introduction

March 15, 2016

1 Introduction to Python

1.1 What is Python?

- a general-purpose, high-level programming language.
- free and open source
- easy to learn and easy to read
- portable to any OS
- saves your time rather than CPU time
- used by many scientists -> large set of modules for data processing/analysis/visualization/...

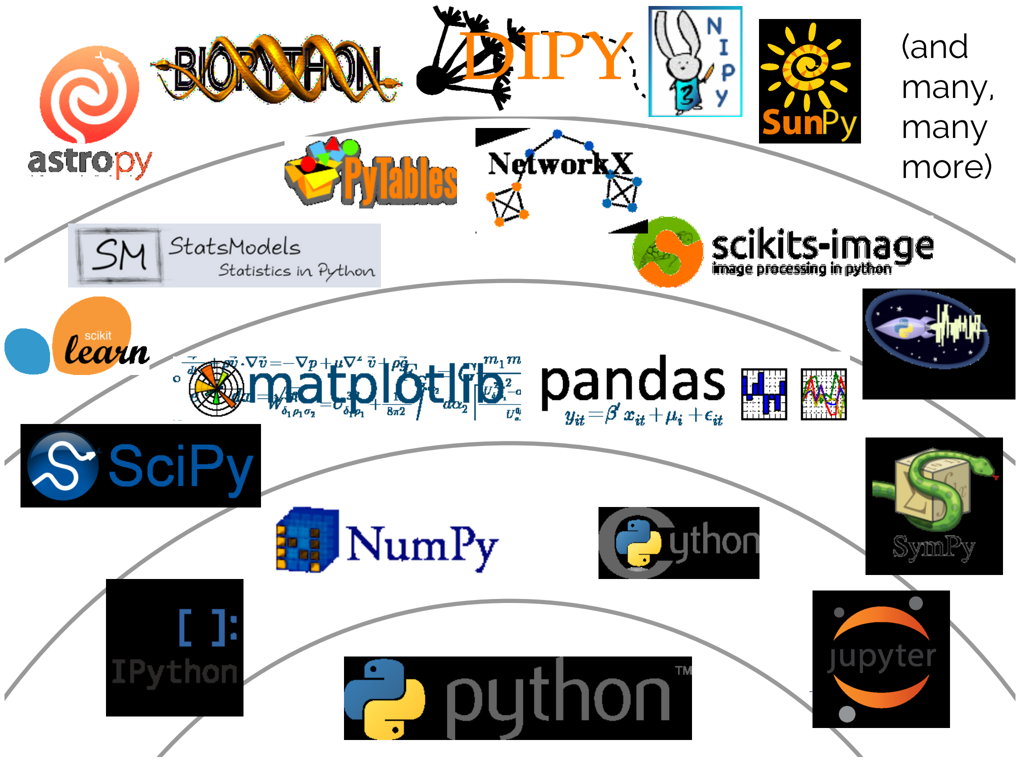
```
In [1]: import this
```

```
The Zen of Python, by Tim Peters

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.
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!
```

```
In [1]: from IPython.display import SVG
        SVG(filename='python-stack.svg')
```

```
Out[1]:
```



1.2 How to use Python

- interactive interpreter (python, ipython)
- scripts (with any text editor, there is no standard IDE)
- jupyter (aka ipython notebook)

1.3 A short intro to jupyter

1.3.1 Markdown

Markdown is a lightweight markup language.

This is a heading This is a list:

- apples
- oranges
- pears

A numbered list:

1. apples
2. oranges
3. pears

LaTeX equations

$$f(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

And this is a link where you find more about markdown syntax.

