# Instalação:

Python 3.5.5

set path=%path%;C:\python35

# Python como interpretador

## Números

Operações ( +, -, \*,/,//, \*\*) 🡺 observação -a\*\*b

Operadores matemáticos (import math)

Múltipla inicialização (a,b = 0,1)

## Strings

Operações (+,\*)

Indexador [], [-], [:] => [a:] e [:a] => mostrar limites

Imutabilidade

## Listas

Var = [1,2,3]

Indexador de lista

Concatenação

Mutabilidade

Adição

## Controle de Fluxo

while

INDENTAÇÃO TAB vs 4spc

If elif else:

For Element in LIST (break, continue)

Mostrar loop em sublista

Range, range(a,b), range (a,b,s)

Len

List

# INTERPRETADORES

Visual Studio 2017

**Visual Code**

Pycharm

Qt

## FUNÇÕES

Def()

Passando argumentos

Chamadas e retorno

## LISTA AVANÇADA

Append, insert, remove, clear, count, sort,reverse

Del

Matrix

# Tuples ()

criando

Tuples = immutable

Objetos diferentes

Packing e unpacking

t = 12345, 54321, 'hello!'

x, y, z = t

# SETs {}

Coleção de objetos não ordenados únicos

Verificando objeto na lista com in

# Dicionários {‘label’:valor, ...}

tel = {'jack': 4098, 'sape': 4139}

tel['jack']

list(tel.keys())

sorted(tel.keys())

Listas duplas com ZIP

questions = ['name', 'quest', 'favorite color']

answers = ['lancelot', 'the holy grail', 'blue']

for q, a in zip(questions, answers):

... print('What is your {0}? It is {1}.'.format(q, a))

.items()

knights = {'gallahad': 'the pure', 'robin': 'the brave'}

for k, v in knights.items():

... print(k, v)

enumarate

for i, v in enumerate(['tic', 'tac', 'toe']):

... print(i, v)

0 tic

1 tac

2 toe

Matplotlib

a = b = np.arange(0, 3, .02)

c = np.exp(a)

d = c[::-1]

# Create plots with pre-defined labels.

fig, ax = plt.subplots()

ax.plot(a, c, 'b--', label='Model length')

ax.plot(a, d, 'r:', label='Data length')

ax.plot(a, c + d, 'k', label='Total message length')

legend = ax.legend(loc='upper center', shadow=True, fontsize='x-large')

# Put a nicer background color on the legend.

legend.get\_frame().set\_facecolor('#00FFCC')

plt.show()

import numpy as np

#import matplotlib

#matplotlib.use('Agg')

import matplotlib.pyplot as plt

# Fixing random state for reproducibility

[np.random.seed](https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.seed.html#numpy.random.seed)(19680801)

fig, ax = [plt.subplots](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplots.html#matplotlib.pyplot.subplots)()

ax.plot([np.random.rand](https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.rand.html" \l "numpy.random.rand" \o "View documentation for numpy.random.rand)(20), '-o', ms=20, lw=2, alpha=0.7, mfc='orange')

ax.grid()

# position bottom right

fig.text(0.95, 0.05, 'Property of MPL',

fontsize=50, color='gray',

ha='right', va='bottom', alpha=0.5)

[plt.show](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.show.html#matplotlib.pyplot.show)()

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

# Fixing random state for reproducibility

[np.random.seed](https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.seed.html#numpy.random.seed)(19680801)

[matplotlib.rcParams](https://matplotlib.org/api/matplotlib_configuration_api.html#matplotlib.rcParams)['axes.unicode\_minus'] = False

fig, ax = [plt.subplots](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplots.html#matplotlib.pyplot.subplots)()

ax.plot(10\*[np.random.randn](https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.randn.html" \l "numpy.random.randn" \o "View documentation for numpy.random.randn)(100), 10\*[np.random.randn](https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.randn.html" \l "numpy.random.randn" \o "View documentation for numpy.random.randn)(100), 'o')

ax.set\_title('Using hyphen instead of Unicode minus')

[plt.show](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.show.html#matplotlib.pyplot.show)()

import numpy as np

import matplotlib.pyplot as plt

# Fixing random state for reproducibility

[np.random.seed](https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.seed.html" \l "numpy.random.seed" \o "View documentation for numpy.random.seed)(19680801)

mu, sigma = 100, 15

x = mu + sigma \* [np.random.randn](https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.randn.html#numpy.random.randn)(10000)

# the histogram of the data

n, bins, patches = [plt.hist](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.hist.html" \l "matplotlib.pyplot.hist" \o "View documentation for matplotlib.pyplot.hist)(x, 50, density=True, facecolor='g', alpha=0.75)

[plt.xlabel](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.xlabel.html" \l "matplotlib.pyplot.xlabel" \o "View documentation for matplotlib.pyplot.xlabel)('Smarts')

[plt.ylabel](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.ylabel.html" \l "matplotlib.pyplot.ylabel" \o "View documentation for matplotlib.pyplot.ylabel)('Probability')

[plt.title](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.title.html" \l "matplotlib.pyplot.title" \o "View documentation for matplotlib.pyplot.title)('Histogram of IQ')

[plt.text](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.text.html" \l "matplotlib.pyplot.text" \o "View documentation for matplotlib.pyplot.text)(60, .025, r'$\mu=100,\ \sigma=15$')

[plt.axis](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.axis.html#matplotlib.pyplot.axis)([40, 160, 0, 0.03])

[plt.grid](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.grid.html#matplotlib.pyplot.grid)(True)

[plt.show](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.show.html#matplotlib.pyplot.show)()