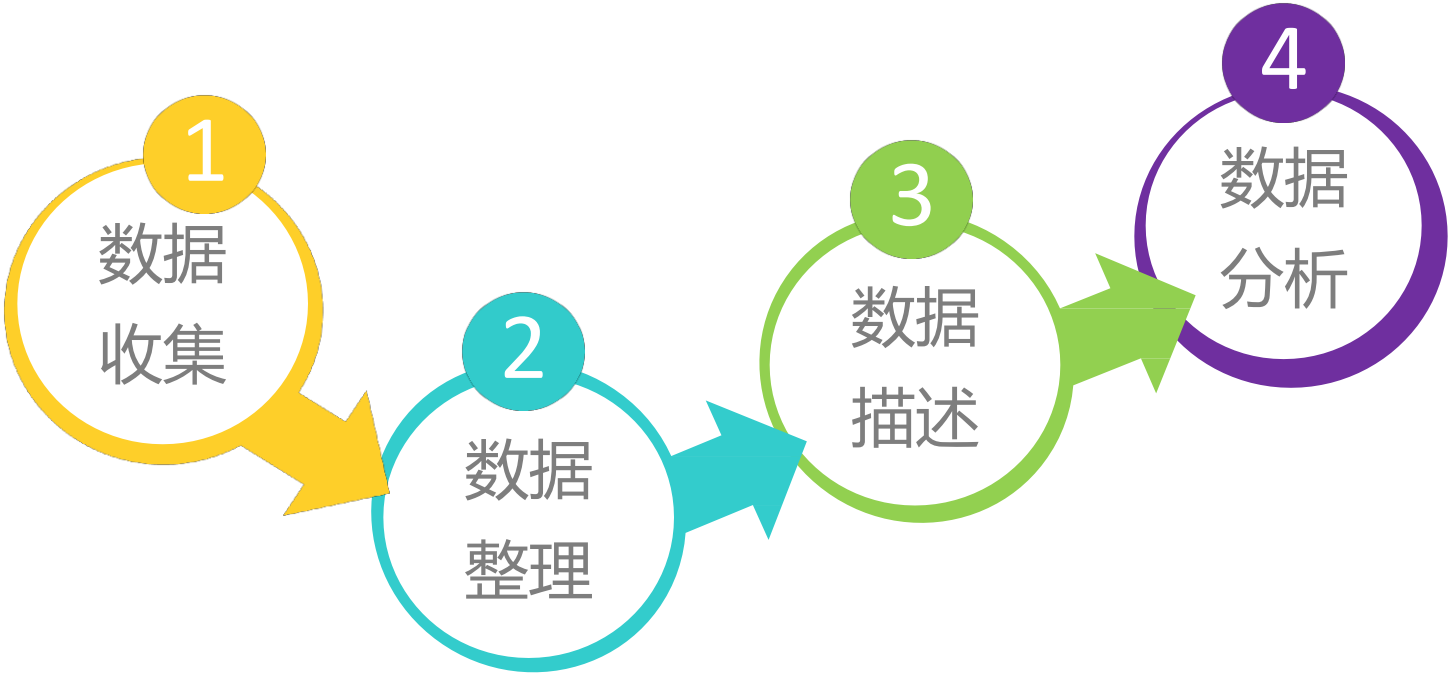
Python大数据编程及开发

Recommend
火龙果软件

大数据处理过程



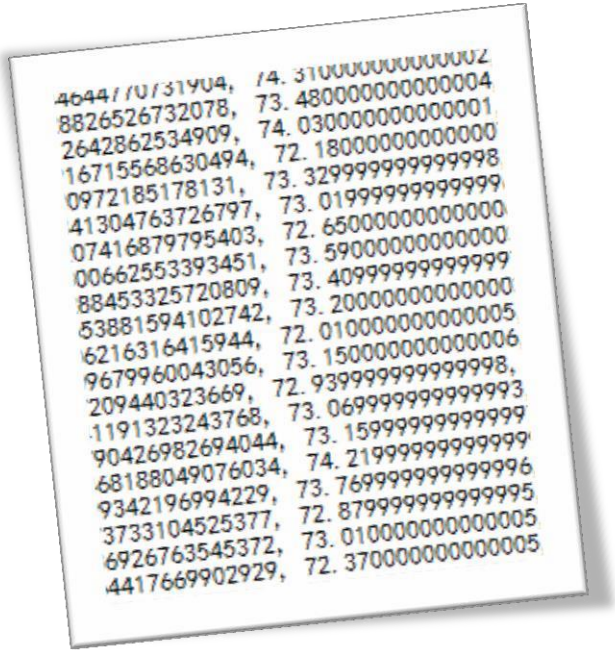
Recommend
火龙果软件

Python大数据编程

便捷数据获取

Recommend
火龙果软件

用Python获取数据



本地数据如何获取?

文件的打开，读写和关闭

- 文件打开
- 读文件

  写文件
- 文件关闭

用Python获取数据

ymbol	Company Name	Pric
OK	Nokia Corporation	8.4
APL	Apple Inc.	104.7
AC	Bank of America Cor...	16.6
	AT&T, Inc.	33.5
BR	Petr	12.5
QQ	PowerShares QQQ	97.9
B	Facebook, Inc.	80.0
IO	iBio, Inc.	1.4
ELP	Yelp, Inc.	58.5
NFN	Infinera Corporation	13.0
ISFT	Microsoft Corporation	44.7
O	The Coca-Cola Comp...	41.2
FE	Pfizer Inc	28.8

网络数据如何获取?

抓取网页，解析网页内容

- urllib
- urllib2
- httplib
- httplib2

Python 3中被urllib.request代替

Python 3中被http.client代替

http://finance.yahoo.com/q/cp?s=%5EDJI+Componen

yahoo

http://finance.yahoo.com/q/cp?s=%5EDJI+Components

^DJI Components | Dow ...

Home | Mail | News | Sports | Finance | Weather | Games | Groups | Answers | Screen | Flickr | Mobile | More

YAHOO! FINANCE

Search Finance | Search Web | Sign In | Mail

Finance Home | My Portfolio | Market Data | Yahoo Originals | Business & Finance | Personal Finance | CNBC | Contributors

Enter Symbol | Look Up

Sat, Oct 25, 2014, 10:48PM EDT - U.S. Markets closed | Report an Issue

Dow ↑0.76% Nasdaq ↑0.69%

More On ^DJI

QUOTES
Summary
Components
Options
Historical Prices

CHARTS
Interactive
Basic Chart
Basic Tech. Analysis

NEWS & INFO
Headlines

Dow Jones Industrial Average (^DJI) - DJI | Watchlist

16,805.41 ↑127.51(0.76%) Oct 24

Beat the market
Get the app

Components

Get Components for: GO

All | A | B | C | D | G | H | I | J | K | M | N | P | T | U | V | W | X

1 - 30 of 30 | First | Previous | Next | Last

Components for ^DJI

Symbol	Name	Last Trade	Change	Volume
AXP	American Express Company	86.40 Oct 24	↑0.79 (0.92%)	3,607,992
BA	The Boeing Company	122.24 Oct 24	↑0.21 (0.17%)	4,016,436
CAT	Caterpillar Inc.	99.44 Oct 24	↑0.17 (0.17%)	4,133,823
CSCO	Cisco Systems, Inc.	23.78 Oct 24	↑0.22 (0.91%)	19,947,265
CVX	Chevron Corporation	115.91 Oct 24	↓0.28 (0.24%)	5,508,970
DD	E. I. du Pont de Nemours and Company	69.00 Oct 24	↑0.41 (0.60%)	4,542,828
DIS	The Walt Disney Company	88.61 Oct 24	↑0.62 (0.70%)	4,937,482
GE	General Electric Company	25.64 Oct 24	↑0.20 (0.79%)	20,563,914
GS	The Goldman Sachs Group, Inc.	183.35 Oct 24	↑3.29 (1.83%)	2,900,871
HD	The Home Depot, Inc.	94.99 Oct 24	↑0.19 (0.20%)	4,825,237
IBM	International Business Machines Corporation	162.08 Oct 24	↓0.10 (0.06%)	6,652,379
INTC	Intel Corporation	33.18 Oct 24	↑0.49 (1.50%)	27,814,900

Sales start today with Yahoo Stores.

Recommend
火龙果软件

利用urllib库获取yahoo财经数据

```
# Filename: dji.py
import urllib  import re
dBytes =
urllib.request.urlopen('http://finance.yahoo.com/q/cp?s=%5EDJI+Components').read()
dStr = dBytes.decode('GBK')
#在python3中urllib.read()返回bytes对象而非str，语句功能是将dBytes转换成Str
m = re.findall('<tr><td class="yfnc_tabledata1"><b><a href=".*?">(.*?)</a></b></td><td
class="yfnc_tabledata1">(.*?)</td>.*?<b>(.*?)</b>.*?</tr>', dStr)
if m:
        print(m)
        print ('\n‘)
        print(len(m))
else:
        print( 'not match‘)
```

数据形式

- 包含多个字符串（dji）
  - 'AXP', 'American Express Company', '86.40'
  - 'BA', 'The Boeing Company', '122.24'
  - 'CAT', 'Caterpillar Inc.', '99.44'
  - 'CSCO', 'Cisco Systems, Inc.', '23.78'
  - 'CVX', 'Chevron Corporation', '115.91'
  - …

mbol	Company Name	Pri
OK	Nokia Corporation	8.4
APL	Apple Inc.	104.7
AC	Bank of America Cor...	16.6
	AT&T, Inc.	33.5
BR	Petr	12.5
QQ	PowerShares QQQ	97.9
B	Facebook, Inc.	80.0
IO	iBio, Inc.	1.4
ELP	Yelp, Inc.	58.5
FN	Infinera Corporation	13.0
SFT	Microsoft Corporation	44.7
D	The Coca-Cola Comp...	41.2
	Pfizer Inc.	

