Short Introduction to Jupyter and Python for Numerical Operations

Why use Python? How does it work?

- Compared to Excel and Matlab, Python has several advantages
  - it is free and developed by a very active community (not a company)
  - it has a large number of specific modules (“packages”) that have been developed for specific applications (like numpy and scipy for numerical calculations)
  - it is easy to learn thanks to many available tutorials

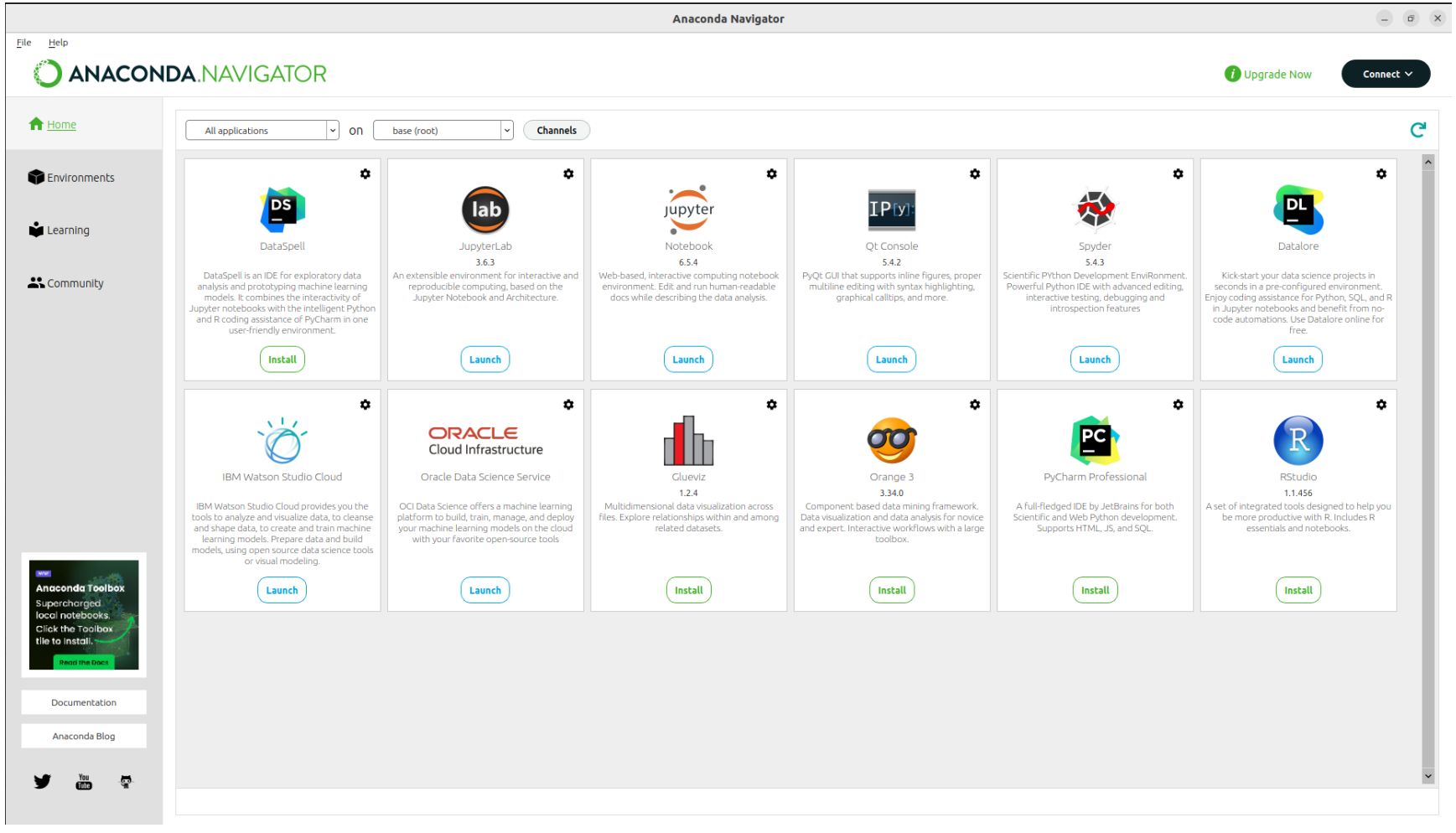
- Python is a programming language that relies on “scripts”, that are text files with code written in successive lines
  - typical Python scripts end with suffix “.py”

- Anaconda is convenient to install Python on any computer (with Linux, Windows, Mac-OS) and handle packages (and dependencies when installing them)

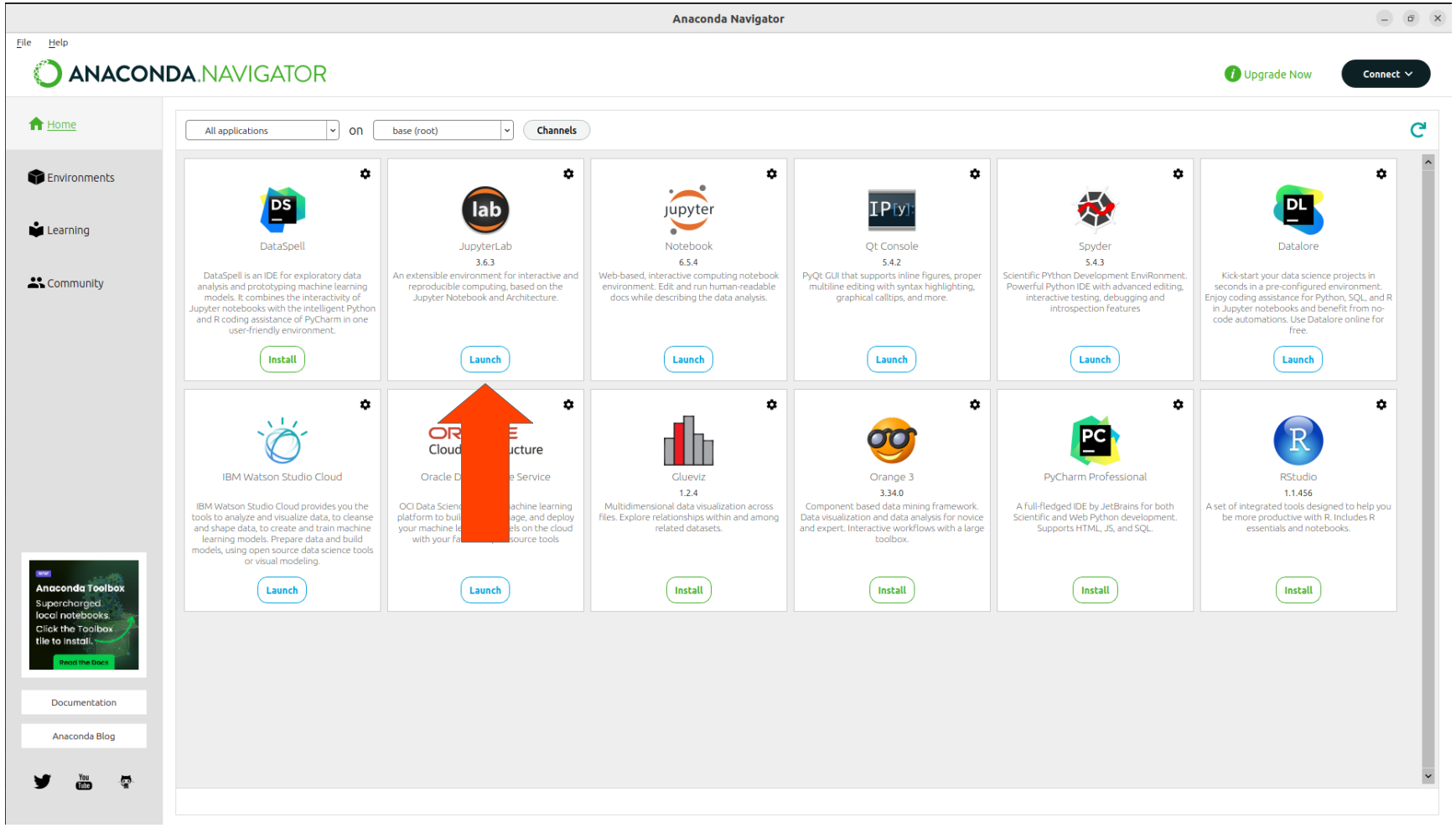
Install Anaconda

- Go to https://www.anaconda.com/download or https://docs.conda.io/projects/miniconda/en/latest/miniconda-install.html
- Follow the instructions to install Anaconda or Miniconda on your computer
  - you may have to go to the folder where conda is installed to run in command line `conda init` for the system to know where conda is located
- Launch anaconda-navigator
- Launch jupyter-lab
  - notebooks are a mixture of code (like Python) and text / images that is very useful to didactic presentations
  - let’s first see how to manipulate the notebook cells, then let’s write Python code

