Intro to Scientific Python

Shankar Kulumani

Flight Dynamics & Control Lab

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

2017 March 3

Get Scientific Python!

- Python is a general language, but we care about the science!
- Easiest way is the Anaconda distribution
- Includes everything we need to for Python and science in a easy to manage package

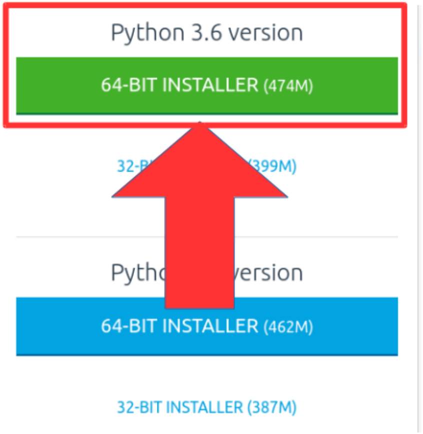


Figure: Just get Python 3

History of Python

- Guido van Rossum started creating Python in 1989

Over six years ago, in December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. . . . I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).

Figure: "Benevolent Dictator For Life"

What is Python?

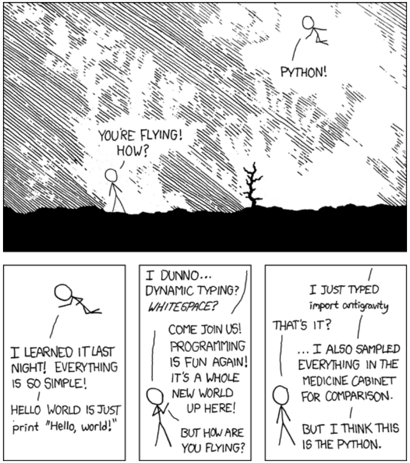
Python is a modern, general-purpose, object-oriented, high-level language.

- *clean and simple language*: Easy to read and easy to learn syntax
- *expressive language*: Fewer lines of code = fewer mistakes
- *dynamically typed*: no need to define variable types or function arguments
- *automatic memory management*: no need to allocate/deallocate memory
- *interpreted*: No need to compile! Fast and easy

Why Python?

- Free - free as in beer AND free as in speech
- General purpose - packages/modules for everything!
- Dynamic - no compiling
- Easy to read - enforces good structure!
- Open - everything is an object

It's harder to read code than to write it!


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Python vs. Matlab

Disadvantages:

- Matlab is a commerical product - entire computing enviornment with code, IDE
- Matlab is expensive - Between $49 - $2150 per license! Extra for toolboxes
- Matlab is proprietary - Cannot inpect source code and restrictions on sharing
- Matlab is closed - difficult to extend functionality

Advantages:

- Matlab handles arrays automatically and by design
- Lots of functionality - control design, linear algebra, optimization, ODEs etc.
- Real engineers (with funding) use it so students have to as well
- Simulink is still unmatched
- Powerful plotting capability

Python can offer all of the same functionality and some extra!

Python vs. Matlab

Disadvantages:

- Matlab is a commerical product - entire computing enviornment with code, IDE
- Matlab is expensive - Between $49 - $2150 per license! Extra for toolboxes
- Matlab is proprietary - Cannot inpect source code and restrictions on sharing
- Matlab is closed - difficult to extend functionality

Advantages:

- Matlab handles arrays automatically and by design
- Lots of functionality - control design, linear algebra, optimization, ODEs etc.
- Real engineers (with funding) use it so students have to as well
- Simulink is still unmatched
- Powerful plotting capability

Python can offer all of the same functionality and some extra!

Python Philosophy - 20 aphorisms from the BDFL

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.

```
>>> import this
```

- In the face of ambiguity, refuse the temptation to guess.
- There should be one– and preferably only one –obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea – let's do more of those!

Python Philosophy - 20 aphorisms from the BDFL

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.

```
>>> import this
```

- In the face of ambiguity, refuse the temptation to guess.
- There should be one– and preferably only one –obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea – let's do more of those!