便捷网络数据

是否能够简单方便并且快速的方式获得雅虎财经上各上市公司股票的历史数据?

File

```
# Filename: quotes.py
from matplotlib.finance import quotes_historical_yahoo
_ochl
from datetime import date
import pandas as pd
today = date.today()
start = (today.year-1,  today.month,  today.day)
quotes = quotes_historical_yahoo_ochl('AXP', start, today)
df = pd.DataFrame(quotes)
print df
```

Recommend
火龙果软件

便捷网络数据

quotes的内容

日期 开盘价 收盘价 最高价 最低价 成交量

	0	1	2	3	4	5
0	735187	82.137528	81.91	82.365056	81.484622	4439700
1	735190	81.954487	81.45	82.083082	81.271946	3104800
2	735191	81.490863	81.57	81.985467	81.382051	2775000
3	735192	82.062317	81.36	82.517340	81.073138	3510500
4	735193	81.744151	83.05	83.099464	81.684795	4355200
5	735194	83.019249	82.95	83.029141	82.475152	2690300
6	735197	83.412905	83.68	83.848171	83.175486	3657700
7	735198	83.833676	84.17	84.664594	83.784217	4896800
8	735199	84.343547	84.67	84.739248	84.066557	2520000
9	735201	84.959142	84.88	85.265818	84.593110	2144600
10	735204	85.092126	84.37	85.596624	84.241402	3620800
11	735205	83.976989	83.70	84.412256	83.294410	3546100
12	735206	83.303123	83.59	84.322031	82.857969	3579700

便捷网络数据

需要先执行nltk.download()下载某一个或多个包，若下载失败，可以在官网（http://www.nltk.org/nltk_data/）单独下载后放到本地python目录的nltk_data\corpora下

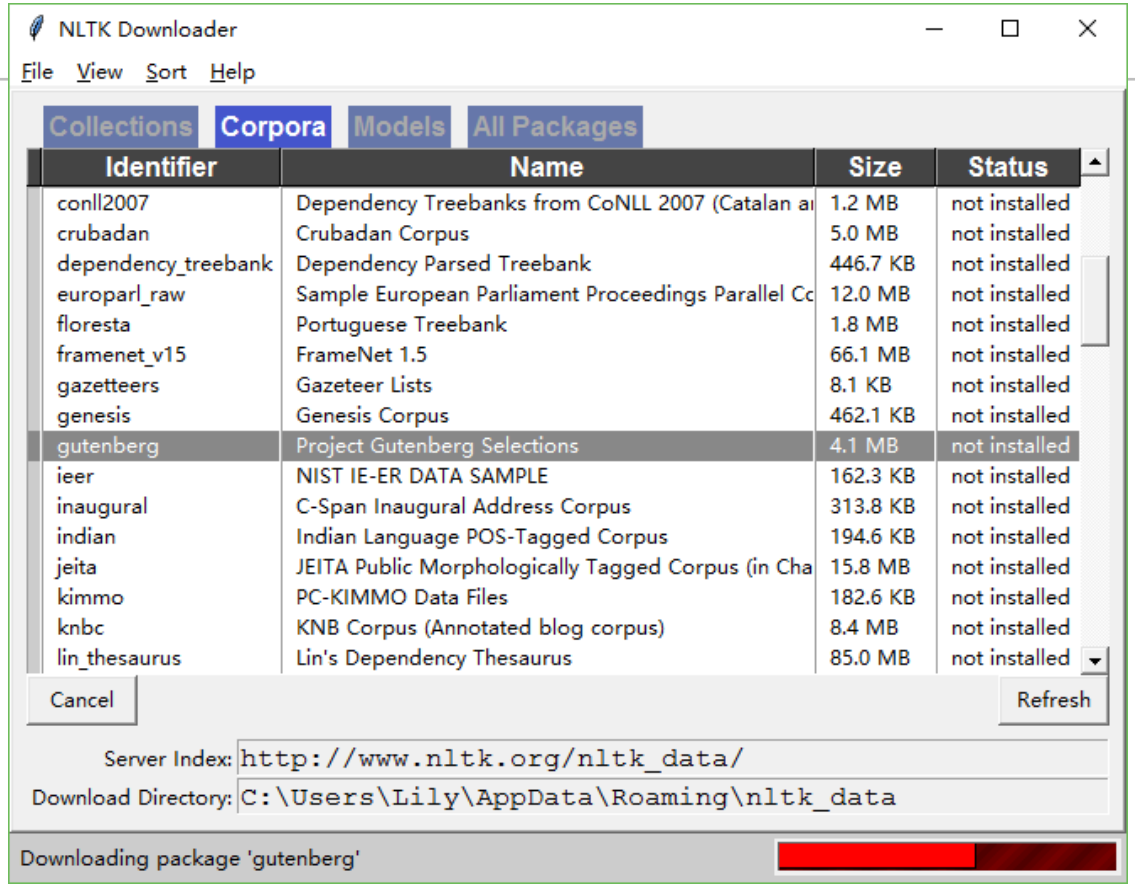
brown

自然语言工具包NLTK

- 古腾堡语料库
- 布朗语料库
- 路透社语料库
- 网络和聊天文本
- …

```
>>> from nltk.corpus import gutenberg
>>> import nltk
>>> print gutenberg.fileids()
[u'austen-emma.txt', u'austen-persuasion.txt', u'austen-sense.txt', u'bible-kjv.txt',
u'blake-poems.txt', u'bryant-stories.txt', u'burgess-busterbrown.txt', u'carroll-
alice.txt', u'chesterton-ball.txt', u'chesterton-brown.txt', u'chesterton-thursday.txt',
u'edgeworth-parents.txt', u'melville-moby_dick.txt', u'milton-paradise.txt',
u'shakespeare-caesar.txt', u'shakespeare-hamlet.txt', u'shakespeare-macbeth.txt',
u'whitman-leaves.txt']
>>> texts = gutenberg.words('shakespeare-hamlet.txt')
[u'[', u'The', u'Tragedie', u'of', u'Hamlet', u'by', ...]
```

Recommend
火龙果软件



Recommend
火龙果软件

Python大数据编程

数据准备

Recommend
火龙果软件

数据形式

30支成分股（dji）股票数据的逻辑结构

公司代码	公司名	最近一次成交价

美国运通公司（quotes）股票详细数据的逻辑结构

日期	开盘价	收盘价	最高价	最低价	成交量

Recommend
火龙果软件

数据整理

quotes数据加属性名

File

```
# Filename: quotesproc.py
from matplotlib.finance import
quotes_historical_yahoo_ochl
from datetime import date
import pandas as pd
today = date.today()
start = (today.year-1, today.month, today.day)
quotes = quotes_historical_yahoo_ochl('AXP', start, today)
fields = ['date','open','close','high','low','volume']
quotesdf = pd.DataFrame(quotes, columns = fields)
print quotesdf
```

	0	1	2	3	4	5
0	735190	81.954487	81.45	82.083082	81.271946	3104800
1	735191	81.490863	81.57	81.985467	81.382051	2775000
2	735192	82.062317	81.36	82.517340	81.073138	3510500
3	735193	81.744151	83.05	83.099464	81.684795	4355200
4	735194	83.019249	82.95	83.029141	82.475152	2690300
5	735197	83.412905	83.68	83.848171	83.175486	3657700
6	735198	83.833676	84.17	84.664594	83.784217	4896800
7	735199	84.343547	84.67	84.739248	84.066557	2520000
8	735201	84.959142	84.88	85.265818	84.593110	2144600
9	735204	85.092126	84.37	85.596624	84.241402	3620800
10	735205	83.976989	83.70	84.412256	83.294410	3546100
11	735206	83.303123	83.59	84.322031	82.857969	3579700
12	735207	83.362906	83.63	84.075156	83.244198	3677800
13	735208	84.663680	85.00	85.158268	84.426277	2666600
14	735211	85.245494	84.83	85.473026	84.434292	2436500
15	735212	84.459029	84.37	84.815146	84.162265	2354200
16	735213	84.365197	83.00	84.751013	82.910965	3782300
17	735214	82.874814	82.40	83.369412	82.310972	3574300
18	735215	82.423872	82.78	83.017419	82.334840	3594900
19	735218	83.056029	83.61	84.055155	83.056029	3599000
20	735219	83.793509	83.20	83.803400	83.081298	2893700
21	735220	83.388278	85.06	85.188594	82.646389	5332300
22	735221	85.015058	85.48	85.598708	84.985381	3956300
23	735222	85.779231	86.63	86.837746	85.512130	6294500
24	735225	87.053335	87.36	87.627095	86.993980	3242600
25	735226	87.383788	87.73	87.779459	87.235412	1042700

