



BB1000

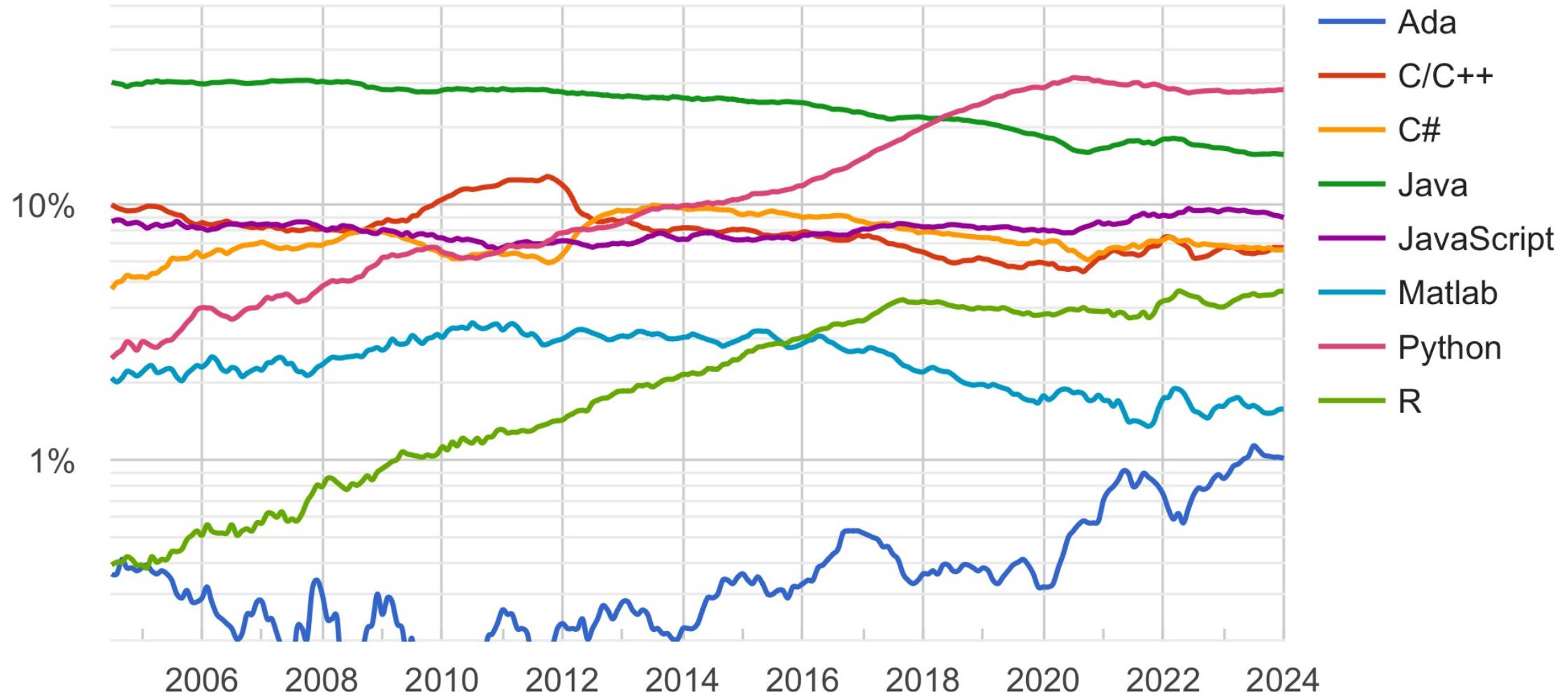
Python för bioteknologi

Patrick Norman

Division of Theoretical Chemistry and Biology

What programming languages are being used?

PYPL PopularitY of Programming Language





10 reasons to learn Python

<https://www.simplilearn.com/>

1. **Career opportunity:** Python language skills **provide job opportunities**.
2. **Data science:** Python is known for being robust, scalable and provides **extensible visualization and graphics** options.
3. **Machine learning:** Python is among the programming languages most preferred for Machine Learning because of its simple syntax and its support of several machine learning libraries.
4. **Web development:** Python provides a vast collection of frameworks to develop web applications.
5. **Scripting and automation:** Python language can be used for **writing scripts and automating workflows**. Yet, it is a full fledged language that **invites to well structured programming and code re-use**.
6. **Libraries and packages:** Python has a **range of libraries, packages, frameworks, and modules**.
7. **Testing frameworks:** Python supports several built-in testing frameworks that help in debugging and speeding up workflows
8. **Portable and extensible:** A Python code is mostly **portable**. It can be integrated with Java and C/C++ libraries.
9. **Active community:** Python has a massive community that can help with issues.
10. **Easy to use:** Python has a **simple syntax** and hence is easy to understand and learn.