1.3.2 Tab completion

```
In [2]: import math
```

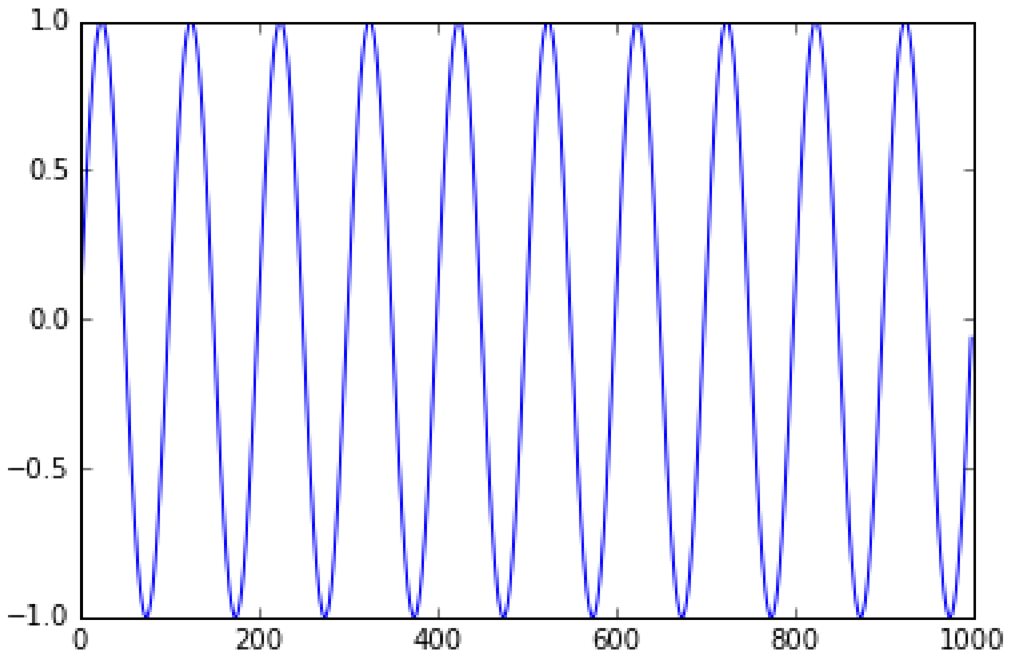
```
In [3]: #type math. and press <tab>
```

1.3.3 Inline plots & interactive widgets

```
In [30]: %matplotlib inline
         import pylab as pl
         import numpy as np
         from IPython.html.widgets import interact

         def plot(frequency):
             x = np.arange(1000)
             pl.plot(x, np.sin(2*np.pi*frequency*x/1000))
         interact(plot, frequency=(0,20,1))
```

```
Out[30]: <function __main__.plot>
```



1.3.4 jupyter is language agnostic

```
In [5]: print("jupyter's native language is python.")
```

```
jupyter's native language is python.
```

```
In [6]: %%javascript
        alert('Jupyter understands javascrip.')
```

```
<IPython.core.display.Javascript object>
```

```
In [7]: %%perl
        print "Jupyter understands perl.";
```

```
Jupyter understands perl.
```

```
In [8]: %%bash
        echo Jupyter can execute bash commands
```

```
Jupyter can execute bash commands
```

Additional kernels can be installed: Octave, Matlab, R, C++, and many more

1.3.5 Presentations

```
In [ ]:
```

1.4 Back to python: a simple example

```
In [10]: def isprime(n):
             """Determine if a given number is a prime number"""
             for x in range(2, n-1):
                 if n % x == 0:       #if n modulo x equals 0
                     return False     #then n is not a prime number
             return True
```

```
In [15]: print(isprime(13))
```

```
True
```

1.5 The Help function

```
In [16]: help(print)
```

```
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

help() can be applyed to any object, try for example:

- help(isprime)
- help(isprime(17))
- help(23)
- help("what is this?")
- help(help)

1.6 Errors messages

```
In [17]: print("Is the word 'prime' a prime number?")
         print(isprime("prime"))
```

```
Is the word 'prime' a prime number?


        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-17-4d0b7f15bf77> in <module>()
          1 print("Is the word 'prime' a prime number?")
    ----> 2 print(isprime("prime"))


        <ipython-input-10-fe20cab3cb49> in isprime(n)
          1 def isprime(n):
          2     """Determine if a given number is a prime number"""
    ----> 3     for x in range(2, n-1):
          4         if n % x == 0:       #if n modulo x equals 0
          5             return False     #then n is not a prime number


        TypeError: unsupported operand type(s) for -: 'str' and 'int'
```

```
In [ ]:
```