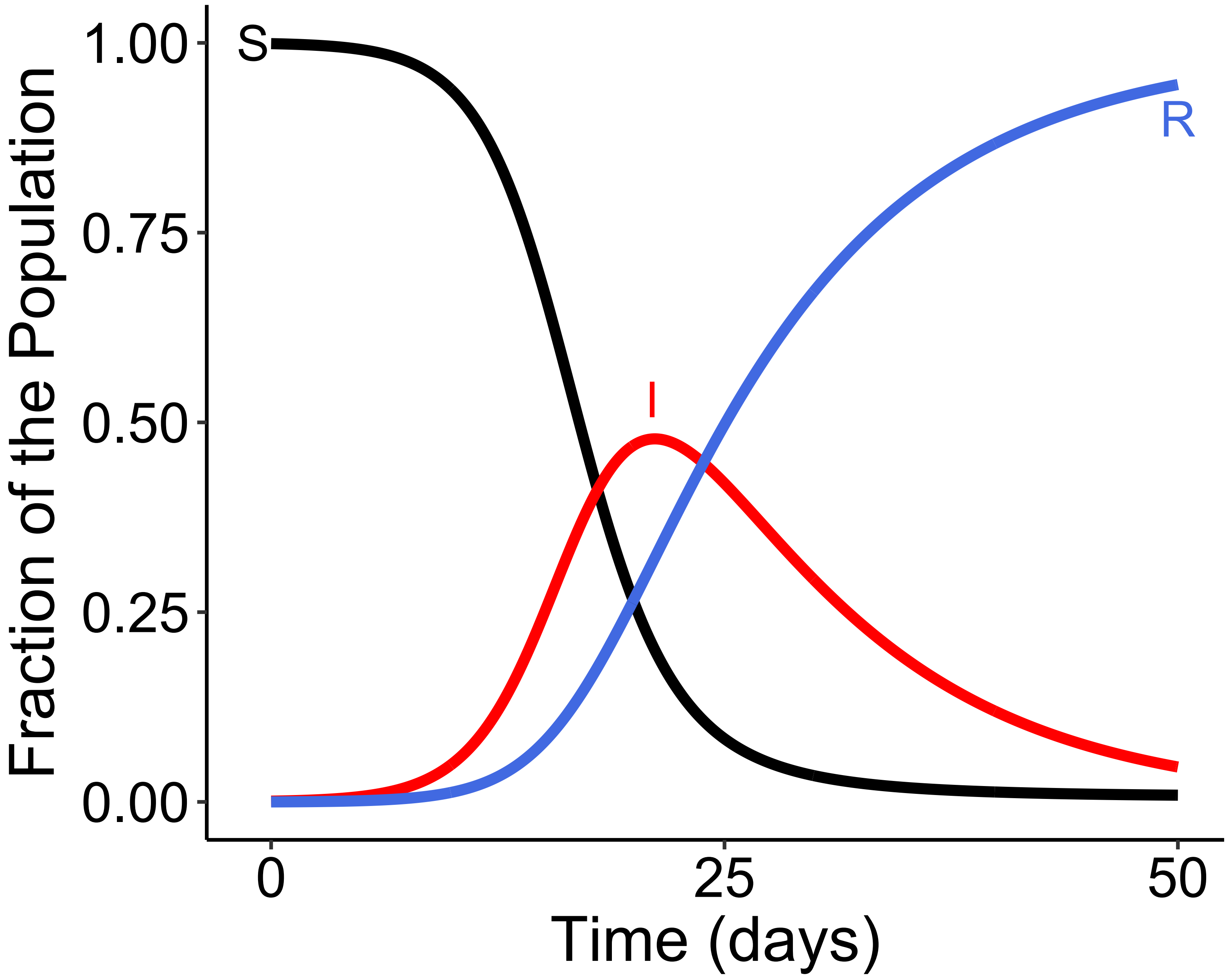
NORTHERN ARIZONA UNIVERSITY | School of Informatics, Computing, and Cyber Systems

