a = []append_zero(a) print(a) # [0] [0] In [15]: def print_value(): print(a) a = 5print_value() # 5 In [16]: def print_value(): print(a) a = 10print(a) a = 5print_value() # UnboundLocalError: local variable 'a' referenced before assignment UnboundLocalError Traceback (most recent call last) <ipython-input-16-4b9b739b1bef> in <module>() $6 \ a = 5$ ---> 7 print_value() # UnboundLocalError: local variable 'a' referenced before assignment <ipython-input-16-4b9b739b1bef> in print_value() 1 def print_value(): ---> 2 print(a) 3 a = 10print(a) UnboundLocalError: local variable 'a' referenced before assignment Множества In [21]: s = set() # создание пустого множестваbasket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'} print(basket) print('orange' in basket) # True print('python' in basket) # False {'apple', 'orange', 'pear', 'banana'} True False In [22]: element = 'apple' s = basket s.add(element) print(s) s.remove(element) print(s) s.discard(element) print(s) s.clear() print(S) print(basket) {'apple', 'orange', 'pear', 'banana'} {'orange', 'pear', 'banana'} {'orange', 'pear', 'banana'} set() set() In [23]: basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'} for x in basket: print(x) apple orange pear banana Словари In [24]: d = dict() $d = \{\}$ $d = {'a': 239, 10: 100}$ print(d['a']) print(d[10]) 239 100 In [25]: dictionary = d key = 'a'print(key in dictionary) print(key not in dictionary) True False In [29]: dictionary[key] = 42 dictionary[key] Out[29]: 42 In [30]: dictionary.get(10) Out[30]: 100 In [31]: del dictionary[key] print(dictionary) {10: 100} In [32]: $d = \{'C': 14, 'A': 12, 'T': 9, 'G': 18\}$ for key in d: print(key, end=' ') T A G C In [33]: for key in d.keys(): print(key, end=' ') TAGC In [34]: for value in d.values(): print(value, end=' ') 9 12 18 14 In [35]: for key, value in d.items(): print(key, value, end='; ') # G 18; C 14; A 12; T 9; T 9; A 12; G 18; C 14; Чтение из файла In []: inf = open('file.txt', 'r') # open('file.txt') s1 = inf.readline() s2 = inf.readline() inf.close() with open('text.txt') as inf: s1 = inf.readline() s2 = inf.readline() # здесь файл уже закрыт In [36]: # s = inf.readline().strip() '\t abc \n'.strip() Out[36]: 'abc' In [38]: import os os.path.join('.', 'dirname', 'filename.txt') Out[38]: '.\\dirname\\filename.txt' In []: # Построчное чтение из файла with open('input.txt') as inf: for line in inf: line = line.strip() print(line) Запись в файл In []: ouf = open('file.txt', 'w') ouf.write('Some text\n') ouf.write(str(25)) ouf.close() with open('text.txt', 'w') as ouf: ouf.write('Some text\n') ouf.write(str(25)) # здесь файл уже закрыт Модули In [39]: import sys print(len(sys.argv)) In [41]: import subprocess subprocess.call(["python", "-h"]) Out[41]: 0 **NumPy** http://wiki.scipy.org/Tentative NumPy Tutorial Пример использования NumPy http://nbviewer.ipython.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-2-Numpy.ipynb In [1]: from numpy import * In [2]: a = array([2,3,4]) # создание одномерного массива из списка целых чисел In [17]: **print**(a) [2 3 4] Out[17]: array([2, 3, 4]) In [4]: a.ndim # размерность массива (одномерный, двумерный итд) Out[4]: 1 In [5]: a.shape # размеры массива (число строк, столбцов итд) Out[5]: (3,) In [6]: b = array([(1.5, 2, 3), (4, 5, 6)]) # создание двумерного массива из двух последовательностей чисел In [18]: **print**(b) # все числа имеют один тип - число с плавающей точкой [[1.5 2. 3.] [4. 5. 6.]] In [8]: b.ndim Out[8]: 2 In [9]: b.shape Out[9]: (2, 3) In [10]: b.size Out[10]: 6 In [12]: z = zeros((3, 2)) # 3, 2 помещаются в дополнительные скобки, чтобы представлять из себя один объект — пару чисел In [13]: z Out[13]: array([[0., 0.], [0., 0.], [0., 0.]]) In [14]: arange(10, 30, 5) # функция arange аналогична функции range, но возвращает массив Out[14]: array([10, 15, 20, 25]) In [15]: linspace(0, 2, 9) # генерирует 9 чисел на отрезке от 0 до 2 с равным шагом Out[15]: array([0. , 0.25, 0.5 , 0.75, 1. , 1.25, 1.5 , 1.75, 2.]) In [20]: b = arange(12).reshape(4, 3) print(b) [[0 1 2] [3 4 5] [678] [9 10 11]] In [21]: a = array([10, 20, 30])b = arange(3) print(a) print(b) [10 20 30] [0 1 2] In [22]: print(a + b) # арифметические операции на массивах выполняются поэлементно print(a - b) [10 21 32] [10 19 28] In [23]: **print**(a ** 2) [100 400 900] In [24]: 2 * sin(a) Out[24]: array([-1.08804222, 1.8258905, -1.97606325]) In [25]: a < 20 Out[25]: array([True, False, False], dtype=bool) Matplotlib Примеры графиков: http://matplotlib.org/gallery.html Python для научных вычислений: https://github.com/jrjohansson/scientific-python-lectures Более полный пример с построением графиков: http://nbviewer.ipython.org/urls/raw.github.com/jrjohansson/scientific-python-lectures/master/Lecture-4- Matplotlib.ipynb In [26]: from pylab import * In [27]: x = linspace(0, 5, 10)y = x ** 2In [28]: figure() plot(x, y, 'r') xlabel('x') ylabel('y') title('title') show() In [29]: %matplotlib inline In [30]: figure() plot(x, y, 'r') xlabel('x') ylabel('y') title('title') show() title In [31]: fig = plt.figure() axes = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # left, bottom, width, height (range 0 to 1) axes.plot(x, y, 'r') axes.set_xlabel('x') axes.set_ylabel('y') axes.set_title('title') # чтобы не отображалась строка <matplotlib.text.Text at ...> в конце можно поставить ; Out[31]: <matplotlib.text.Text at 0x6567510> title In [32]: fig = plt.figure() axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # main axes axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3]) # inset axes # main figure axes1.plot(x, y, 'r') axes1.set_xlabel('x') axes1.set_ylabel('y') axes1.set_title('title') # insert axes2.plot(y, x, 'g') axes2.set_xlabel('y') axes2.set_ylabel('x') axes2.set_title('insert title'); title insert title 15 20 In [33]: fig, axes = plt.subplots(nrows=1, ncols=2) for ax in axes: ax.plot(x, y, 'r')ax.set_xlabel('x') ax.set ylabel('y') ax.set_title('title') fig.tight_layout() In [34]: fig, axes = plt.subplots(figsize=(12,3)) axes.plot(x, y, 'r') axes.set_xlabel('x') axes.set_ylabel('y') axes.set_title('title'); title In [35]: fig, ax = plt.subplots() ax.plot(x, x**2, label="y = x**2")ax.plot(x, x**3, label="y = x**3")ax.legend(loc=2); # upper left corner ax.set_xlabel('x') ax.set_ylabel('y') ax.set_title('title'); — y = x**2 — y = x**3 100 In [39]: # Построение гистограмм from numpy import * n = random.randn(100000) # функция из numpy fig, axes = plt.subplots(1, 2, figsize=(12,4)) axes[0].hist(n) axes[0].set_title("Default histogram") axes[0].set_xlim((min(n), max(n))) axes[1].hist(n, cumulative=True, bins=50) axes[1].set_title("Cumulative detailed histogram") axes[1].set_xlim((min(n), max(n))); Default histogram Cumulative detailed histogram 35000 100000 30000 80000 25000 60000 20000 15000 40000 10000 20000 5000 -4 -3 -2 -1 0 1 2 3 4 -4 -3 -2 -1 0 1 2 3 4 XKCD plots http://nbviewer.ipython.org/url/jakevdp.github.com/downloads/notebooks/XKCD_plots.ipynb

