

Python - Cheat Sheet

Recursos

<https://www.w3schools.com/> Help Basic Commands
<https://colab.research.google.com/> Google Python notebooks

Print & Input

```
# Print & Input...
text = input('Entre com um texto: ')
numero = float(input('Entre com um número: '))
print('Seu texto: ', text, '\n', 'Seu nr: ', numero)
```

Integer, Float-point, Boolean

```
print(3 * 4, 3 + 4, 3 - 4, 3 / 4, 3.14, 4.2e-4 )
print(type(3 * 4), type(3.14), type(4.2e-4) )
print(3 ** 2, 3 // 4, 3 % 4 )
print(4 > 3, 4 >= 3, 3 == 3.0, 3 != 4, 3 <= 4 )
```

Math functions

```
import math as ma
print(math.pi) # pi
```

Strings

```
text = 'Beatles'
# Strings are immutable
# text[0] = 'T' # TypeError: ...
tex2 = '012345'
# print(text[<inicio>, <inicio>+<quantidade>])
# print(text[<inicio>, <fim>+1])
```

```
print(text[0:1]) # B
print(text[0:2]) # Be
print(tex2[0:2]) # 01
print(text[1:3]) # ea
print(tex2[1:3]) # 12
```

String Operations

```
str1 = 'To be or not to be'
print(str1.find('be')) # =3
print(str1.lower()) # to be or...
str2 = str1.replace('be', 'Be') # to Be or not to Be
print(str(4.2e-4)) # convert number to string
```

Lists []

Lista (**list**) é uma coleção que é ordenada e mutável. Permite membros duplicados.

```
mylist = ['A', 'B', 'C', 'A', 'B']
'A' in mylist # True
mylist.count('B') # 2
mylist.index('B') # 1
mylist.index('B', 2) # 4
mylist.append('D') # ['A', 'B', 'C', 'A', 'B', 'D']
mylist.sort() # ['A', 'A', 'B', 'B', 'C', 'D']
mylist.insert(1, 'X') # ['A', 'X', 'A', 'B', 'B', 'C', 'D']
del mylist[0] # ['X', 'A', 'B', 'B', 'C', 'D']
mylist.remove('B') # ['X', 'A', 'B', 'C', 'D']
mylist[1] = 'ZZ' # ['X', 'ZZ', 'B', 'C', 'D']
```

```
mylist = [[1,2,3], [11,22,33], [9,9,9]]
print(mylist[1]) # [11,22,33]
print(mylist[1][0]) # 11
mylist1 = mylist # NÃO USE
mylist2 = mylist.copy() # USE
```

Percorrendo uma lista

```
mylist = ['A', 'B', 'C', 'A', 'B']
for i in range(len(mylist)):
    print(mylist[i])
for item in mylist:
    print(item)
```

Tuples ()

Tupla (**tuple**) é uma coleção que é ordenada e imutável. Permite membros duplicados. No mais para nossos propósito Listas e Tuplas são bastante semelhantes.

```
mytuple = ('A', 'B', 'C', 'A', 'B')
'A' in mytuple # True
mytuple.count('B') # 2
# mytuple[1] = 'X' # TypeError: ...
```

Sets { }

Conjunto (**set**) é uma coleção não ordenada e não indexada. Nenhum membro duplicado. Para **Sets** são implementadas a funções conhecidas de conjuntos como união, intersecção etc.

```
myset = {'A', 'B', 'C', 'A', 'B'}
print(myset) # {'C', 'A', 'B'}, sem duplicatas
'A' in myset # True
```

Dictionary {key:values}

Dicionário (**dictionary**) é uma coleção desordenada, mutável e indexada. Nenhum membro duplicado. Você já teve algum contato com uma estrutura de dados JSON essa uma estrutura bastante semelhante em Python.

```
mydict = {'A':2, 'B':2, 'C':1}
print(mydict.keys()) # dict_keys(['A', 'B', 'C'])
'A' in mydict # True
print(mydict['B']) # 2
for item in mydict: # percorrendo mydict
    print(item, mydict[item]) # A 2 , B 2, C 1
```

If-Then-Else

```
if b > a:
    print("b is greater than a")
elif a == b: # opcional
    print("a and b are equal")
else:
    print("a is greater than b")
```

For & While

```
i = 1
while i < 6:
    print(i)
    if i == 3: # break opcional
```

```
break
i += 1
for i in [0,1,2,3,4]: # 0,1,2,3,4
    print(i)
for i in range(10): # 0,1,2,..., 9
    print(i)
```

Funções

```
def mymax(a,b):
    if a > b:
        return a
    else:
        return b
mymax(1984,1964) # 1984
```

File Handling

```
note = ['This is a note']
f = open("note.txt", "w") # write a file
for line in note:
    f.write(line)
f.close()
f = open("note.txt", "r") # read a file
for line in f:
    print(line)
```

Acesso dados Web

```
import urllib.request
data = urllib.request.urlopen('http://.../file.txt')
for line in data:
    print(line.decode('utf-8'))
```

Basic imports

```
import math
print(dir(math))
help(math.pow)
import numpy as np
import os
os.getcwd()
```

Numpy Basic & Seed

```
import numpy as np
```

```
x = np.array([1,2,3])
x[0] # 1
x[1:3] # array([2, 3])
```

Operations

```
x_list = x.tolist()
x = x * 2 + 1 # [3,5,7]
# mas x_list = x_list * 2 + 1 # TypeError:
np.mean(x) # 2
np.sum(x) # 6
x = np.insert(x, len(x), 4) # array([1, 2, 3, 4])
x = np.delete(x, len(x)-1) # array([1, 2, 3])
x = np.append(x, [4,5,6]) # array([1, 2, 3, 4, 5, 6])
x[0] = 9 # array([9, 2, 3, 4, 5, 6])
```

```
np.random.seed(1984) # semente de geração aleatória
x = np.array(np.random.randint(low=1, high=6, size=10))
print(x) # [1 2 1 3 5 5 1 2 3 4]
```