Recommend
火龙果软件

数据整理

dji数据：加属性名

	name	lasttrade
BA		
CAT		
…		
XOM		

quotes数据：加属性名

date	open	close	high	low	volume
735190.0					
735191.0					
735192.0					
…					
735551.0					

Recommend
火龙果软件

数据整理

用1,2,...作为索引

```
quotesdf = pd.DataFrame(quotes, columns = fields)
```

```
quotesdf = pd.DataFrame(quotes, index = range(1,len(quotes)+1),columns =  fields)
```

	date	open	close	high	low	volume
0	735190	81.954487	81.45	82.083082	81.271946	3104800
1	735191	81.490863	81.57	81.985467	81.382051	2775000
2	735192	82.062317	81.36	82.517340	81.073138	3510500
3	735193	81.744151	83.05	83.099464	81.684795	4355200
4	735194	83.019249	82.95	83.029141	82.475152	2690300

	date	open	close	high	low	volume
1	735190	81.954487	81.45	82.083082	81.271946	3104800
2	735191	81.490863	81.57	81.985467	81.382051	2775000
3	735192	82.062317	81.36	82.517340	81.073138	3510500
4	735193	81.744151	83.05	83.099464	81.684795	4355200
5	735194	83.019249	82.95	83.029141	82.475152	2690300

Recommend
火龙果软件

数据整理

如果可以直接用date作为索引，quotes的时间能否转换成常规形式（如下图中的效果）？

	open	close	high	low	volume
2013-11-18	81.954487	81.45	82.083082	81.271946	3104800
2013-11-19	81.490863	81.57	81.985467	81.382051	2775000
2013-11-20	82.062317	81.36	82.517340	81.073138	3510500
2013-11-21	81.744151	83.05	83.099464	81.684795	4355200
2013-11-22	83.019249	82.95	83.029141	82.475152	2690300
2013-11-25	83.412905	83.68	83.848171	83.175486	3657700
2013-11-26	83.833676	84.17	84.664594	83.784217	4896800
2013-11-27	84.343547	84.67	84.739248	84.066557	2520000
2013-11-29	84.959142	84.88	85.265818	84.593110	2144600
2013-12-02	85.092126	84.37	85.596624	84.241402	3620800
2013-12-03	83.976989	83.70	84.412256	83.294410	3546100
2013-12-04	83.303123	83.59	84.322031	82.857969	3579700
2013-12-05	83.362906	83.63	84.075156	83.244198	3677800
2013-12-06	84.663680	85.00	85.158268	84.426277	2666600
2013-12-09	85.245494	84.83	85.473026	84.434292	2436500
2013-12-10	84.459029	84.37	84.815146	84.162265	2354200
2013-12-11	84.365197	83.00	84.751013	82.910965	3782300
2013-12-12	82.874814	82.40	83.369412	82.310972	3574300

Source

```
>>> from datetime import date
>>> firstday = date.fromordinal(735190)
>>> lastday = date.fromordinal(735551)
>>> firstday
datetime.date(2013, 11, 18)
>>> lastday
datetime.date(2014, 11, 14)
```

时间序列

File

```
# Filename: quotesproc.py
from matplotlib.finance import quotes_historical_yahoo_ochl
from datetime import date
from datetime import datetime
import pandas as pd
today = date.today()
start = (today.year-1, today.month, today.day)
quotes = quotes_historical_yahoo_ochl('AXP', start, today)
fields = ['date','open','close','high','low','volume']
list1 = []
for i in range(0,len(quotes)):
    x = date.fromordinal(int(quotes[i][0]))
    y = datetime.strftime(x,'%Y-%m-%d')
    list1.append(y)
quotesdf = pd.DataFrame(quotes, index = list1, columns = fields)
quotesdf = quotesdf.drop(['date'], axis = 1)
print quotesdf
```

转换成常规时间

转换成固定格式

删除原date列

Recommend
火龙果软件

创建时间序列

Source

```
>>> import pandas as pd
>>> dates = pd.date_range('20141001', periods=7)
>>> dates
<class 'pandas.tseries.index.DatetimeIndex'>
[2014-10-01, ..., 2014-10-07]
Length: 7, Freq: D, Timezone: None
>>> import numpy as np
>>> dates = pd.DataFrame(np.random.randn(7,3),index=dates,columns =  list('ABC'))
>>> dates
                   A         B         C
2014-10-01  1.302600 -1.214708  1.411628
2014-10-02 -0.512343  2.277474  0.403811
2014-10-03 -0.788498 -0.217161  0.173284
2014-10-04  1.042167 -0.453329 -2.107163
2014-10-05 -1.628075  1.663377  0.943582
2014-10-06 -0.091034  0.335884  2.455431
2014-10-07 -0.679055 -0.865973  0.246970
[7 rows x 3 columns]
```

Python大数据编程

数据显示

Recommend
火龙果软件

数据显示

	code	name	lasttrade
0	AXP	American Express Company	90.67
1	BA	The Boeing Company	128.86
2	CAT	Caterpillar Inc.	101.34
3	CSCO	Cisco Systems, Inc.	26.32
4	CVX	Chevron Corporation	116.32
5	DD	E. I. du Pont de Nemours and Company	70.80
6	DIS	The Walt Disney Company	90.80
7	GE	General Electric Company	26.46
8	GS	The Goldman Sachs Group, Inc.	189.98
9	HD	The Home Depot, Inc.	98.24
10	IBM	International Business Machines Corporation	164.16
11	INTC	Intel Corporation	33.95
12	JNJ	Johnson & Johnson	108.16
13	JPM	JPMorgan Chase & Co.	60.28
14	KO	The Coca-Cola Company	42.73
15	MCD	McDonald's Corp.	96.21
16	MMM	3M Company	158.85
17	MRK	Merck & Co. Inc.	59.07
18	MSFT	Microsoft Corporation	49.58
19	NKE	Nike, Inc.	95.50
20	PFE	Pfizer Inc.	30.34
21	PG	The Procter & Gamble Company	88.11
22	T	AT&T, Inc.	35.90
23	TRV	The Travelers Companies, Inc.	102.43
24	UNH	UnitedHealth Group Incorporated	95.11
25	UTX	United Technologies Corporation	107.45
26	V	Visa Inc.	248.84
27	VZ	Verizon Communications Inc.	51.50
28	WMT	Wal-Mart Stores Inc.	82.96
29	XOM	Exxon Mobil Corporation	95.09

djidf

	open	close	high	low	volume
2013-11-18	81.954487	81.45	82.083082	81.271946	3104800
2013-11-19	81.490863	81.57	81.985467	81.382051	2775000
2013-11-20	82.062317	81.36	82.517340	81.073138	3510500
2013-11-21	81.744151	83.05	83.099464	81.684795	4355200
2013-11-22	83.019249	82.95	83.029141	82.475152	2690300
2013-11-25	83.412905	83.68	83.848171	83.175486	3657700
2013-11-26	83.833676	84.17	84.664594	83.784217	4896800
2013-11-27	84.343547	84.67	84.739248	84.066557	2520000
2013-11-29	84.959142	84.88	85.265818	84.593110	2144600
2013-12-02	85.092126	84.37	85.596624	84.241402	3620800
2013-12-03	83.976989	83.70	84.412256	83.294410	3546100
2013-12-04	83.303123	83.59	84.322031	82.857969	3579700
2013-12-05	83.362906	83.63	84.075156	83.244198	3677800
2013-12-06	84.663680	85.00	85.158268	84.426277	2666600
2013-12-09	85.245494	84.83	85.473026	84.434292	2436500
2013-12-10	84.459029	84.37	84.815146	84.162265	2354200
2013-12-11	84.365197	83.00	84.751013	82.910965	3782300
2013-12-12	82.874814	82.40	83.369412	82.310972	3574300

quotesdf

Recommend
火龙果软件

数据显示

显示方式：

- 显示索引
- 显示列名
- 显示数据的值
- 显示数据描述

Source

```
>>> djidf.index
Int64Index([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, 25, 26, 27, 28, 29], dtype='int64')
>>> djidf.columns
Index([u'code', u'name', u'lasttrade'], dtype='object')
>>> dijdf.values
array([['AXP', 'American Express Company', '90.67'],
        ['BA', 'The Boeing Company', '128.86'],
         ...
        ['XOM', 'Exxon Mobil Corporation', '95.09']], dtype=object)
>>> djidf.describe
<bound method DataFrame.describe of   code                    name  lasttrade
0    AXP                American Express Company      90.67
1     BA                     The Boeing Company     128.86
...
29   XOM                 Exxon Mobil Corporation      95.09
```

Recommend
火龙果软件

数据显示

索引的格式

Source

```
>>> quotesdf.index
Index([u'2013-11-18', u'2013-11-19', u'2013-11-20',
u'2013-11-21',
u'2013-11-22', u'2013-11-25', u'2013-11-26',  u'2013-
11-27',
...
-04-08', u'2014-04-09', u'2014-04-10', u'2014-04-11',
...],
dtype='object')
```

Recommend
火龙果软件

数据显示

查看道琼斯工业股中前5只和后5只的股票基本信息?

