Enough Python to Do Something Useful With

Chris Woodley

Session 1 – Using Conda Environments and Basic Python

What is this course? And Why?

- Code literacy is likely an important future skill. Only limited opportunities in undergraduate Chemistry to gain skills
- Demand for people with coding experience is likely to increase
- Coding is useful for chemists – applications in graph plotting, cheminformatics, machine learning, automating tedious tasks
- This course is designed to be largely **self taught** with a focus on **practical experience** through examples
- Many people find it difficult to learn a new skill **unless they have to**

AIM: In this course, you'll learn Python essentials to leverage its use for chemistry projects and feel comfortable listing Python on your CV

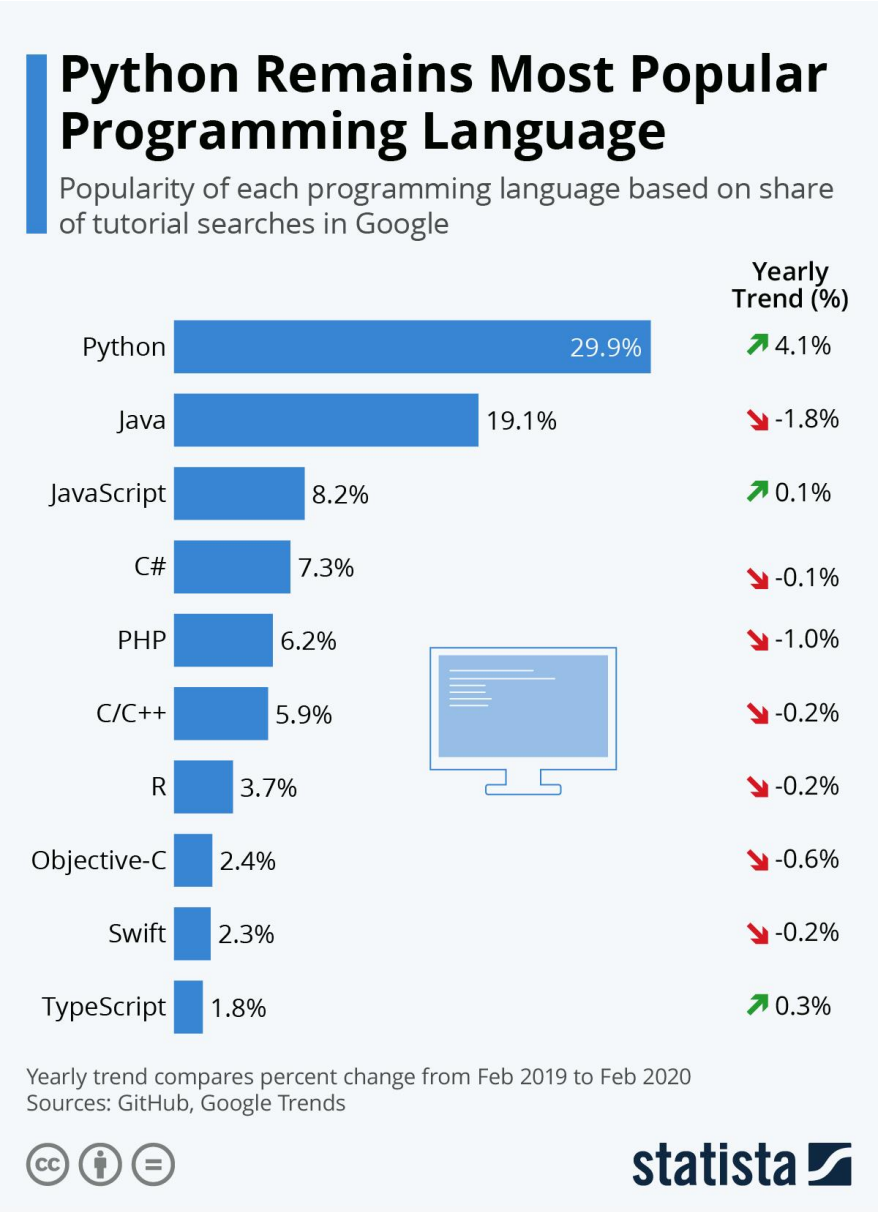
Why Python?

Advantages

- Automatically deals with data types and memory management
- Basically English
- Rapid development time
- Ability to import modules expands Python's utility for scientific computation

Disadvantages

- Slower to run than other programming languages
- Wealth of modules can lead to issues with inconsistency and incompatibilities