Kernel integration

In DFT, the exchange-correlation (xc) kernel integr

$$E_{xc}[n(\mathbf{r})] = \int n(\mathbf{r}) \varepsilon_{xc}(n, \nabla n, \nabla^2 n) d^3\mathbf{r}$$

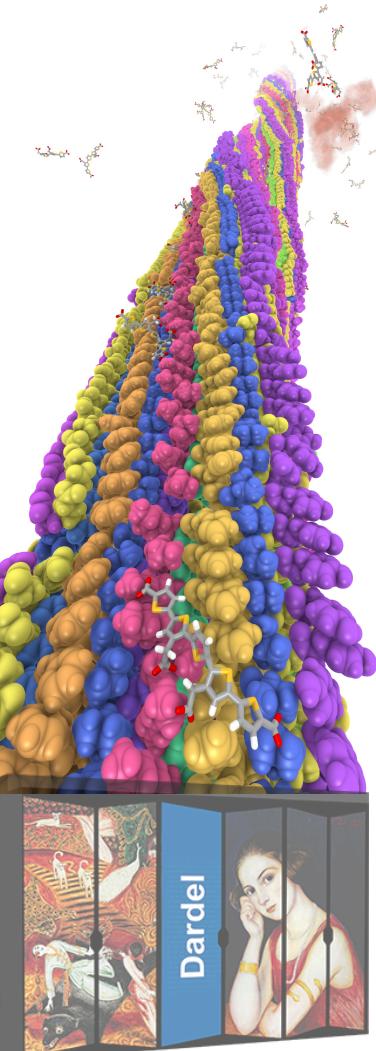
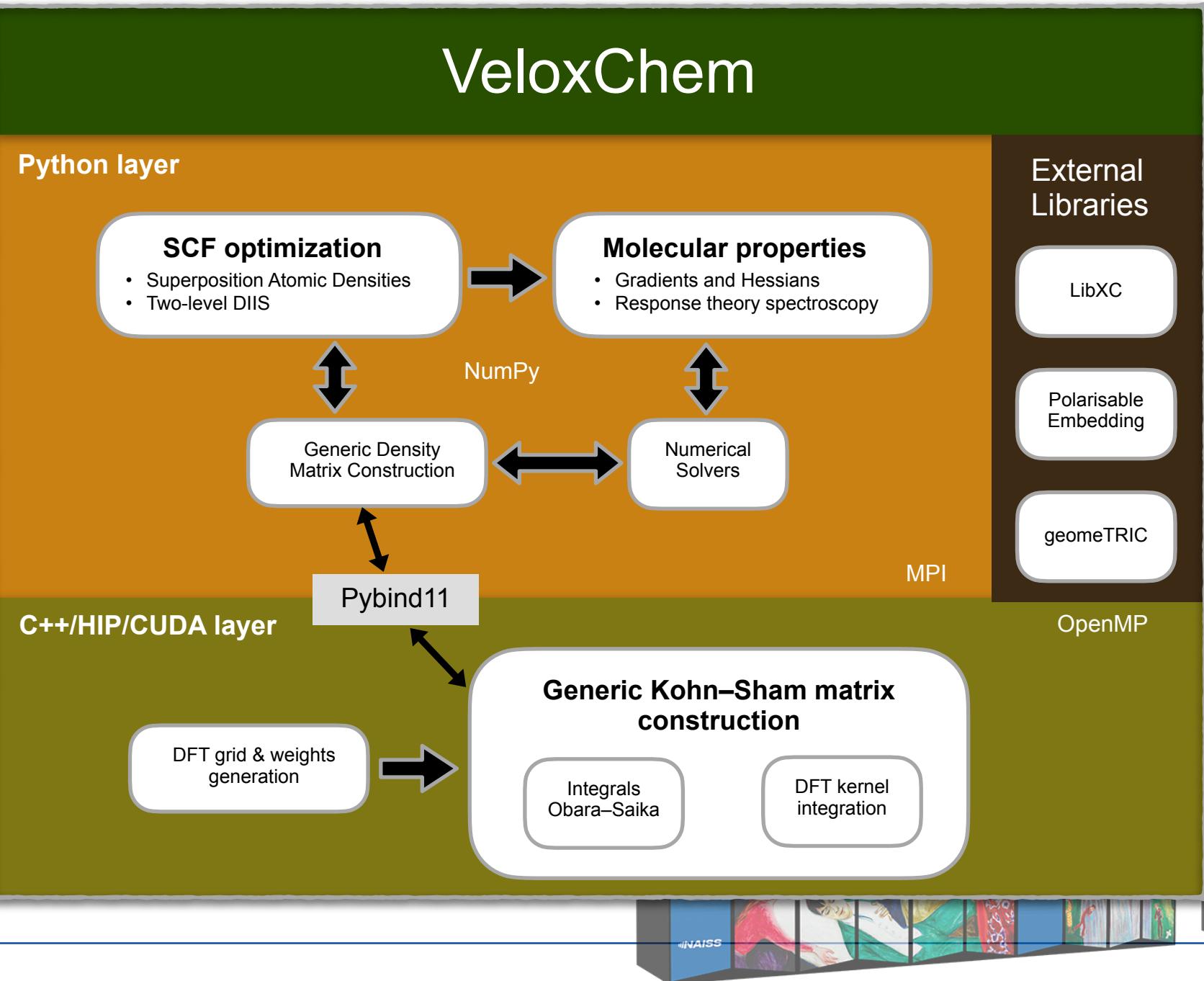
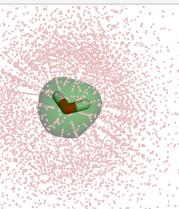
In VeloxChem, the set of grid points and the assoc using modified versions of the Log3-quadrature [T [SSF96]. In practical work, the number of grid poin

