Python

For Fun and Professional Profit

Ryan T. Hamilton, Instrument Scientist
Lowell Observatory, July 2018

Step 1:

Figure out what Python is.

Step 1:

Figure out what Python is.

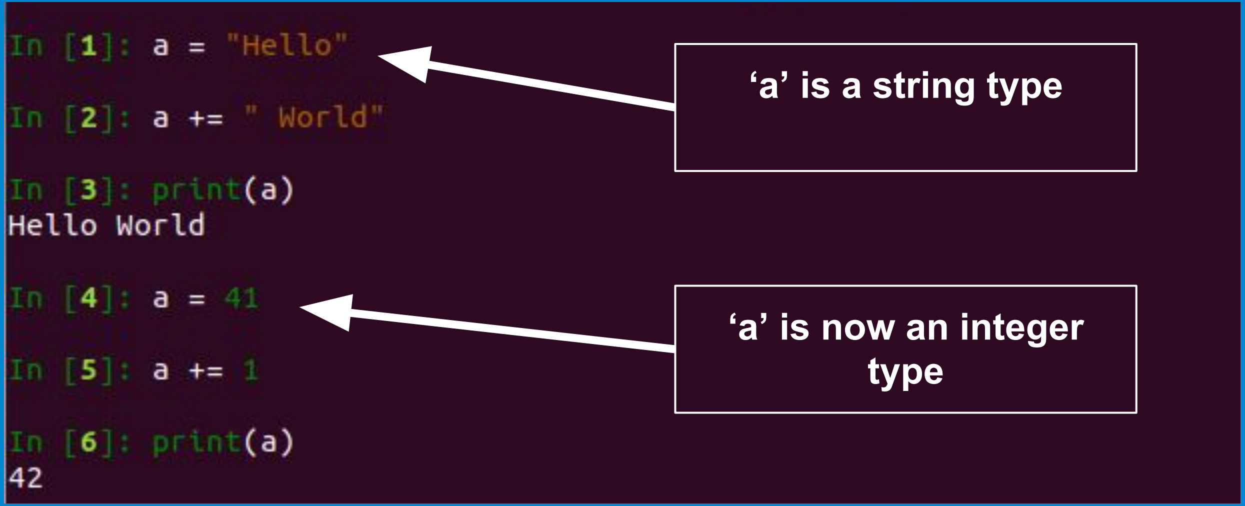
Tip: Don’t do a Google image search. It’s kinda gross.

Python’s Goals:

An easy and intuitive language just as powerful as major competitors.
Open source, so anyone can contribute to its development.
Code that is as understandable as plain English.
Suitability for everyday tasks, allowing for short development times.

Python Quick Rundown

- Interpreted, not compiled
  - Means errors discovered at runtime, unless you incorporate a linter into your workflow/IDE
    - Most good IDEs will either bundle one or nag you to incorporate one
- Flexible typing

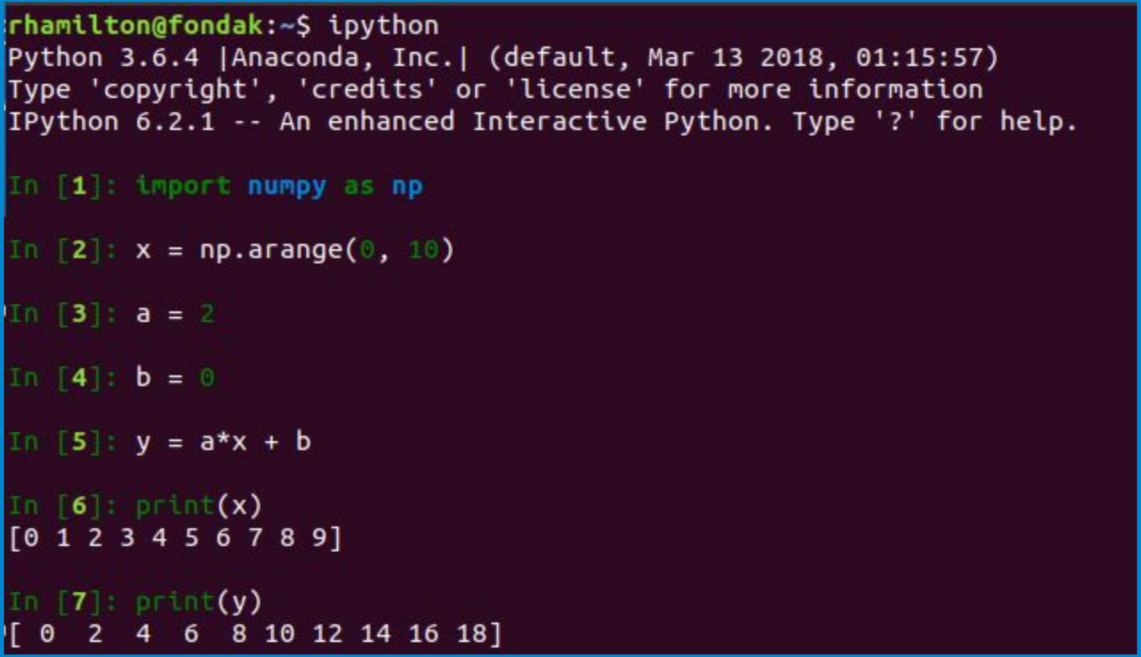


```
In [1]: a = "Hello"

In [2]: a += " World"

In [3]: print(a)
Hello World

In [4]: a = 41

In [5]: a += 1

In [6]: print(a)
42
```