Infectious Disease Ecology & Epidemiology

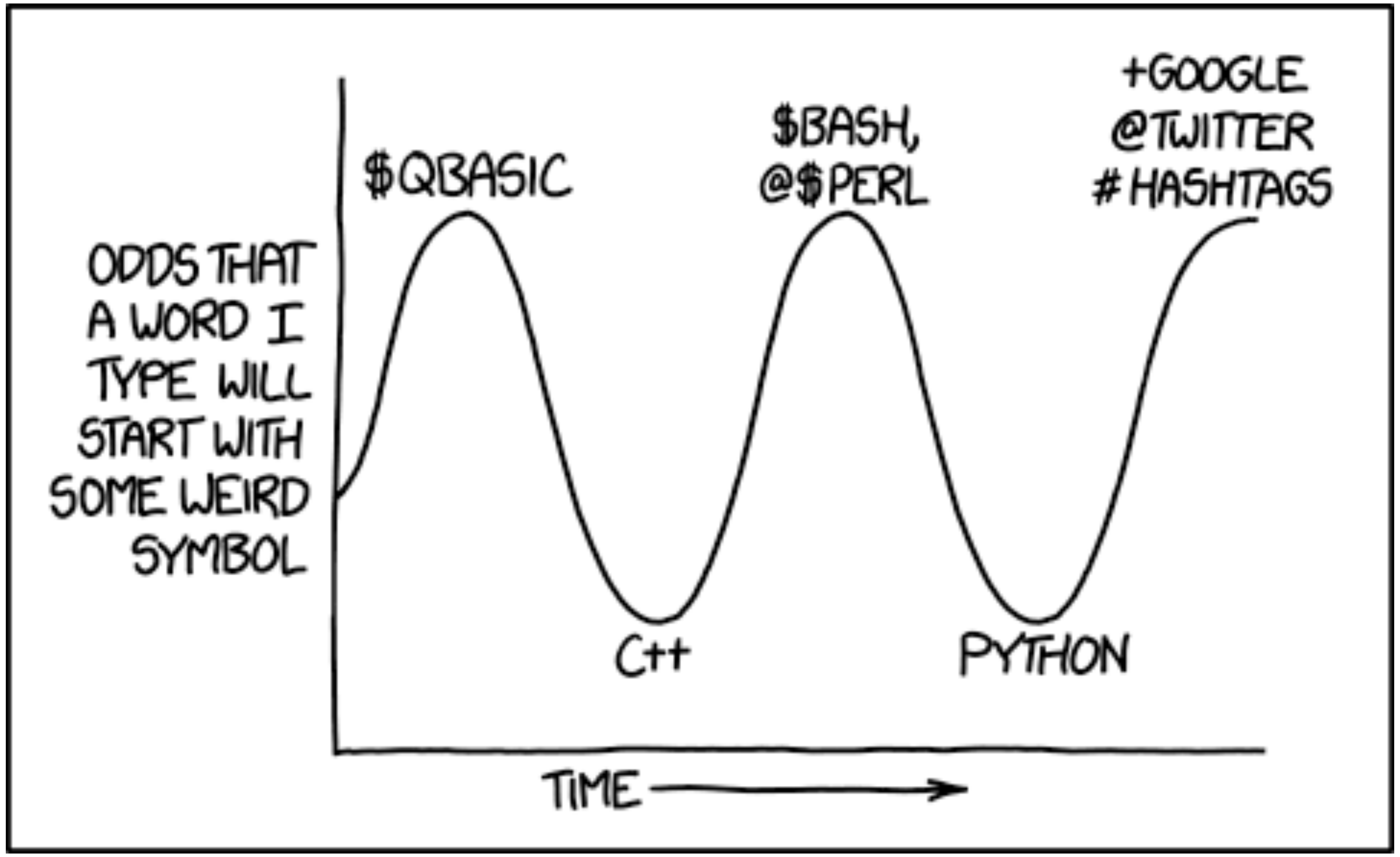
INF 599-001 Spring 2019
T/Th 2:20p – 3:35p, SICCS 102
Contact: Joseph.Mihaljevic@nau.edu



