Python Minicourse

L G Brunnet and Samoel R M da Silva

Instituto de Física - Ufrgs

March, 2019

Python Software Foundation

https://docs.python.org/

Variable type is defined on its attribution

```
text,a,b,c="Hello World", 1, 2, 2.3
```

```
print(type(text), type(a),type(b),type(c))# python3
```

Lists

Lists may be composed by a mix of integers, floats, strings or objects.

```
lista=[1,2,3,"texto",2.]

print(lista, type(lista[0]))

lista2 = lista[3]

print(lista2+"\t"+str(len(lista[3])))

lista3=lista[0:2]

print(lista3)
```

While

Indentation is essential!

```
a,b=0,1
while (b<100) and (a<50):
        print(b)
        a,b=b,a+b
```

```
1
1
2
3
5
```

For, if, else, enumerate

```
words = [ "cat", "window", "defenestrar" ]
for i in words:
        print(i)
for i,w in enumerate(words):
        print(i,w)
        if w=="cat":
                words[i]="dog"
        elif w=="window":
                words[i]="toto"

print(words)
#How to get help on Python?!
help(enumerate)
```

For, else, range

```
for i in range(100,10,-3):
        print(i)

# Uso de else associado com for!

for n in range(2,100):
        for x in range(2,n):
                if(n%x)==0:
                        break
        else:
                print( n, "is a prime number")
```

For, continue

```
for n in range(2,30):
        if n%2==0:
                print(n, "eh par")
                continue
        print(n, "e impar")
```

Functions

```
def f(name):
        print("hello")
        return len(name), name[0], 1

x,a,y=f("Willian of Ocham")
print(x,a,y)
```

Dictionary

```
tel = { ″leon″:7251, ″prado″:7255, ″TI″:6559}

tel[″TI″]

del tel[″TI″]

tel[″if″]=7111

for k,v in tel.items():
        print(k,v)

type(tel)

help(tel)
```

Dictionary of dictionary

```
bunge = {"idade":98, "nome":"Mario Bunge"}
francis = {'idade':65+, 'nome':'Francis Bacon'}
leonardo = {'idade':67+, 'nome':'Leonardo di ser Piero da Vinci'}

dicdic = {'bunge':bunge, 'francis':francis, 'da vinci':leonardo}

user = 'da vinci'
dicdic[user]['nome']
dicdic[user]['idade']
```

Geometric functions, exponentials, log

```
import math

print(math.pi)

import math as m

print(m.pi)

help(m)
```

Data input from console

```
import sys
a = sys.argv[0:4] #read 3 values of console input

print(a)

print(a[0],a[1]) # Note: a[0] → python program name
```

Data input from file

```
f=open("Initial_DivTime.csv")

print(f.readline()) #read first line
print(f.readline()) #read second line

f.close() #close file
```

Reading a whole file

```
f=open(″Initial_DivTime.csv″)

whole-file-unsplitted-string=f.read()

line-separator-free-string=\# line continues below
whole-unsplitted-string.replace(′\n′,″)

list-string=line-separator-free-string.split()

# If the original file is composed only of float numbers
list-float=list(map(float,list-string))
```

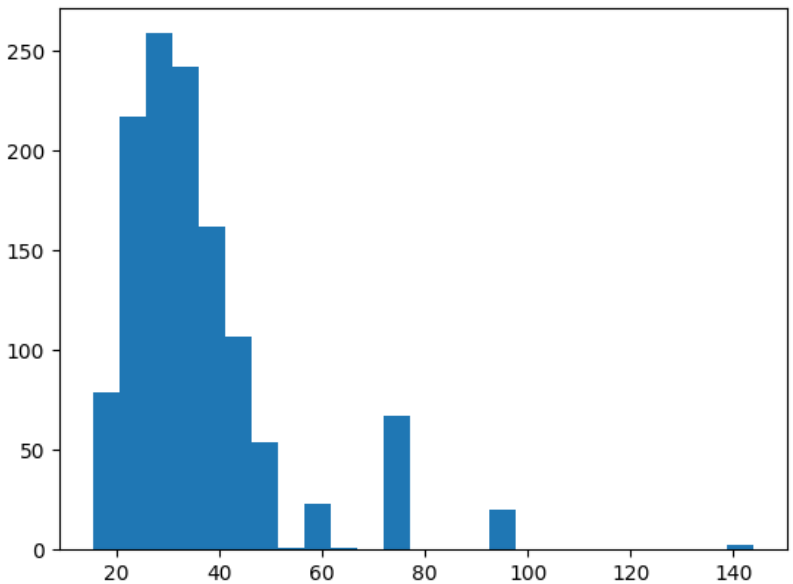
Reading in detail

- Here we open and read a file line by line
- We avoid blank lines and break if end of file
- First string of the lines is converted to float and goes to x

```
f=open("Initial_DivTime.csv") # Path+filename to open
x=[]
while 1:
line = f.readline()
if not line:
. . .     break #EOF
if line.replace( '\r',"") == '\n' : # skip a blank line
. . .     while line.replace( '\r' , '' ) == '\n' :
. . .         line = fn.readline()
else:
. . .     x.append(float(fn.readline.split()[0]))
```