显示方式:

- 显示行
  - 专用方式
  - 切片

Source

df[:5]

```
>>> djidf.head(5)
    code                    name  lasttrade
0    AXP  American Express Company      90.67
1     BA       The Boeing Company     128.86
2    CAT             Caterpillar Inc.  101.34
3   CSCO        Cisco Systems, Inc.     26.32
4    CVX        Chevron Corporation    116.32
[5 rows x 3 columns]
```

df[25:]

```
>>> djidf.tail(5)
     code                    name  lasttrade
25    UTX  United Technologies Corporation     107.45
26      V                   Visa Inc.     248.84
27     VZ  Verizon Communications Inc.      51.50
28    WMT            Wal-Mart Stores Inc.     82.96
29    XOM        Exxon Mobil Corporation     95.09
[5 rows x 3 columns]
```

Recommend
火龙果软件

Python大数据编程

数据选择

Recommend
火龙果软件

数据选择

	code	name	lasttrade
0	AXP	American Express Company	90.67
1	BA	The Boeing Company	128.86
2	CAT	Caterpillar Inc.	101.34
3	CSCO	Cisco Systems, Inc.	26.32
4	CVX	Chevron Corporation	116.32
5	DD	E. I. du Pont de Nemours and Company	70.80
6	DIS	The Walt Disney Company	90.80
7	GE	General Electric Company	26.46
8	GS	The Goldman Sachs Group, Inc.	189.98
9	HD	The Home Depot, Inc.	98.24
10	IBM	International Business Machines Corporation	164.16
11	INTC	Intel Corporation	33.95
12	JNJ	Johnson & Johnson	108.16
13	JPM	JPMorgan Chase & Co.	60.28
14	KO	The Coca-Cola Company	42.73
15	MCD	McDonald's Corp.	96.21
16	MMM	3M Company	158.85
17	MRK	Merck & Co. Inc.	59.07
18	MSFT	Microsoft Corporation	49.58
19	NKE	Nike, Inc.	95.50
20	PFE	Pfizer Inc.	30.34
21	PG	The Procter & Gamble Company	88.11
22	T	AT&T, Inc.	35.90
23	TRV	The Travelers Companies, Inc.	102.43
24	UNH	UnitedHealth Group Incorporated	95.11
25	UTX	United Technologies Corporation	107.45
26	V	Visa Inc.	248.84
27	VZ	Verizon Communications Inc.	51.50
28	WMT	Wal-Mart Stores Inc.	82.96
29	XOM	Exxon Mobil Corporation	95.09

选择方式：

- 选择行
- 选择列
- 选择区域
- 筛选（条件选择）

	open	close	high	low	volume
2013-11-18	81.954487	81.45	82.083082	81.271946	3104800
2013-11-19	81.490863	81.57	81.985467	81.382051	2775000
2013-11-20	82.062317	81.36	82.517340	81.073138	3510500
2013-11-21	81.744151	83.05	83.099464	81.684795	4355200
2013-11-22	83.019249	82.95	83.029141	82.475152	2690300
2013-11-25	83.412905	83.68	83.848171	83.175486	3657700
2013-11-26	83.833676	84.17	84.664594	83.784217	4896800
2013-11-27	84.343547	84.67	84.739248	84.066557	2520000
2013-11-29	84.959142	84.88	85.265818	84.593110	2144600
2013-12-02	85.092126	84.37	85.596624	84.241402	3620800
2013-12-03	83.976989	83.70	84.412256	83.294410	3546100
2013-12-04	83.303123	83.59	84.322031	82.857969	3579700
2013-12-05	83.362906	83.63	84.075156	83.244198	3677800
2013-12-06	84.663680	85.00	85.158268	84.426277	2666600
2013-12-09	85.245494	84.83	85.473026	84.434292	2436500
2013-12-10	84.459029	84.37	84.815146	84.162265	2354200
2013-12-11	84.365197	83.00	84.751013	82.910965	3782300
2013-12-12	82.874814	82.40	83.369412	82.310972	3574300

数据选择

美国运通公司2013年12月2日至2013年12月6日间的股票交易信息?

选择方式:

- 选择行
  - 切片
  - 索引

Source

```
>>> quotesdf[u'2013-12-02':u'2013-12-06']
                 open  close       high       low  volume
2013-12-02  85.092126  84.37  85.596624  84.241402  3620800
2013-12-03  83.976989  83.70  84.412256  83.294410  3546100
2013-12-04  83.303123  83.59  84.322031  82.857969  3579700
2013-12-05  83.362906  83.63  84.075156  83.244198  3677800
2013-12-06  84.663680  85.00  85.158268  84.426277  2666600

[5 rows x 5 columns]
```

数据选择

道琼斯工业股公司代码?

选择方式:

- 选择列
  - 列名

```
>>> djidf['code']
0      AXP
1       BA
2      CAT
...
29     XOM
Name: code, dtype: object
>>> djidf.code
0      AXP
1       BA
2      CAT
...
29     XOM
Name: code, dtype: object
```

不支持

djidf['code', 'lasttrade']

djidf['code':'lasttrade']

数据选择

道琼斯工业股中标号是1至5的股票信息以及所有股票的代码和最近一次交易价?

选择方式:

- 行、列
  - 标签label（loc）

Source

```
>>> djidf.loc[1:5,]
   code                                   name  lasttrade
1    BA                     The Boeing Company     128.86
2   CAT                          Caterpillar Inc.     101.34
3  CSCO                       Cisco Systems, Inc.      26.32
4   CVX                     Chevron Corporation     116.32
5    DD  E. I. du Pont de Nemours and Company      70.80
[5 rows x 3 columns]
>>> djidf.loc[:,['code','lasttrade']]
    code  lasttrade
0    AXP      90.67
1     BA     128.86
2    CAT     101.34
...
29   XOM      95.09
[30 rows x 2 columns]
```

数据选择

道琼斯工业股中标号是1至5的股票代码和最近一次交易价?标号是1的股票的最近一次交易价?

选择方式：

- 行和列的区域
  - 标签label（loc）
- 单个值
  - at

Source

```
>>> djidf.loc[1:5,['code','lasttrade']]
    code  lasttrade
1     BA     128.86
2    CAT     101.34
3   CSCO      26.32
4    CVX     116.32
5     DD      70.80

[5 rows x 2 columns]
>>> djidf.loc[1,'lasttrade']
'128.86‘
>>> djidf.at[1,'lasttrade']
'128.86'
```

Recommend
火龙果软件

数据选择

选择方式：

- 行、列和区域
  - 用iloc（位置）
- 取某个值
  - iat

Source

```
>>> djidf.loc[1:5,['code','lasttrade']]
   code  lasttrade
1    BA     128.86
2   CAT     101.34
3  CSCO      26.32
4   CVX     116.32
5    DD      70.80
```

Source

```
>>> djidf.loc[1,'lasttrade']
'128.86'
>>> djidf.at[1,'lasttrade']
'128.86'
```

Source

```
>>> djidf.iloc[1:6,[0,2]]
   code  lasttrade
1    BA     128.86
2   CAT     101.34
3  CSCO      26.32
4   CVX     116.32
5    DD      70.80
```

Source

```
>>> djidf.iloc[1,2]
'128.86'
>>> djidf.iat[1,2]
'128.86'
```

Recommend
火龙果软件

数据选择

美国运通公司2014年的股票信息？进一步寻找美国运通公司2014年收盘价大于等于95的记录？

选择方式：

- 条件筛选

Source

```
>>> quotesdf[quotesdf.index >= u'2014-01-01']
                open   close       high        low   volume
2014-01-02 89.924438  88.49 90.102506  88.420751  5112000
2014-01-03 88.186377  88.77 89.106325  87.671998  3888500
2014-01-06 88.730000  88.73 89.274052  88.413460  2844700
...
2014-03-28 89.531554  89.72 90.811002  89.263763  3138900
           ...   ...    ...       ...       ...
[221 rows x 5 columns]
>>> quotesdf[(quotesdf.index >= u'2014-01-01') & (quotesdf.close >= 95)]
                open   close       high        low   volume
2014-06-09 94.532820  95.02 95.328216  94.105295  3825200
2014-06-18 94.204662  95.01 95.039827  93.538518  2454800
2014-07-03 95.031492  95.29 95.389426  94.673558  1633800
[3 rows x 5 columns]
```

Python大数据编程

简单统计与处理

Recommend
火龙果软件

简单统计与筛选

求道琼斯工业股中30只股票最近一次成交价的平均值？股票最近一次成交价大于等于120的公司名？

Source

```
>>> djidf.mean(columns = 'lasttrade')
lasttrade    91.533667
dtype: float64
>>> djidf[djidf.lasttrade >= 120].name
1                                   The Boeing Company
8                        The Goldman Sachs Group, Inc.
10    International Business Machines Corporation
16                                          3M Company
26                                             Visa Inc.
Name: name, dtype: object
```

简单统计与筛选

统计美国运通公司近一年股票涨和跌分别的天数?