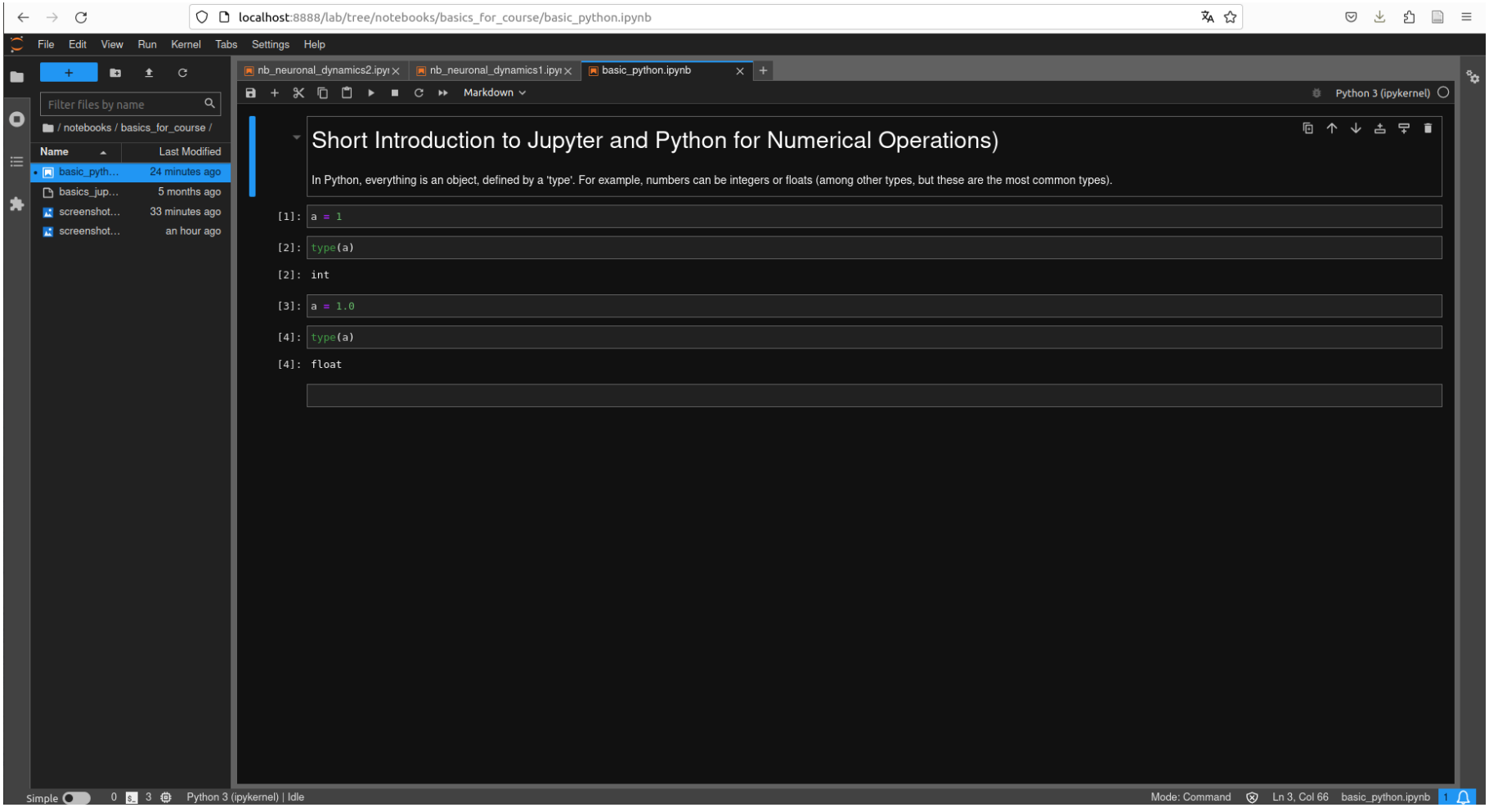
Launch Jupyter Lab



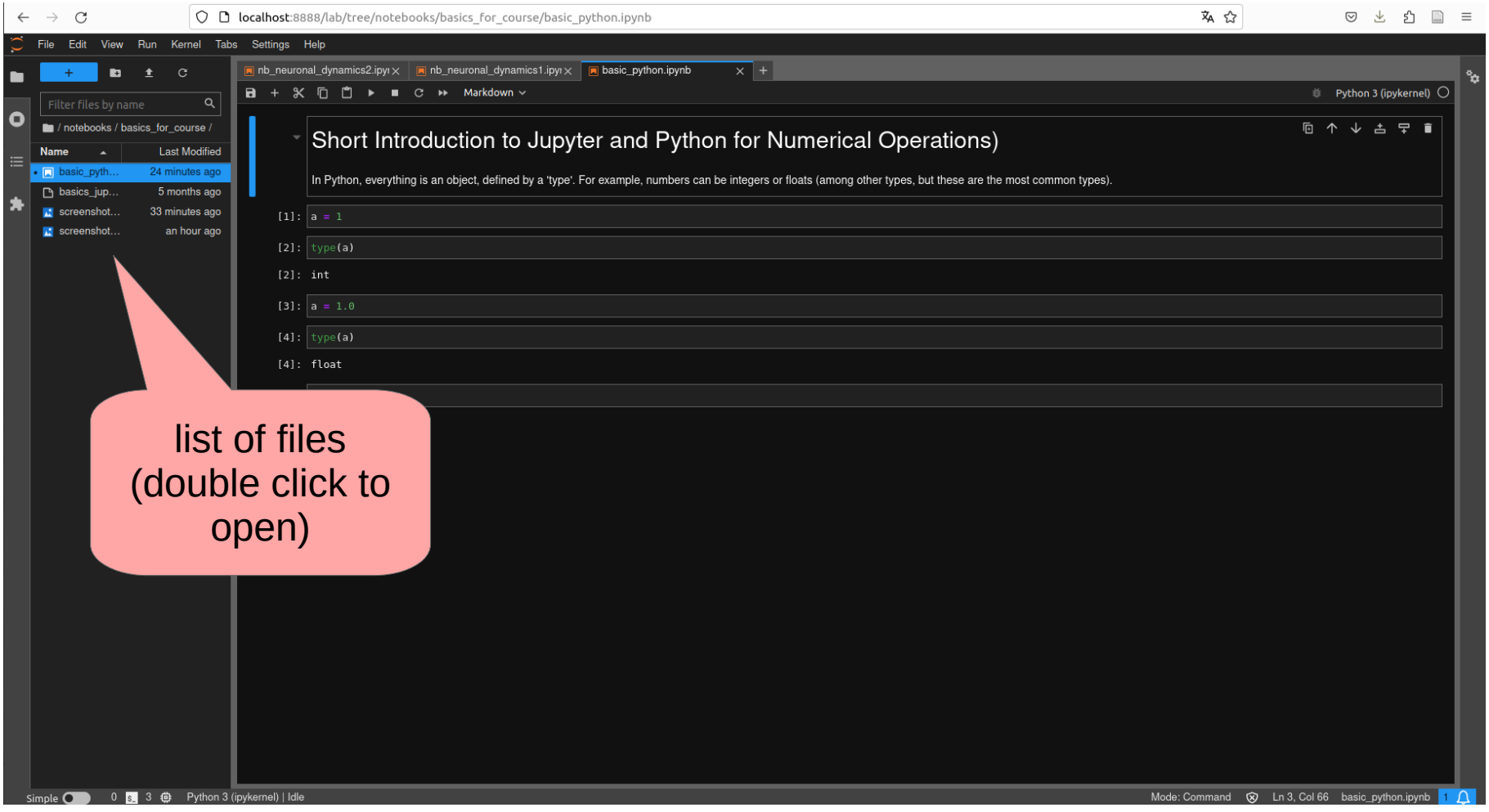
Launch Jupyter Lab



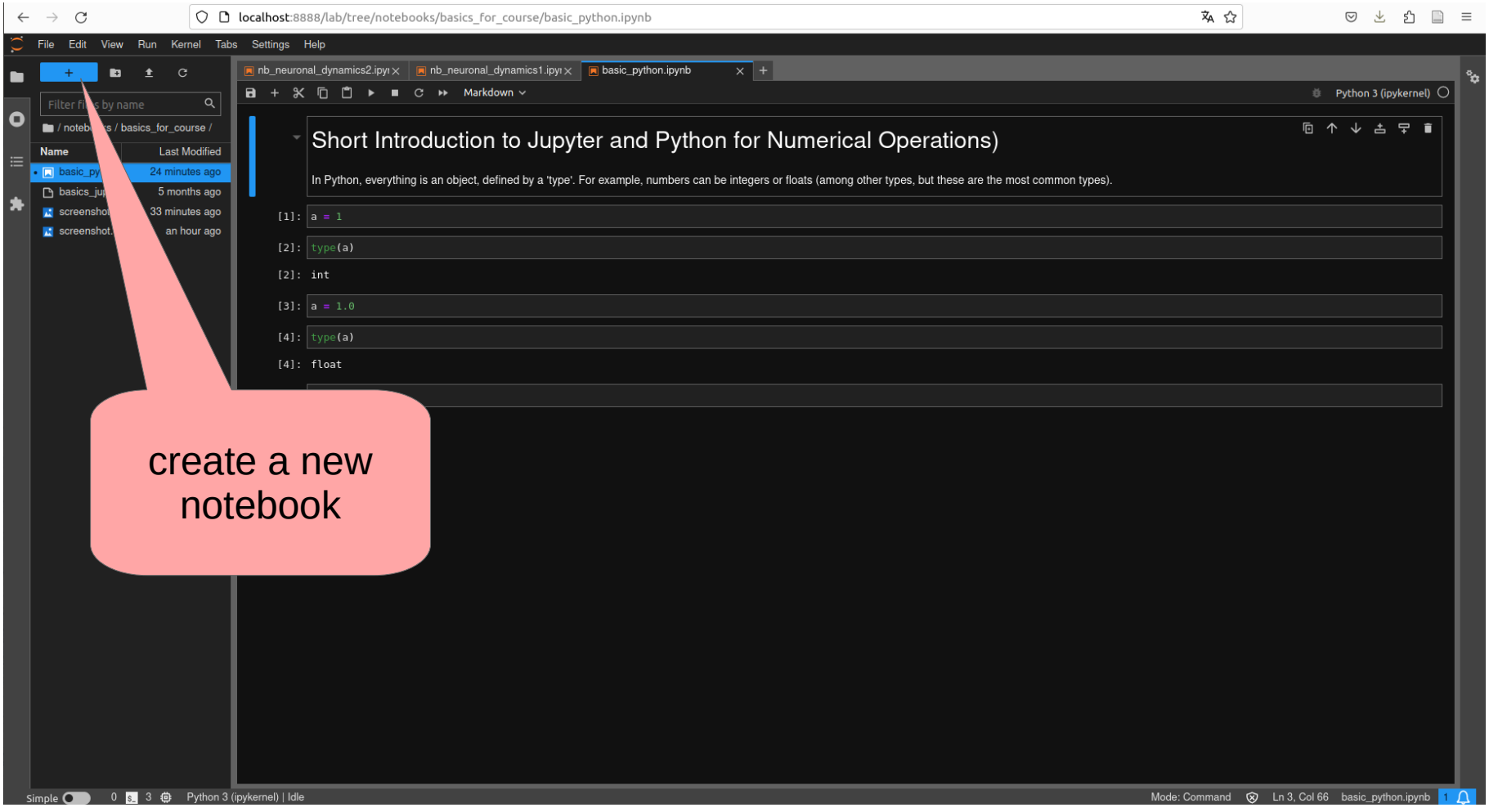
Cells in Jupyter



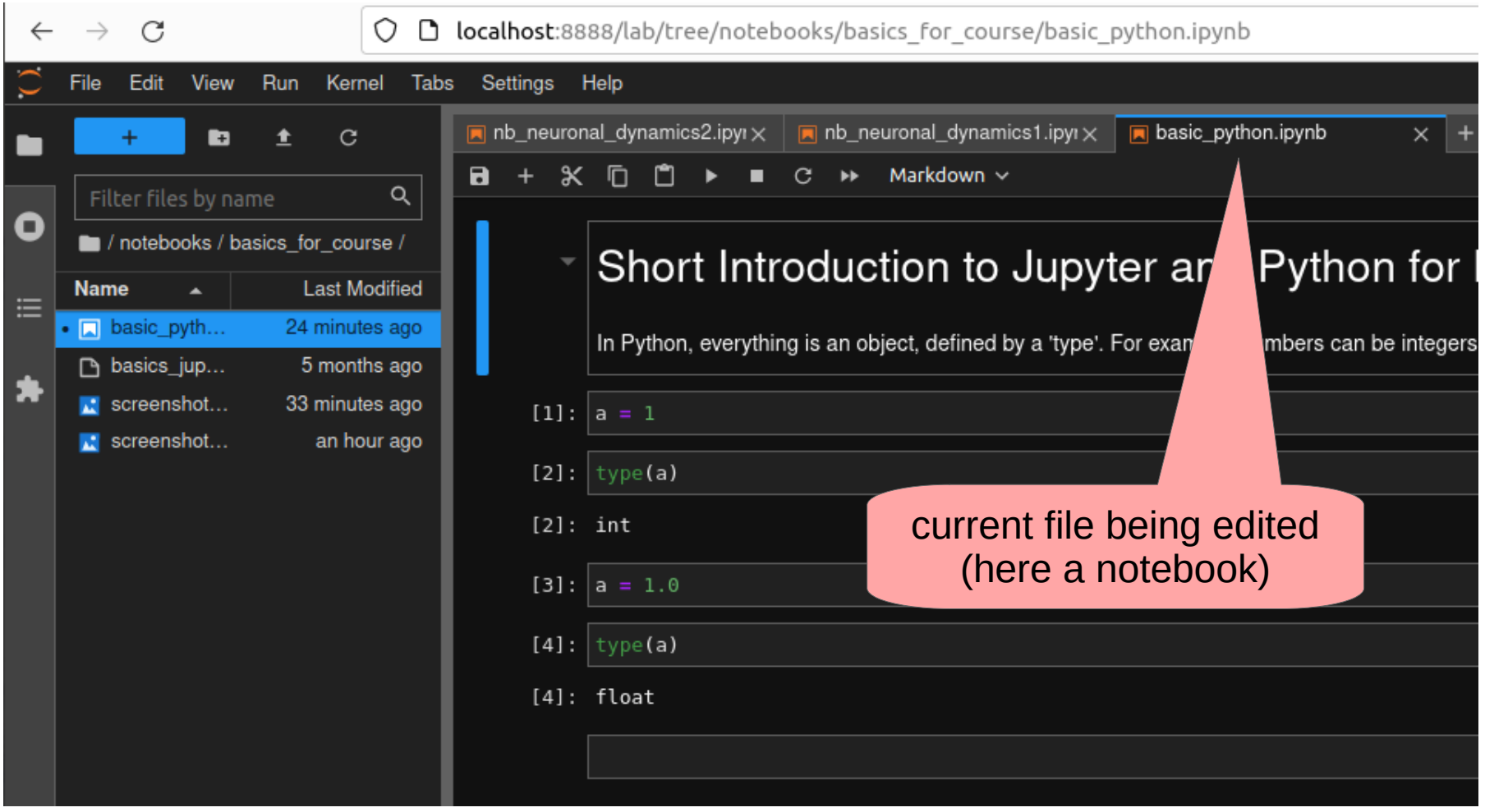
Cells in Jupyter



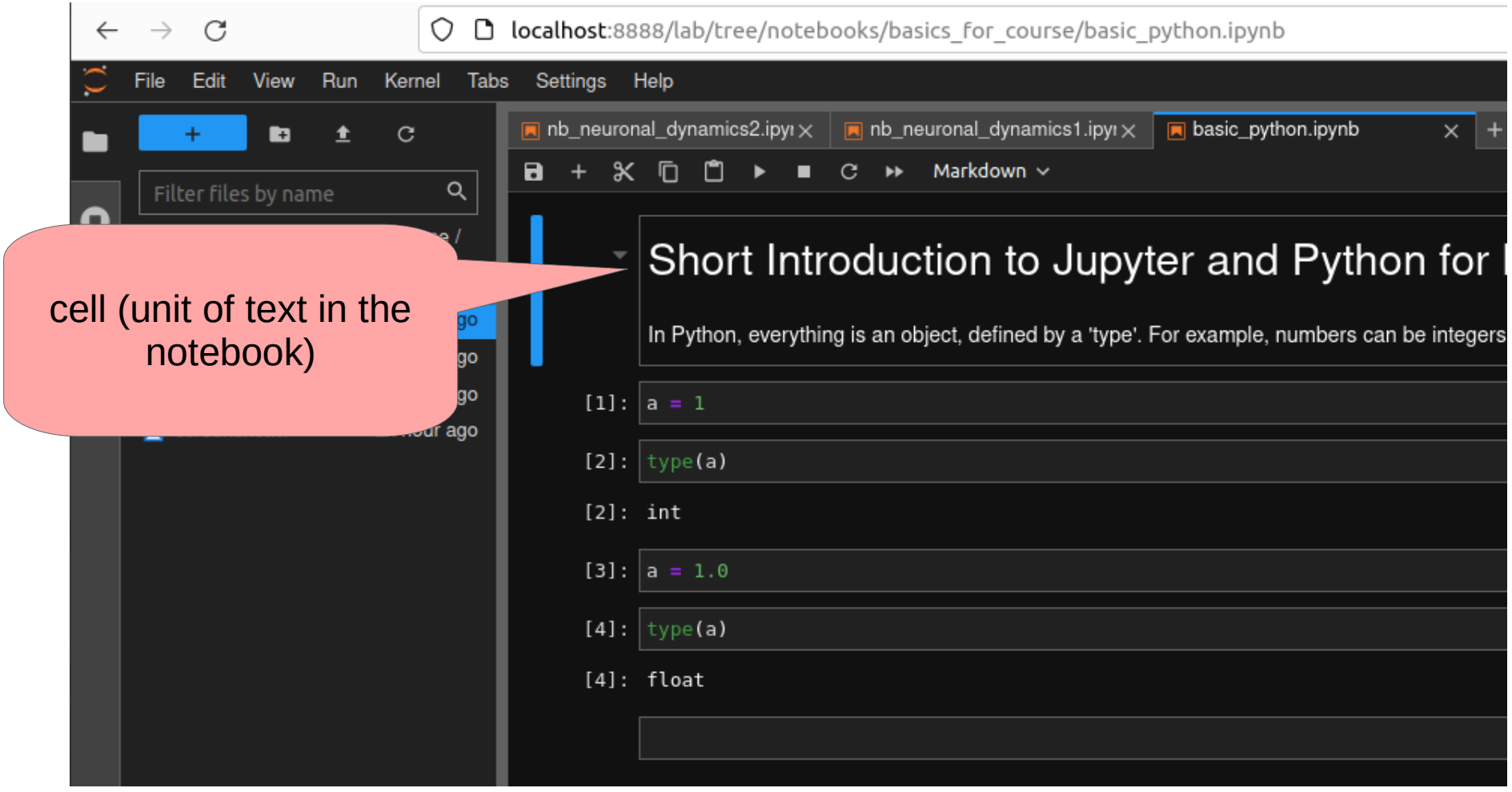