Функции

if a <= b:

else:

m = a[0] for x in a:

return m

print(min(5, 3))

[5, 3, 6, 10]

res = []

if step > 0:

return a

return b

if m > x:

print(min(5, 3, 6, 10))
print(min([5, 3, 6, 10]))

In [5]: def my_range(start, stop, step=1):

x = start

elif step < 0:</pre>

return res

[2, 5, 8, 11, 14] [15, 12, 9, 6, 3]

> a = 100b = 200

init_values()

In [7]: print(my_range(2, 5))

[2, 3, 4]

In [10]: def init_values():

In [11]: def init_values():

a = 0

a = 100

init_values()
print(a) # 0

In [12]: def init_values(a):

b = 0

a = 100

init_values(b)
print(b) # 0

In [13]: def append_zero(xs):

In [14]: def append_zero(xs):

append_zero(a)
print(a) # [0]

a = []

[0]

xs.append(0)

xs.append(0)xs = [100]

x = start

print(my_range(2, 15, 3))
print(my_range(15, 2, -3))

5 init_values()

NameError: name 'a' is not defined

while x < stop:</pre>

while x > stop:

res += [x] x += step

print(a + b) # Ошибка, переменные a и b не объявлены

---> 6 print(a + b) # Ошибка, переменные а и b не объявлены

<ipython-input-10-b3d11a658b04> in <module>()

Traceback (most recent call last)

res += [x] x += step

m = x

print(min2(min2(42, 30), 25))

In [1]: **def** min2(a, b):

In [2]: m = min2(42, 30)

25

In [3]: **def** min(*a):

In [4]: print(min(5))

3