‘a’ is a string type

‘a’ is now an integer type

Python Quick Rundown

- Array broadcasting!
  - 

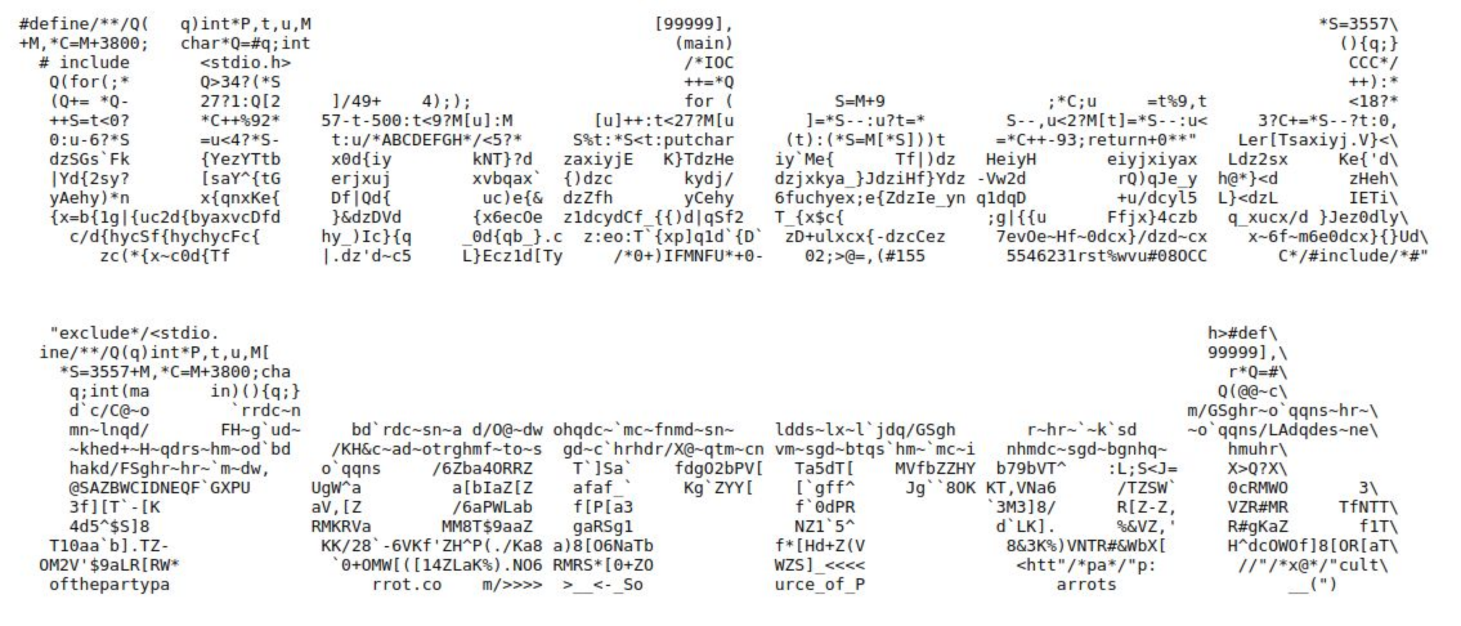
```
rhamilton@fondak:~$ ipython
Python 3.6.4 |Anaconda, Inc.| (default, Mar 13 2018, 01:15:57)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.2.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import numpy as np

In [2]: x = np.arange(0, 10)

In [3]: a = 2

In [4]: b = 0

In [5]: y = a*x + b

In [6]: print(x)
[0 1 2 3 4 5 6 7 8 9]

In [7]: print(y)
[ 0  2  4  6  8 10 12 14 16 18]
```

Python Quick Rundown

- Other quick boons:
    - Multiple return values
    - Pass any variable or object to a function trivially
        - Pass a function to function which bundles it up in a class and returns it to be passed to another function
    - Array strides and array slices
    - Object introspection/discovery (dir() and .__dict__)
    - Memory is managed for you (no malloc concept)
    - It’s taking over even the embedded microprocessor space! Great for education!
        - MicroPython and CircuitPython

