Allah says

عَلَّمَهُ الْبَيَانَ o

He has taught him speech (and intelligence).

Al-Qur'an, 055.004 (Ar-Rahman)

Lecture-1.2
Control Flow in Python

Fayyaz ul Amir Afsar Minhas, Ph.D.

fayyazafsar@gmail.com
https://piazza.com/pieas.edu.pk/summer2015/python/home

Department of Computer and Information Sciences
Pakistan Institute of Engineering and Applied Sciences (PIEAS)
P.O. Nilore, Islamabad.

Python myths

- **But Python is slow…**
  - **Yes, because it is interpreted**
  - **But you can also, very easily, use packages such as PyPy and Numba to do Just-In-Time Compilation to compile the code and make it fast**
  - **It also integrates with C (through Cython or SWIG)**

http://blog.dhananjaynene.com/2008/07/performance-comparison-c-java-python-ruby-jython-jruby-groovy/

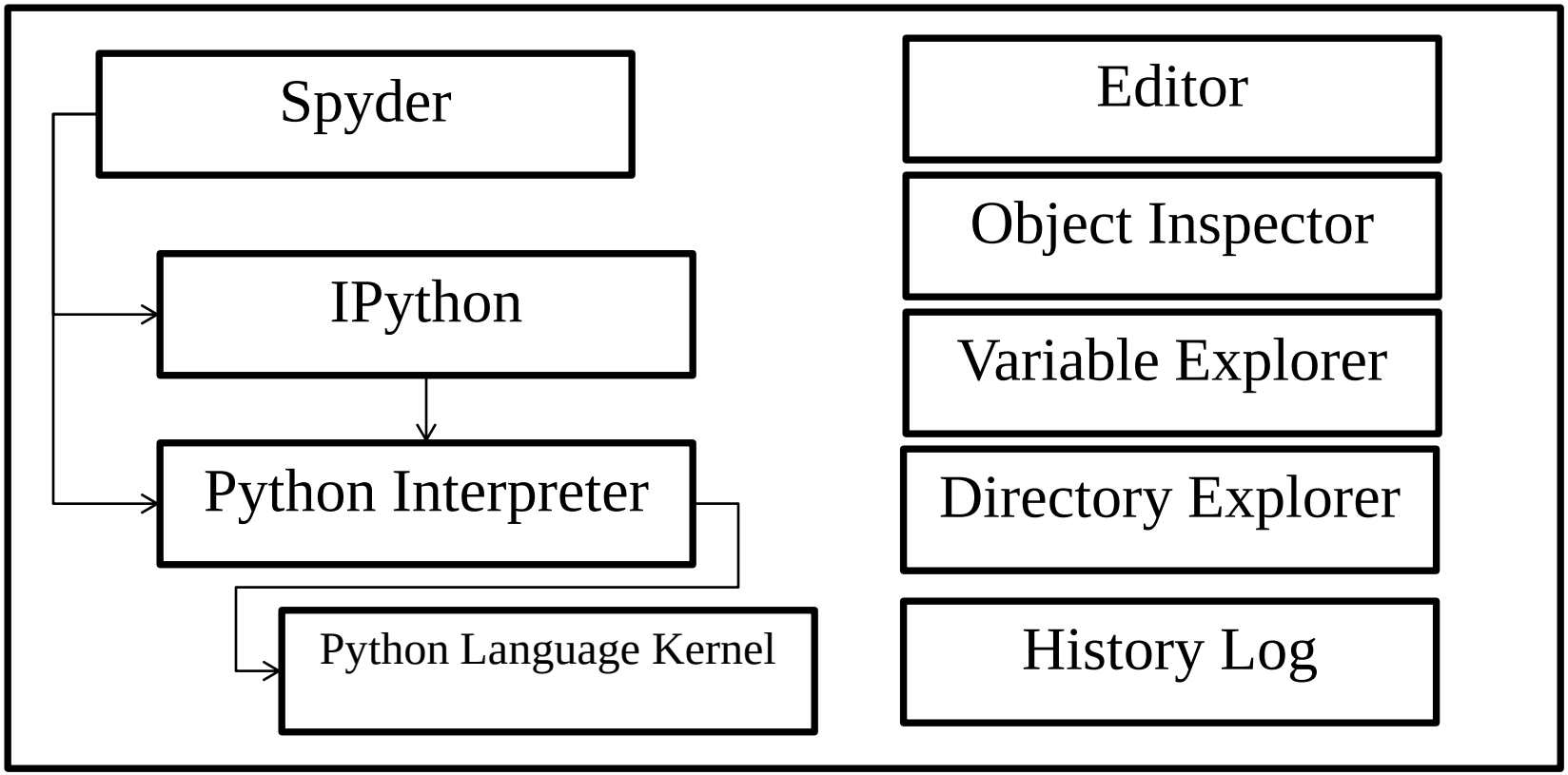
Language	Version	Lines of Code	Time per iteration (microseconds)
Java	Sun JDK 1.6.0.03	86	1.6
C++	4.1.3 20070929 (prerelease) (Ubuntu 4.1.2-16ubuntu2) Compiled with optimisation -O3	86	3
	gcc version 4.2.3 (Ubuntu 4.2.3-2ubuntu7) Compiled with optimisation -O3	124	~0
Python	2.5.1		192
	2.5.1 with psyco	41	33

