

A Flock of Birds...



A Heard of Cows...



A Huge Mess of Pythons...

*A quick look at why nothing ever
works on your machine*



A Huge Mess of Pythons...

Pycharm

IDE

Python2

System Python

Notebook

Python3.4

numpy
v1.18.4

Python3.8

pip

Anaconda

Jupyter kernel

numpy
v1.22.4

conda

matplotlib

Spyder



A Huge Mess of Pythons...

Pycharm

IDE

System Python

Python2

Notebook

Python3.4

numpy
v1.18.4

Python3.8

pip

Anaconda

Jupyter kernel

numpy
v1.22.4

conda

matplotlib

Spyder



Python Installations

A Huge Mess of Pythons...

Pycharm

IDE

Python2

System Python

Python3.4

numpy
v1.18.4

Python3.8

pip

Anaconda

Jupyter kernel

numpy
v1.22.4

conda

matplotlib

Spyder



Python Installations

package managers

A Huge Mess of Pythons...



Pycharm

IDE

System Python

Python2

Notebook

Python3.4

numpy
v1.18.4

Python3.8

pip

Anaconda

Jupyter kernel

numpy
v1.22.4

conda

matplotlib

Spyder

Python Installations

package managers

packages

A Huge Mess of Pythons...



Pycharm

IDE

System Python

Python2

Notebook

Python3.4

numpy
v1.18.4

Python3.8

pip

Anaconda

Jupyter kernel

numpy
v1.22.4

conda

matplotlib

Spyder

Python Installations

package managers

packages

editors

A Huge Mess of Pythons...

Python Installations	package managers	packages	editors
System Python	conda	numpy v1.18.4	IDE
Python2	pip	numpy v1.22.4	Pycharm
Python3.4		matplotlib	Spyder
Python3.8			Notebook
Anaconda			Jupyter kernel

A Huge Mess of Pythons...



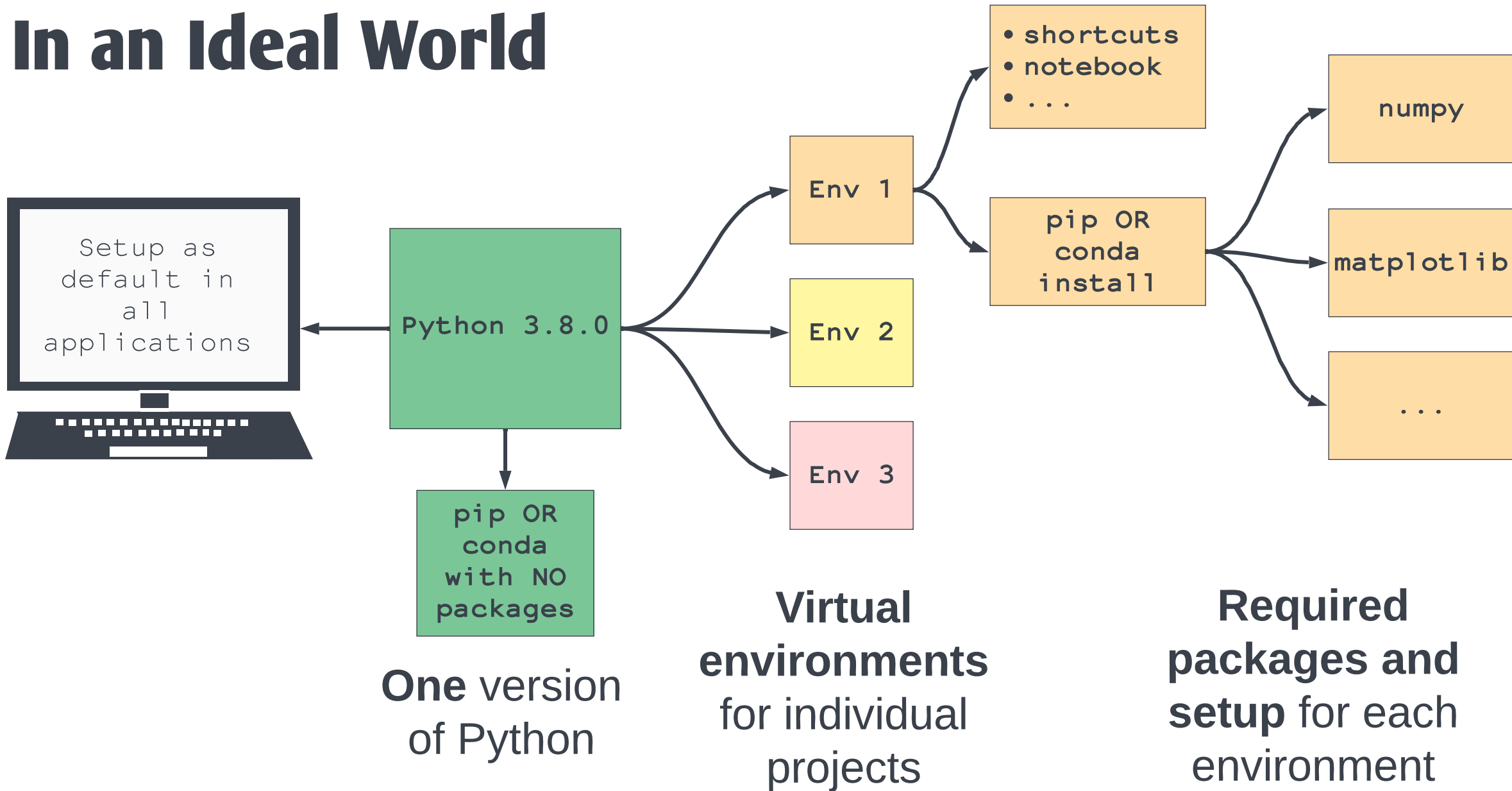
- Python is not like R or Matlab; your computer may actually be running Python to do it's own thing sometimes (e.g. MacOS System Python)
- You may have (inadvertently) installed **multiple versions of Python** and Python related tools.
- Having a mixed bunch of Pythons makes for hard to solve, **highly individual problems**