Python Quick Rundown

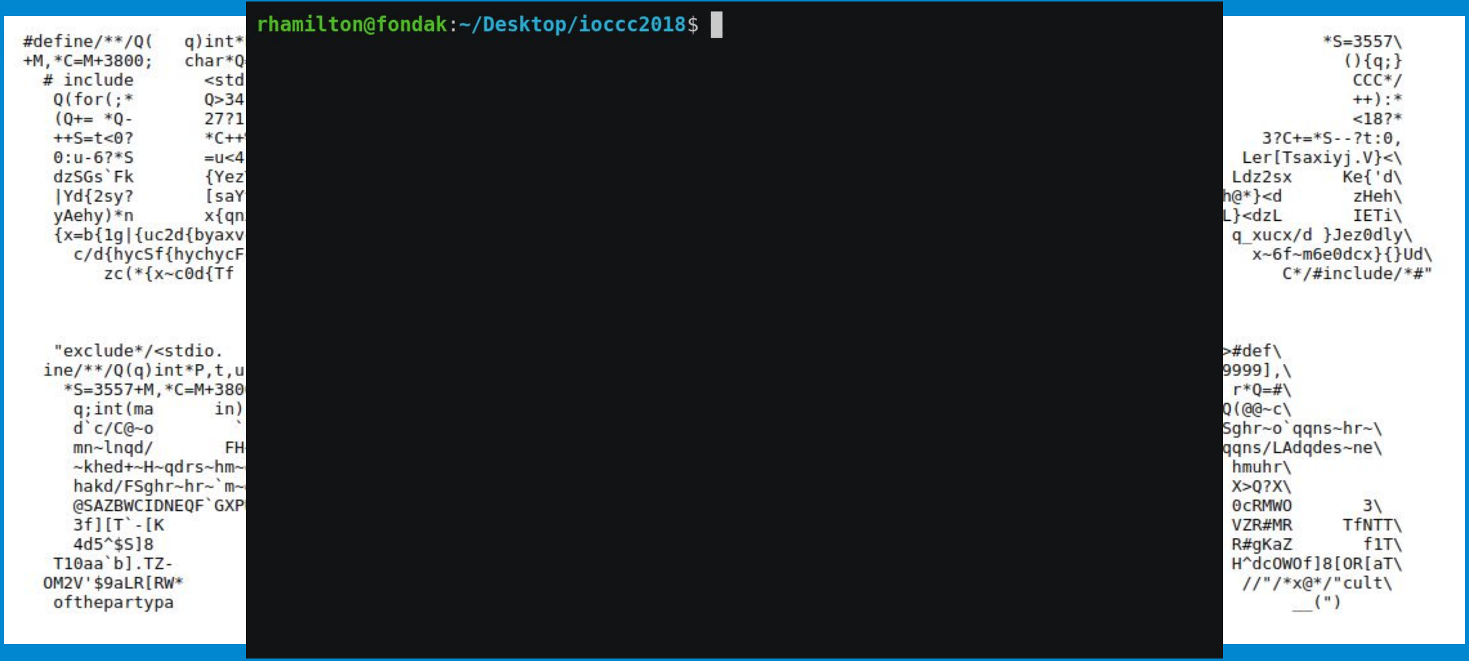
- FORCED code style (via indentation/line spacing)
    - Avoids this (https://www.ioccc.org/2018/endoh2); it’s valid C code!



```
#define/**/Q(   q)int*P,t,u,M                                  [99999],                                                          *S=3557\
+M,*C=M+3800;   char*Q=#q;int                                    (main)                                                            (){q;}
  # include       <stdio.h>                                       /*IOC                                                             CCC*/
   Q(for(;*       Q>34?(*S                                        ++=*Q                                                             ++):*
   (Q+= *Q-       27?1:Q[2     ]/49+    4););                     for (          S=M+9                ;*C;u     =t%9,t              <18?*
   ++S=t<0?       *C++%92*    57-t-500:t<9?M[u]:M        [u]++:t<27?M[u       ]=*S--:u?t=*        S--,u<2?M[t]=*S--:u<     3?C+=*S--?t:0,
   0:u-6?*S       =u<4?*S-     t:u/*ABCDEFGH*/<5?*     S%t:*S<t:putchar     (t):(*S=M[*S]))t     =*C++-93;return+0**"    Ler[Tsaxiyj.V}<\
   dzSGs`Fk       {YezYTtb     x0d{iy        kNT}?d   zaxiyjE   K}TdzHe    iy`Me{      Tf|)dz   HeiyH       eiyjxiyax   Ldz2sx     Ke{'d\
   |Yd{2sy?       [saY^{tG     erjxuj        xvbqax`  {)dzc       kydj/    dzjxkya_}JdziHf}Ydz -Vw2d         rQ)qJe_y  h@*}<d       zHeh\
   yAehy)*n       x{qnxKe{     Df|Qd{         uc)e{&  dzZfh       yCehy    6fuchyex;e{ZdzIe_yn q1dqD         +u/dcyl5  L}<dzL       IETi\
   {x=b{1g|{uc2d{byaxvcDfd     }&dzDVd       {x6ec0e  z1dcydCf_{{)d|qSf2   T_{x$c{              ;g|{{u      Ffjx}4czb   q_xucx/d }Jez0dly\
     c/d{hycSf{hychycFc{      hy_)Ic}{q     _0d{qb_}.c  z:eo:T`{xp]q1d`{D`  zD+ulxcx{-dzcCez     7ev0e~Hf~0dcx}/dzd~cx    x~6f~m6e0dcx}{}Ud\
        zc(*{x~c0d{Tf         |.dz'd~c5     L}Ecz1d[Ty     /*0+)IFMNFU*+0-    02;>@=,(#155        5546231rst%wvu#08OCC       C*/#include/*#"



   "exclude*/<stdio.                                                                                                  h>#def\
  ine/**/Q(q)int*P,t,u,M[                                                                                             99999],\
    *S=3557+M,*C=M+3800;cha                                                                                             r*Q=#\
     q;int(ma      in)(){q;}                                                                                           Q(@@~c\
     d`c/C@~o        `rrdc~n                                                                                        m/GSghr~o`qqns~hr~\
     mn~lnqd/       FH~g`ud~     bd`rdc~sn~a d/O@~dw ohqdc~`mc~fnmd~sn~    ldds~lx~l`jdq/GSgh       r~hr~`~k`sd     ~o`qqns/LAdqdes~ne\
     ~khed+~H~qdrs~hm~od`bd    /KH&c~ad~otrghmf~to~s  gd~c`hrhdr/X@~qtm~cn vm~sgd~btqs`hm~`mc~i   nhmdc~sgd~bgnhq~      hmuhr\
     hakd/FSghr~hr~`m~dw,     o`qqns     /6Zba4ORRZ    T`]Sa`    fdgO2bPV[   Ta5dT[    MVfbZZHY  b79bVT^    :L;S<J=     X>Q?X\
     @SAZBWCIDNEQF`GXPU      UgW^a         a[bIaZ[Z    afaf_`     Kg`ZYY[    [`gff^     Jg``8OK KT,VNa6      /TZSW`     0cRMWO       3\
     3f][T`-[K               aV,[Z         /6aPWLab    f[P[a3                f`0dPR             `3M3]8/      R[Z-Z,     VZR#MR     TfNTT\
     4d5^$S]8                RMKRVa       MM8T$9aaZ    gaRSg1                NZ1`5^              d`LK].      %&VZ,'     R#gKaZ       f1T\
   T10aa`b].TZ-               KK/28`-6VKf'ZH^P(./Ka8 a)8[O6NaTb            f*[Hd+Z(V              8&3K%)VNTR#&WbX[      H^dcOWOf]8[OR[aT\
  OM2V'$9aLR[RW*               `0+OMW[([14ZLaK%).NO6 RMRS*[0+ZO            WZS]_<<<<               <htt"/*pa*/"p:        //"/*x@*/"cult\
   ofthepartypa                    rrot.co    m/>>>>  >__<-_So             urce_of_P                  arrots                 __(")
```

Python Quick Rundown

- FORCED code style (via indentation/line spacing)
  - Avoids this (https://www.ioccc.org/2018/endoh2); it's valid C code!



```
#define/**/Q(   q)int*
+M,*C=M+3800;   char*Q
  # include       <std
   Q(for(;*       Q>34
   (Q+= *Q-       27?1
   ++S=t<0?       *C++
   0:u-6?*S       =u<4
   dzSGs`Fk       {Yez
   |Yd{2sy?       [saY
   yAehy)*n       x{qn
   {x=b{1g|{uc2d{byaxv
     c/d{hycSf{hychycF
        zc(*{x~c0d{Tf



   "exclude*/<stdio.
  ine/**/Q(q)int*P,t,u
    *S=3557+M,*C=M+380
     q;int(ma      in)
     d`c/C@~o        `
     mn~lnqd/        FH
     ~khed+~H~qdrs~hm~
     hakd/FSghr~hr~`m~
     @SAZBWCIDNEQF`GXP
     3f][T`-[K
     4d5^$S]8
   T10aa`b].TZ-
  OM2V'$9aLR[RW*
   ofthepartypa
```

```
rhamilton@fondak:~/Desktop/ioccc2018$
```

```
          *S=3557\
            (){q;}
             CCC*/
             ++):*
             <18?*
     3?C+=*S--?t:0,
   Ler[Tsaxiyj.V}<\
  Ldz2sx     Ke{'d\
h@*}<d        zHeh\
L}<dzL        IETi\
  q_xucx/d }Jez0dly\
    x~6f~m6e0dcx}{}Ud\
       C*/#include/*#"



>#def\
9999],\
 r*Q=#\
Q(@@~c\
Sghr~o`qqns~hr~\
qqns/LAdqdes~ne\
 hmuhr\
 X>Q?X\
 0cRMWO        3\
 VZR#MR     TfNTT\
 R#gKaZ       f1T\
 H^dcOWOf]8[OR[aT\
  //"/*x@*/"cult\
      __(")
```