This course will provide students with a basic understanding of disease modeling. We will derive and simulate fundamental insights about infectious disease dynamics, with topics including: predicting the size of epidemics, understanding cyclical disease outbreaks, and simulating spatially-explicit pathogen spread. Throughout, we will reinforce quantitative insights with biological concepts, computer-based assignments, and relevant readings from the literature.

Python Scripting - Part 3

Fall 2018
PCfB Class 6
October 5, 2018


ODDS THAT A WORD I TYPE WILL START WITH SOME WEIRD SYMBOL
$QBASIC
C++
$BASH, @$PERL
PYTHON
+GOOGLE @TWITTER #HASHTAGS
TIME

Sets

- Unordered lists, CANNOT contain duplicate values
- Easy to convert between lists and sets
  - `set(alistvariable), list(asetvariable)`
- Quick way to obtain unique values in list
- Built-in set methods for set comparison
  - `.difference(), .union(), .intersection()`

`while` loops

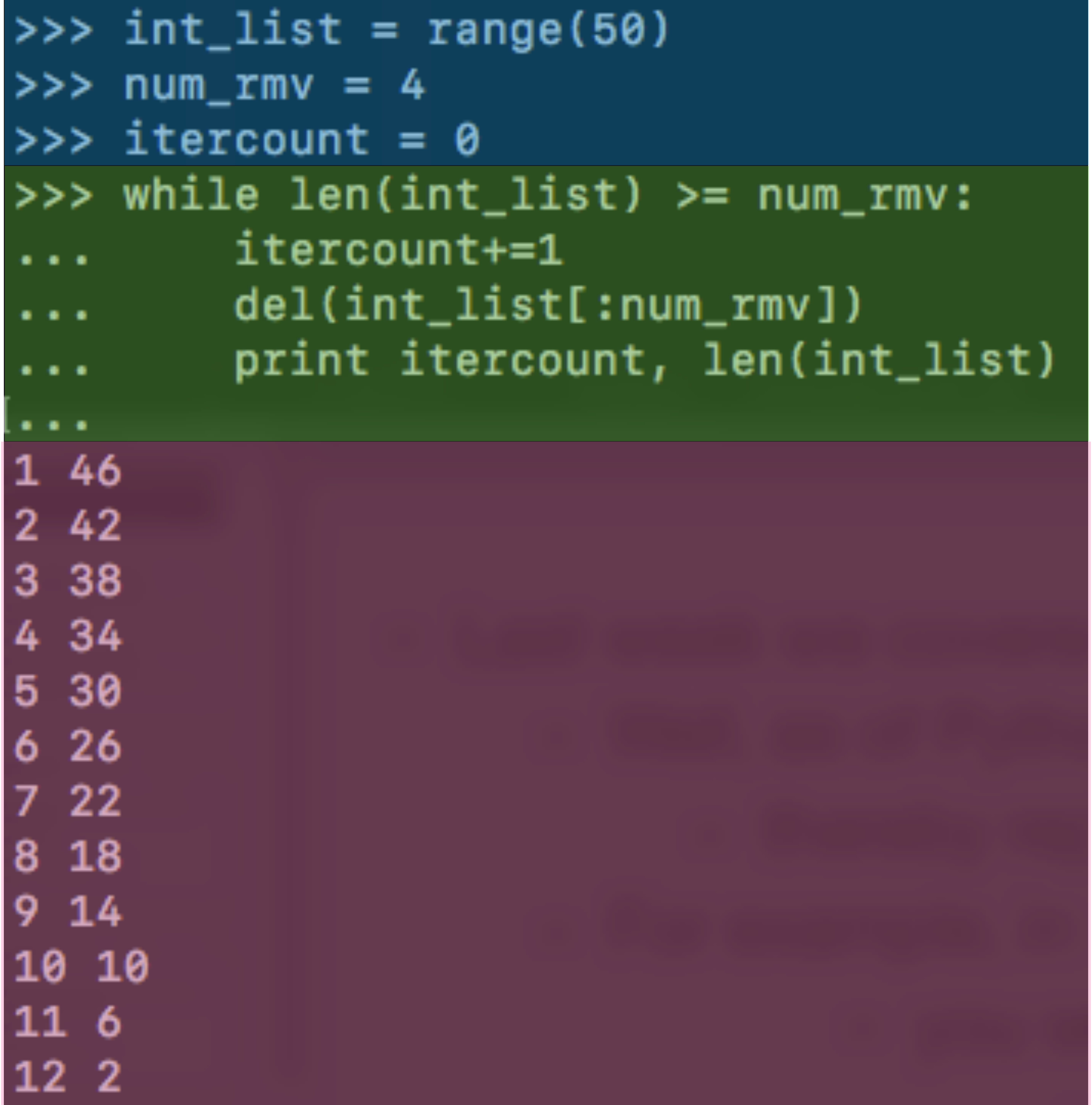
- Iterate over the loop until logical expression == False
- Beware of infinite loops!
- Contents must be indented

```
int_list = range(100)               # Creates list with 100 integers
num_rmv = 4                         # # of items to remove from list in each iteration
itercount = 0                       # To keep track of the number of loop iterations
while len(int_list) >= num_rmv:     # Initiate while loop
    itercount+=1                    # Increment counter
    del(int_list[:num_rmv])         # Delete items from the int_list variable
    print itercount, len(int_list)  # Print current iteration # and length of int_list
```

