Introduction to Python part II

Peter Alonzi

Research Data Services

2016-10-11

Last Time

- 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

Outline

- Dictionary type
- If,while,for
- File i/o
- User defined functions
- User defined classes
- User defined packages
- Programming Example: Let’s play blackjack

Dictionary

- Mutable, helper functions, key access ***(not index)***
- `dictionary = {string:anything}`
- `variable = { key : value }`
  - Anything means anything, it can be a list, a dict, a string, etc.

Program Exercise – 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

Populations={“NY”:8175133,”LA”:3792621,”CHI”:2695598}

So far our programs have just been straight through line by line. Let’s mix it up.

Programming, the usual suspects: if, while, for

- If statements

```
 8 a = 7
 9 if a > 5:
10     print 'hello'
11 else:
12     print 'goodbye'
```

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

Programming, the usual suspects: if, while, for

- For loops

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

We created the variable `i` on the fly.

Programming Exercise - Combine for and if

- Write a program that prints out all the even numbers from 2 to 10.

Programming, the usual suspects: if, while, for

- For loops

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

Programming, the usual suspects: if, while, for

- While loops

```
21 i = 0
22 while i<5:
23     print i
24     i+=1
```

File input/output (I/O)

- Reading from a file

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

File input/output (I/O)

- Writing to a file

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

Exercise

- Write a program that writes to file all the odd numbers from 1 to 11.
- Run the program.
- Open the file and confirm that it worked.

Functions

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

Functions

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

```
26 def get_kinetic_energy(m,v):
27     return (1./2.)*m*v**2
28
29 mass = 2
30 velocity = 2
31 get_kinetic_energy(mass,velocity)
```

Programming Exercise - How to write a function

- The first step is to determine what you want the function to do
- Then write the skeleton and make the code function
- Then worry about if it gets the right answer.

- Example: calculating GDP
  - GDP=C+G+I+X-M

Classes

- Buzzword alert: Object Oriented Programming
- What it really means is that information is compartmentalized.
- So far we have only worked with global variables, OOP introduces the idea of scope.
- Scope: “the part of a **computer** program where the binding [variable name] is valid: where the name can be used to refer to the entity.”

Classes

Class: Jedi

lightsabre color: blue

master: Yoda

alignment: light side

Classes

Class: Jedi

lightsabre color: green

master: ???

alignment: light side

Classes

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

Packages

- To turn your code into a package you have to add some boilerplate. It prevents loose code from running.
- If you have no loose code then you can skip the boilerplate.

- Then you may import your code just you would any other package.
  - Nb: you have to make sure the path is there

Programming Exercise – Building a Package

- Here we will take from our previous examples and build a package
- Then we will import that package and use it in some new code

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