Integration of the density itself results in the numb this simple case to assess the accuracy in the num electron density of the water molecule in our exam

```
import matplotlib.pyplot as plt
import numpy as np
import veloxchem as vlx
```

```
# define the molecular system
h2o_xyz = """3
 0  0.000000000000  0.000000000000
H  0.000000000000  0.740848095288
H  0.000000000000 -0.740848095288
"""
molecule = vlx.Molecule.read_xyz_string(h2o_xyz)

# determine the method
scf_drv = vlx.ScfRestrictedDriver()
scf_drv.xcfun = "b3lyp"
```



Lecture	Topic	Teacher	Homework
1	Introduction and software installation	PN	
2	Computers, filesystems, terminals, environments	PN	
3	Jupyter notebooks – some basics	VK	OV
4	Datatypes, statements, conditions, iterations, NumPy	VK	OV
5	Debugging, stack overflow	JHA	OV
6	Pandas, file I/O, binary vs ascii	JHA	OV
7	Matplotlib, seaborn, vector vs bitmap	JHA	OV
8	Functions and classes, object oriented programming	ZR	OV
9	Programming practises, style and formatting, modules, import	ZR	OV
10	AI-tools for programming (chatGPT, github-copilot)	BvH	OV
11	Biotechnology-specific modules (PyCobra, Biopython, etc)	MA	
12	Outlook: Toward professional programming (Spyder, git, hybrid language, etc)	BvH	
Exercise	Topic	Teacher	
1	Differential equations	VK + BvH	
2	MD + visualization	MA	
3	Bioinformatics - FASTA-file	ML	
4	Dataanalysis HPA-data	MA + JHA	



From where do we get the needed software?



Our History

Anaconda was founded in 2012 by Peter Wang and Travis Oliphant out of the need to bring Python into business data analytics. Since that time, the use of Python has exploded, becoming the most popular programming language used today. Anaconda now has over 300 full-time employees worldwide and is proud to serve over 40 million users!



GUI or CLI?

Anaconda Navigator is a desktop application that is included with every installation of Anaconda Distribution. It is built on top of conda, the open-source package and environment manager, and allows you to manage your packages and environments from a graphical user interface (**GUI**). This is especially helpful when you're not comfortable with the command line.

