Python
数据分析与机器学习

余力

buaayuli@ruc.edu.cn

为什么学?

- 当今最为流行的编程语言
- Python = 大数据分析、大数据分析利器？
- 直译式、交谈式 (Interpreted, Interactive)
- 与多种语言接口
- 易学易用 (Easy Learning)
- 大量的开源代码(Open Source)

学什么?

- Python语法基础
  - https://docs.python.org/
  - Python3.7从零开始学，清华大学出版社
  - Python基础教程（第三版），人民邮电出版社
- Python数据分析
  - 利用Python进行数据分析，机械工业出版社
- Python机器学习

Python语法基础

- 简介与安装：
- 数据类型：Number、String、List、Dictionary、Tuple
- 控制语句：赋值、表达式、Print、If、While、For
- 函数：作用域、变量传递、return、lambda、map
- 模块与包：概念、import(from)、reload、__name__
- 对象与类：运算符重载

Python数据分析

- 描述数据分析：Numpy、Pandas
- 文件读写：Text、Excel、csv
- 可视化：Matplotlib、Worcloud
- 爬虫：re、Urllib、 BeautifulSoup
- 自然语言处理：Jieba、NLTK
- 数据分析案例：电商、金融

Python机器学习

- 机器学习概述
- 机器学习基础算法
- 深度学习技术
- 机器学习应用

软件安装

- Python安装
- Pip安装包
- PyCharm安装
- Anaconda安装
  - https://www.anaconda.com/distribution/#download-section
  - 安装说明：https://jingyan.baidu.com/article/eae078275a31851fec5485b8.html
  - 安装说明：https://www.jianshu.com/p/62f155eb6ac5
- Jupyter Nootbook

包安装

- 1. 单文件模块：直接把文件拷贝到$python_dir/lib
- 2. 多文件模块，带setup.py：python setup.py install

pip 、conda

```
pip install scikit-learn
pip install jupyter_contrib_nbextensions
pip install --upgrade SomePackage
pip uninstall  SomePackage
pip install xxxx.whl
```

第1编 Python语法基础

中国人民大學
RENMIN UNIVERSITY OF CHINA

第1讲 数据类型

余力

buaayuli@ruc.edu.cn

内容

- Number
- String
- List
- Dictionary
- Tuple

01. Number

Number-运算符以及优先权

运算符	说明
x or y,	or的逻辑运算,
x and y	and的逻辑运算
not x	否定的逻辑运算
<, <=, >, >=, ==, <>, !=	比较运算符
is, is not	对等测试
in, not in	序列的成员关系

Number-运算符以及优先权

运算符	说明
x + y, x - y	加法, 减法
x * y, x / y, x % y	乘法, 除法, 余数运算
-x, +x, ~x	变号, 本身, 补码的位运算
x[i], x[i:j],	索引, 切片,
x.y, x(…)	名称评定用法, 函数调用
(…), […], {…}, `…`	Tuple, 序列, 字典, 转换成字符串

02. String

String-Introduction

- 双引号 or 单引号

  'hh'

  '混合型'

  'gp': '股票型',

  'zq': '债券型',

  'zs': '指数型',

  'qdii': 'QDII型',

  'bb': '保本型',

  'lof': 'LOF型',

  'fof': 'FOF型'

"单位净值","累积净值","日增长","近1周","近1月","近3月","近6月","近1年","近2年","近3年","今年来","成立来"

"002367","000001","399001"

String-常见的字符串运算

- **基本运算**

S1 + S2	串接
S2 * 3	重复串接
for x in S2	循环的迭代
"m"in S2	成员关系

- **索引参考和切片运算**

S2[i]	索引参考
S2[i : j]	切片运算
len(S2)	求字符串长

- **内容变更与书写格式**

"a %s parrot" % "dead"	字符串的输出方式

String-基本运算(1/2)

- 长度：字符个数

```
>>> len ("LaRC" )
    4
```

- 串接：形成新字符串

```
>>>"LaRC" + "EE"
    "LaRCEE "
```

- 重复串接：等于

```
>>>"LaRC" * 3      →   "LaRC "+ "LaRC "+ "LaRC "
    "LaRCLaRCLaRC "
```

- Python 看不懂的东西：
  - "LaRC"+ 3   #字符串与数字混用

String-基本运算(2/2)

- 有没有包括：判断成员关系是否成立

```
mylab = "LaRC"
 "R" in mylab
      True
        “000123”    “123”                           # means true
for x in mylab:    #循环的迭代
      print (x)
      L                            #一次print一个字母
      a
      R
      C
```

String-索引参考&切片运算(1/3)