Another look at Python IDE

- We are using Python (x,y)
  - Which is a Python IDE
  - Contains tools for writing and executing code (Spyder)
  - Tools for building GUI (QtDesigner)
  - Notebooks
  - Debugging (PDB)
  - …
- However, for this course you will be using Spyder primarily
- Now Spyder has a file editor, a variable explorer (you can save variables), a directory explorer, an object inspector (to view help by pressing ctrl+I)
- Consoles
  - Python Console: Python Interpreter
  - Ipython Console: Interactive Python , Enhanced capabilities – “magic” functions
- The consoles have intellisense
- Unlike Matlab, you can have multiple consoles running at the same time
- You can also run consoles from your operating system directly

Concept Diagram

Python (x,y)



Assignment-1

- **Finding distance between two cities**

Assignment-1 Solution

```
from math import sin,cos,atan2,sqrt,radians

phi_1, lambda_1 = 33.6547, 73.2500
phi_2, lambda_2 = -22.9083, -43.1964

d_phi = phi_2-phi_1
d_lambda = lambda_2-lambda_1

phi_1,phi_2 = radians(phi_1),radians(phi_2)
lambda_1,lambda_2 = radians(lambda_1),radians(lambda_2)
d_phi,d_lambda = radians(d_phi),radians(d_lambda)

a = sin(d_phi/2.0)**2+cos(phi_1)*cos(phi_2)*(sin(d_lambda/2.0)**2)
B = 2*atan2(sqrt(a),sqrt(1-a))
R = 6371
D = R*b

print "Distance is",d
```

Modules

Syntactic Sugar

Imports

```
from math import sin,cos,atan2,sqrt,radians
r = radians(180)

import math
r = math.radians(180)

from math import radians as rad
r = rad(180)

from math import *
```

Important concept

- Stuff defined or declared in a file can be used in console after the file is run for the first time
- Stuff available in the console can be used in the file as well

Functions

- If you want to write a big program, you may want to reuse your existing code
- One way is to create functions
- For example
  - In a program that will compute distance of one city from all other cities you can easily call our function with different parameters which will depend on the cities

Python Functions

- Python allows the creation of functions
  - Allow multiple inputs of different types
    - Even infinite or different number
  - Allows multiple outputs of different types

Python functions

```
def getDistance(phi_1,lambda_1,phi_2,lambda_2):
    """
    This function computes the distance between two points on the surface of
    the earth.
    Input arguments:
        phi_1: latitude of the first point
        lambda_1: longitude of the first point
        phi_2: latitude of the second point
        lambda_2: longitude of the second point
        All the above values are in degrees.
    Return value:
        Distance in kilometers between the two points
    """
    phi_1,phi_2=radians(phi_1),radians(phi_2)
    lambda_1,lambda_2=radians(lambda_1),radians(lambda_2)

    d_phi=phi_2-phi_1
    d_lambda=lambda_2-lambda_1

    a=sin(d_phi/2.0)**2+cos(phi_1)*cos(phi_2)*sin(d_lambda/2.0)**2
    b=2*atan2(sqrt(a),sqrt(1-a))
    R=6371
    d=R*b
    return d
```