A Huge Mess of Pythons...



- Python is not like R or Matlab; your computer may actually be running Python to do it's own thing sometimes (e.g. MacOS System Python)
- You may have (inadvertently) installed **multiple versions of Python** and Python related tools.
- Having a mixed bunch of Pythons makes for hard to solve, **highly individual problems**
- **The only way to fix it is to wipe your entire hard drive, burn your computer, and start from scratch**
- **well, maybe not...**

In an Ideal World



Python installations

- Ideally you have **one Python**, that you have chosen
- Installing tools will sometimes install its own version of Python
- It's also fine to have multiple Pythons, if you know and keep track
- **Anaconda** is a Software bundle that makes the install (slightly) easier
 - It includes **Python**, a bunch of preinstalled packages, and their own package manager **conda**

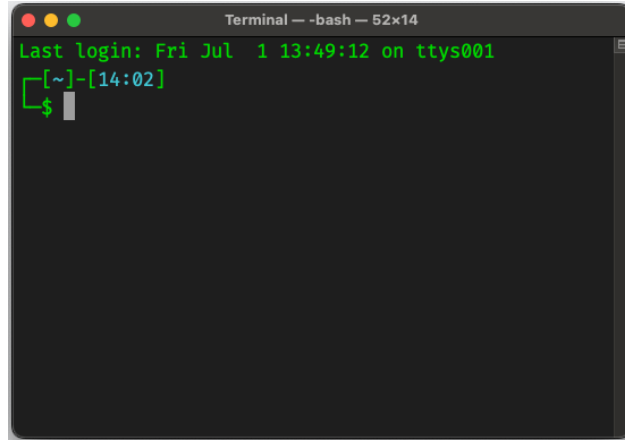


Package Managers



- Typically **pip** and **conda**
- **Don't mix and match using pip and conda.** It will work. But each package manager doesn't know what the other has done and you may end up with a huge mess.
 - **Except:** sometimes you have to mix and match! If things are not available in conda you have to use pip. But then only use pip for individual problematic cases.