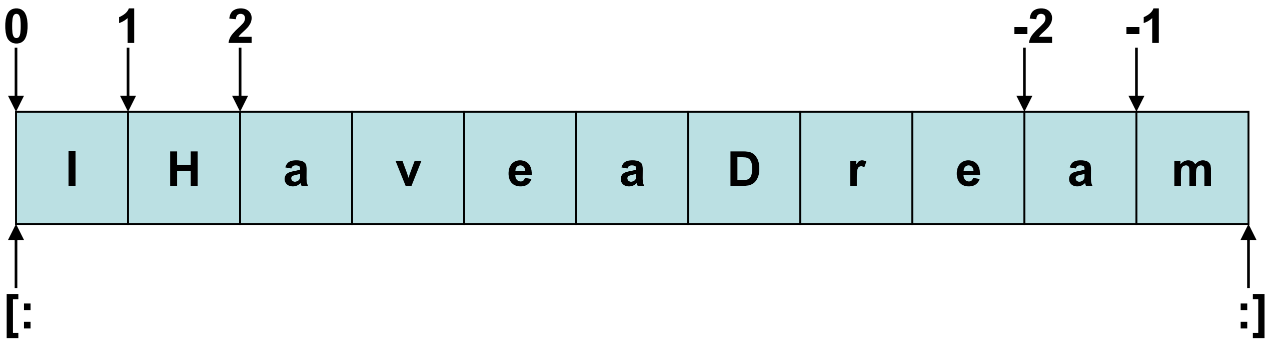
- 基本上与C类似, 都从 [0] ~ [n-1]
- 增加了负值的表示方法 [-n] ~ [-1]
  - Offset_p = Offset_n + n

```
>>> mylab = "LaRC"
>>> mylab[2], mylab[-1]        #前面&后面索引
    ( "L ", "R ")
>>> mylab[4:10]                 #切片运算
    "aR "
```

- 预设边界值 [(lower buond), :] [: (upper bound)]

```
>>>mylab[10:], mylab[ : -1 ]      #省略写法
    ( "aRC ", "LaR ")
```

String-索引参考&切片运算(2/3)



- **索引参考 (S[i])**
  - 以偏移量把字符读出来
  - 负索引值是从字符串的尾端倒数回来
  - S[-2] = S[len(S)-2]
- **切片运算 (S[i:j])**
  - 从序列中把某一片断的节区抽取出来
  - 切片的分界值预设是0和len(S)
  - S[:-1] 会把除了最后一个字符以外的都包含进来

String-书写格式

```
>>> Mylab = "LaRC"
>>> "I am in %s now!" % Mylab
    "I am in LaRC now! "
>>> "%d %s %d = ? " % (1, "+", 1)
    "1 + 1 = ? "
>>> "%s - %s - %s " % (33, 3.1415926, [1, 2, 3])
    "33 – 3.1415926 – [1, 2, 3] "
```

- %s这个符号会将所有的对象型态都转成字符串

```
>>>l = 3.1415926
>>>l = "%s" % l
    "3.1415926 "
```

常用的字符串工具

import string

S ="immediaTely"

S.**upper**()　　　　　"IMMEDIATELY　“

S.**lower**()　　　　　"immediately "

S.**find**("mm")　　　　　#传回"mm

S.split("mm")

print “ ".join("hello")

h:e:l:l:o

"xyz".**join**("abc")

　　"axyzbxyzc"

"ll".**join**(S.split("mm"))　#以mm分割, 再用ll合并起来

S.**replace**("mm","ll")

输入：input()

- x = input(‘Enter an integer : ’)#输入字符\字符串
  - print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
  - print("123","456","789",sep='-')
  - t=245
  - print(t,end=" end")
  - print(t,end="")

格式输出

```
x=9.8
print ("x= %.4f" % x)
format= "the value of x is %f"
value=x

print (format % value)
print ("the value of x is %f" % x)
```

03. List

List-Introduction

```
['000227', '华安年年红债券A', 'HANNHZQA', '2020-02-14', '1.0680', '1.4730', '', '1.6175', '2.8865', '4.2661', '6.3655', '9.3036', '17.2048', '20.9574', '2.9838', '55.7880', '2013-11-14', '6', '', '0.60%', '0.06%', '1', '0.06%', '1',  '30.5295']
```

```
title2=["单位净值","累积净值","日增长","近1周","近1月","近3月","近6月","近1年","近2年","近3年","今年来","成立来","成立时间","未知","成立来2","折前手续费","手续费","折数","手续费2","折数2","未知2"]
```

```
l_weight = [0.0, 0.1, 0.2, 0.3, 0.4]
```

List-常见的序列运算(1/2)

- 基本运算
  - L1 = [ ]
  - L2 = [0, 1, 2, 3]
  - L3 = ["abc", ["def","ghi"]]