Input Arguments

Note colon

Help string AKA Doc string

Notice indentation

Return Value

Task

- Create a file “distance_script.py”
- Copy paste your code from yesterday and create a function called getDistance as on the previous slide

- This function can be called as
  - `getDistance(phi_1,lambda_1,phi_2,lambda_2)`

- Make a call to the function
- Run the file
- Do not copy from this presentation. Do it for your own code

distance_script.py

```
from math import sin,cos,atan2,sqrt,radians

phi_1, lambda_1=33.6547, 73.2500
phi_2, lambda_2=-22.9083, -43.1964

def getDistance(phi_1,lambda_1,phi_2,lambda_2):
    """
    This function computes the distance between two points on the surface of
    the earth.
    Input arguments:
        phi_1: latitude of the first point
        lambda_1: longitude of the first point
        phi_2: latitude of the second point
        lambda_2: longitude of the second point
        All the above values are in degrees.
    Return value:
        Distance in kilometers between the two points
    """
    phi_1,phi_2=radians(phi_1),radians(phi_2)
    lambda_1,lambda_2=radians(lambda_1),radians(lambda_2)

    d_phi=phi_2-phi_1
    d_lambda=lambda_2-lambda_1

    a=sin(d_phi/2.0)**2+cos(phi_1)*cos(phi_2)*sin(d_lambda/2.0)**2
    b=2*atan2(sqrt(a),sqrt(1-a))
    R=6371
    d=R*b
    return d

print "Distance is", getDistance(phi_1,lambda_1,phi_2,lambda_2)
```

Things to remember

- Arguments
  - Python functions can take multiple arguments
  - Even a variable number of arguments (*args, **kwargs)
  - Setting default values

```
def getDistance(phi_1,lambda_1,phi_2=33.6547,lambda_2=73.2500):
            ...
print "Distance is", getDistance(phi_2,lambda_2)
```

  - Named argument calling

```
getDistance(phi_2=33.6547,lambda_2=73.2500,phi_1=-22.9083,lambda_1=-43.1964)
```

- Return values
  - Can return multiple values

```
return d, d**2
```

- Help
  - Can add description of the function as comments
  - Use: <func_name>? Or help(func_name)
- Scope: `locals()` and `globals()`

Developing your first module

- Create a file ‘distance.py’
- Copy the import statement and the getDistance Function into it
- Create another file called ‘distance_script2.py’ in the same directory
- Write ‘from distance import getDistance’ and then you can call the function

Task

- Change the function to return both the distance and the square of the distance

Concepts on control flow in Python

- Python is an interpreted language
- It executes the code one line at a time
- When a Python code encounters an import statement it compiles the code of the required module then
  - If a module file is named “distance.py”, its compiled file will be “distance.pyc” which contains the byte code
  - The byte code is then executed by the Python Virtual Machine

https://en.wikipedia.org/wiki/Bytecode

Concepts on control flow in Python

- If you change a module file but do not “compile” a new .pyc file prior to importing it, the code will not reflect the changes
  - The solution is “reloading”
  - Recompile the code every time it is imported
- Not a problem when not using the console

Problems in Reloading modules?

- If you make a change to the module file and run the script it does not reflect these changes unless it is 'reloaded'

- Easy to reload using Ipython Magic functions

- `%load_ext autoreload`
- `%autoreload 2`

Adding a module tester to a module

```
from math import sin,cos,atan2,sqrt,radians

def getDistance(phi_1,lambda_1,phi_2=33.6547,lambda_2=73.2500):
    """
    This function computes the distance between two points on the surface of
    the earth.
    Input arguments:
        phi_1: latitude of the first point
        lambda_1: longitude of the first point
        phi_2: latitude of the second point
        lambda_2: longitude of the second point
        All the above values are in degrees. If phi_2 and lambda_2 are not
                    provided, the function uses the default coordinates of Nilore
                    as point 2.
    Return value:
        Distance in kilometers between the two points
    """
    phi_1,phi_2=radians(phi_1),radians(phi_2)
    lambda_1,lambda_2=radians(lambda_1),radians(lambda_2)
    d_phi=phi_2-phi_1
    d_lambda=lambda_2-lambda_1
    a=sin(d_phi/2.0)**2+cos(phi_1)*cos(phi_2)*sin(d_lambda/2.0)**2
    b=2*atan2(sqrt(a),sqrt(1-a))
    R=6371
    d=R*b
    return d

if __name__=='__main__':
    print "Distance is", getDistance(phi_2=33.6547,lambda_2=73.2500,phi_1=-22.9083,lambda_1=-43.1964)
```