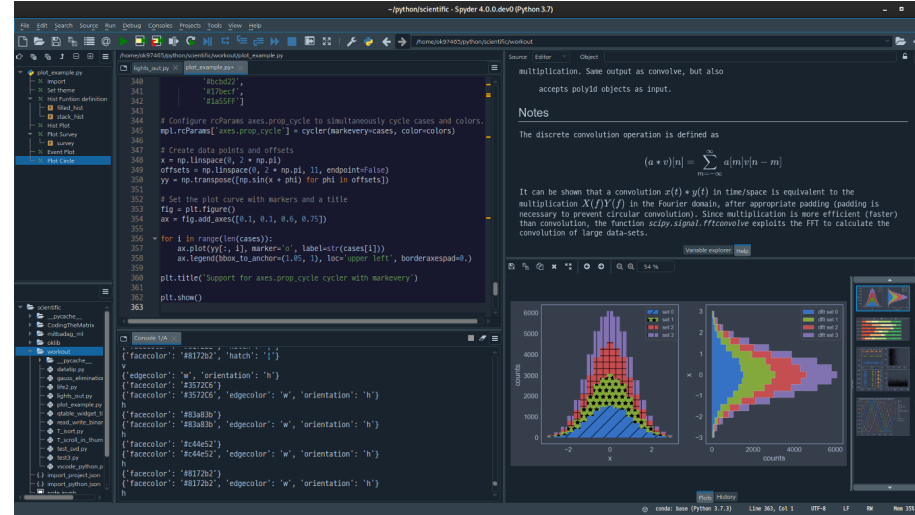
But how do I WORK with it?



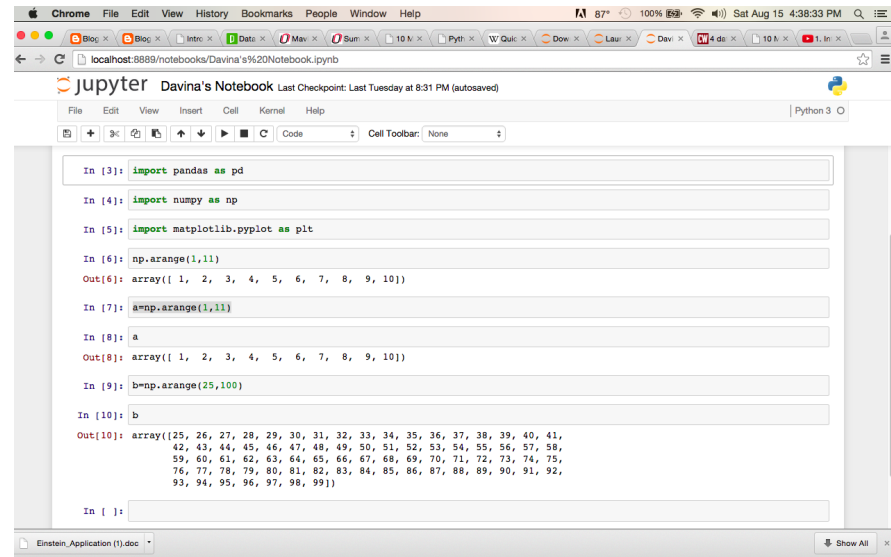
Back to
the roots:
Terminal

```
1  ***
2  SAW
3  ***
4  import numpy as np
5  # from numba import jit
6  # import matplotlib.pyplot as plt
7  # import time
8
9  np.seterr(under='warn', over='warn', divide='warn')
10
11
12  class saw_model():
13      def __init__(self, parameters, seed=None):
14          if seed is None:
15              self.seed = np.random.randint(10000)
16          else:
17              self.seed = seed
18          self.RS = np.random.RandomState(self.seed)
19
20          self._L = None
21          self.initialize = 2000 # number of steps to take for initializ
22
23          # parameters
24          self._ri = None
25          self._rj = None
26          self.eta = None # transition probs
27          self.delay = None
28          self.TSIZE = None # width of ellipse
29          self._VMAX = None
30          self._vrad = None
```