  - L1 + L2
  - L2 * 3
  - For x in L2
  - 3 in L2
- 索引值参考和切片运算
  - L2[i], L3[i][j]
  - L2[i : j]   **L[i:j:d]**

通用序列操作（内置函数)

- list(L)：变成列表
- tuple(L)：转成元组
- str(L)：转化字符串
- len(L)
- max(L) 、min(L)
- sum(L)：对可迭代对象求和
- sorted(L) 、reversed(L)

常用序列方法

- L.append(“ABC”)
- L.sort( )   L.reverse( )
- L.count( )   L.index(1)
- L.pop(2)   L.remove(“A”)

List-基本运算(1/2)

- 跟字符串的运算方法一样

Num = [1,2,3,4,5,6,7,8,9]

Num[2]   Num[-2]   Num[1:]   Num[: : 2]

- 以新的对象参考地址来替代原先的

  Num[1] = "AA"

- 切片指定运算:

  Num[0 : 2] = ["I′m", "a"]

- 也可以用多个取代一个

  Num[1 : 2] = ["student", "in"]

List-基本运算(2/2)

- 串接
  - [1, 2, 3] + [4, 5, 6]
  - [1, 2] + list("34") #等于 [1, 2] + ["3", "4"]

- 重复串接
- ["Oh~"] * 4
  - ["Oh~", "Oh~", "Oh~", "Oh~"]
- 循环迭代
  - for x in [1, 2]:
  -     print x

List生成： range()

```
list("abc")

range(10)     range(1,10)     range(1,10,2)

list(range(10))

[x*x for x in range(1,10)]

[x*x for x in range(1,10) if x%2==0]

[x+y for x in "123" for y in "abc"]

for x in range(4):
        print (x, "little,")
else:
       print ("Indians!!")
```

List方法：append、extend、insert

mylab.append(“D")

mylab

["I"m", "student", "in", "LaRC!", “D"]

extend:

a=[1,2,3] b=[4,5,6]

a.extend(b)

insert: List.insert(3,”four”)

List方法：count、index

- count：
  - 计算某个元素出现的次数 list("good").count("o")
- index:
  - 第一个匹配项的索引位置list("good").index("o")

List方法：sort、inverse

- **排序**
  - 会产生新序列, 且一定存回原序列

mylab.sort( )　　#依照ASCII来排序

mylab　→　["I"m", "LaRC!", "in", "nthu", "student"]

- sort: **x.sort()** 不要y=x.sort() 不返回值
- 　可以**y=sorted(x)**
- reverse: a.reverse()　y=reversed(x)　list(y)
  - a=["r","s","t","p"]
  - t=**reversed**(a)
  - print t
  - list(t)

List方法：L.pop() + L.remove() +del L[]

```
del mylab[-1]                    #删去一个项目
mylab      #→     [ "I"m", "LaRC!", "in", "nthu"]
del mylab[:-1]                   #删去整个片段

mylab       #→   [ "nthu"]       #等于mylab[:-1] = [ ]
```

```
a=["r","s","t","p"]

a.pop(1)


a=["r","s","t","p"]

del a[1]

a.remove("s")
```

pop:移除列表中的某个索引，默认是最后一个

```
x.append(x.pop())
```

remove:移除列表中某个值的第一个匹配项

List应用-例1

```
for fund_type in ["zs","gp","hh","zq"]:
    #mainurl = "http://fund.eastmoney.com/data/rankhandler.aspx"
    mainurl = "http://fund.eastmoney.com/data/fundranking.html"
    fund_url = mainurl+"?op=ph&dt=kf&ft={0}&rs=&gs=0&sc={1}zf" \
           "&st=desc&qdii=|&pi=1&pn={2}&dx=1".format(fund_type, "1n", 10000)
    response = urllib.request.urlopen(fund_url).read().decode()
    data = response.split("[\"")[1].split("\"]")[0]
    for line in data.split("\",\""):
        item_list = line.split(",")
        if len(item_list) > 0:
            item_list.append(fund_type)
            fund_list.append(item_list)
```

List应用-例1

```
### Check a user name and PIN code

database = [
    ['albert',  '1234'],
    ['dilbert', '4242'],
    ['smith',   '7524'],
    ['jones',   '9843'] ]

username = input('User name: ')
pin = input('PIN code: ')

if [username, pin] in database:
    print ('Access granted')
else:
    print ('No Access')
```