Initiate variables

While loop

Printed results



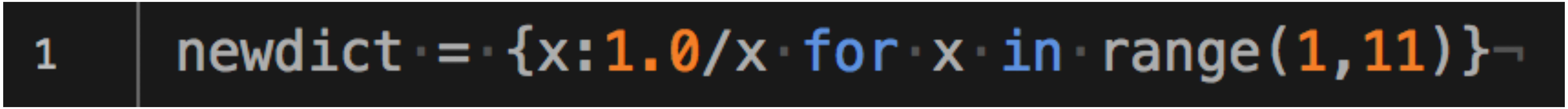
```
>>> int_list = range(50)
>>> num_rmv = 4
>>> itercount = 0
>>> while len(int_list) >= num_rmv:
...     itercount+=1
...     del(int_list[:num_rmv])
...     print itercount, len(int_list)
...
1 46
2 42
3 38
4 34
5 30
6 26
7 22
8 18
9 14
10 10
11 6
12 2
```

Control statement within loops

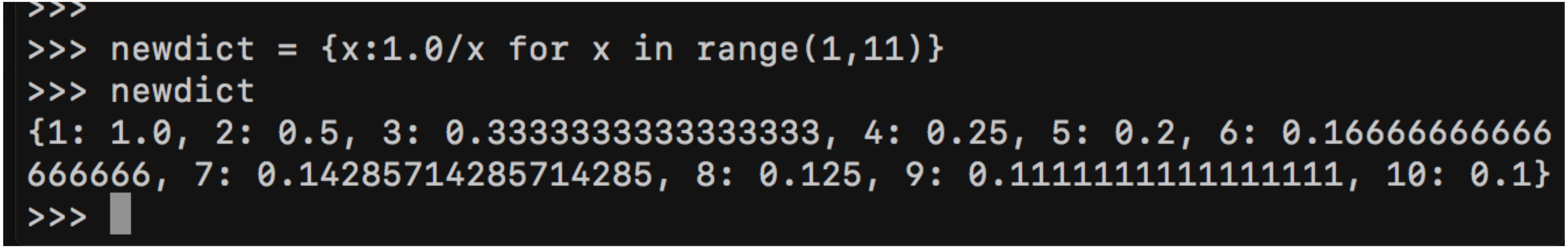
- `break`: internal command to exit loop
- `continue`: end current loop iteration