Basic
code
editor



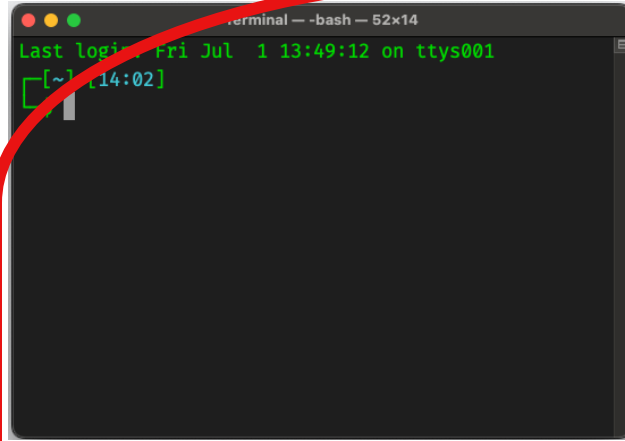
IDE



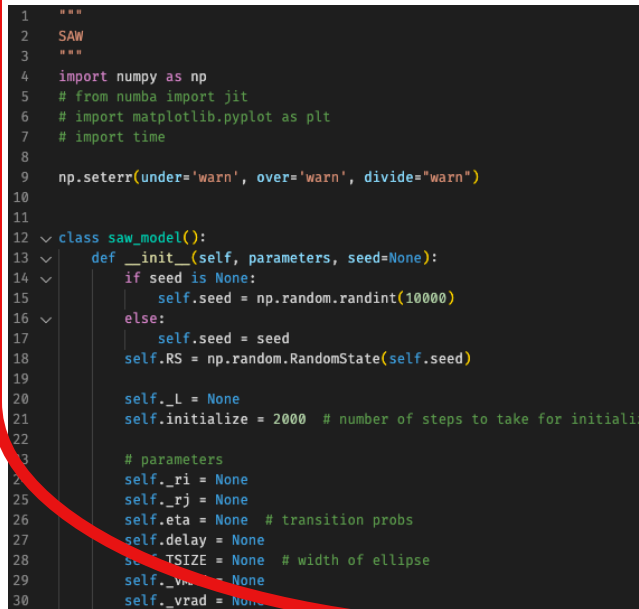
Notebook

But how do I WORK with it?

Back to
the roots:
Terminal

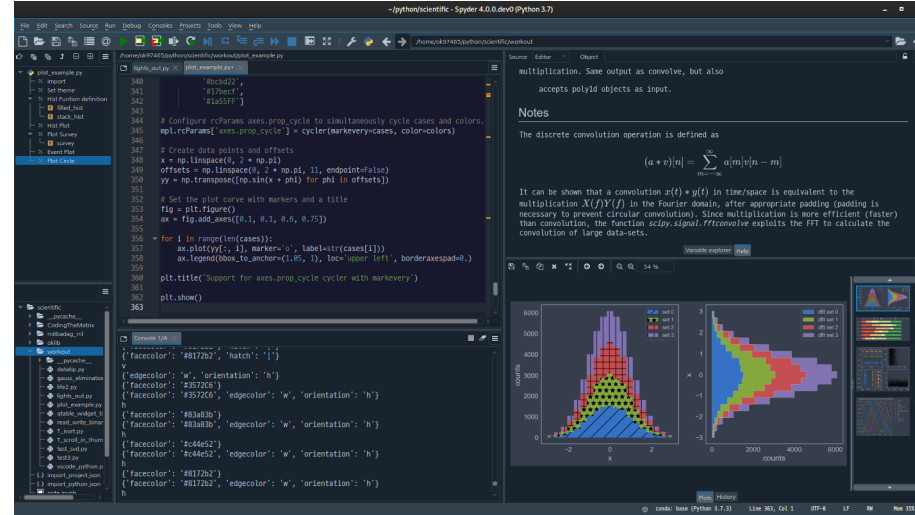


```
terminal --bash-- 52x14
Last login: Fri Jul 1 13:49:12 on ttys001
[~] 14:02
```

