Python Fundamentals

Learning CS with Python Series - Day 2

Basic Tools in a Language

- Variables - Review of day 1
  - Ints / Floats / Decimals
  - Strings
  - Advanced types: Lists, Dictionaries (Arrays)
- Functions
  - A collection of instructions with an input and an output

What is a function?

- It is a built in tool provided by the language

```
variablelength = len(variable)

open('/tmp/file','w')
```

- But the real power is in creating your own functions

```
def MyFunction (inputvarone = '', inputvartwo=True):

        privatevar = 'Some Value'

        ...

        return output
```

Why create a function

- You have a task that takes a few steps to complete
- You might use that block of code over and over again
- You might use that block of code in another program
- A core idea in programming is to ***never repeat yourself***

In most languages there are...

- Functions
- Variables
- Classes
  - Functions within a Class (method)
  - Variables within a Class (property)

Classes

```
class MyClass:

    MyVariable = ‘Hello World’

    def SayHello(self):

        return self.MyVariable
```

```
myinstance = MyClass()

mysecondinstance = MyClass()

myinstance.MyVariable = ‘Say Goodbay’

myinstance.SayHello()

mysecondinstance.SayHello()
```

What do you think these output?

Including a Class

Filename

from MyClass import *

Import everything in the file

Understanding Namespace

```
from MyClass import *

myinstance = MyClass()

mysecondinstance =
MyClass()

myinstance.MyVariable = 'Say
Goodbay'

myinstance.SayHello()

mysecondinstance.SayHello()
```

```
import MyClass

myinstance =
MyClass.MyClass()

mysecondinstance =
MyClass.MyClass()

myinstance.MyVariable = 'Say
Goodbay'

myinstance.SayHello()

mysecondinstance.SayHello()
```

Using others' classes

```
import math
...
b = math.sqrt(a)
```

```
from math import sqrt
...
b = sqrt(a)
```

Making Decisions

```
if x >0:
    print ‘x is non-negative’
elif x > -1 and x < -0.5:
    print ‘x is between -1 and -0.5’
else:
    print ‘x is ‘ + x
```

Comparisons

- ==
- !=
- <=
- >=
- <
- >

Lists

- A list is a collection of more primitive data types
- They are natively supported in python (no import)
- In other languages this called an array

Lists

```
myList = [‘One’,‘Two’,
‘Three’]
print myList[1]
```

```
myList = []
myList.append(‘One’)
myList.append(‘Two’)
myList.append(‘Three’)
print myList[1]
```

Dictionaries

- Sort of like lists in that they are a collection of more primitive datatypes
- But it is about naming items

Dictionaries

```
myDict = {‘one’:1,
‘two’:2, ‘three’:3}

print myDict[‘one’]
```

```
myDict = {}

myDict[‘one’] = 1

myDict[‘two’] = 2

myDict[‘three’] = 3

print myDict[‘one’]
```

Loops

- Two types of loops in Python
  - For
    - Loops over a range
  - While
    - Continues to loop until a condition is not true

For Loop

```
a =
['One','Two','Three']
for x in a:
    ...
    print x
```

```
for x in range(3):
    ...
    print x
```

While Loop

```
while myVariable != True:
    ...
    if x > 1:
        myVariable = True
```

Example - Reading a CSV

```
import csv

...

f = open(filename, 'rb')
reader = csv.DictReader(f)
for item in reader:
    print item[‘name’]
f.close()
```