Course Outline

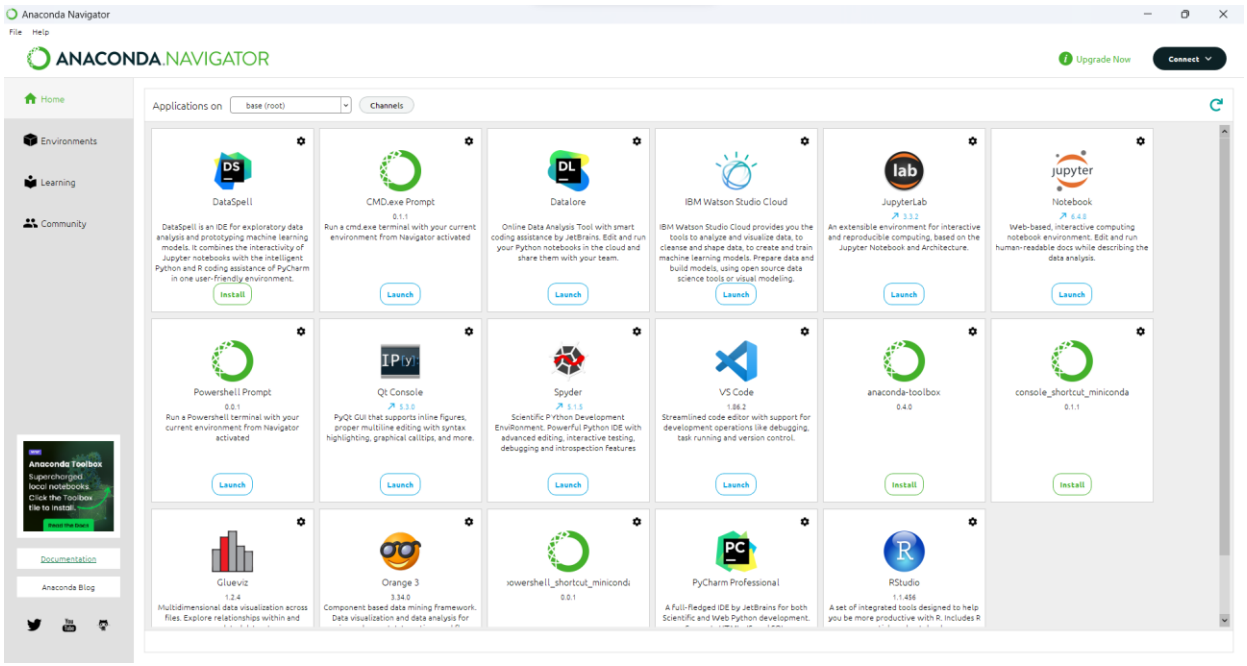
1. Introduction to Python, Anaconda and Jupyter Notebooks
2. Handling data in Python: Numpy, Pandas and Matplotlib
3. Chemistry Specific Modules: RDKit
4. Git and using other peoples code

Scope of this Session

1. Installation of anaconda
2. Installation and setup of an integrated developer environment
3. Introduction to Python in Jupyter Notebooks
4. Self Guided Python skills session
   - Basic python skills examples
   - Basic skills exercises

1. Anaconda

- For different projects different modules (and different versions of Python might be required)
- Anaconda provides functionality for this – implements virtual environments
- Most importable packages pre-installed
- Also provides a GUI for managing environments (command line usually more useful)



1. Anaconda

- Install Anaconda:
  - https://www.anaconda.com/download
- Create an environment
  - Open “Anaconda Powershell Prompt”
  - “conda create –n my_env”
- Activate the environment
  - “conda activate my_env”
- Install modules
  - “conda install ipykernel”

Anaconda Powershell Prompt

```
(base) PS C:\Users\chris> conda create -n my_env
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\chris\anaconda3\envs\my_e
nv



Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate my_env
#
# To deactivate an active environment, use
#
#     $ conda deactivate

Retrieving notices: ...working... done
(base) PS C:\Users\chris> conda activate my_env
(my_env) PS C:\Users\chris>
```

2. Integrated Development Environment (IDE)

- An Integrated Development Environment (IDE) is a software which provides tools and features to support software development
- Typical features include code-editing, debugging tools, version control and file management bundled in a single user-interface
- Many IDEs are available and available in Anaconda Navigator – not a one size fits all approach.
  - Visual Studio code – I like this one
  - Spider
  - PyCharmm

3. Jupyter Notebooks

- Python scripts (.py files) are standalone files designed to be ran sequentially and are suited to automation and reproducibility
- Jupyter Notebooks offer an interactive environment combining code, visualisations and formatted text
  - Good for exploratory data analysis
  - Allows running codes in “chunks”
  - Useful for producing graphics
  - Shareable format – almost like a web page
  - Not as good for reproducibility
- The majority of this course will use jupyter notebooks

In [1]:
```
%matplotlib inline
import pandas as pd
import numpy as np
import plotly
from IPython.display import display, Markdown as md
```

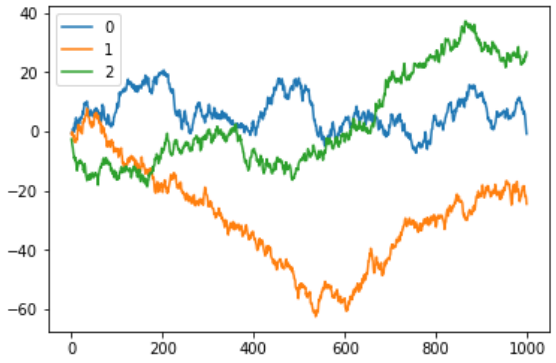
In [2]:
```
title = "My Shiny Report"
x = 1000
y = 3
```

In [3]:
```
display(md("# Just look at this graph from {}".format(title)))
```

Just look at this graph from My Shiny Report

In [4]:
```
df = pd.DataFrame(np.random.randn(x, y))
df.cumsum().plot()
```

Out[4]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f127adda278>
```



4. Session Content

- Course content is provided at https://github.com/cmwoodley/Python_UoL_Medchem
- Each session will be accompanied by two jupyter notebooks:
  - Examples – a notebook containing Python examples for reference
  - Exercises – partially completed Python snippets
  - Exercise solutions will be made available before the next session
- This session will cover:

- Variables - What are variables? - Naming conventions	- Data Types - Integers - Floats - Strings - Lists - Tuples - Dictionaries	- Functions - Mathematical functions - Iterating through lists - Searching in strings

NB – not an exhaustive list of python syntax and functionality but **enough to be useful**

5. Troubleshooting/ Further Reading

- This session covers mostly in-built functionality of Python
- The w3 schools python reference is a good resource for looking up syntax: https://www.w3schools.com/python/python_reference.asp
- ChatGPT can find the solution to the exercises:
  - **Try not to use this –** manually troubleshooting is a good way to learn python