Introduction to Python part I

Peter Alonzi

Research Data Services

2016-09-27

Outline

- Installation
- ipython interpreter
- Arithmetic and logic (int,float,bool)
- Built in functions
- Packaged functions
- Strings and methods
- Programs – area of a rectangle
- Magic 8 ball and lists
- Population of cities and dictionary

Extra Time

If,else
For,while
Functions
Classes
Tuple - mutability

Programming vs Python

- For this workshop we are not requiring any previous knowledge of python or of computer programming. As a result we will be explaining fundamental concepts of each. Our emphasis is on the python but we will spend time on programming as well.

Brief history of Python - https://www.python.org/

- **Designed by** Guido van Rossum
- 1991
- successor to the ABC language
- Python 2.0 was released on 16 October 2000
- Python 3.0 was released on 3 December 2008

Installation of Python

- Two Notes
  - Language (python) vs language distribution (anaconda)
  - Python 2 vs Python 3
- https://www.continuum.io/downloads
  1. Scroll to your operating system
  2. Follow the instructions

ipython interpreter

- The ‘Input, process, return’ model
- To lauch open a terminal on mac or cmd on windows and type ‘ipython’

- Now you can input python code and see what happens

Arithmetic and logic (int,float,bool)

- Let’s start with basic math
  - + , - , * , / , //
- Now add some variables
  - Use ’=‘ to assign values to variables
- Now compare variables
  - Use < , > , <= , >= , … and what to do for =

- Now we tease the next section, functions.
  - Use type(…) to learn what type of object things are

Built in Functions

- Python has several functions built in to make your life easier.
  - https://docs.python.org/2/library/functions.html
- We're going to look at
  - abs(…)
  - float(…)
  - int(…)
  - str(…)

Packaged functions

- Some functions don't exist in python but do exist in packages written by upstanding members of the community. To use those packages use the word <u>import</u>.
  - `import numpy`
  - `numpy.random.randint(0,8)` # random number from 0 to 7

Strings

- Immutable (more on this later)
- Methods
- Indexable
- slicing

Indexing

```
fruit = ' a  p  p  l  e '
         0  1  2  3  4
        -5 -4 -3 -2 -1
```

```
In [19]: fruit = 'apple'

In [20]: fruit[0]
Out[20]: 'a'

In [21]: fruit[-2]
Out[21]: 'l'
```

Slices

```
Indexable[X:Y]
   ":" makes it a slice
   "X" is the start point (beginning if omitted)
   "Y" is the up to and not including point (end +1 if omitted)
```

Programming

- The rest of our workshops will focus on writing programs. We now shift from using the interpreter to using the integrated development environment (spyder).

- Translating a program from a description in English to computer code is tricky and involves skill. It may feel uncomfortable if it is new to you, that’s normal.

Program 1 – Compute the Area of a Rectangle

1. Store the length and width in variables
2. Store their product in a variable
3. Print out the area

Program 2 – The Magic 8 Ball

- Randomly select a phrase
- Print out the phrase

http://www.foundmyself.com/gallery/albums/userpics/21106/0vgnfwsz2ky5a03k9nst0j61jvvjn6.png

Lists

- Mutable, Indexable, Helper Functions
- Let’s look at some examples
  - List.pop()
  - List.append(…)
  - import random random.shuffle(list)

Program 3 – city population

- Print out the population of a city from a given list
  - New York - 8,175,133
  - Los Angeles - 3,792,621
  - Chicago - 2,695,598
  - Houston - 2,100,263

Dictionaries 101

- Mutable, helper functions, key access ***<u>(not index)</u>***
- `dictionary = {string:anything}`
- `variable = key : value`

Resources

- https://developers.google.com/edu/python/strings
- https://developers.google.com/edu/python/lists
- https://codehabitude.com/2013/12/24/python-objects-mutable-vs-immutable/
- https://developers.google.com/edu/python/dict-files
- https://docs.python.org/3/library/stdtypes.html

Programming, the usual suspects: if, while, for

- If statements

```
a = 7
if a > 5:
    print 'hello'
else:
    print 'goodbye'
```

Notice the coloring of words. My editor put that in.

Programming, the usual suspects: if, while, for

- For loops

```
for i in range(5):
    print i
```

We created the variable `i` on the fly.

Programming, the usual suspects: if, while, for

- For loops

```
fruits = ['apple','banana','cherry','kiwi']
for fruit in fruits:
    print fruit
```

Programming, the usual suspects: if, while, for

- While loops

```
i = 0
while i<5:
    print i
    i+=1
```

File input/output (I/O)

- Reading from a file

```
f = open('lyrics.txt','r')
for line in f:
    print line
f.close()
```

File input/output (I/O)

- Writing to a file

```
f = open('numbers.txt','w')
for i in range(5):
    f.write(str(i))
f.close()
```

Functions

- Extremely useful
- Used for code replication and compartmentalization
- Functions are called and they take arguments
- Functions return values

Functions

- Extremely useful
- Used for code replication and compartmentalization
- Functions are called and take arguments
- Functions return values

```
def get_kinetic_energy(m,v):
    return (1./2.)*m*v**2

mass = 2
velocity = 2
get_kinetic_energy(mass,velocity)
```

Classes

- Buzzword alert: Object Oriented Programming

Class: Jedi

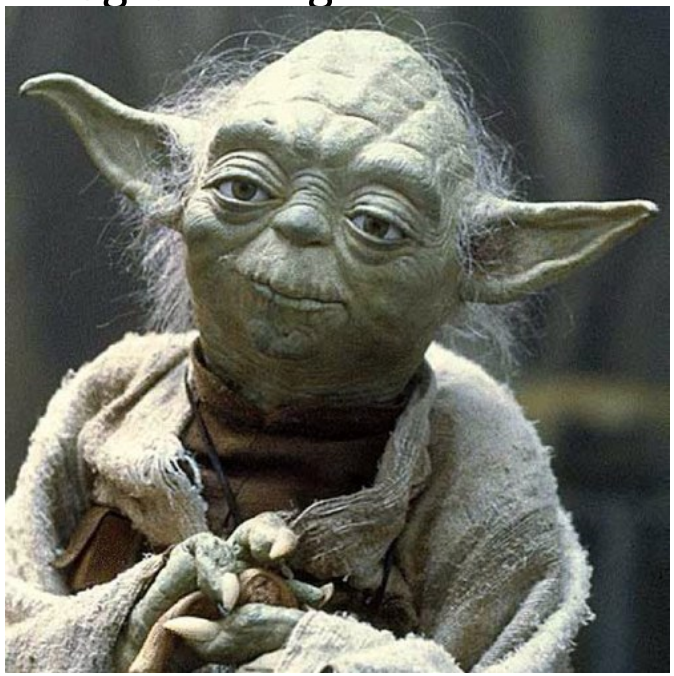
lightsabre color: blue
master: Yoda
alignment: light side

Classes

- Buzzword alert: Object Oriented Programming

Class: Jedi

lightsabre color: green
master: ???
alignment: light side



Classes

- Buzzword alert: Object Oriented Programming

Class: Jedi

lightsabre color: red
master: Obi-Wan
alignment: dark side

Classes

- Recurring set of attributes
- Different values for the attributes

Now we can look at some code … jediCode.py