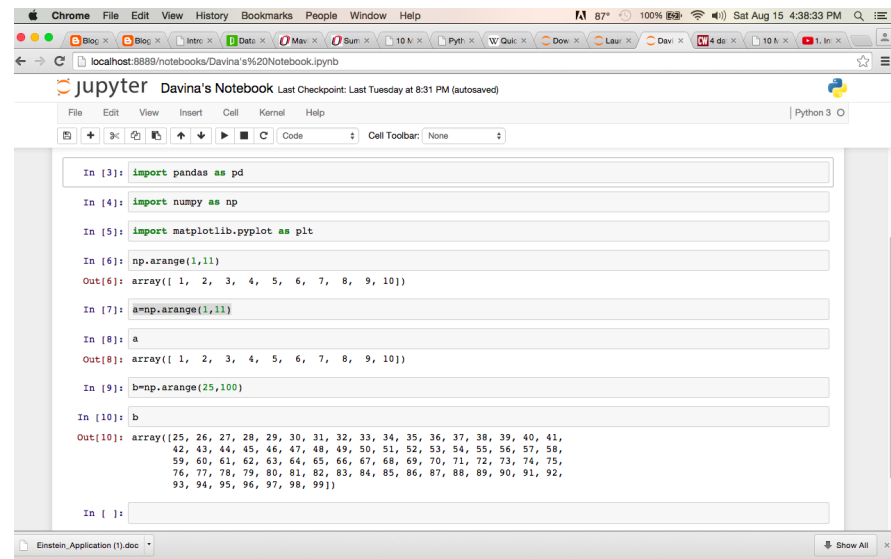


```
1  """
2  SAW
3  """
4  import numpy as np
5  # from numba import jit
6  # import matplotlib.pyplot as plt
7  # import time
8
9  np.seterr(under='warn', over='warn', divide='warn')
10
11
12 class saw_model():
13     def __init__(self, parameters, seed=None):
14         if seed is None:
15             self.seed = np.random.randint(10000)
16         else:
17             self.seed = seed
18         self.RS = np.random.RandomState(self.seed)
19
20         self._L = None
21         self.initialize = 2000 # number of steps to take for initializ
22
23         # parameters
24         self._ri = None
25         self._rj = None
26         self.eta = None # transition probs
27         self.delay = None
28         self._TSIZE = None # width of ellipse
29         self._vmax = None
30         self._vrad = None
```

