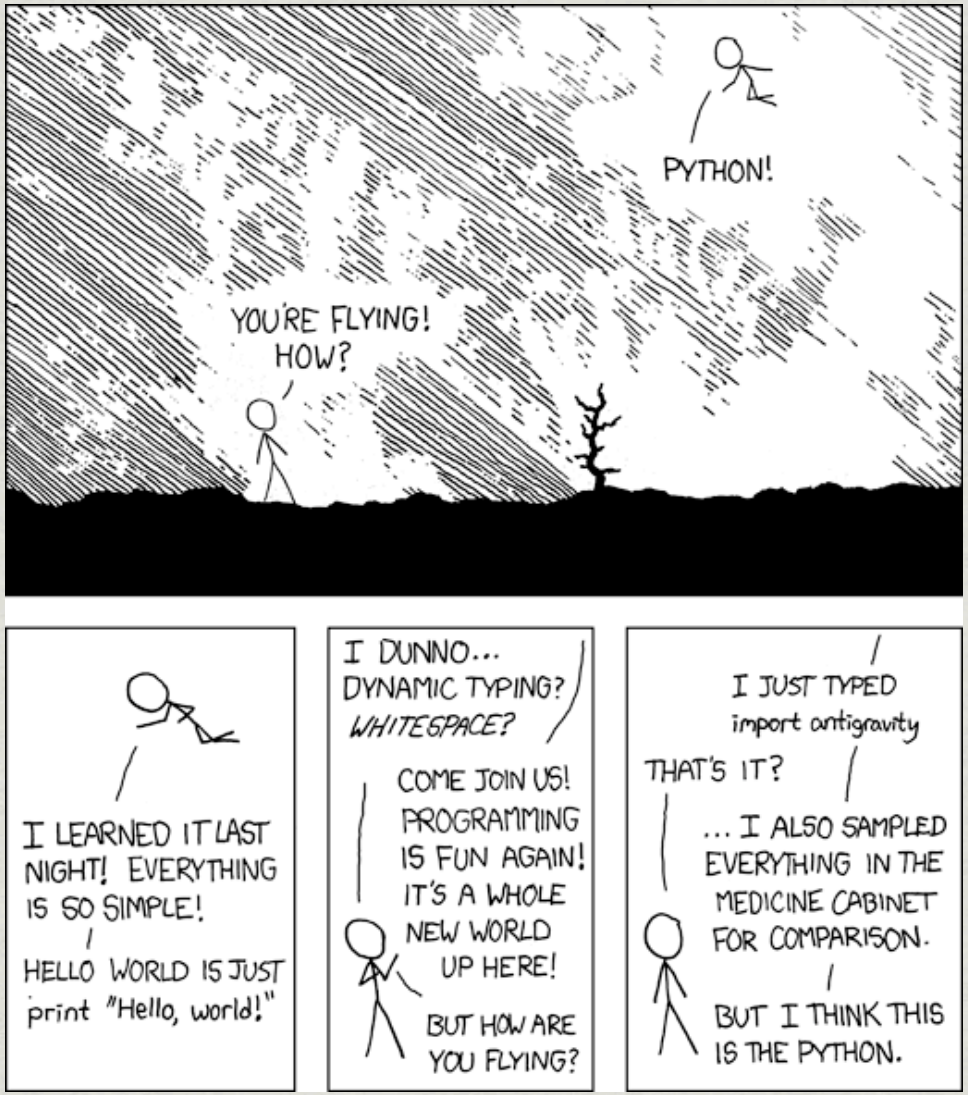
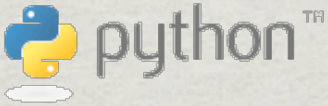
Python for Astronomers

Lauren Palladino
February 22, 2011



HTTP://XKCD.COM/353/



Today's Topics

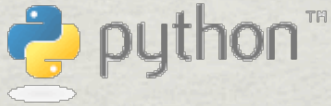
- Review
- Scripting
- Functions
- Loops
- Reading/Writing Files & Formatting Output
- Libraries
  - math, string, os, PyFits

python™

Review

- Numeric input: <variable>=input<prompt>
- String input: <variable>=raw_input<prompt>
- Lists, mutable, mixed-type: [<item1>,<item2>,...]
- Tuples, immutable list: (<item1>,<item2>,...)
- Dictionaries, mutable key-value pairs: {<key>:<value>,...}
- Indexing: <variable>[i]
- Slicing: <variable>[begin:end]
- If statement: if <condition>:

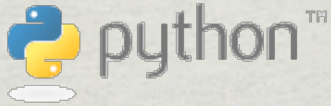
  <statements>



Scripts

To run your Python script:

- ./scriptname.py
  - needs **#!/usr/bin/env python** at top of script
- python scriptname.py