Dictionary comprehension

- Can be used to create dictionaries from scratch



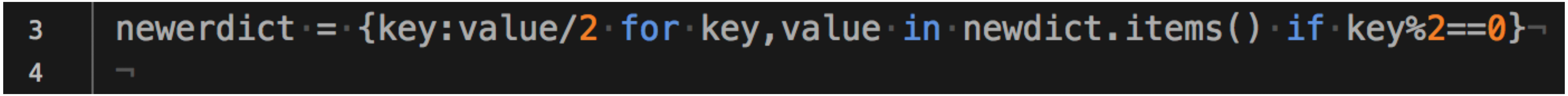
```
newdict = {x:1.0/x for x in range(1,11)}
```



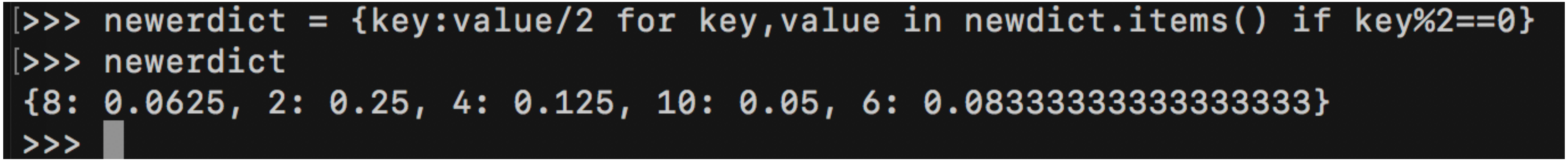
```
>>>
>>> newdict = {x:1.0/x for x in range(1,11)}
>>> newdict
{1: 1.0, 2: 0.5, 3: 0.3333333333333333, 4: 0.25, 5: 0.2, 6: 0.16666666666
666666, 7: 0.14285714285714285, 8: 0.125, 9: 0.1111111111111111, 10: 0.1}
>>> 
```

Dictionary comprehension

- Can be used to make changes to existing dicts
- Can incorporate if statements



```python
newerdict = {key:value/2 for key,value in newdict.items() if key%2==0}

```



```
[>>> newerdict = {key:value/2 for key,value in newdict.items() if key%2==0}
[>>> newerdict
{8: 0.0625, 2: 0.25, 4: 0.125, 10: 0.05, 6: 0.08333333333333333}
>>> 
```

Python modules

- Python functions that can be imported, as needed for use within your scripts
- Standard modules: included with Python installation
- 3rd party modules: must be installed
- Always include import statements at beginning of script (just after #! line)

Method #1: `import numpy`

- Import the entire module and link functions to the module name

- Example usage:

  - `numpy.mean(range(4,19,2))`

  - `numpy.std(range(4,19,2))`

Method #2: `import numpy as np`

- Import the entire module and link functions to a name specified by the user
- Example usage:
  - `np.mean(range(4,19,2))`
  - `np.std(range(4,19,2))`

Method #3: `from numpy import mean`

- Import select functions from a module

- Functions exist on their own, NOT linked to module name

- Example usage:

  - `mean(range(4,19,2))`

Method #4: `from numpy import *`

- Import ALL functions from a module
- Functions exist on their own, NOT linked to module name
- Example usage:

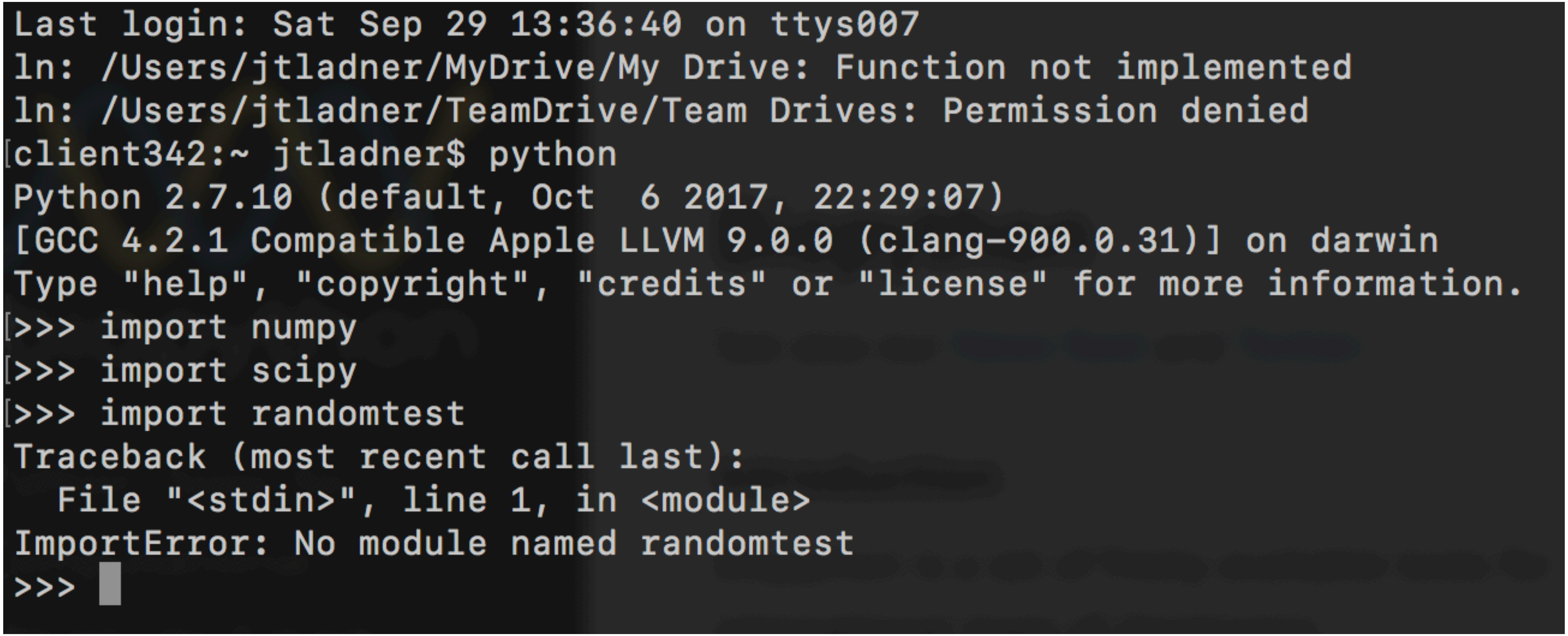
```
-mean(range(4,19,2))

-std(range(4,19,2))
```

- **Potential for name conflicts, especially when importing multiple modules**
- **Unclear which functions are from which module**

Recommended 3rd party modules

- **NumPy & SciPy** (https://scipy.org/)
- **Biopython** (https://biopython.org/)
- **Matplotlib** (https://matplotlib.org/)

Checking to see if module is installed



```
Last login: Sat Sep 29 13:36:40 on ttys007
ln: /Users/jtladner/MyDrive/My Drive: Function not implemented
ln: /Users/jtladner/TeamDrive/Team Drives: Permission denied
client342:~ jtladner$ python
Python 2.7.10 (default, Oct  6 2017, 22:29:07)
[GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> import scipy
>>> import randomtest
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named randomtest
>>> 
```

Exercises

- Stand-alone analysis scripts
- Feel free to work on today's exercises of those from previous weeks