Basic
code
editor

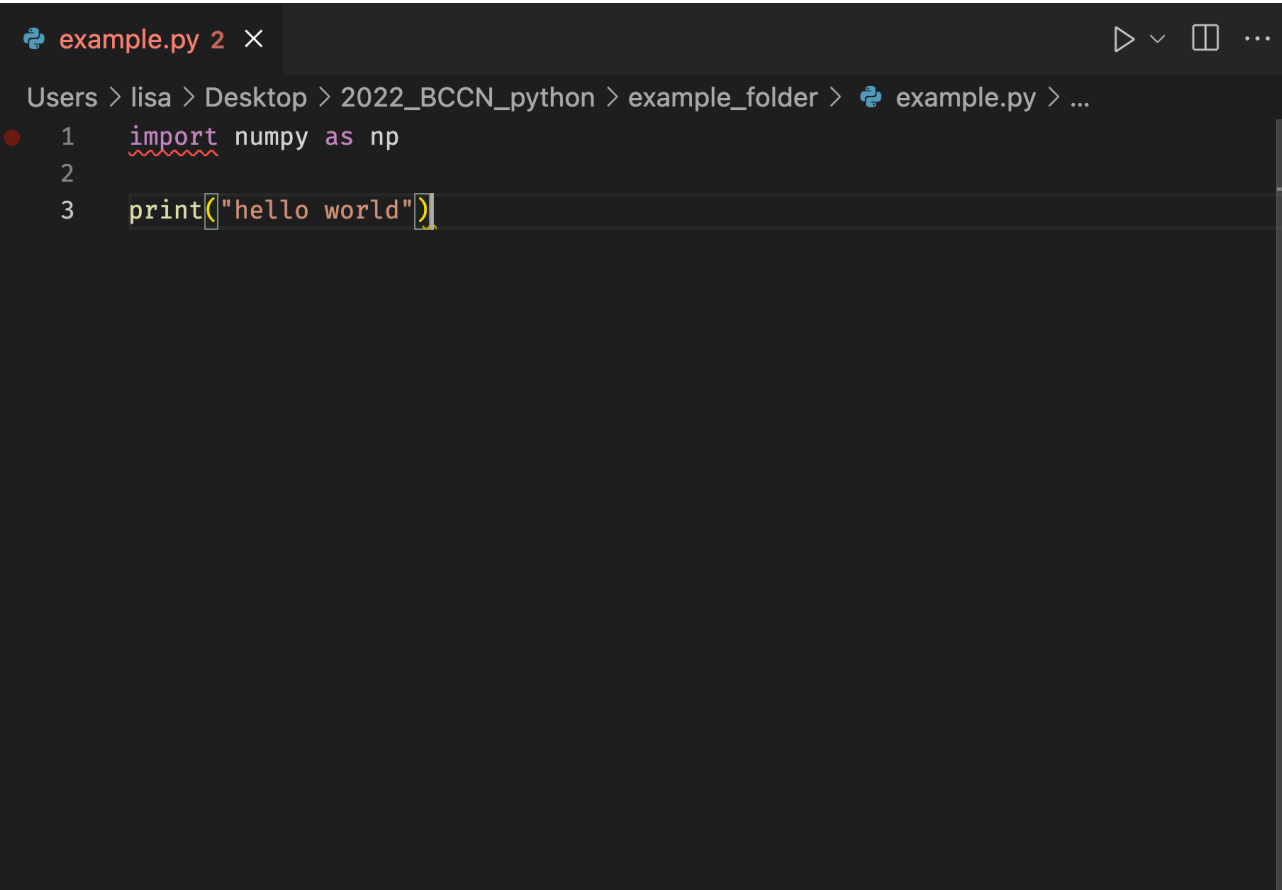


IDE



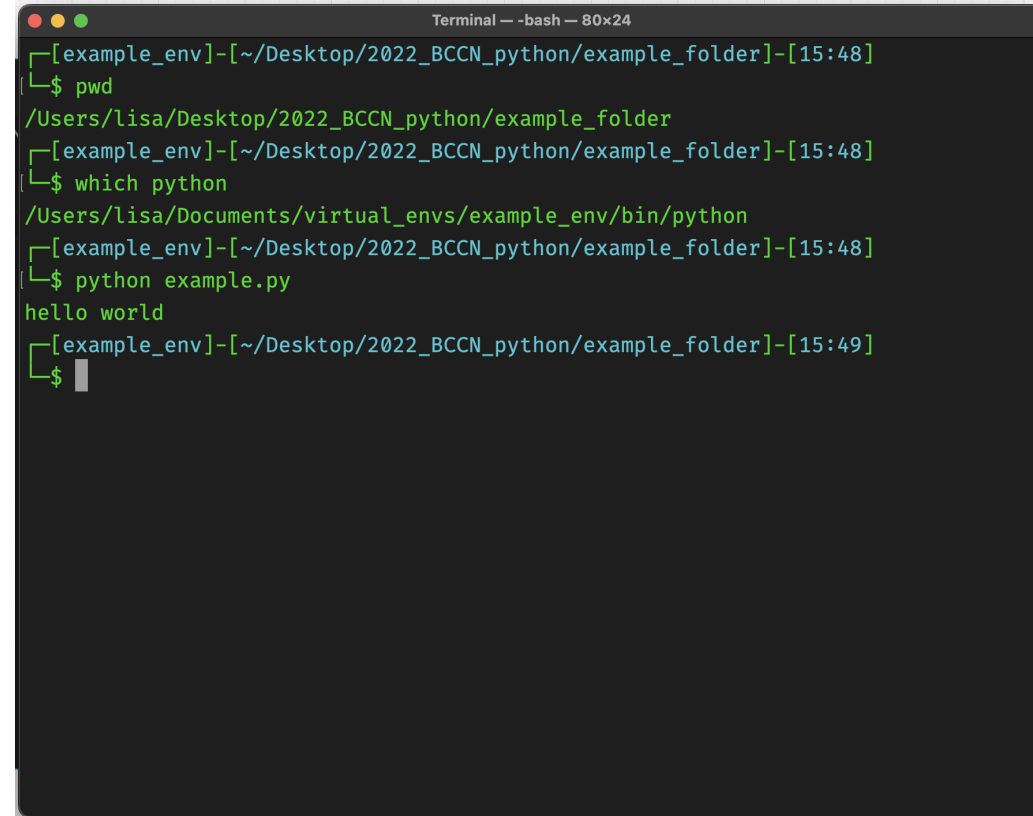
Notebook

Terminal



```
example.py 2 x
Users > lisa > Desktop > 2022_BCCN_python > example_folder > example.py > ...
1  import numpy as np
2
3  print("hello world")
```