Cells in Jupyter



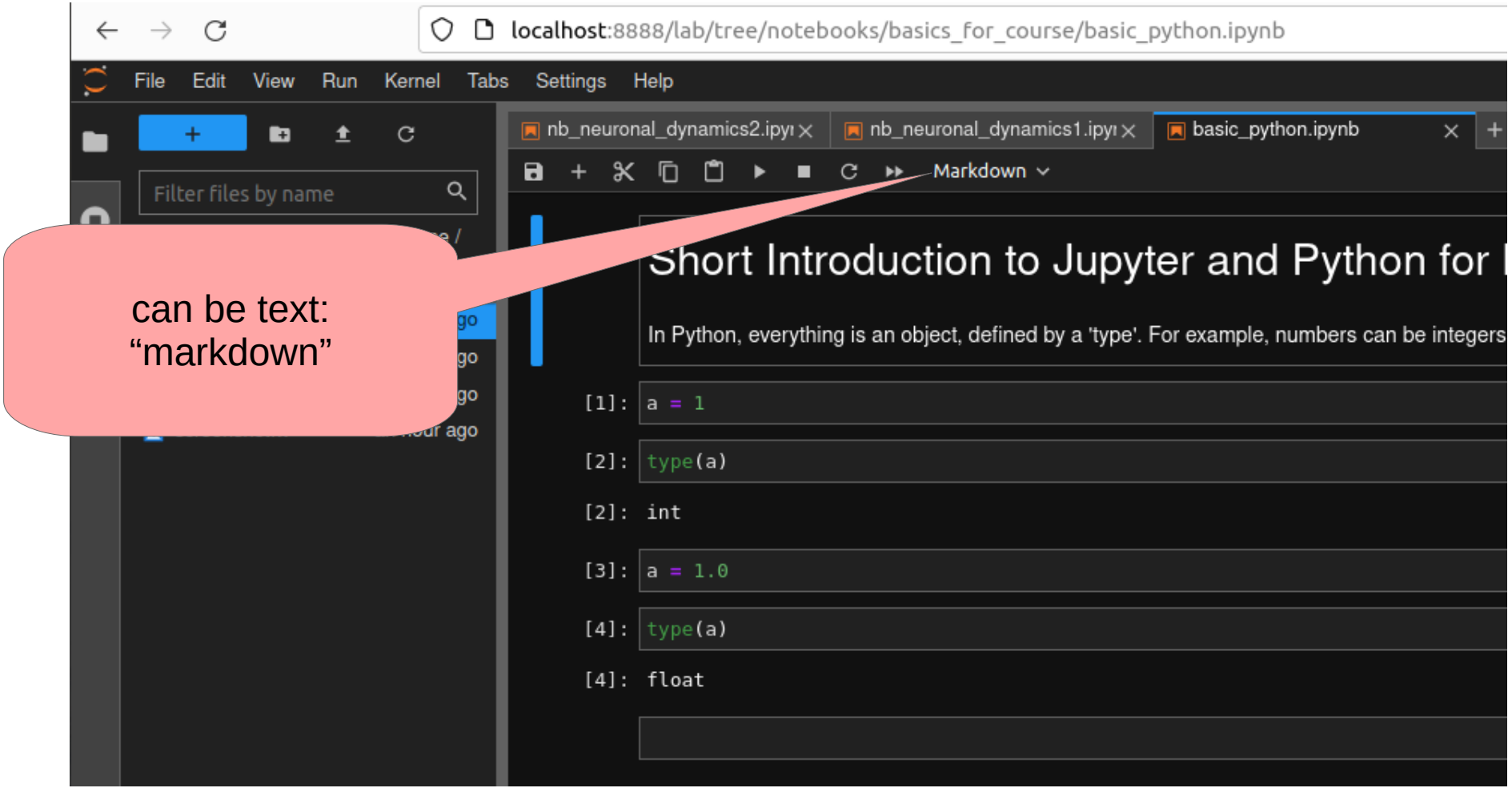
Cells in Jupyter



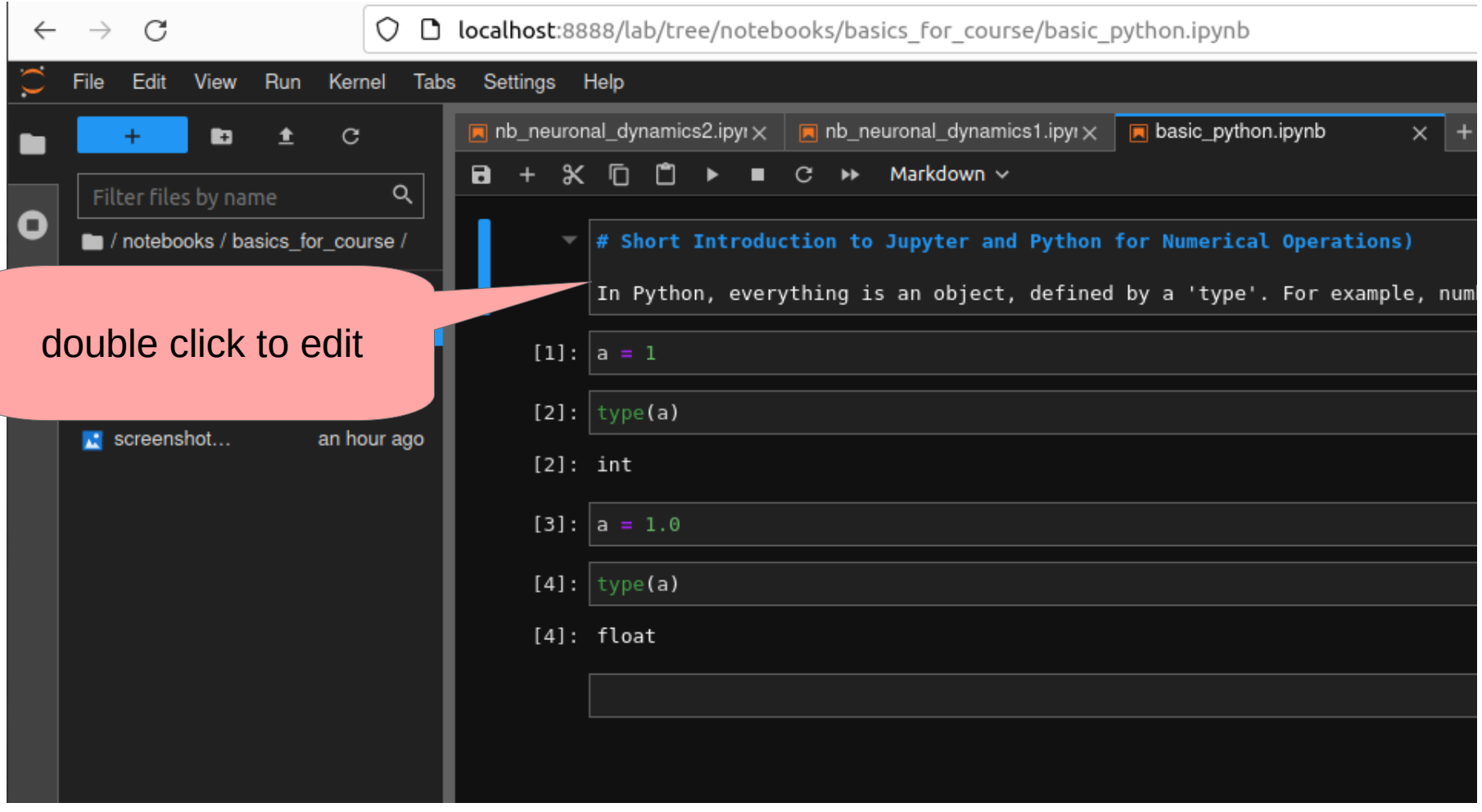
Cells in Jupyter



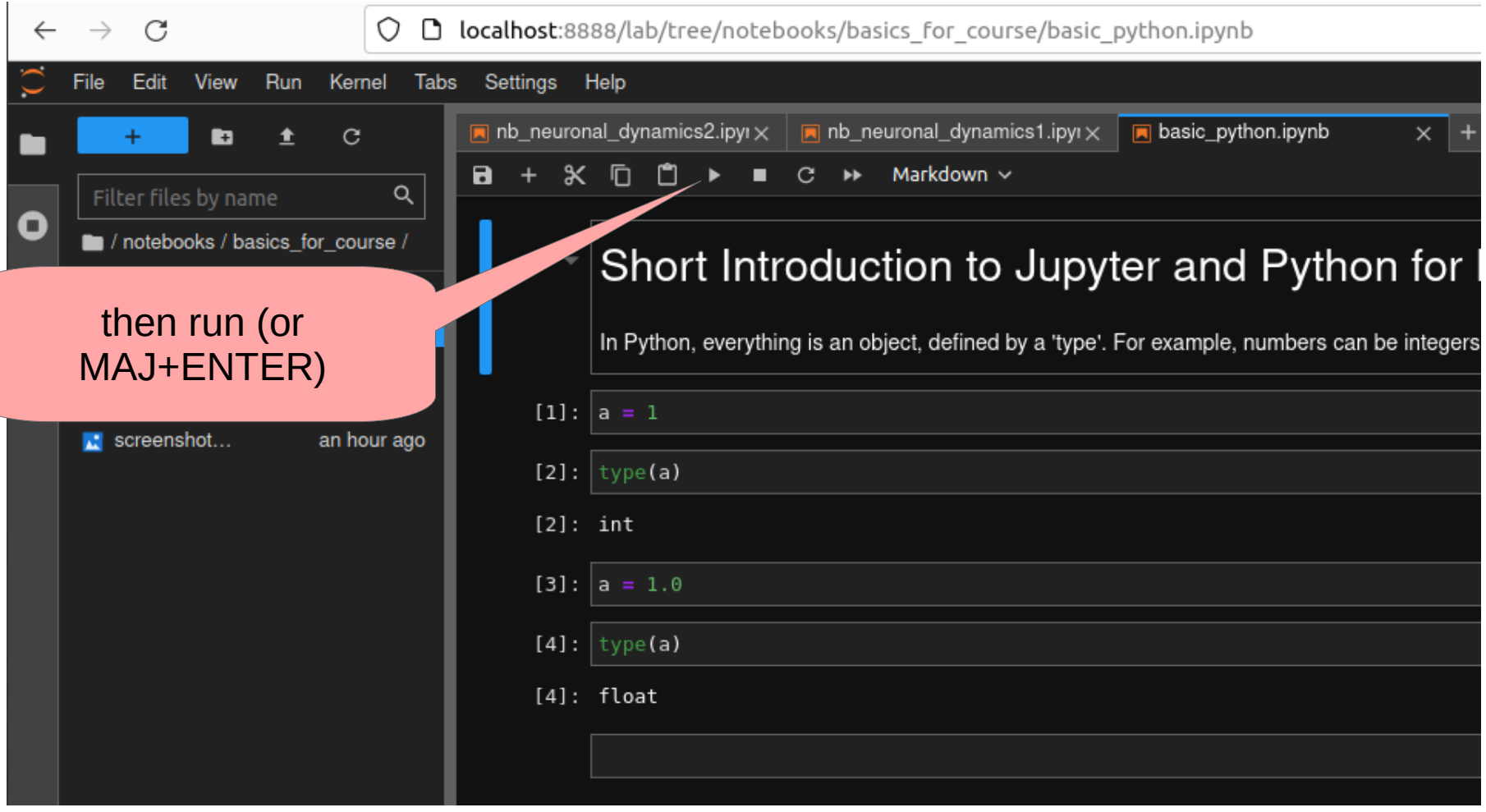
Cells in Jupyter



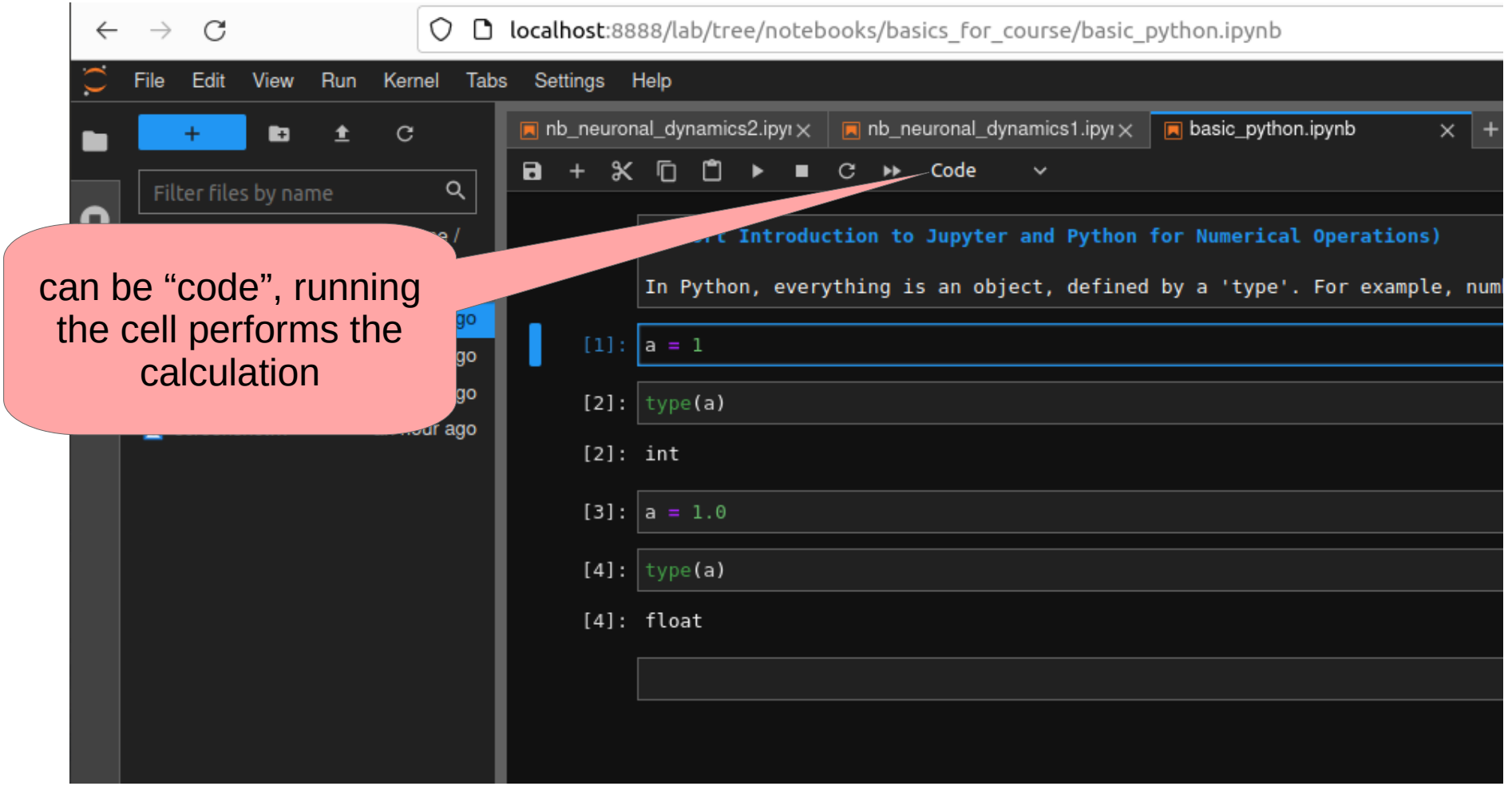
Cells in Jupyter