A command line interface (or **CLI**) is a program on your computer that processes text commands to do various tasks. Conda is a CLI program, which means it can only be used via the command line. On Windows computers, Anaconda recommends that you use the Anaconda Prompt CLI to work with conda. MacOS and Linux users can use their built-in command line applications.

Should I use Anaconda or Miniconda?

Choose Anaconda if you:

- Are new to conda or Python.
- Like the convenience of having Python and over 1,500 scientific packages automatically installed at once.
- Have the time and disk space—a few minutes and 3 GB.
- Do not want to individually install each of the packages you want to use.
- Wish to use a set of packages curated and vetted for interoperability and usability.

Choose Miniconda if you:

- Do not mind installing each of the packages you want to use individually.
- Do not have time or disk space to install over 1,500 packages at once.
- Want fast access to Python and the conda commands and you wish to sort out the other programs later.



What do you get with Miniconda?

Miniconda is a free minimal installer for conda. It is a small bootstrap version of Anaconda that includes only **conda**, **Python**, the packages they both depend on, and a small number of other useful packages (like pip, zlib, and a few others).

If you need more packages, use the **conda install** command to install from thousands of packages available by default in Anaconda's public repo, or from other channels.

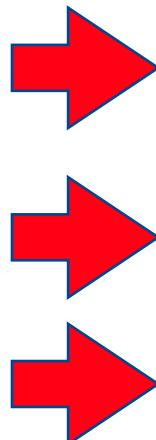
The conda command searches a set of channels. By default, packages are automatically downloaded and updated from the default channel that may require a paid license. The **conda-forge** channel is free for all to use.



Download and install the latest Miniconda

<https://docs.conda.io/projects/miniconda/en/latest/>

Latest - Conda 23.11.0 Python 3.11.5 released December 20, 2023



Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	c9b32faa9262828702334b16bcb5b53556e630d54e5127f5c36c7ba7ed43179a
macOS	Miniconda3 macOS Intel x86 64-bit bash	2b7f9e46308c28c26dd83abad3e72121ef63916eaf17b63723b5a1f728dc3032
	Miniconda3 macOS Intel x86 64-bit pkg	74ab9e8c3e9b3c2fc7c44d710ed9bad19085d951d819c1284a46eeb0bdfe2578
	Miniconda3 macOS Apple M1 64-bit bash	5694c382e6056d62ed874f22692224c4f53bca22e8135b6f069111e081be07aa
	Miniconda3 macOS Apple M1 64-bit pkg	912c0b58e800f26e08d515526a8d3455755e83963b40e78597176540ea2401ca
Linux	Miniconda3 Linux 64-bit	c9ae82568e9665b1105117b4b1e499607d2a920f0aea6f94410e417a0eff1b9c
	Miniconda3 Linux-aarch64 64-bit	decd447fb99dbd0fc5004481ec9bf8c04f9ba28b35a9292afe49ecef400237f
	Miniconda3 Linux-s390x 64-bit	53a9e9eb97cd6e318f4f184add069436e1a46124cf864bf2d7bd67043e50e471



Open a command-line interface (a.k.a. terminal)

A command-line interface (CLI) is a text-based user interface used to

- run programs
- manage computer files
- interact with the computer

On Windows, we will use
[Anaconda Powershell](#).

On macOS, we will use [Terminal](#).

A screenshot of a macOS Terminal window. The window title bar says "panor — zsh — 80x24". The window content shows the following text:

```
Last login: Wed Jan 17 10:29:53 on ttys003
(base) ~ %
```