Source

```
>>> len(quotesdf[quotesdf.close > quotesdf.open])
131
>>> len(quotesdf)-131
120
```

统计美国运通公司近一年相邻两天收盘价的涨跌情况?

Source

```
>>> status = np.sign(np.diff(quotesdf.close))
>>> status
array([ 1., -1.,  1., -1.,  1.,  1.,  1.,  1., -1., -1., -1.,  1.,  1.,
        ...
        -1., -1., -1.])
>>> status[np.where( status == 1.)].size
130
>>> status[np.where( status == -1.)].size
120
```

排序

DataFrame的sort()函数已不推荐使用，相同功能推荐使用sort_index()函数

按最近一次成交价对30只道琼斯工业股股票进行排序。根据排序结果列出前三甲公司名。

Source

```
>>> djidf.sort(columns = 'lasttrade')
     code                                   name  lasttrade
3    CSCO                     Cisco Systems, Inc.      26.32
7      GE              General Electric Company      26.46
20    PFE                              Pfizer Inc.      30.34
11   INTC                     Intel Corporation      33.95
...
8      GS        The Goldman Sachs Group, Inc.     189.98
26      V                                Visa Inc.     248.84
[30 rows x 3 columns]
>>> djidf.sort(columns = 'lasttrade')[27:].name
10    International Business Machines Corporation
8                  The Goldman Sachs Group, Inc.
26                                      Visa Inc.
Name: name, dtype: object
```

可以添加sort()函数的ascending属性控制顺序/逆序排序，默认该属性=True，即顺序排列

计数统计

统计2014年1月份的股票开盘天数?

Source

```
>>> t = quotesdf[(quotesdf.index >= '2014-01-01') & (quotesdf.index < '2014-02-01')]
>>> t
                   open  close       high        low    volume
2014-01-02  89.924438  88.49  90.102506  88.420751  5112000
2014-01-03  88.186377  88.77  89.106325  87.671998  3888500
...
2014-01-30  85.741393  85.91  86.267049  85.364508  4259000
2014-01-31  84.577859  84.32  85.202672  84.210906  4778000
[21 rows x 5 columns]
>>> len(t)
21
```

计数统计

统计近一年每个月的股票开盘天数?

File

```
# Filename: quotesmonth.py
import time
...
listtemp = []
for i in range(0,len(quotesdf)):
    temp = time.strptime(quotesdf.index[i],"%Y-%m-%d")
    listtemp.append(temp.tm_mon)
print listtemp
tempdf = quotesdf.copy()
tempdf['month'] = listtemp
print tempdf['month'].value_counts()
```

Output:

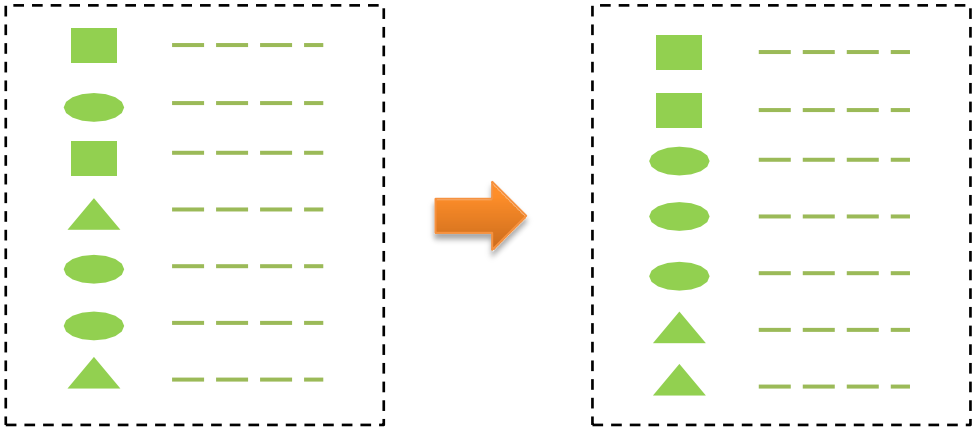
```
10    23
7     22
12    21
9     21
8     21
6     21
5     21
4     21
3     21
1     21
11    19
2     19
dtype: int64
```

Python大数据编程

GROUPING

Recommend
火龙果软件

分组



Grouping的顺序

① Splitting

② Applying

③ Combining

Recommend
火龙果软件

分组

统计近一年每个月的股票开盘天数?

Source

```
>>> tempdf.groupby('month').count()
        open  close  high  low  volume  month
month
1         21     21    21   21      21     21
2         19     19    19   19      19     19
3         21     21    21   21      21     21
...
10        23     23    23   23      23     23
11        20     20    20   20      20     20
12        21     21    21   21      21     21
[12 rows x 6 columns]
>>> tempdf.groupby('month').count().month
```

如果没有特殊处理的话则没有month这一列数据

如果没有month这一列数据，则此处month需要改成其他属性名如open

Output:

```
month
1     21
2     19
3     21
4     21
5     21
6     21
7     22
8     21
9     21
10    23
11    20
12    21
Name: month, dtype: int64
```

Recommend
火龙果软件

分组

统计近一年每个月的总成交量?

Source

```
>>> tempdf.groupby('month').sum().volume
month
        1      104874000
        2       76173800
        3       71488400
        4       84786400
...
        9       85341400
       10      120822100
       11       67906300
       12       67589400
Name: volume, dtype: float64
```

mean()

min()

max()

...

Recommend
火龙果软件

分组

如果更高效统计近一年每个月的总成交量?

Source

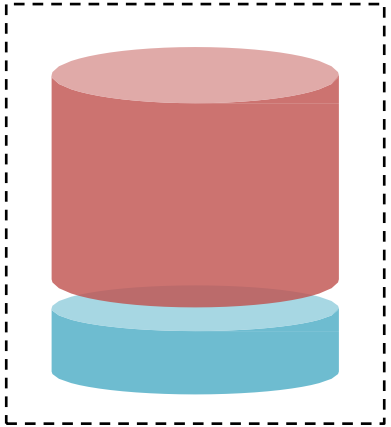
```
>>> g = tempdf.groupby('month')
>>> gvolume = g['volume']
>>> print gvolume.sum()
month
       1      104874000
       2       76173800
       3       71488400
...
      10      120822100
      11       67906300
      12       67589400
Name: volume, dtype: float64
```

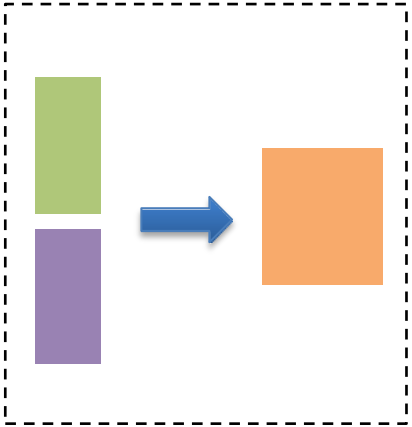
Python大数据编程

MERGE

Recommend
火龙果软件

合并





Merge的形式

- Append
  - 加行到DataFrame
- Concat
  - 连接pandas对象
- Join
  - SQL类型的连接

Append

把美国运通公司2014年1月1日至2014年1月5日间的股票交易信息合并到近一年中前两天的股票信息中?

Source

```
>>> p = quotesdf[:2]
>>> p
                   open  close       high        low   volume
2013-11-18  81.954487  81.45  82.083082  81.271946  3104800
2013-11-19  81.490863  81.57  81.985467  81.382051  2775000
[2 rows x 5 columns]
>>> q = quotesdf[u'2014-01-01':u'2014-01-05']
>>> q
                   open  close       high        low   volume
2014-01-02  89.924438  88.49  90.102506  88.420751  5112000
2014-01-03  88.186377  88.77  89.106325  87.671998  3888500
[2 rows x 5 columns]
>>> p.append(q)
                   open  close       high        low   volume
2013-11-18  81.954487  81.45  82.083082  81.271946  3104800
2013-11-19  81.490863  81.57  81.985467  81.382051  2775000
2014-01-02  89.924438  88.49  90.102506  88.420751  5112000
2014-01-03  88.186377  88.77  89.106325  87.671998  3888500
[4 rows x 5 columns]
```

Recommend
火龙果软件

Concat

将美国运通公司近一年股票数据中的前5个和后5个合并。

Source

```
>>> pieces = [tempdf[:5], tempdf[len(tempdf)-5:]]
>>> pd.concat(pieces)
                  open  close       high       low  volume month
2013-11-18  81.954487  81.45  82.083082  81.271946  3104800    11
2013-11-19  81.490863  81.57  81.985467  81.382051  2775000    11
2013-11-20  82.062317  81.36  82.517340  81.073138  3510500    11
2013-11-21  81.744151  83.05  83.099464  81.684795  4355200    11
2013-11-22  83.019249  82.95  83.029141  82.475152  2690300    11
2014-11-11  92.280000  91.74  92.590000  91.490000  2719800    11
2014-11-12  91.160000  91.55  91.670000  91.100000  3825200    11
2014-11-13  91.700000  91.07  91.700000  90.940000  3637900    11
2014-11-14  91.070000  90.67  91.240000  90.350000  2361600    11
2014-11-17  90.240000  90.13  90.260000  89.650000  2620500    11
[20 rows x 6 columns]
```