Python Quick Rundown

- Object oriented
  - **Objects** have **properties/attributes** to hold information and **methods** to act on stuff
  - Python is **class-based**; kinda-sorta similar to **structures** in C or **modules** in Fortran
    - C++ and Java are also **class-based** languages
    - Typical lingo: “Object X is an ***instance*** of class Y”
- Extendable via **packages**
  - Well-done packages spread via a central repository (pypi.org) and have well-defined versioning and dependencies
    - Half- or quarter-assed packages have some of that
      - Zero-assed packages are just blobs of code that do something

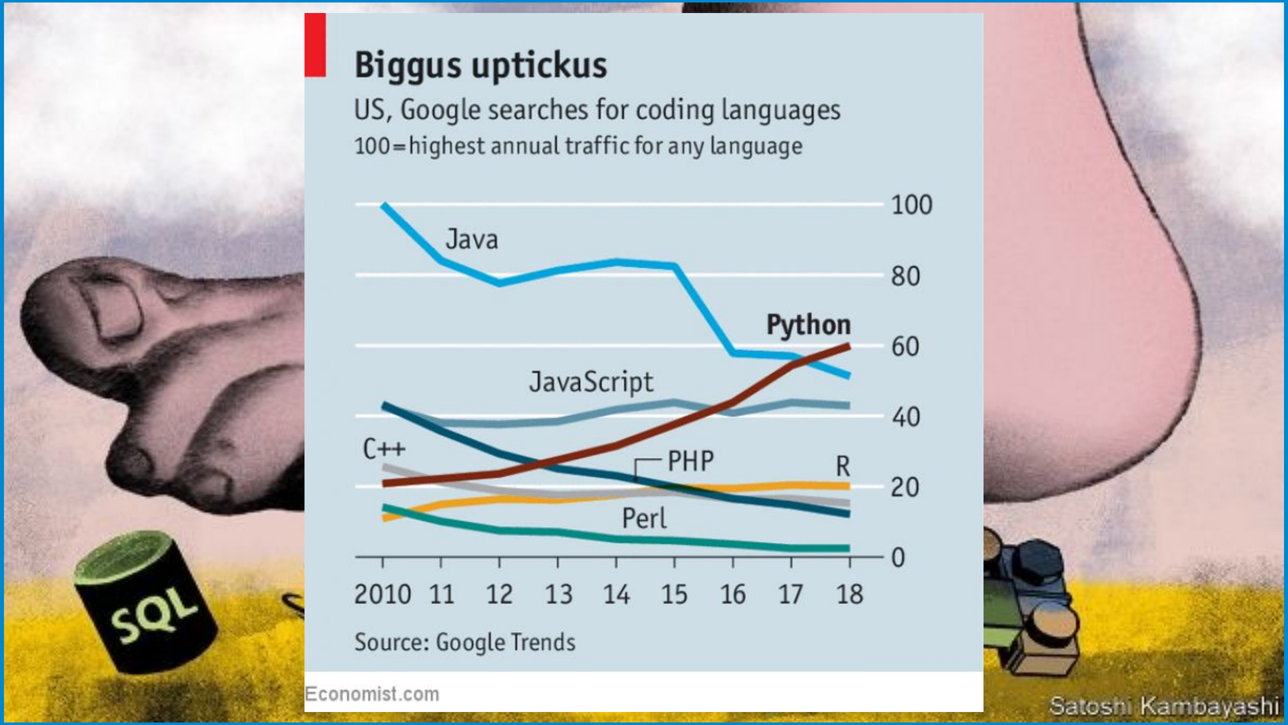
Python Quick Rundown

- Useful stuff that comes ***out-of-the box*** with Python:
  - String manipulation that won’t make you want to scream at the mountains
  - A date/time library (“datetime”) that is follows established standards (strftime, strptime)
    - Can take care of routine calendar math for you
    - A separate package (“tzinfo”) that takes care of timezone/DST annoyances
  - Utilities for interacting with the filesystem/host machine in semi-easy ways
    - Some quirks, same attention to race conditions as any other language needed
  - Robust logging support (with log rotation!)
  - Complex generators and tools for iterating across large multidimensional data
  - Exception/error handling in a sane and graceful way
  - Basic data compression support (zlib, gzip, bz2, zip files, tar files)
  - Interaction with sockets, simple email support (including data encoding)
  - Support for parsing CSV, HTML, XML

Python Quick Rundown

- Useful stuff that can be simply added via **packages**:
    - N-dimensional arrays, linear algebra, regressions, FFTs, complex math, statistics
        - numpy, scipy
    - Full plotting support to a variety of output types (including native PDF/PS)
        - matplotlib (apes the Matlab plotting interface)
    - Complex data structure support for dealing with large/weird datasets (like stellar catalogs)
        - pandas
    - WCS, VOTable, Tables, FITS, text parsing, statistics, fitting, convolutions, visualizations
        - astropy
            - Growing *rapidly* to encompass imaging and spectroscopic data reductions, even a telescope scheduling optimization module in the works
    - GUIs
        - pyqt5

SQL
JS
C++
C
php
Satoshi Kambayashi



Biggus uptickus

US, Google searches for coding languages

100=highest annual traffic for any language

Source: Google Trends

Economist.com

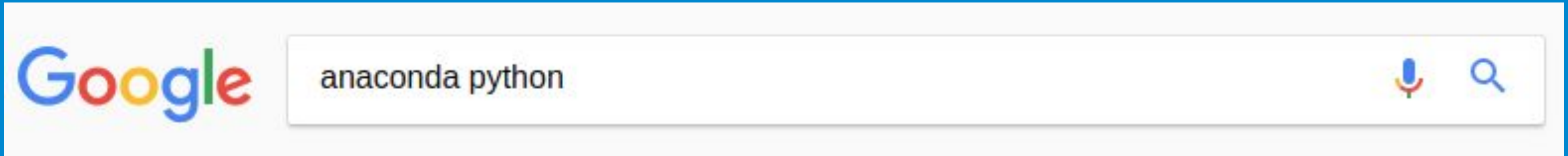
Satoshi Kambayashi

Step 2:

Figure out how to run Python on your daily work computer.