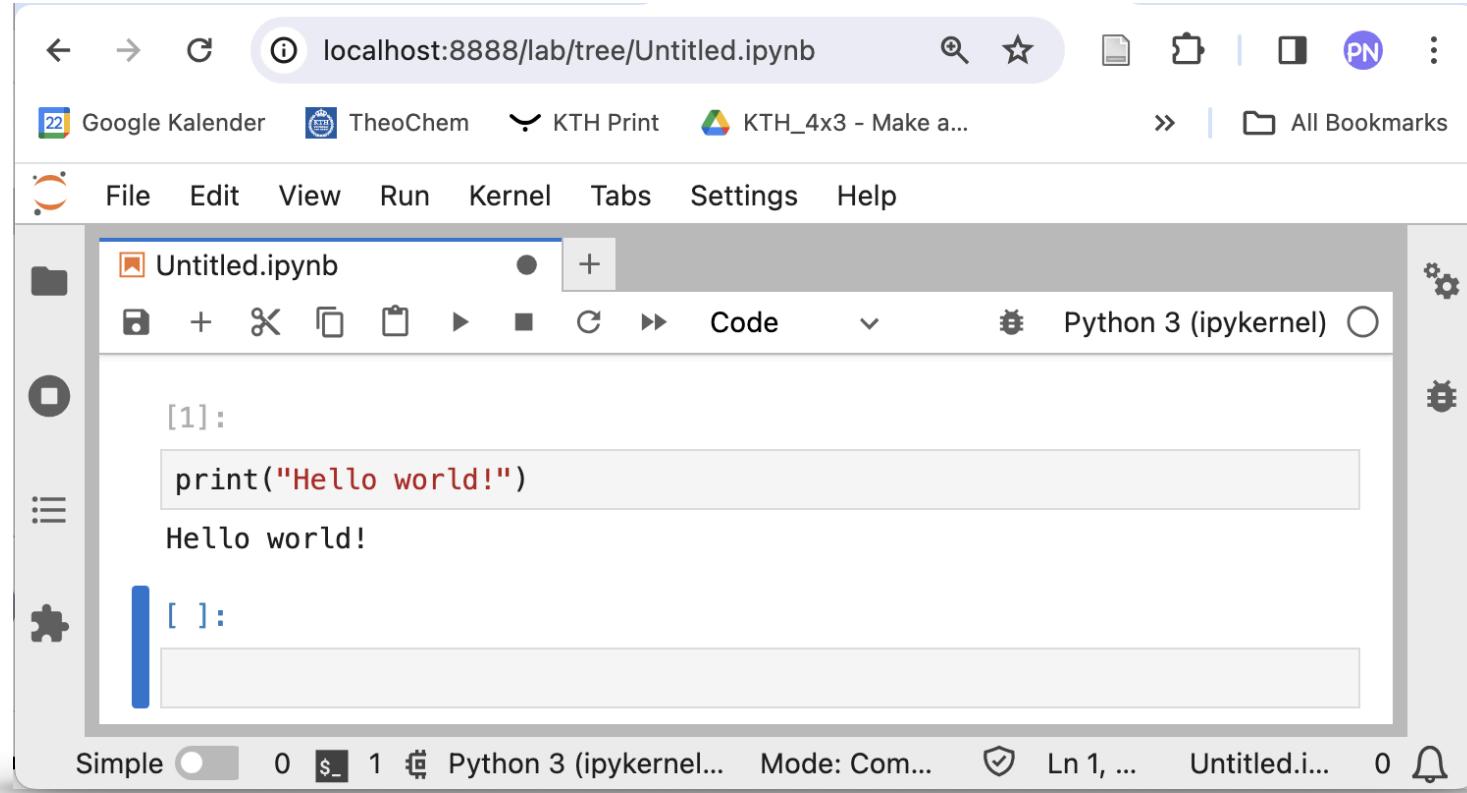
The window has the standard OS X look with red, yellow, and green close buttons.

Conda environments

- With conda, you can create, export, list, remove, and update environments that have different versions of Python and/or packages installed in them.
- Switching or moving between environments is called activating the environment.
- **Do not install any additional packages in the base environment.**
- Keep **base** up-to-date with the command:
`% conda update conda`
- To create an environment:
`% conda create -n myenv`
- To remove an environment:
`% conda remove -n myenv --all`

Testing things out

```
% conda create -n test  
% conda activate test  
% conda env list  
% conda install jupyterlab -c conda-forge  
% jupyter-lab
```



The screenshot shows a Jupyter Notebook interface running in a web browser. The address bar indicates the URL is `localhost:8888/lab/tree/Untitled.ipynb`. The notebook has one cell, [1], containing the Python code `print("Hello world!")`. The output of this cell is "Hello world!". Below the notebook, the status bar shows "Simple" mode, "Python 3 (ipykernel...)", "Mode: Com...", and "Untitled.i...".



End of introduction