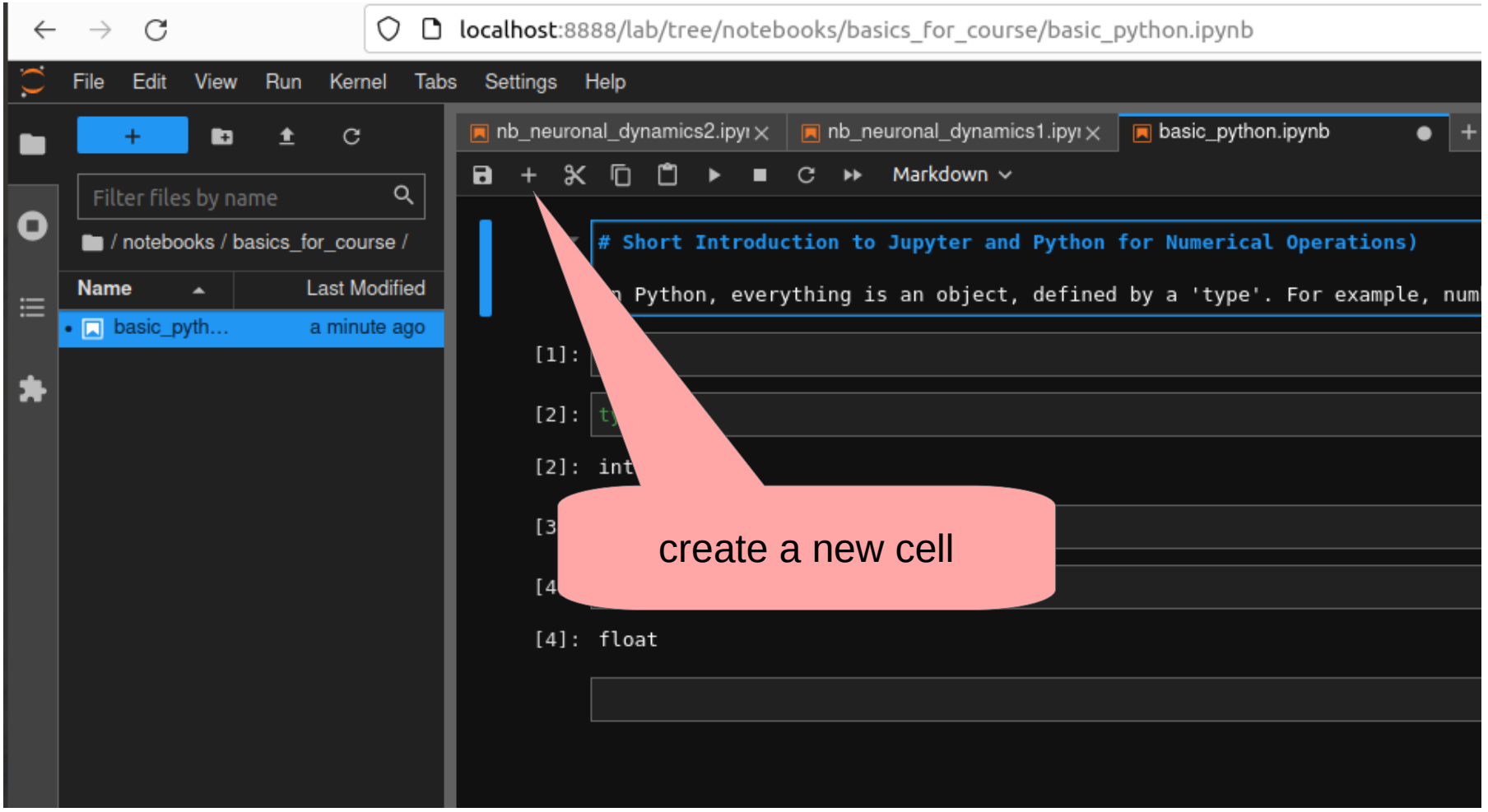
Cells in Jupyter



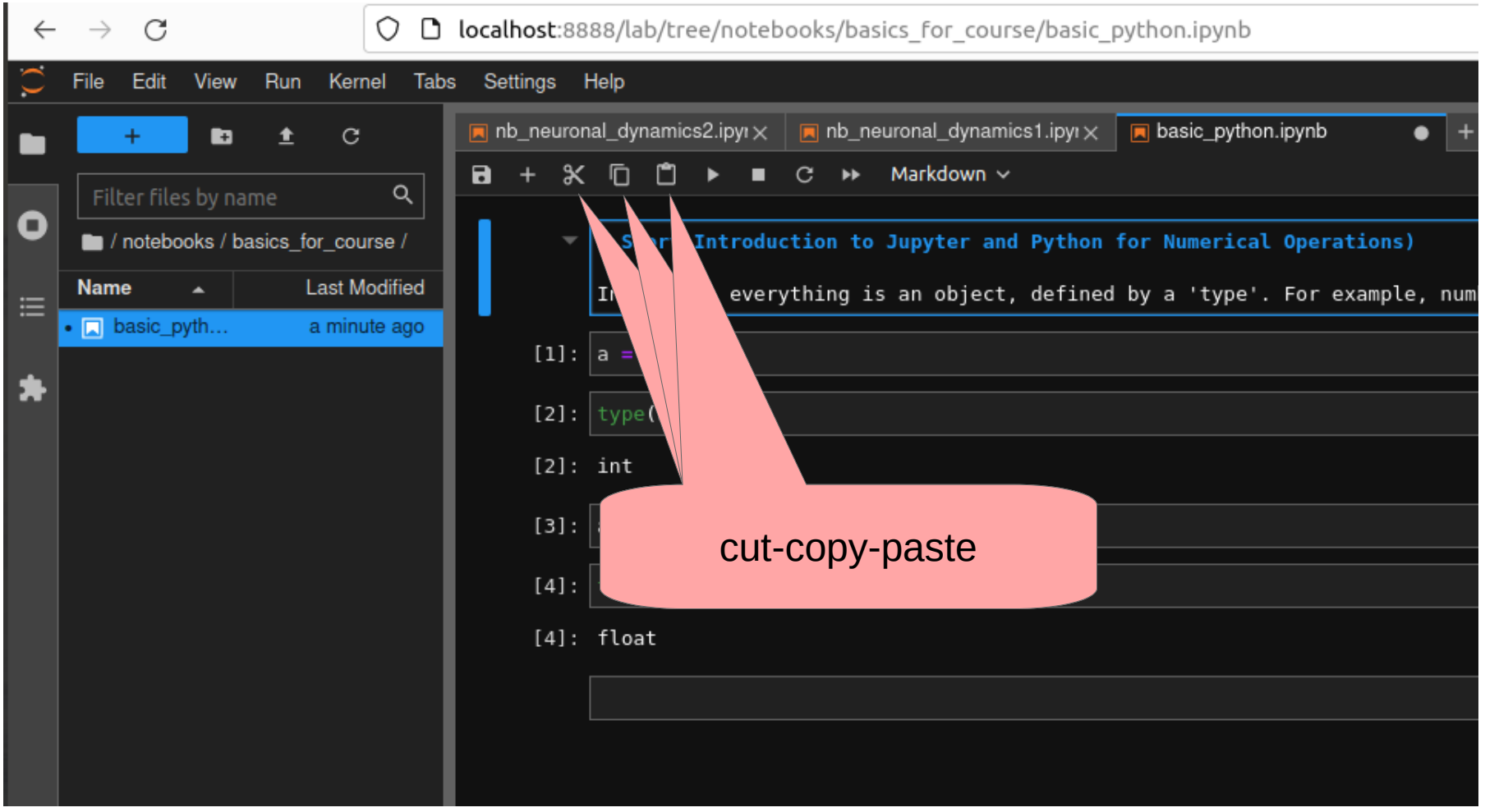
Cells in Jupyter



Cells in Jupyter



Cells in Jupyter



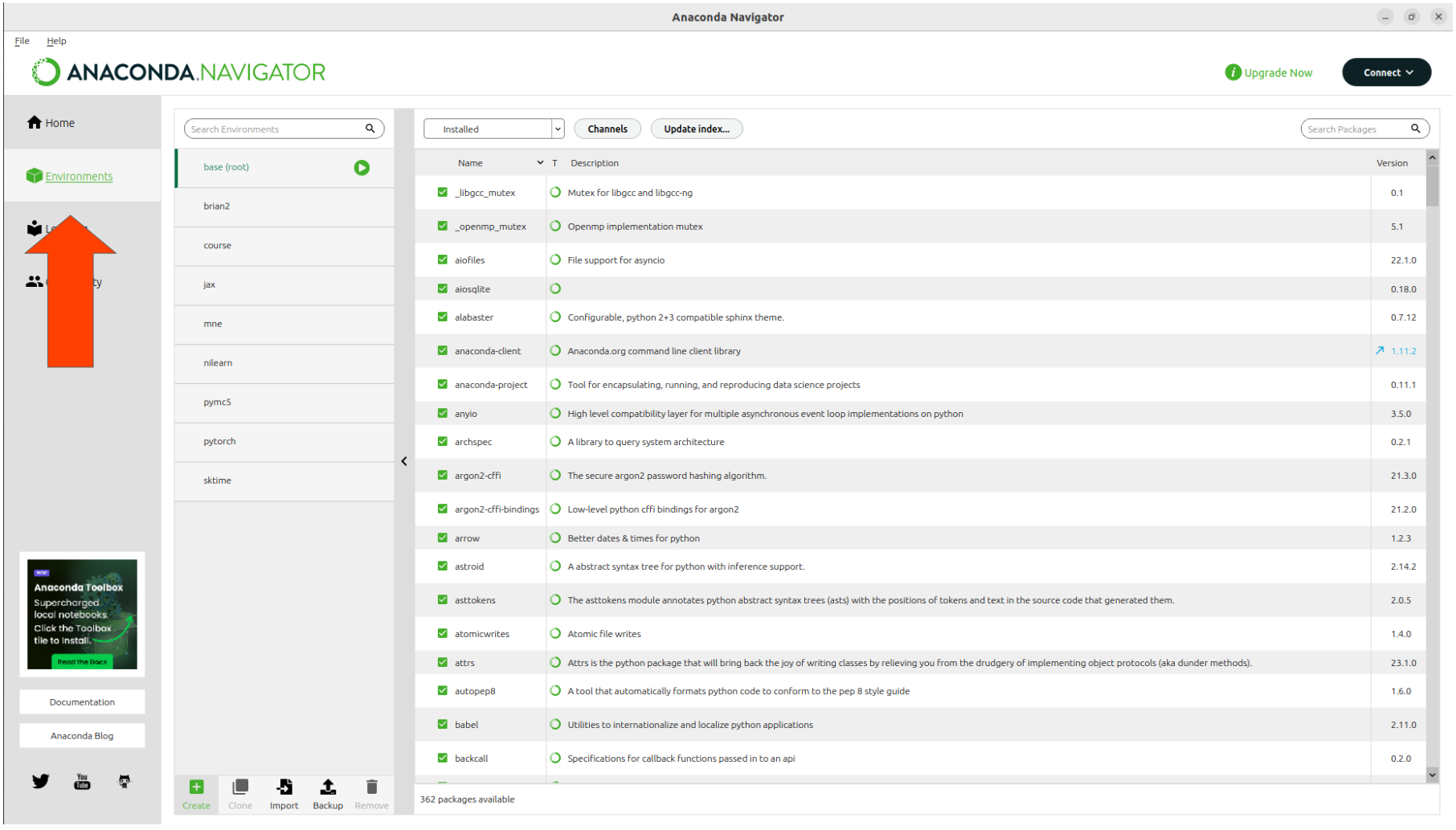
Useful Tips

- Shortcut MAJ+ENTER to run current cell