Step 2:

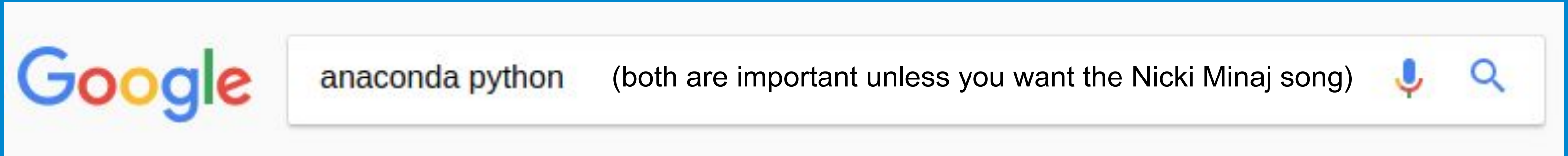
Figure out how to run Python on your daily work computer.



anaconda python

Step 2:

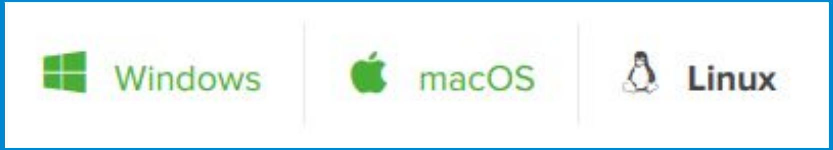
Figure out how to run Python on your daily work computer.



anaconda python (both are important unless you want the Nicki Minaj song)

Step 2:

ANACONDA

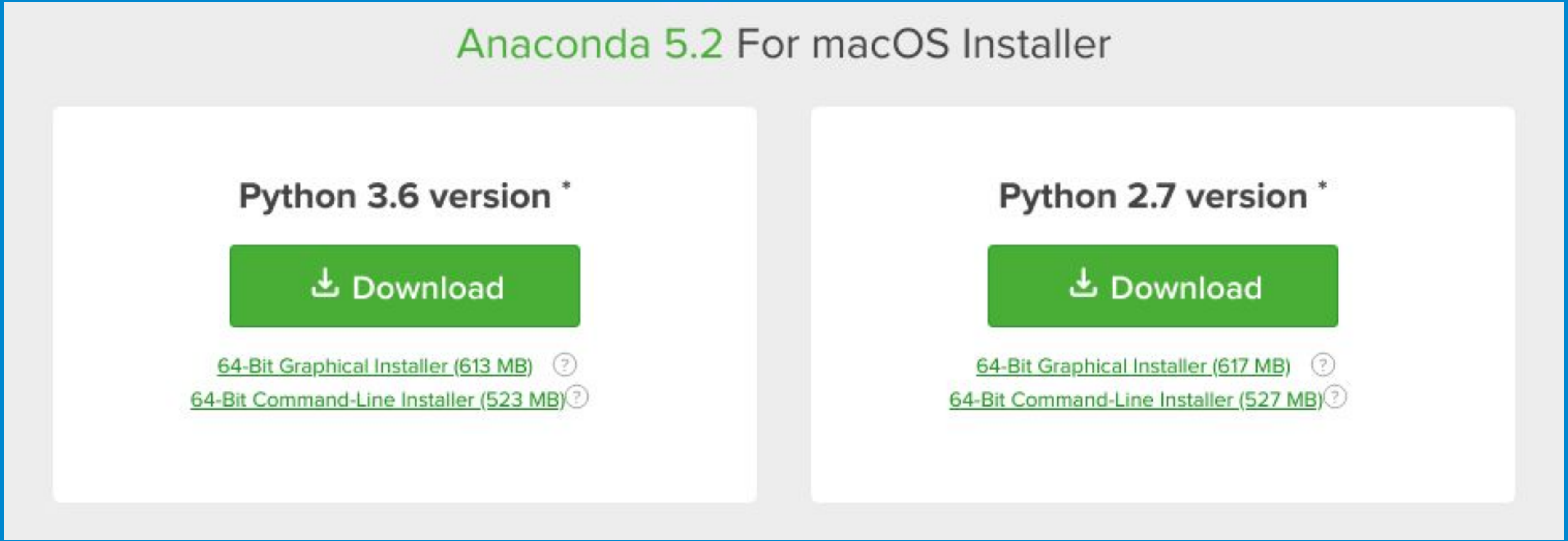


Why Anaconda Python?

Because the Python version by default on your Mac is *old*.

Because Windows doesn’t come with *any* version of Python

Because it’s so easy to get going and doesn’t require root access.