例2

```
### Print out a date, given year, month, and day as numbers

months = [ 'January',  'February', 'March', 'April', 'May', 'June',
   'July',  'August', 'September',  'October',  'November',   'December']

# A list with one ending for each number from 1 to 31
endings = ['st', 'nd', 'rd'] + 17 * ['th']  + ['st', 'nd', 'rd'] +  7 * ['th']  + ['st']

year     = input('Year: ')
month    = input('Month (1-12): ')
day      = input('Day (1-31): ')

month_number = int(month)
day_number = int(day)

# Remember to subtract 1 from month and day to get a correct index
month_name = months[month_number-1]
ordinal = day + endings[day_number-1]

print (month_name + ' ' + ordinal + ', ' + year)
```

```
Year: 2018

Month (1-12): 9

Day (1-31): 18

September 18th, 2018
```

04. Dictionary

Dictionary-概念

```
FUND_TYPE =  {'hh': '混合型'，  'gp'： '股票型'， 'zq': '债券型'， 'zs': '指数型' }
```

```
market_dict =
 {"399006": ["399006", "创业板指"],
"000001": ["000001", "上证指数"],
"399001": ["399001", "深证成指"]}
```

```
phonebook = {“Alice”:[22,1 ,”山东” ,“东风楼”  , , ], "Beth":"9102",
"Cecil":"3258"}
```

Dictionary 创建

- D1 = { }
- D2 = {   “LaRC":1, "NTHU":2 }
- D3 = { "TW":{"LaRC":1, "NTHU":2}}
- D2["NTHU"], D3["TW"]["LaRC"]

- items=[("name","Gumb"),("age",42)]
- d=dict(items)

- **my_dict = {i: i * i for i in range(10)}**

Dictionary-基本运算

- 取值
  - d = { "A":1, "B":2, "C":3 }
  - d["A"]　　d.get("A")
  - d["A"] =4
- 字典数据项数量：
  - len(D2)
- 删除:
  - del d[key]　　d.clear()
- 新增：d[8]="floor"

Dictionary-基本运算

- d.keys()   d.values()   d.items()
- list(d.keys())   list(d.items())

```
d = { "Python": "Guido van Rossum",
      "Perl": "Larry Wall",
      "Tcl": "John Ousterhout"}
d.items()
list(d.items())
```

[('Python', 'Guido van Rossum'), ('Tcl', 'John Ousterhout'), ('Perl', 'Larry Wall')]

- 关系测试： ("B") in d.keys()

字典方法：d.pop(“A”)+d.popitem()

- pop()：删除字典给定键 key 所对应的值
  - d ={ "A":1, "B":2, "C":3 }
  - d.pop("A")

- popitem() #随机返回并删除一对键和值
  - student={'name':'小萌','number':'1001'}
  - key,value=student.popitem()
  - key

例子：通讯查询

```
people = {
    'Alice': {   'phone': '2341',      'addr': 'Foo drive 23'   },
    'Beth': {    'phone': '9102',      'addr': 'Bar street 42'  },
    'Cecil': {   'phone': '3158',      'addr': 'Baz avenue 90'  }   }

labels = {  'phone': 'phone number',    'addr': 'address'  }
name = input('Name: ')
request = input('Phone number (p) or address (a)? ')

# Use the correct key:
if request == 'p': key = 'phone'
if request == 'a': key = 'addr'

if name in people.keys():
    print ("%s's %s is %s." %  (name, labels[key], people[name][key]))
else:
    print ("No this person")
```

Dictionary-例子

```
for fund_type in FUND_TYPE:
    temp_pd=pd.read_csv(u'data/基金_基础数据_分类
_{}_{}.csv'.format(fund_type, DATE_NOW), encoding='gbk')
```

```
                    FUND_TYPE = {'hh': u'混合型',
                                 'gp': u'股票型',
                                 'zq': u'债券型',
                                 'zs': u'指数型',
                                 'qdii': u'QDII型',
                                 'lof': u'LOF型',
                                 'fof': u'FOF型' }
```

05. Tuple

Tuple-Introduction

(0, 1, 2, 3)

- 差别---不可变更
  - 不能变更某个项目的内容
  - 无法自行增长项目or删减项目
- 不可变更可以提供某种程度上的保证
- 使用于function上面的传递
- tuple(seq): 序列转换成无组

Tuple-基本tuple运算

```
( )                             #空tuple
T1  = (0 )              #1个项目的tuple(与表达式区别)
T2  = (0, 1, 2, 3)             #4个项目的tuple
T3  = 0, 1, 2, 3               #4个项目的tuple(与上式相同)
T4  = ("abc", ("def", "ghi"))   #巢状tuples
T1[ i ], T3[ i : j ]           #索引值参考
T1[ i:j ], len(t1)              #切片运算, 长度
T1  + T2                        #串接
T2 * 3                          #重复串接
for x in T2                     #循环反复
3 in T2                         #成员关系
```

谢谢大家！