Recommend
火龙果软件

Concat

两个不同逻辑结构的对象能否连接?

objs	axis
join	join_axes
keys	levels
names	verify_integrity
ignore_index	

Source

```
>>> piece1 = quotesdf[:3]
>>> piece2 = tempdf[:3]
>>> pd.concat([piece1,piece2],ignore_index
=True)
```

	close	high	low	month	open	volume
0	81.45	82.083082	81.271946	NaN	81.954487	3104800
1	81.57	81.985467	81.382051	NaN	81.490863	2775000
2	81.36	82.517340	81.073138	NaN	82.062317	3510500
3	81.45	82.083082	81.271946	11	81.954487	3104800
4	81.57	81.985467	81.382051	11	81.490863	2775000
5	81.36	82.517340	81.073138	11	82.062317	3510500

[6 rows x 6 columns]

Join

code	name	lasttrade
AXP		
KO		

code	month	volume
AXP		
AXP		
KO		
KO		

code	name	lasttrade	month	volume
AXP				
AXP				
KO				
KO				

Recommend
火龙果软件

Join

将美国运通公司和可口可乐公司近一年中每个月的交易总量表（包含公司代码）与30只道琼斯 成分股股票信息 合并。

code|name|month|volumes

```
>>> djidf
    code                                        name  lasttrade
0    AXP                     American Express Company      90.13
1     BA                           The Boeing Company     128.42
2    CAT                              Caterpillar Inc.     101.70
3   CSCO                          Cisco Systems, Inc.      26.47
4    CVX                          Chevron Corporation     115.75
5     DD          E. I. du Pont de Nemours and Company      70.47
6    DIS                       The Walt Disney Company      90.41
7     GE                     General Electric Company      26.61
8     GS                 The Goldman Sachs Group, Inc.     189.93
9     HD                         The Home Depot, Inc.      98.03
10   IBM  International Business Machines Corporation     164.16
11  INTC                             Intel Corporation      34.24
12   JNJ                         Johnson &amp; Johnson     108.30
13   JPM                     JPMorgan Chase &amp; Co.      60.38
14    KO                        The Coca-Cola Company      42.92
15   MCD                             McDonald's Corp.      95.97
16   MMM                                    3M Company     158.69
17   MRK                          Merck &amp; Co. Inc.      59.46
18  MSFT                         Microsoft Corporation      49.46
19   NKE                                   Nike, Inc.      96.06
20   PFE                                  Pfizer Inc.      30.32
21    PG              The Procter &amp; Gamble Company      87.84
22     T                                AT&amp;T, Inc.      35.85
23   TRV                 The Travelers Companies, Inc.     102.75
24   UNH                UnitedHealth Group Incorporated      96.47
25   UTX                United Technologies Corporation     107.37
26     V                                     Visa Inc.     249.80
27    VZ                   Verizon Communications Inc.      51.40
28   WMT                          Wal-Mart Stores Inc.      83.57
29   XOM                        Exxon Mobil Corporation      95.11

[30 rows x 3 columns]
```

```
>>> AKdf
           volume code  month
month
1       104874000  AXP      1
2        76173800  AXP      2
3        71488400  AXP      3
4        84786400  AXP      4
5        65091900  AXP      5
6        60522000  AXP      6
7        68452900  AXP      7
8        83077800  AXP      8
9        85341400  AXP      9
10      120822100  AXP     10
11       67906300  AXP     11
12       67589400  AXP     12
1       289400900   KO      1
2       385231900   KO      2
3       349073700   KO      3
4       382724700   KO      4
5       213626400   KO      5
6       281965200   KO      6
7       282905200   KO      7
8       247854800   KO      8
9       314443100   KO      9
10      529541900   KO     10
11      265863000   KO     11
12      301144500   KO     12

[24 rows x 3 columns]
```

Recommend
火龙果软件

Join

Source

```
>>> pd.merge(djidf, AKdf, on = 'code')
  code                          name  lasttrade   volume  month
0  AXP  American Express Company      90.13  104874000   1
1  AXP  American Express Company      90.13   76173800   2
2  AXP  American Express Company      90.13   71488400   3
3  AXP  American Express Company      90.13   84786400   4
4  AXP  American Express Company      90.13   65091900   5
...
19  KO      The Coca-Cola Company     42.92  247854800   8
20  KO      The Coca-Cola Company     42.92  314443100   9
21  KO      The Coca-Cola Company     42.92  529541900  10
22  KO      The Coca-Cola Company     42.92  265863000  11
23  KO      The Coca-Cola Company     42.92  301144500  12
[24 rows x 5 columns]
>>> pd.merge(djidf,AKdf,on = 'code').drop(['lasttrade'],axis = 1)
```

```
   code                     name     volume  month
0   AXP  American Express Company  104874000      1
1   AXP  American Express Company   76173800      2
2   AXP  American Express Company   71488400      3
3   AXP  American Express Company   84786400      4
4   AXP  American Express Company   65091900      5
5   AXP  American Express Company   60522000      6
6   AXP  American Express Company   68452900      7
7   AXP  American Express Company   83077800      8
8   AXP  American Express Company   85341400      9
9   AXP  American Express Company  120822100     10
10  AXP  American Express Company   67906300     11
11  AXP  American Express Company   67589400     12
12   KO     The Coca-Cola Company  289400900      1
13   KO     The Coca-Cola Company  385231900      2
14   KO     The Coca-Cola Company  349073700      3
15   KO     The Coca-Cola Company  382724700      4
16   KO     The Coca-Cola Company  213626400      5
17   KO     The Coca-Cola Company  281965200      6
18   KO     The Coca-Cola Company  282905200      7
19   KO     The Coca-Cola Company  247854800      8
20   KO     The Coca-Cola Company  314443100      9
21   KO     The Coca-Cola Company  529541900     10
22   KO     The Coca-Cola Company  265863000     11
23   KO     The Coca-Cola Company  301144500     12

[24 rows x 4 columns]
```

Recommend
火龙果软件

merge函数的参数

left	right	how
on	left_on	right_on
left_index	right_index	sort
suffixes	copy	

Recommend
火龙果软件

Python大数据编程

条件

Recommend
火龙果软件

if 语句

语 法	expression	expr_true_suite
if expression: expr_true_suite	条件表达式: • 比较运算符 • 成员运算符 • 逻辑运算符	• expression 条 件 为 True时执行的代码块 • 代码块必须缩进（通常为4个空格）

File

```
# Filename: ifpro.py
sd1 = 3
sd2 = 3
if sd1 == sd2:
    print "the square's area is:%d" % (sd1*sd2)
```

else 语句

语法

```
if expression :
    expr_true_suite
else:
    expr_false_suite
```

expr_false_suite

- expression 条件为False时执行的代码块
- 代码块必须缩进
- else语句不缩进

File

```
# Filename: elsepro.py
sd1 = int(raw_input('the first side:'))
sd2 = int(raw_input('the second side:'))
if sd1 == sd2:
    print "the square's area is:%d" %(sd1*sd2)
else:
    print "the rectangle's area is:%d" %(sd1*sd2)
```

Input and Output

```
the first side:4
the second side:4
the square's area is:16
```

Recommend
火龙果软件

elif 语句

语法

```
if  expression :
    expr_true_suite
elif expression2:
    expr2_true_suite
        :
        :
elif expressionN :
    exprN_true_suite
else:
   none_of_the_above_suite
```

expr2_true_suite

- expression2为True时执行的代码块

exprN_true_suite

- expressionN 为 True 时执行的代码块

else

- none_of_the_above_suite是以上所有条件都不满足时执行的代码块

elif 语句

File

```
# Filename: elifpro.py
k = raw_input('input the index of shape:')
if k == '1':
    print 'circle'
elif k == '2':
    print 'oval'
elif k == '3':
    print 'rectangle'
elif k == '4':
    print 'triangle'
else:
    print 'you input the invalid number'
```

Input and Output

```
input the index of shape:3
rectangle
```

Input and Output

```
input the index of shape:8
you input the invalid number
```

Recommend
火龙果软件

条件嵌套

- 同等缩进为同一条件结构

Input and Output

```
input the index of shape:3
the first side:3
the second side:4
the rectangle's area is:12
```

Input and Output

```
input the index of shape:2
oval
```

File

```
# Filename: ifnestpro.py
k = raw_input('input the index of shape:')
if k == '1':
    print 'circle'
elif k == '2':
    print 'oval'
elif k == '3':
    sd1 = int(raw_input('the first side:'))
    sd2 = int(raw_input(:'the second side'))
    if sd1 == sd2:
        print "the square's area is:%d" %(sd1*sd2)
    else:
        print "the rectangle's area is:%d" %(sd1*sd2)
    print 'rectangle'
elif k == '4':
    print 'triangle'
else:
    print 'you input the invalid number'
```