- Menu ‘Cells’: ‘Run All Cells’ to run the whole notebook
  - the memory is updated by the execution of each cell, so if you don’t run cells in order (or just go to a cell after opening the notebook), the variables in memory may not be what you think…
- Restart the kernel when something went wrong in the notebook (like no cell output although there should be...)

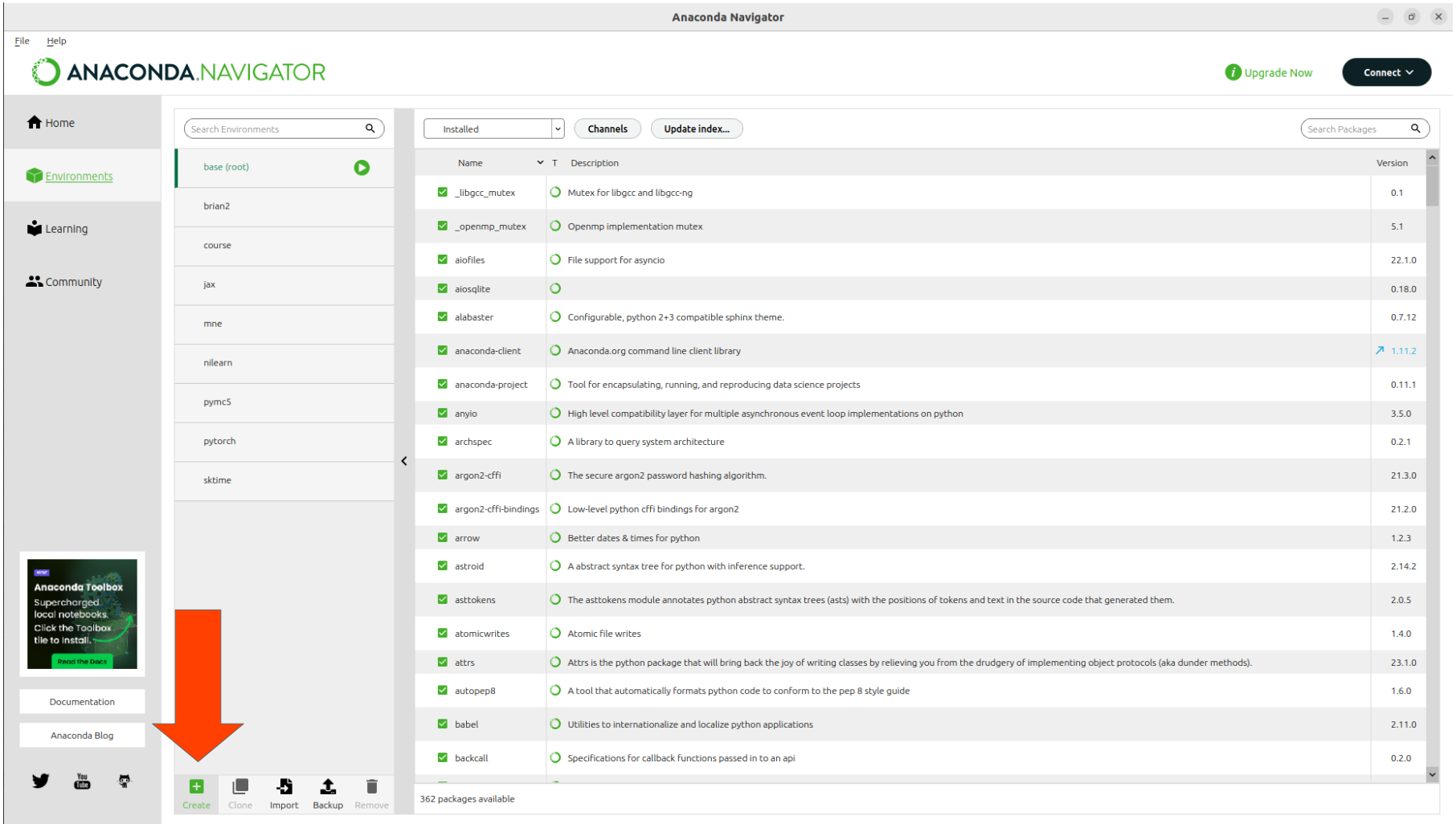
Create Environment with Selected Packages

- Back to anaconda-navigator: follow the following steps described in the following slides
- Go to ‘Environments’ tab and create a new environment
  - name it for example ‘env1’ and go with the latest version of Python as proposed
- Search packages that we are going to use and install them:
  - ‘numpy’ and ‘scipy’ for calculation
  - ‘matplotlib’ for visualization
  - ‘jupyter lab’