Anaconda 5.2 For macOS Installer

Python 3.6 version *

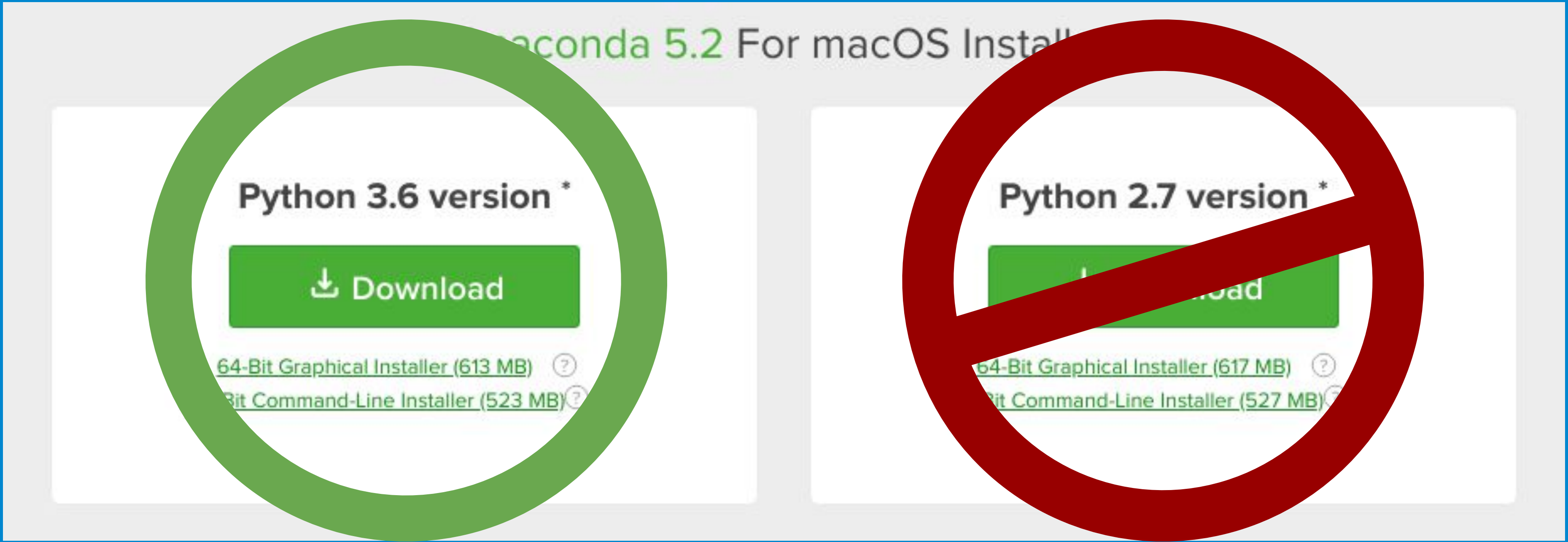
64-Bit Graphical Installer (613 MB) ?

64-Bit Command-Line Installer (523 MB) ?

Python 2.7 version *

64-Bit Graphical Installer (617 MB) ?

64-Bit Command-Line Installer (527 MB) ?



aconda 5.2 For macOS Install

Python 3.6 version *

64-Bit Graphical Installer (613 MB)

Bit Command-Line Installer (523 MB)

Python 2.7 version *

64-Bit Graphical Installer (617 MB)

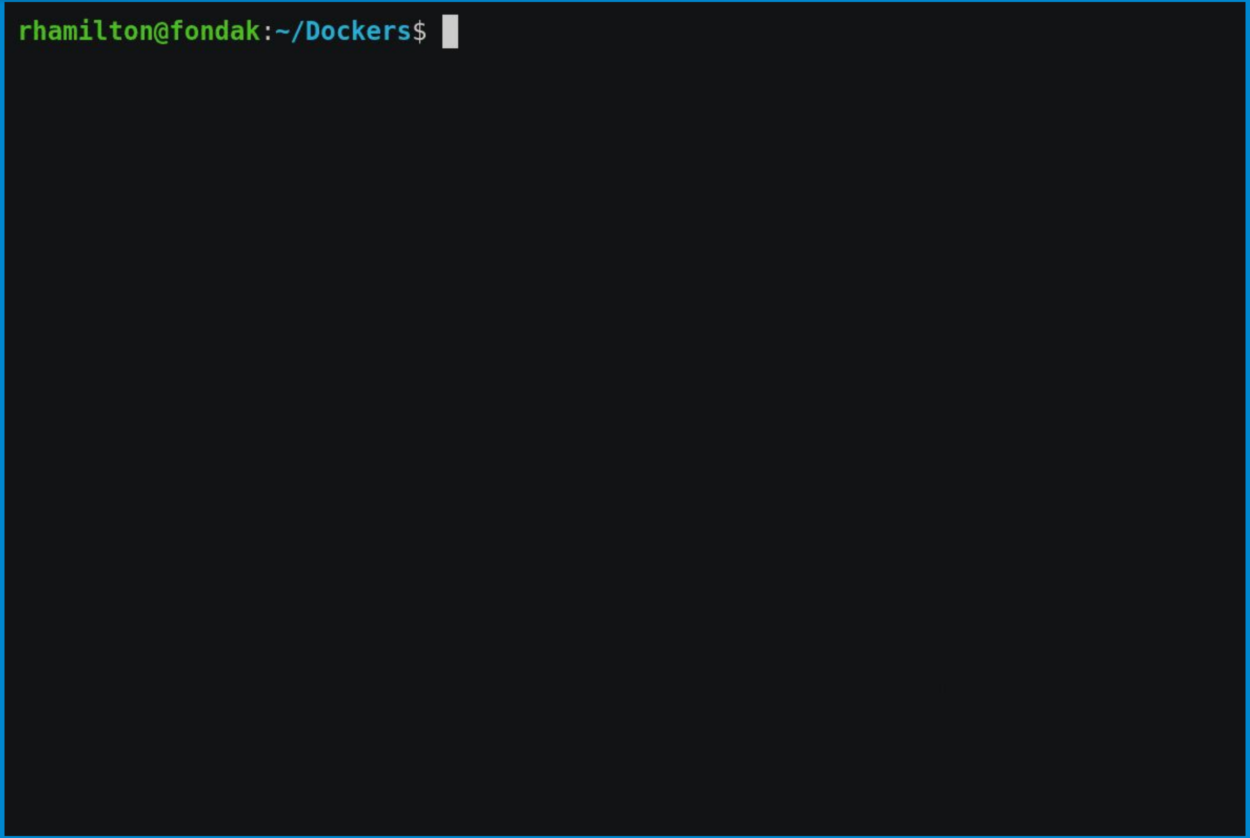
it Command-Line Installer (527 MB)

Python 2 vs. Python 3

- Don’t use Python 2.X for new code
    - Python 2.7 will be ***unsupported*** on January 1, 2020.
- Python 3.1 (first usable 3.x version) released in **June 2009**
    - As of 2018-07-25, 341 out of the top 360 Python packages support Python 3.x (http://py3readiness.org/)
        - Looking at the list of the remaining 19, I only recognized four as useful ones
            - *beautifulsoup*, but it is obsoleted in favor of *BeautifulSoup4* which is Python 3.x
                - Parse some truly ugly HTML
            - *ansible* and *fabric* are listed, but they ***do*** have Python 3.x support.
                - sysadmin/IT automation and deployment utilities
            - *MYSQL-python* is still on Python 2.x (but there are other DB fish that can be fried)