Recommend
火龙果软件

猜数字游戏

- 程序随机产生一个0~300间的整数，玩家竞猜，系统给出“猜中”、“太大了”或“太小了”的提示。

File

```
# Filename: guessnum1.py
from random import randint

x = randint(0, 300)
print 'Please input a number between 0~300:'
digit = input()
if digit == x :
    print 'Bingo!'
elif digit > x:
    print 'Too large,please try again.'
else:
    print 'Too small,please try again.'
```

Recommend
火龙果软件

Python大数据编程

RANGE和XRANGE

Recommend
火龙果软件

range()

语法

range (start, end, step=1)

range (start, end)

range (end)

- 生成一个真实的列表

Source

```
>>> range(3,11,2)
[3, 5, 7, 9]
>>> range(3,11)
[3, 4, 5, 6, 7, 8, 9, 10]
>>> range(11)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

start

- 起始值（包含）

end

- 终值（不包含）

step

- 步长（不能为0）

range (start, end, step=1)

- 不包含end的值

range (start, end)

- 缺省step值为1

range (end)

- 缺省了start值为0，step为1

Recommend
火龙果软件

xrange()

异同	range()	xrange()
语法	一样	
返回	列表	生成器
生成	真实列表	用多少生成多少

在Python 3.x中不支持xrange()函数，都使用range()函数，其返回值为range对象，并且需要显式调用，常用list(range(10))形式

Source

```
>>> xrange(3,11,2)
xrange(3, 11, 2)
>>> print xrange(3,11,2)
xrange(3, 11, 2)
>>> for i in xrange(3,11,2):
        print i
```

Output:

```
3
5
7
9
```

Recommend
火龙果软件

Python大数据编程

循环

Recommend
火龙果软件

while 循环

语法

while expression:

suite_to_repeat

expression

- 条件表达式
- 当 expression 值为 True 时执行 suite_to_repeat 代码块

Source

```
>>> sumA = 0
>>> j = 1
>>> while j < 10:
        sumA += j
        j += 1
>>> sumA
45
>>> j
10
```

for 循环（一）

语 法

```
for iter_var in iterable_object:
    suite_to_repeat
```

可以明确循环的次数

- 遍历一个数据集内的成员
- 在列表解析中使用
- 生成器表达式中使用

iterable_object

- String
- List
- Tuple
- Dictionary
- File

for 循环（二）

- 字符串就是一个*iterable_object*
- range()返回的也是 *iterable_object*

Source

```
>>> s = 'python'
>>> for c in s:
        print c
p
y
t
h
o
n
>>> for i in range(3,11,2):
        print i,
3 5 7 9
```

Python 3.x中print函数用法与Python 2.x中的语句用法也有所改变，例如此处变成print(i, end = ' ')

Recommend
火龙果软件

猜数字游戏

- 程序随机产生一个0~300间的整数，玩家竞猜，允许猜多次，系统给出“猜中”、“太大了”或“太小了”的提示。

File

```
# Filename: guessnum2.py
from random import randint

x = randint(0, 300)
for count in range(0,5):
    print 'Please input a number between 0~300:'
    digit = input()
    if digit  == x :
        print 'Bingo!'
    elif digit > x:
        print 'Too large,please try again.'
    else:
        print 'Too small,please try again.'
```

Recommend
火龙果软件

Python大数据编程

循环中的BREAK,CONTINUE和ELSE

Recommend
火龙果软件

break 语句

- break语句终止当前循环，转而执行循环之后的语句

File

```
# Filename: breakpro.py
sumA = 0
i = 1
while True:
    sumA += i
    i += 1
    if sumA > 10:
        break
print 'i=%d,sum=%d' % (i,sumA)
```

Output:
i=6,sumA=15

Recommend
火龙果软件

while 循环和break

- 输出2-100之间的素数

Output:
2 3 5 7 11 13 17 19
23 29 31 37 41 43
47 53 59 61 67 71
73 79 83 89 97

File

```
# Filename: prime.py
from math import sqrt
j=2
while j <=100:
    i = 2
    k= sqrt(j)
    while( i <= k ):
        if j%i == 0: break
         i = i+1
     if( i > k):
        print j,
    j += 1
```

Recommend
火龙果软件

for 循环和break

- 输出2-100之间的素数

Output:
2 3 5 7 11 13 17 19
23 29 31 37 41 43
47 53 59 61 67 71
73 79 83 89 97

File

```
# Filename: prime.py
from math import sqrt
for i in range(2,101):
    k = int(sqrt(i))
    for j in range(2,k+1):
        if i%j == 0:
            flag = 0
            break
    if(
        flag    ):
        print i,
```

flag = 1

Recommend
火龙果软件

continue 语句

- 在while和for循环中，continue语句的作用：
  - 停止当前循环，重新进入循环
  - while循环则判断循环条件是否满足
  - for循环则判断迭代是否已经结束

Recommend
火龙果软件

continue语句

- 循环中的break：

File

```
# Filename: breakpro.py
sumA = 0
i = 1
while i <= 5:
    sumA += i
    if i  ==  3:
        break
    print 'i=%d,sum=%d' % (i,sumA)
    i += 1
```

- 循环中的continue：

File

```
# Filename: continuepro.py
sumA = 0
i = 1
while i <= 5:
    sumA += i
    i += 1
    if i == 3:
        continue
    print 'i=%d,sum=%d' % (i,sumA)
```

Recommend
火龙果软件

猜数字游戏（想停就停，非固定次数）

- 程序随机产生一个0~300间的整数，玩家竞猜，允许玩家自己控制游戏次数，如果猜中系统给出提示并退出程序，如果猜错给出“太大了”或“太小了”的提示，如果不想继续玩可以退出。

File

```
# Filename: guessnum3.py
from random import randint
x = randint(0, 300)
go = 'y'
while (go == 'y'):
    print 'Please input a number between 0~300:'
    digit = input()
    if digit == x :
        print 'Bingo!‘
        break
    elif digit > x:
        print 'Too large,please try again.'
    else:
        print 'Too small,please try again.‘
    print 'If you do not want to continue,input n,or input y.'
    go = raw_input()
    print go
else:
    print 'Goodbye!'
```

Recommend
火龙果软件

循环中的else语句

- 循环中的else：
  - 如果循环代码从break处终止，跳出循环
  - 正常结束循环，则执行else中代码

Source

```
>>> k=5
>>> for i in range(1,10):
        if k==3:
            break
    else:
        print i
9
```

Recommend
火龙果软件

Python大数据编程

自定义函数

Recommend
火龙果软件

函数

内置函数

函数调用之前必须先定义

自定义函数

Recommend
火龙果软件

自定义函数的创建

语 法

```
def function_name([arguments]):
"optional documentation string"
function_suite
```

Source

```
>>> def addMe2Me(x):
        'apply operation + to argument'
        return (x+x)
```

Recommend
火龙果软件

自定义函数的调用

- 函数名加上函数运算符， 一对小括号
  - 括号之间是所有可选的参数
  - 即使没有参数， 小括号也不能省略

Source

```
>>> addMe2Me()

Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    addMe2Me()
TypeError: addMe2Me() takes exactly 1 argument (0 given)
```

Source

```
>>> addMe2Me(3.7)
7.4
>>> addMe2Me(5)
10
>>> addMe2Me('Python')
'PythonPython'
```

Recommend
火龙果软件

自定义函数

- 输出1-100之间的素数

Output:
2 3 5 7 11 13 17 19
23 29 31 37 41 43
47 53 59 61 67 71
73 79 83 89 97

File

```
# Filename: prime.py
from math import sqrt
def isprime(x):
    if x == 1:
        return False
    k = int(sqrt(x))
    for j in range(2,k+1):
        if x%j == 0:
            return False
    return True
if __name__== "__main__":
    for i in range(2,101):
        if isprime(i):
            print i,
```

与Python2.x中一样，用“if __name__== '__main__'”来判断是否是在直接运行该.py文件

Recommend
火龙果软件

默认参数（一）

- 函数的参数可以有一个默认值， 如果提供有默认值，在函数定义中，默认参数以赋值语句的形式提供

Source

```
>>> def f(x = True):
        '''whether x is a correct word or not'''
        if x:
            print 'x is a correct word'
        print 'OK'
>>> f()
x is a correct word
OK
>>> f(False)
OK
```

Recommend
火龙果软件

默认参数（二）

- 默认参数的值可以改变

Source

```
>>> def f(x ,  y = True):
        '''x and y both correct words or not '''
        if y:
            print x, 'and y both correct '
        print x, 'is OK'
>>> f (68)
68 and y both correct
68 is OK
>>> f(68,False)
68 is OK
```

默认参数（三）

- 默认参数一般需要放置在参数列表的最后

Source

```
def f(y = True, x):
    '''x and y both correct words or not '''
    if y:
        print x, 'and y both correct '
    print x, 'is OK'
```

SyntaxError: non-default argument follows default argument

Recommend
火龙果软件

关键字参数

- 关键字参数是让调用者通过使用参数名区分参数。允许改变参数列表中的参数顺序

Source

