

Python Minicourse

L G Brunnet

Instituto de Física - Ufrgs

October, 2017

- 1 Variables and Lists
- 2 Flow control
- 3 Functions
- 4 Dictionary
- 5 Input
- 6 Classes and Objects

Python Software Foundation

<https://docs.python.org/>

The type of variable is defined on its attribution

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

```
>>>print text, type(a),type(b),type(c)
```

```
>>>print(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]
```

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
```

Input is always string

```
>>>f=open("Initial_DivTime.csv") # reopen
```

```
>>>f
```

```
>>>f=f.read()
```

```
>>>type(f)
```

```
>>>f
```

```
>>>help(str)
```

```
>>>f=f.replace('\ n','')
```

```
>>>f
```

```
>>>f=f.split()
```

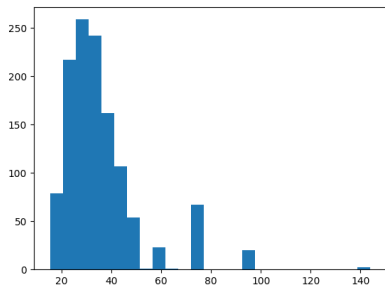
```
>>>f
```

```
>>>f=map(float,f)
```

```
>>>f
```


Histograma

```
>>>import matplotlib.pyplot as m  
>>>max(f)  
>>>min(f)  
>>>h=m.hist(f,bins=int((144-15.3)/5))  
>>>h  
>>>m.show()
```



Treating exceptions

```
>>>a = raw_input("Quero dividir:")
>>>b= raw_input("por:")
>>>try:
...     c=float(a)/float(b)
...     print c
>>>except ValueError:
...     print "Ei! Nao posso fazer isto!"
>>>except ZeroDivisionError:
...     print "Ei! 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

```
>>> 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]
```

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

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

```
>>> print reduce(lambda x, y: x + y, zoo)  
368
```

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 m
```

```
>>>a = map(lambda y:y.x, ob)
```

```
>>>b = m.hist(a)
```

Moving 1000 steps more

```
>>>for i in range(10000): # Parallel way ...  
map(lambda x:x.mov(),ob)  
  
>>>c = map(lambda z:z.x, ob)  
>>>d = m.hist(c,bins=33)  
  
>>>plt.show()
```