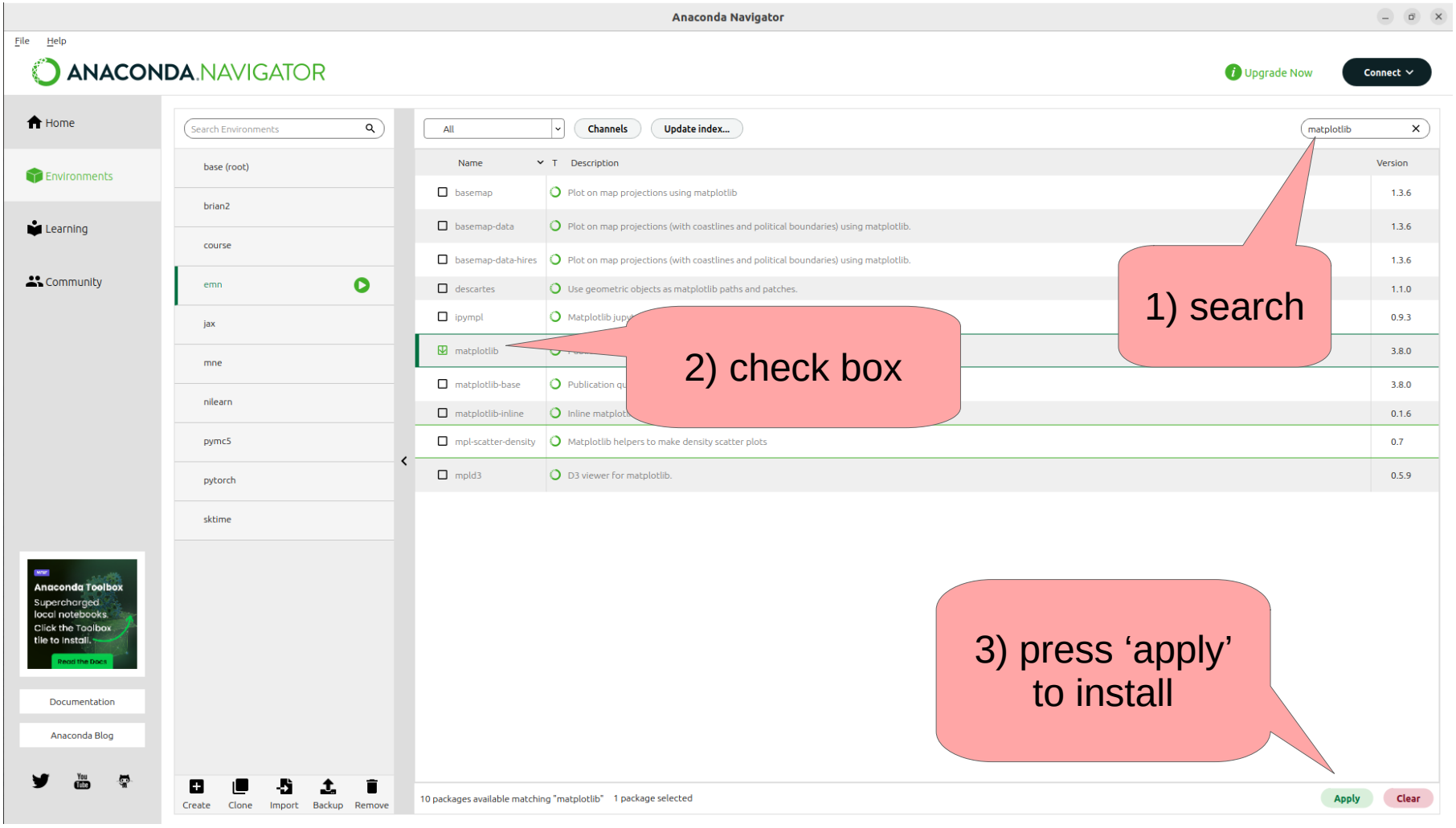
Create Environment and Select Packages Manually



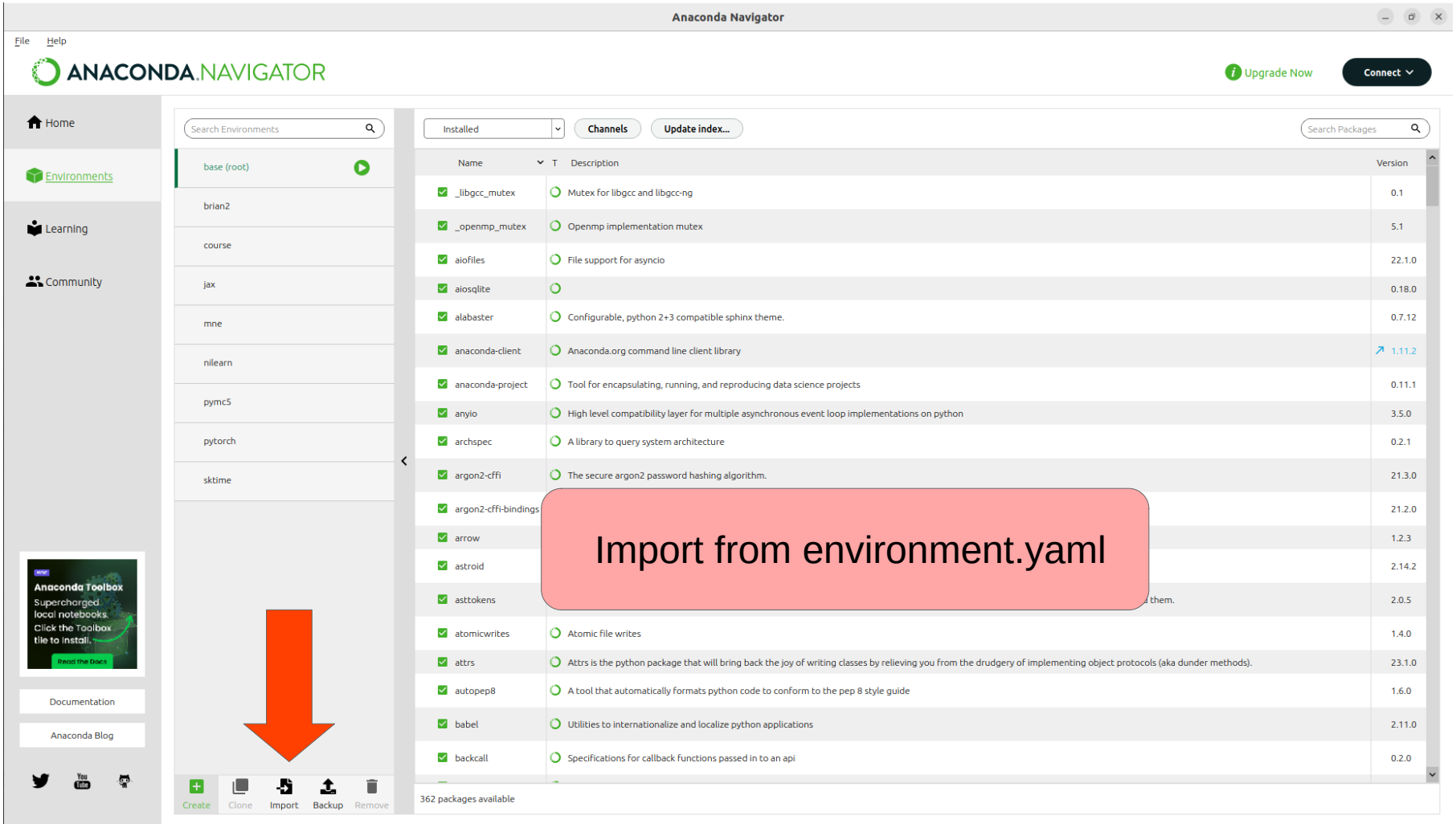
Create Environment and Select Packages Manually



Create Environment and Select Packages Manually



Create Environment Using YAML File



Now to Practice with Python Programming

- Download and open the notebook ‘basic_python.ipynb’ in folder ‘basics/python/’
- For further information about notebooks, check https://docs.jupyter.org/en/latest/
  - markdown syntax: https://fr.wikipedia.org/wiki/Markdown
- Numpy documentation: https://numpy.org/doc/stable/reference/
  - tutorial: https://numpy.org/devdocs/user/absolute_beginners.html
- Scipy documentation: https://docs.scipy.org/doc/scipy/reference/index.html
- Matplotlib tutorial: https://matplotlib.org/stable/users/explain/quick_start.html