Histograma

```
import matplotlib.pyplot as plt
b=max(f)
a=min(f)
h=plt.hist(f,bins=int(b-a)/5) # 5 is the bin width
h
plt.show()
```



Treating exceptions

```
a = input("Quero dividir:")
b = input("por:")
try:
        c=float(a)/float(b)
        print(c)
except ValueError:
        print("Opa! Nao posso fazer isto!")
except ZeroDivisionError:
        print("Opa! Nao posso dividir por zero!")
```

Lambda Function (Anonymous)

```
def f(x): return x**2
f(7)
49

g= lambda x: x**2
print(g(7))
49

g = lambda x,y : x**2+y**2
g(2,3)
13
```

No need of “return” or assigning to a variable!

Examples of Lambda Functions and Combinations with Normal Functions

```
def make_inc(n): return lambda x: x + n

f = make_inc(3)
g = make_inc(7)
print(f(20), g(30))
23 37
```

Examples of Lambda Functions

```
zoo = [2, 18, 9, 22, 17, 24, 8, 12, 27]

out=list(filter(lambda x: x % 3 == 0, zoo)))
print(out) [18, 9, 24, 12, 27]

print(list(map(lambda x: x * 2 + 10, zoo)))
 [14, 46, 28, 54, 44, 58, 26, 34, 64]

import functools help(functools.reduce)
print(functools.reduce(lambda x, y: x + y, zoo))
139
```

Classes, Objects, what a hell is that?!

```
class MyClass:
        i = 2
        j = 3
        def doSomething(self):
                self.k = self.i+self.j

MyObject=MyClass()
help(MyObject)

print MyObject.i
print MyObject.j

MyObject.doSomething()

print MyObject.k
```

Atributing values for object variables

```
class Complex:
        def __init__(self, realpart, imagpart):
                self.r = realpart
                self.i = imagpart

z = Complex(3.0, -4.5)
help(z)
z.r, z.i
```

Random walker class example

```
import random as p

class part():
        x=0.0
        def mov(zz):
                zz.x=zz.x+(p.random()-0.5)

#Objects as instances of that class

ob1=part()
ob2=part()

#Asking their positions
ob1.x
ob2.x
```

Making them walk

```
ob1.mov()
ob2.mov()

#Asking their positions

ob1.x
ob2.x
```

Creating a list of objects

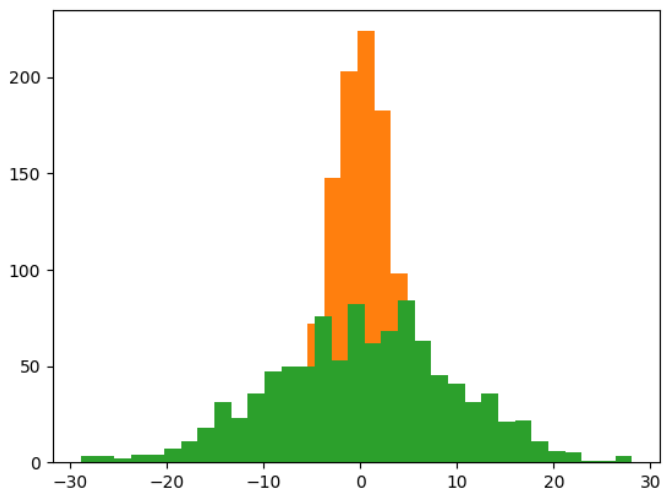
```
ob = list(part() for i in range(1000))

#Moving 1000 random walkers for 100 steps (simple way)

for i in range(100):
        for j in range(1000):
                ob[j].mov()

print ob[10].x #checking if 10 has walked
```

Histograms

```
import matplotlib.pyplot as plt

a = list(map(lambda y:y.x, ob))
b = plt.hist(a)
```

Moving 1000 steps more

```
for i in range(10000): # Smarter way
    list(map(lambda x:x.mov(),ob))

c = list(map(lambda z:z.x, ob))
d = plt.hist(c,bins=33) #new histogram

plt.show()
```



A more complex example: Random walkers with spatial exclusion

```
import random as p
class walker():
    def __init__(self,ident,x):
        self.x=x
        self.newx=x
        self.ident=ident
    def mov(self):
        self.newx += int(10*(p.random()-0.5))
        for i in wk :
            if i.ident != self.ident :
                if i.x == self.newx :
                    break
        else:
            self.x=self.newx
```

Create walker instances and move them

```
# Create a list with 10 walkers with index from 0 to 9
# and initial position from 0 to 18
wk = list(walker(i,2*i) for i in range(10))

# Print their initial positions
print(list(map(lambda i:i.x, wk)))
output: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]

# Make them move 100 times
for i in range(100) :
    list(map(lambda i:i.mov(), wk))

# Print their final positions
print(list(map(lambda i:i.x, wk)))
output_example: [3, -4, 8, 5, -16, 23, 6, 1, 9, 10]
```