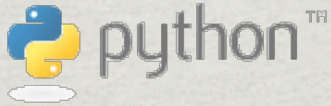


Scripts

FROM LAST TIME:

```
>>> spam = raw_input("2 letter word for (and rhymes with) you & me: ")
>>> guido = [spam, 'eggs']
>>> if guido[1] == 'eggs':
        guido[1] = 'are the'
>>> x = guido
>>> sausage = raw_input("Like a time of day, but begins with 'k': ")
>>> y = sausage + 's'
>>> bacon = raw_input("Fill in the blank in this children's game: Simon ____: ")
>>> z = ['who', bacon[0:-1]]
>>> print x, y, z, "Ni!"
```

~palladl2/python/example.py



Scripts

```
def main():

    spam = raw_input('2 letter word for (and rhymes with) you & me: ')
    guido = [spam, 'eggs']

    if guido[1] == 'eggs':
        guido[1] = 'are the'

    x = guido
    sausage = raw_input("Like a time of day, but begins with 'k': ")
    y = sausage + 's'
    bacon = raw_input("Fill in the blank in this children's game: Simon ____: ")
    z = ['who', bacon[0:-1]]

    print x, y, z, 'Ni!'


main()
```

palladl2/python/example.py

python™

Functions

- Define a function with def <function>():
  - called a header line, every statement "belonging" to a header line must be indented
- call a function with <function>()

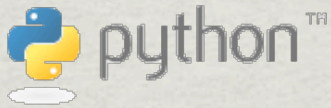
For example:

```
def spam():
    print "This is a function."

spam()
```

Write your own:

Hello, World!



Functions

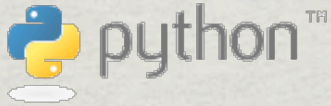
- Can pass arguments between functions.
- def <function>(<argument1>,<argument2>):

For example

```
def greet(person):
    print “Hello”, person

def main():
    name=raw_input(“What is your name?: “)
    greet(name)

main()
```



Functions

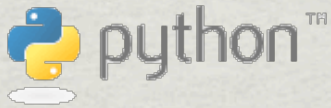
- Can return values.

For example

```
def square(x):
    z = x**2
    return z

def main():
    number = input("Pick a number, any number: ")
    y = square(number)
    print "The square of your number is: ", y

main()
```



Loops

FOR LOOP

- for <variable> in <sequence>:

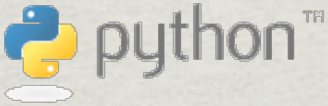
  <body>

For example

```
word = 'spam'
for letter in word:
    print letter

for i in [1,3,5,7,9]:
    print i*i
```

```
for i in range(10):
    x = 3.9 * i * (1 - i)
    print x
```



Loops

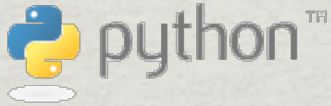
WHILE LOOP

- while <condition>:

  <body>

For example

```
i = 0
while i < 10:
    print i
    i = i + 1
```



Reading Files

- <filevar>.read() Returns the entire remaining contents of the file as a single string.
- <filevar>.readline() Returns the the next line of the file, up to and including the next newline character (\n).
- <filevar>.readlines() Returns a list of the remaining lines in the file. Each item in the list is a line including the newline character.

Reading Files

~palladl2/python/sample.dat

For example

filevar

```
infile = open('filename.dat', 'r')
data = infile.read()
print data
infile.close()
```

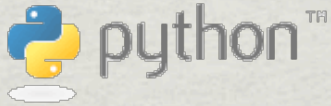
how to open a file

read the file

how to close the file

Print the first 6 lines of sample.dat:

```
infile = open('sample.dat', 'r')
for i in range(6):
    line = infile.readline()
    print line[:-1]
infile.close()
```



Writing Files

- <filevar>.write(<string>)

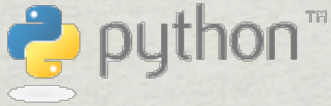
For example

write to the file

```
outfile = open('out.dat', 'w')
outfile.write("Spam and eggs\n")
outfile.write("Spam, bacon, eggs, and spam\n")
outfile.close()
```

Watch out... 'w' will overwrite the file each time it is reopened.

Can use 'a' instead to append to the file.



Formatting

- <format specifier> % (<values>)
- format specifier: %<width>.<precision><datatype>

Possible data types:

- decimal
- float
- string

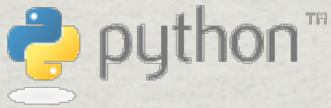
width: How many places to use in display. If not enough, Python will expand to fit the value. Width of '0' means "use as much space as needed."

precision: with floats, indicates number of digits after the decimal.

width and *precision* are optional

For example

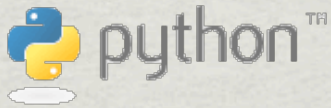
```
print "%.2d %s %0.4f you" % (1, 'spam', 4)
```



Libraries

ADDITIONAL FUNCTIONS AVAILABLE TO PYTHON

- Math- ‘advanced’ math functions
- String- string manipulation functions
- Os- operating system interface
- PyFits- reading, writing, manipulating FITS files
- etc.