__name__

- Add the following line to the module and the script
- print "__name__ in", __file__, "is",__name__
-

- __name__ in distance.py is distance
- __name__ in distance_script2.py is __main__
- Distance is 13772.847015

Control flow statements

```
if condition:
        # Perform some actions
        print "Condition is True"
elif condition != True:
        # Perform some other actions
        print "Condition is not True"
else:
        # Perform default or fall-through actions
        print "Anomaly: Condition is neither True nor False"
```

Control flow statements

- **While loop**

```
while condition:
    do-something
```

- **Example**

```
a = 10
while a>0:
    print a
    a = a - 1
```

Python lists

- A list is a data structure in Python
- Required here for understanding loops

Control flow statements

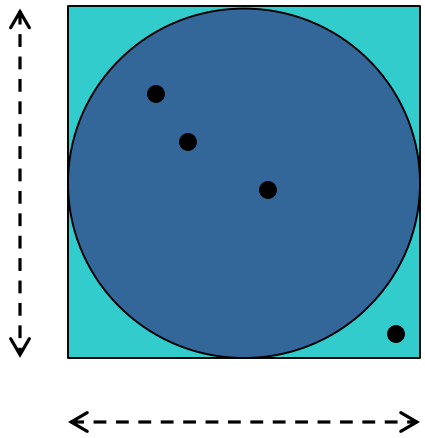
- The `range` function
- Returns a 'list'
  - More on lists later
- The 'for' loop iterates over a 'list': any list

```
print range(10)

for a in range(10):
        print a

for a in ['Mary', 'had', 10, 'little', 'lambs']:
        print a
```

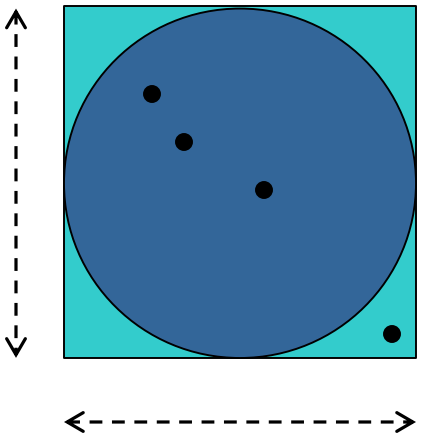
Estimating the value of Pi



- Montecarlo methods
- Can be used to integrate any function

Finding the value of π

- $r = 1.0$
- $A_{square} = (2r)^2 = 4.0$
- $A_{circle} = \pi r^2 = \pi$
- $\frac{A_{circle}}{A_{square}} = \frac{\pi}{4.0}$
- $\pi = 4\left(\frac{A_{circle}}{A_{square}}\right)$



- If we throw darts at random at an object, the percentage of darts that land inside it is proportional to its area
- Throw darts at random within the square and calculate the percentage of darts that land inside the circle. This gives an approximate value of the ratio
- Use this to calculate π

Throwing darts

- Generate coordinates using “random” module
  - x=random.random()
  - y=random.random()
- How to see if (x,y) is inside the circle
  - If $x^2 + y^2 < 1.0$
- Throw a large number of darts and see what proportion lands inside

Python: Do Not Forget

- Everything is an object
  - dir (<obj>)
  - type(sqrt)

Challenge

- Make a python program that takes as input any function and computes the area under it using Monte Carlo Simulation
- Example:

```
def circle(x,y):
      r = 10
      return x**2+y**2 <=r**2

def findArea(func, trials = 10000):
      # Implementation
      return area

Area = findArea(circle)
```

Summary of topics covered today

Want to read?

- Control Flow
  - https://docs.python.org/3/tutorial/controlflow.html
- Books
  - Do exercises in the end of the books
  - Deitel!

End of Lecture-2

"Python's a drop-in replacement for BASIC in the sense that Optimus Prime is a drop-in replacement for a truck."

- Cory Dodt

Next

- **Handling files and strings**