1.python基础试题[¶](http://localhost:8888/notebooks/Desktop/QTC2019/1%E3%80%81python%E7%9B%B8%E5%85%B3%E8%AF%95%E9%A2%98.ipynb" \l "1.python基础试题)

1.1 python 中boolean、float 和 int 分别表示什么？

boolean表示布尔型，float表示浮点型，int表示整型

1.2 设计求1-2+3-4+5 ... 99的所有数的和

start = 1

sum = 0

while start <100:

temp = start % 2

if temp ==1:

sum = sum + start

else:

sum = sum - start

#print(start)

#sum = sum + 1

start += 1

print(sum)

50

. . .

1.3 将字符串 s="yoyo" 转换成列表

s = 'yoyo'

s.split()

['yoyo']

list(s)

['y', 'o', 'y', 'o']

. . .

2. python进阶基础试题[¶](http://localhost:8888/notebooks/Desktop/QTC2019/1%E3%80%81python%E7%9B%B8%E5%85%B3%E8%AF%95%E9%A2%98.ipynb" \l "2.-python进阶基础试题)

2.1     for i in range(1,100)[2::3][-10:]:

​

               print i

       理解这段代码，并说出它是如何取数的

在1-99的数中，选择从第2个数到倒数第三个数，每隔10选一个数。

2.2 使用init 实例化时自动运行 分别计算单只股票最高价和收盘价两个时间点差值问题,可统一为one、two两个时间点,其最高价和收盘价赋值为 one(15,7) two(66,20)

class stock:

def \_\_init\_\_(high price, value):

return {'time': high price-value}

​

one = (15, 7)

print(high price-value)

​

two = stock(66, 20)

print(high price-value)

​

​

​

{'one\_dif': 8}

{'two\_dif': 46}

. . .

3. pandas数据处理试题[¶](http://localhost:8888/notebooks/Desktop/QTC2019/1%E3%80%81python%E7%9B%B8%E5%85%B3%E8%AF%95%E9%A2%98.ipynb" \l "3.-pandas数据处理试题)

3.1 如何查看列名、怎么对数据转置

3.1 如何查看列名、怎么对数据转置

def query(self, sql):

connect = self.connect()

cur = connect.cursor()

cur.execute(sql)

index = cur.description

result = []

for res in cur.fetchall():

row = {}

for i in range(len(index)-1):

row[index(i)(0)] = res[i]

result.append(row)

connect.close()

return result

​

​

3.2 读取data里的600029这只股票的DataFrame,将其收盘价转换成用Numpy的Array格式，并用talib计算10日EMA值，返回ndarray的最后五个值

3.2 读取data里的600029这只股票的DataFrame,将其收盘价转换成用Numpy的Array格式，并用talib计算10日EMA值，返回ndarray的最后五个值

In [28]:

​

import talib as ta

import pandas as pd

import warnings

warnings.filterwarnings('ignore')

​

data = pd.read\_excel('sz50.xlsx', sheet\_name='600029.XSHG', index\_col='datetime')

​

print(ta.MA(data.close.values, 5)[-5:])

​

import talib as ta

import numpy as np

​

EMA = pd.Series(ta.EMA(np.array(Prices, dtype=np.float64),5))

print(EMA)

<class 'numpy.ndarray'>

[ 15.08227205 15.2564044   15.44433088 15.72172526 15.96322976]

Type *Markdown* and LaTeX: *α* 2  α2

3.3 读取sz50.xlsx的['600029.XSHG','600050.XSHG','601318.XSHG']的全数据做成Panel

3.3 读取sz50.xlsx的['600029.XSHG','600050.XSHG','601318.XSHG']的全数据做成Panel

In [6]:

​

Out[6]:

<class 'pandas.core.panel.Panel'>

Dimensions: 3 (items) x 215 (major\_axis) x 5 (minor\_axis)

Items axis: 600029.XSHG to 601318.XSHG

Major\_axis axis: 2017-01-03 15:00:00 to 2017-11-20 15:00:00

Minor\_axis axis: close to volume

. . .

3.4把Panel转成ndim为3的Numpy，然后用array的切片读取ndim为2的三只股票最近20天的收盘价

3.4把Panel转成ndim为3的Numpy，然后用array的切片读取ndim为2的三只股票最近20天的收盘价

In [7]:

​

(3, 215, 5)

[[ 13.85 13.82 13.83 14.11 14.25 14.39 15.1 15.15 15.22

14.97 14.89 14.99 14.71 15.07 15.35 16. 16.04 16.29

16.97 17.05]

[ 8.53 8.56 8.62 8.88 9.25 9.24 9.43 9.25 9.02

9.15 9.63 9.5 9.9 9.97 9.96 9.49 9.68 9.61

9.63 9.8 ]

[ 134.83 134.56 137.5 143.18 143.48 144.15 143.61 143.65 144.35

143.36 146.41 144.71 148.56 156.53 157.12 157.72 154.99 163.52

168.58 169.57]]

. . .

3.5 建立一个5*\*5的矩阵，值从0到24*

3.5 建立一个5\*5的矩阵，值从0到24

In [8]:

​

[[ 0 1 2 3 4]

[ 5 6 7 8 9]

[10 11 12 13 14]

[15 16 17 18 19]

[20 21 22 23 24]]