```
>>> def f(x , y):
        '''x and y both correct words or not '''
        if y:
            print x, 'and y both correct '
        print x, 'is OK'
>>> f(68,False)
68 is OK
>>> f(y = False,x = 68)
68 is OK
>>> f(y = False,68)
SyntaxError: non-keyword arg after keyword  arg
>>> f(x = 68, False)
SyntaxError: non-keyword arg after keyword  arg
```

Recommend
火龙果软件

传递函数

- 函数可以像参数一样传递给另外一个函数

Source

```
>>> def addMe2Me(x):
            return (x+x)
>>> def self(f, y):
            print f(y)
>>> self(addMe2Me, 2.2)
4.4
```

Recommend
火龙果软件

lambda函数

- 匿名函数

Source

```
>>> def addMe2Me(x):
        'apply operation +  to argument'
        return (x + x)
>>> addMe2Me(5)
10
```

Source

```
>>> r = lambda x : x + x
>>> r(5)
10
```

lamda函数

```
def my_add(x, y) : return x + y
```

```
lambda     x, y  :  x + y
```

```
my_add = lambda x, y :  x + y
```

```
>>> my_add(3, 5)
8
```

Recommend
火龙果软件

Python大数据编程

递归

Recommend
火龙果软件

递归

循环

生成斐波那契数列的方法

递归

递归是最能表现计算思维的算法之一

循环和递归

- 递归必须要有边界条件，即停止递归的条件
  - n == 0 or n == 1
- 递归的代码更简洁，更符合自然逻辑，更容易理解

Source

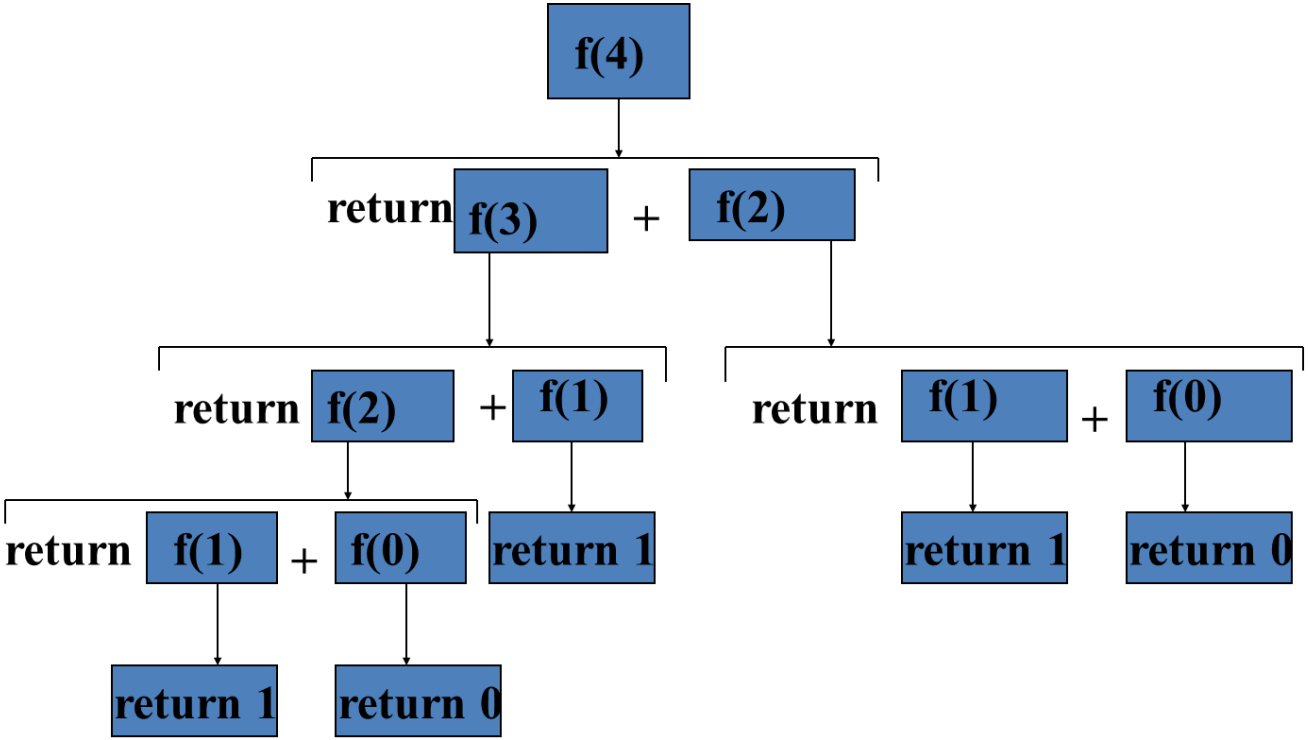
```
# the nth Fibonacci number
def fib(n):
    a, b = 0, 1
    count = 1
    while count < n:
        a, b = b, a+b
        count = count + 1
    print a
```

Source

```
# the nth Fibonacci number
def fib(n):
    if n == 0 or n == 1:
        return n
    else:
        return (fib(n – 1)  +  fib(n – 2))
```

Recommend
火龙果软件

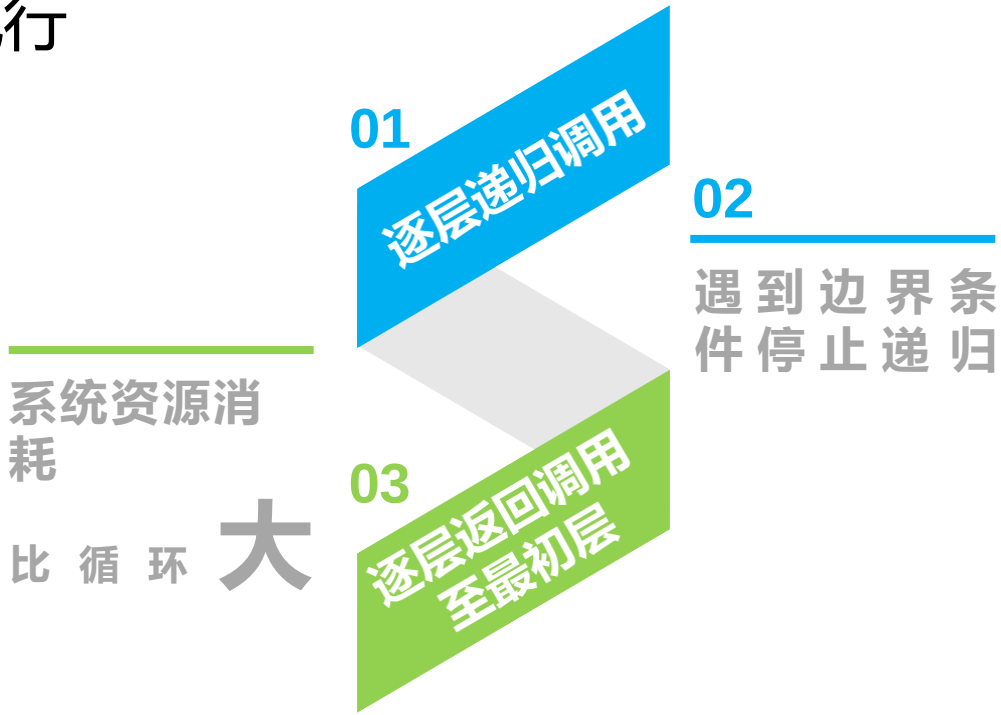
递归



Recommend
火龙果软件

递归

- 递归的执行



汉诺塔

- 汉诺塔游戏

  三个塔座A、B、C上各有一根针，通过C把64个盘子从A针移动到B针上，移动时必须遵循下列规则：

  （1）圆盘可以插入在A、B或C塔座的针上

  （2）每次只能移动一个圆盘

  （3）任何时刻都不能将一个较大的圆盘压在较小的圆盘之上

File

```
# Filename: Hanoi.py
def hanoi(a,b,c,n):
    if n==1:
        print a,'->',c
    else:
        hanoi(a,c,b,n-1)
        print a,'->', c
        hanoi(b,a,c,n-1)

hanoi('a','b','c',4)
```

Output:

```
a -> b
a -> c
b -> c
a -> b
c -> a
c -> b
a -> b
a -> c
b -> c
b -> a
c -> a
b -> c
a -> b
a -> c
b -> c
```

Recommend
火龙果软件

Python大数据编程

变量作用域

Recommend
火龙果软件

变量作用域

- 全局变量
- 局部变量

File

```
# Filename: global.py
global_str = 'hello'
def foo():
    local_str = 'world'
    return global_str + local_str
```

Source

```
>>> foo()
'helloworld'
```

Recommend
火龙果软件

同名变量

- 全局变量和局部变量用同一个名字

File

```
# Filename: samename.py
a = 3
def f( ):
    a = 5
    print a ** 2
```

Recommend
火龙果软件

改变全局变量的值

- 方法是否可行?

File

```
# Filename: scopeofvar.py
def f(x):
    print a
    a = 5
    print a + x

a = 3
f(8)
```

UnboundLocalError: local variable 'a' referenced before assignment

Recommend
火龙果软件

global语句

- global语句强调全局变量

File

```
# Filename: scopeofvar.py
def f(x):
    global a
    print a
    a = 5
    print a + x

a = 3
f(8)
print  a
```

Output:

3
13
5

Recommend
火龙果软件

谢谢

Recommend
火龙果软件