basic python demo

Knowing your tools

Notebook environments: pros & cons

Numpy: math in python

Pandas: organize & manipulate data

Matplotlib (& Seaborn): visualize & analyze data

Notebook Pros:

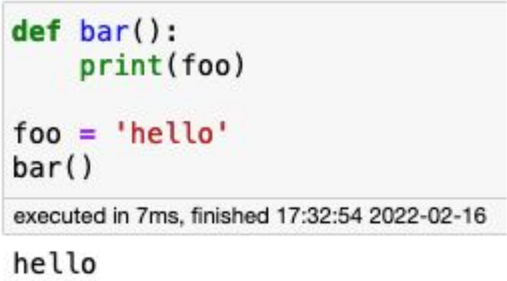
Great for beginners, development and communication

- Immediate results, interactive cells
- Inline plots

Notebook Cons:

Hard to stay reproducible

- Hidden state
  - Can run cells “out of order” and reach unreproducible states
- Reliance on global variables (can cause nasty bugs)
- Scaling issues, difficult to:
  - Run code with different parameters
  - Parallelize tasks over a cluster
  - Version control
- Not a great code editor



```python
def bar():
    print(foo)

foo = 'hello'
bar()
```

executed in 7ms, finished 17:32:54 2022-02-16

```
hello
```

Over-reliance on notebooks can build bad habits

They are ideally used:

- As a “living document” for communication of code/results
- To prototype and develop new code
  - Keep notebooks small and restart the kernel often
  - Refactor and then version control code

Once you feel comfortable, consider:

- Learning to use an editor, eg vscode, pycharm, (Neo)vim, emacs, etc
- Becoming familiar with the command line

Until then, there is nothing “wrong” with working inside notebooks

For those interested:

“I dont like notebooks” slideshow

Working efficiently with JupyterLab Notebooks covers refactoring

Refs for getting started at the command line (automation, reproducibility, git, etc)

- https://missing.csail.mit.edu/
- https://merely-useful.tech/py-rse/index.html
- https://datascienceatthecommandline.com/

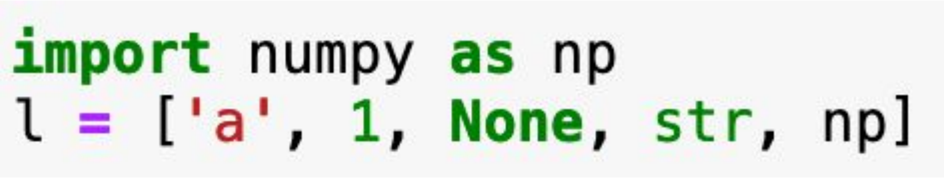
notebook tips and tricks demo

command line arguments & configs

NumPy: **Num**erical **Py**thon

Python is dynamically typed:

Why numpy?



```
import numpy as np
l = ['a', 1, None, str, np]
```

Nothing wrong here …

This makes `for` loops very slow! Why? What would C do?

numpy gives us access to highly optimized procedures

- implemented in a low-level language (eg C)
- **vectorization**: do the for loop in C instead of python, see example notebook
- when writing new functions, find out if it already exists in numpy/pandas/etc

numpy demo

pandas & plotting demos