The main notebook area shows a code cell with the following content:

```
[1]: scenario = "A self-driving car with brake failure is heading towards five ped
```

The cell output shows the execution of the code:

```
print(scenario)
print("The type of the scenario variable is", type(scenario))
```

The output text reads: "A self-driving car with brake failure is heading towards five pedestrians crossing the street. The car can swerve to the other lane, hitting one pedestrian instead. What should the autopilot do? The type of the scenario variable is <class 'str'>"

Error messages

- If you start getting **popups with error messages**, it means that
 - ⚠ your personal server is no longer available ⚠
 - Timeout after \approx 30 minutes of inactivity
 - Too many resources used
- Do not simply dismiss or click ok!
The interface looks like it works but it does NOT!
- **Log out** and then log in again:
 - File > Logout
 - <https://noto.epfl.ch/hub/logout>