Editor



```
Terminal — -bash — 80x24
[example_env]-[~/Desktop/2022_BCCN_python/example_folder]-[15:48]
$ pwd
/Users/lisa/Desktop/2022_BCCN_python/example_folder
[example_env]-[~/Desktop/2022_BCCN_python/example_folder]-[15:48]
$ which python
/Users/lisa/Documents/virtual_envs/example_env/bin/python
[example_env]-[~/Desktop/2022_BCCN_python/example_folder]-[15:48]
$ python example.py
hello world
[example_env]-[~/Desktop/2022_BCCN_python/example_folder]-[15:49]
$
```

Terminal

Terminal

- Let's say you're running python from the Terminal
- You can find out what program is called when you type "python"

```
Lisas-MBP-4:~ root# which python  
/usr/local/bin/python
```

- You can find out what program is called when you type "pip"
- Those should be in the same directory

```
Lisas-MBP-4:~ root# which pip  
/usr/local/bin/pip
```

Terminal

- If you are inside an environment instead it will look like this

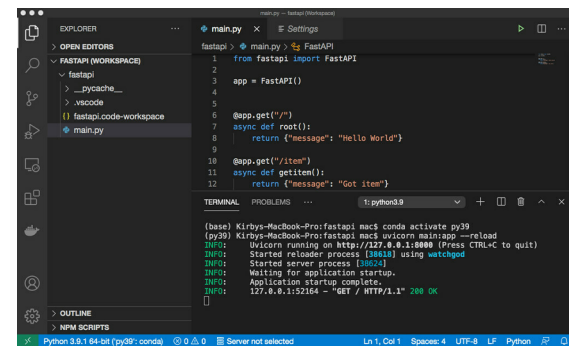
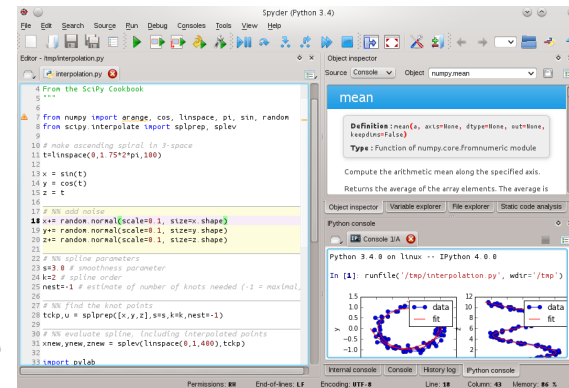
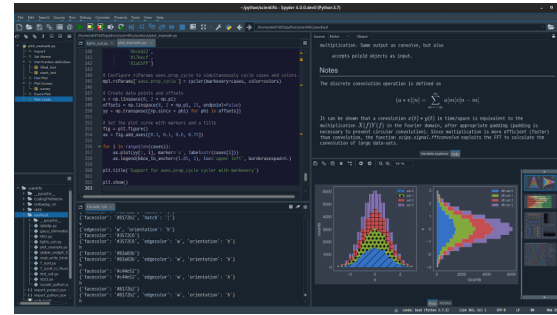
```
└─[~]-[15:59]  
└─$ source Documents/virtual_envs/example_env/bin/activate  
└─[example_env]-[~]-[15:59]  
└─$ which python  
/Users/lisa/Documents/virtual_envs/example_env/bin/python  
└─[example_env]-[~]-[15:59]  
└─$ █
```

Terminal

- In the terminal (and in all other programs) you can and should **deliberately set up which version it uses**
 - i.e. which Python the terminal calls when you type `python`
- depending on which Terminal shell you are using it may be set up in
 - `bashprofile`
 - `.bashrc`
 - `.zshprofile`
 - `.profile`
 - ...