Python 2 vs. Python 3

- Don’t use Python 2.X for new code
    - ○



```
rhamilton@fondak:~/Dockers$
```

Step 3→N:

Learn Python. There are already great resources out there!

https://git.io/fN0ir

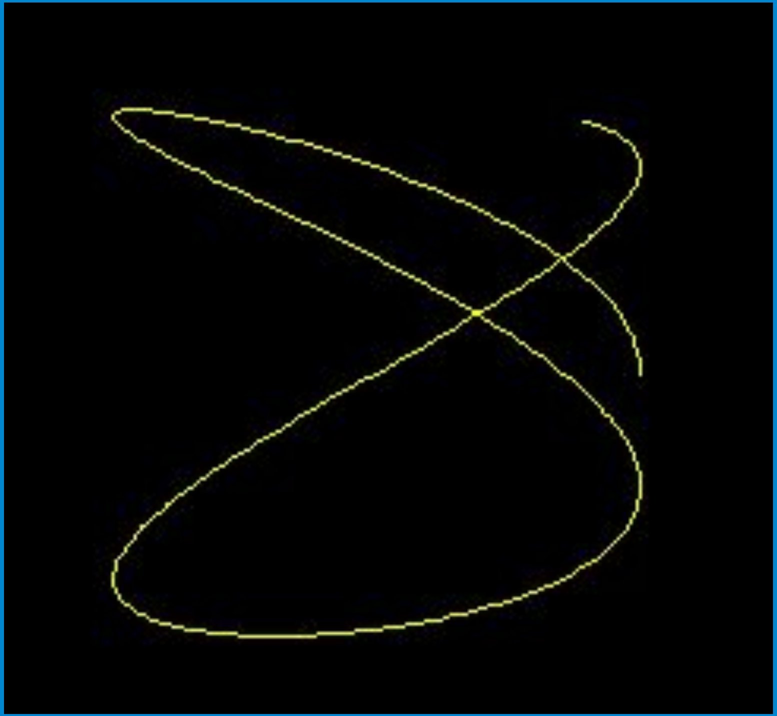
https://github.com/astropy/astropy-workshops/tree/master/aas231_workshop/02-PythonIntro

Time for a few “Real World” examples

To better show the concepts

State and Define “The Problem”

- A FIR inst. on Earth needs to subtract the sky
- Need to avoid systematic effects:
  - Sample the background randomly
  - Average signals across as many detectors
- Telescopes hate to move randomly! Hard to do
  - Lissajous curves provide the best balance
    - https://arxiv.org/abs/0806.4888
- Want to maximize coverage on detectors given:
  - Scan speed (RMS),
  - Scan amplitude (Alt/Az or RA/DEC)
  - Phase
  - Duration
- How do we pick good values? How do we avoid crappy values accidentally? Code it up!

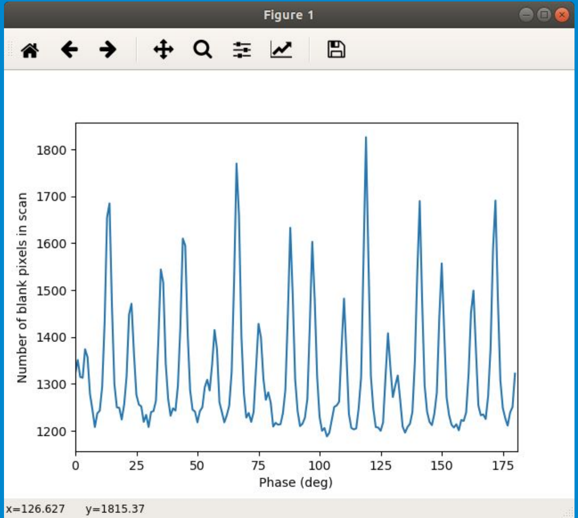


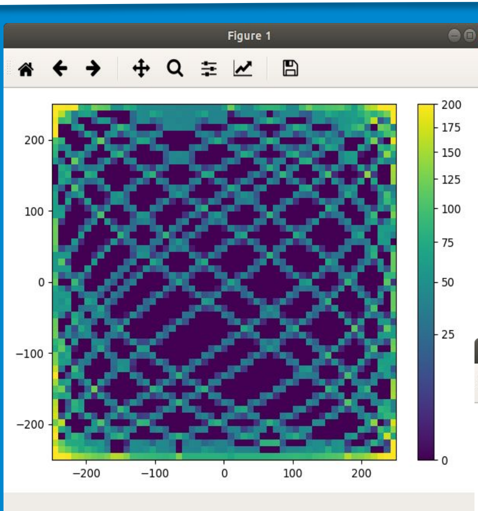
The Code!

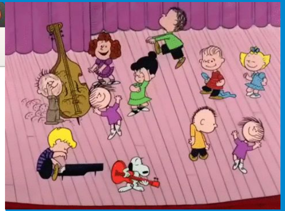
(stop looking at this page)

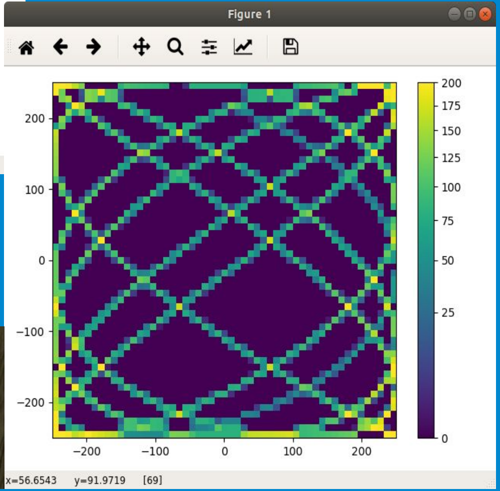
(don’t look over here either, go load the code)