Libraries

ADDITIONAL FUNCTIONS AVAILABLE TO PYTHON

How to use a library:

- import <library>

  import math

  <library>.<function>()

  math.sqrt(x)

- from <library> import <function1,function2,...>

  <function1>()

  sqrt(x)

- from <library> import *

Libraries

MATH

some math functions:

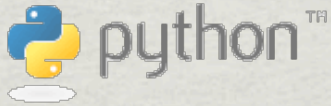
- constants- math.e, math.pi
- trig- math.sin(), math.sinh(), math.asin(), etc.
- logarithmic- math.log(), math.log10()
- power- math.exp(), math.sqrt()
- angular conversion- math.degrees(), math.radians()

Libraries

STRING

some string functions:

- capitalize(s)- copy of string s with 1st character capitalized
- capwords(s)- capitalize each word in s
- count(s, sub)- counts number of occurrences of sub in s
- find(s, sub)- find 1st position of sub in s
- replace(s, oldsub, newsub)- replace all occurrences of oldsub with newsub
- split(s)- split s into a list of substrings



Libraries

os

Example using os

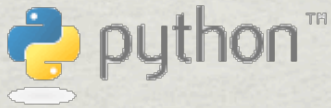
```
import os

def main():

  #use this to capture stdout
  out = os.popen("pwd","r")
  while 1:
    line = out.readline()
    if not line: break
    print line

main()
```

Courtesy of Daniel Sissom



Libraries

PYFITS

Example using PyFits

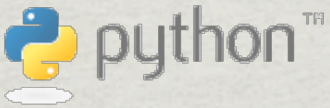
```
import pyfits

def convertfile():
        fitfile = pyfits.open("filename.fit")
        infile = fitfile[1].data
        i = 0
        line = infile[i]
        while line != "":
                outfile = open("textfile.dat", "a")
                outfile.write(str(line) + "\n")
                outfile.close()
                i = i + 1
                line = infile[i]
        fitfile.close()

def main():
        convertfile()

main()
```

0 for header
1 for data



```
import math
import string

def deredmags(mag_u, mag_g, mag_r, mag_i, mag_z, ext_u, ext_g, ext_r, ext_i, ext_z):
        dered_u = float(mag_u) - float(ext_u)
        dered_g = float(mag_g) - float(ext_g)
        dered_r = float(mag_r) - float(ext_r)
        dered_i = float(mag_i) - float(ext_i)
        dered_z = float(mag_z) - float(ext_z)
        return dered_u, dered_g, dered_r, dered_i, dered_z

def calccolors(dered_u, dered_g, dered_r, dered_i, dered_z):
        umg = dered_u - dered_g
        gmr = dered_g - dered_r
        rmi = dered_r - dered_i
        imz = dered_i - dered_z
        return umg, gmr, rmi, imz

def calcerrors(err_u, err_g, err_r, err_i, err_z):
        umg_e = math.sqrt(float(err_u)**2 + float(err_g)**2)
        gmr_e = math.sqrt(float(err_g)**2 + float(err_r)**2)
        rmi_e = math.sqrt(float(err_r)**2 + float(err_i)**2)
        imz_e = math.sqrt(float(err_i)**2 + float(err_z)**2)
        return umg_e, gmr_e, rmi_e, imz_e

def main():
        infile = open("data.dat", "r")
        line = infile.readline()
        while line != "":
                a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z = string.split(line, ",")
                dered_u, dered_g, dered_r, dered_i, dered_z = deredmags(i,j,k,l,m,s,t,u,v,w)
                umg, gmr, rmi, imz = calccolors(dered_u, dered_g, dered_r, dered_i, dered_z)
                umg_e, gmr_e, rmi_e, imz_e = calcerrors(n,o,p,q,r)
                outfile = open("colorsanderrors.dat", "a")
                outfile.write(str(umg)+" "+str(umg_e)+" "+str(gmr)+" "+str(gmr_e)+" "+str(rmi)+" "+str
                (rmi_e)+" "+str(imz)+" "+str(imz_e)+"\n")
                outfile.close()
                line = infile.readline()
        infile.close()

main()
```

~palladl2/python/data.dat

Useful References

- Python Programming: An Introduction to Computer Science, by John Zelle
- Learning Python, by Mark Lutz
- http://www.python.org/
- http://www.stsci.edu/resources/software_hardware/pyfits/