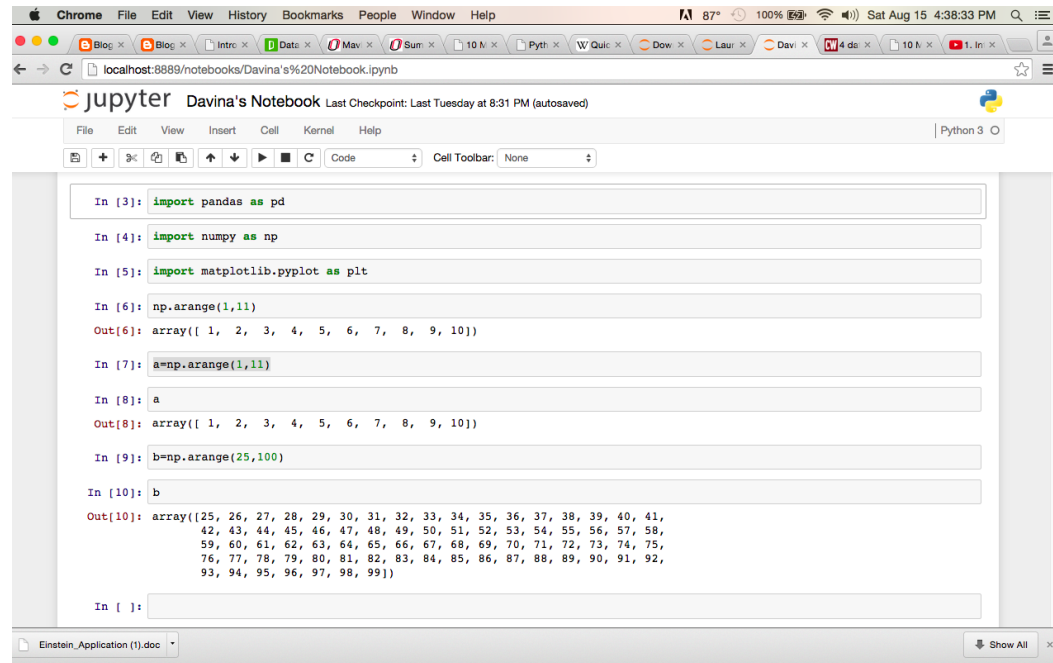
Editors

- There are any number of editors you might use. Notably
 - VScode
 - pycharm
 - Spyder
- Most have **IDE** capabilities
 - set it up to run your code **directly** from the editor
 - tell the IDE **which version of python** you want to use
(usually which *path* to use)



Notebooks

- Ipython/Jupyter Notebooks are really nice for data exploration
 - start it from inside the environment you want to use
 - `pip install jupyter`
 - `jupyter notebook`
 - (OR set up a **kernel**)



The screenshot shows a Jupyter Notebook running in a Chrome browser. The browser's address bar shows the URL `localhost:8889/notebooks/Davina's%20Notebook.ipynb`. The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations and execution. The notebook content consists of several code cells:

```
In [3]: import pandas as pd

In [4]: import numpy as np

In [5]: import matplotlib.pyplot as plt

In [6]: np.arange(1,11)
Out[6]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])

In [7]: a=np.arange(1,11)
In [8]: a
Out[8]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])

In [9]: b=np.arange(25,100)
In [10]: b
Out[10]: array([25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
         42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58,
         59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75,
         76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92,
         93, 94, 95, 96, 97, 98, 99])

In [ ]:
```

The bottom of the browser window shows a taskbar with a file named 'Einstein_Application (1).doc' and a 'Show All' button.

How to stay sane

- **Pick one system and stick with it!**
 - conda OR pip
- Identify which is your **base Python**
 - **Do not** install any packages directly there!
 - **Do** set up the base Python version as the **default** in the terminal and other programs like IDEs
- Whenever you want to start a project, make a **virtual environment** using the base Python
- In that environment install all your dependencies

Now let's get started!



Other Problems

- But I NEED Multiple pythons!
 - You could get really familiar with using the terminal and defining your own commands
 - Look into using pyenv or something similar
-