Code Outputs



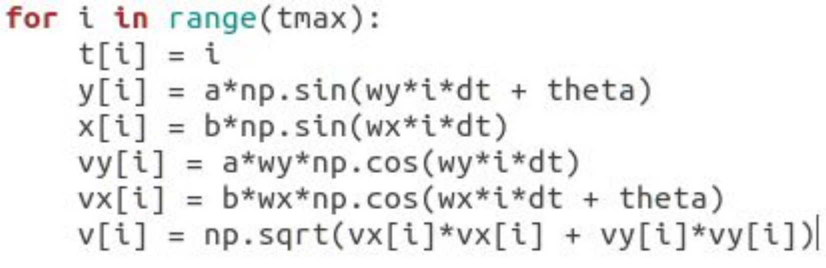






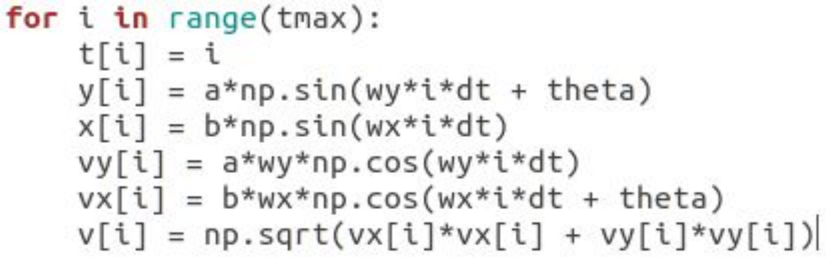
Lowest num. of empty bins: 1188 (phase = 103 deg)
Highest num. of empty bins: 1826 (phase = 119 deg)

Don’t Write Python Like You’d Write Fortran

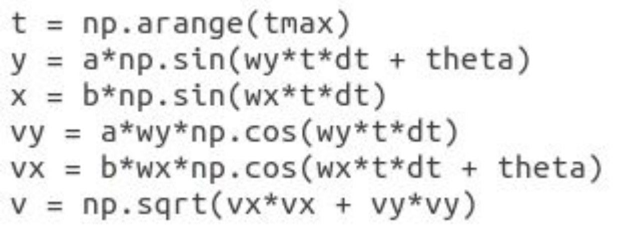


```python
for i in range(tmax):
    t[i] = i
    y[i] = a*np.sin(wy*i*dt + theta)
    x[i] = b*np.sin(wx*i*dt)
    vy[i] = a*wy*np.cos(wy*i*dt)
    vx[i] = b*wx*np.cos(wx*i*dt + theta)
    v[i] = np.sqrt(vx[i]*vx[i] + vy[i]*vy[i])
```

Don’t Write Python Like You’d Write Fortran

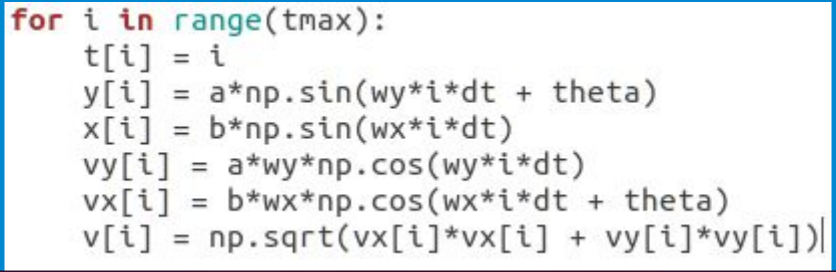


```python
for i in range(tmax):
    t[i] = i
    y[i] = a*np.sin(wy*i*dt + theta)
    x[i] = b*np.sin(wx*i*dt)
    vy[i] = a*wy*np.cos(wy*i*dt)
    vx[i] = b*wx*np.cos(wx*i*dt + theta)
    v[i] = np.sqrt(vx[i]*vx[i] + vy[i]*vy[i])
```

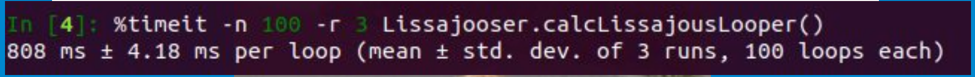


```python
t = np.arange(tmax)
y = a*np.sin(wy*t*dt + theta)
x = b*np.sin(wx*t*dt)
vy = a*wy*np.cos(wy*t*dt)
vx = b*wx*np.cos(wx*t*dt + theta)
v = np.sqrt(vx*vx + vy*vy)
```

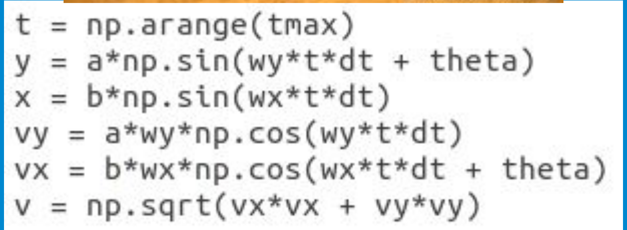
Don’t Write Python Like You’d Write Fortran



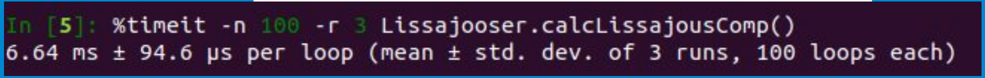
```python
for i in range(tmax):
    t[i] = i
    y[i] = a*np.sin(wy*i*dt + theta)
    x[i] = b*np.sin(wx*i*dt)
    vy[i] = a*wy*np.cos(wy*i*dt)
    vx[i] = b*wx*np.cos(wx*i*dt + theta)
    v[i] = np.sqrt(vx[i]*vx[i] + vy[i]*vy[i])
```



```
In [4]: %timeit -n 100 -r 3 Lissajooser.calcLissajousLooper()
808 ms ± 4.18 ms per loop (mean ± std. dev. of 3 runs, 100 loops each)
```



```python
t = np.arange(tmax)
y = a*np.sin(wy*t*dt + theta)
x = b*np.sin(wx*t*dt)
vy = a*wy*np.cos(wy*t*dt)
vx = b*wx*np.cos(wx*t*dt + theta)
v = np.sqrt(vx*vx + vy*vy)
```



```
In [5]: %timeit -n 100 -r 3 Lissajooser.calcLissajousComp()
6.64 ms ± 94.6 μs per loop (mean ± std. dev. of 3 runs, 100 loops each)
```

Python Tips

- Create classes and functions liberally
  - Modular code is code you can reuse later! Be nice to future you.
  - AVOID GLOBALS. You’ll thank yourself later!
- Think really hard whether you can eliminate explicit loops
  - Nested loops are even worse, just like in other languages
- Experiment! And learn to read the traceback/exceptions
  - Debugging is ESSENTIAL and knowing (how to read it) is half the battle
- Use an IDE (vscode, spyder)
  - For complicated codes it’ll change your life. Guaranteed!

That’s It!