

# Python 数据分析

---

Cloudera 大数据培训基地

重庆翰海睿智大数据科技有限公司

# 交互式计算和开发环境

---

## 交互式计算和开发环境

# 交互式计算和开发环境

---

在本章中，您将了解到：

1. 交互式计算和开发环境的安装
2. 数据分析常用类库
3. IPython 基础运用
4. jupyter notebook web 应用程序的运用
5. 利用 jupyter notebook 文件读取和图表绘制展示
6. markdown 简单语法

## 交互式计算和开发环境

交互式计算和开发环境安装

IPython 基础

内省

使用命令历史

jupyter notebook

结论

# 交互式计算和开发环境介绍 (1)

IPython 是一个交互式计算系统。主要包含三个组件：增加的交互式“Python shell”，解耦的双过程通信模型，交互式并行计算的架构。支持变量自动补全、自动缩进等。

IPython 官方网站: <http://www.ipython.org/>

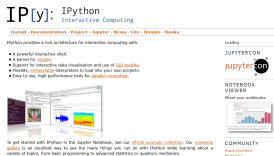


图 1: 'IPython 官网'

## 交互式计算和开发环境介绍 (2)

Jupyter Notebook 是 IPython 团队开始开发一种基于 Web 技术的交互式计算文档格式 (此前被称为 IPython Notebook), 支持运行 40 多种编程语言。

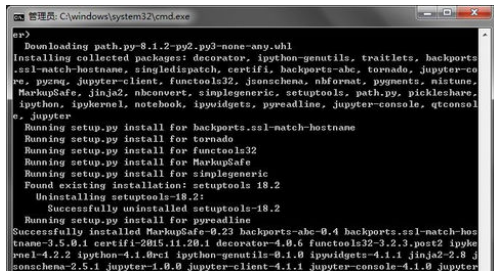
Jupyter Notebook 官方网站:<https://www.jupyter.org/>



图 2: 'Jupyter Notebook 官网'

## 交互式计算和开发环境安装方式

Windows 平台: 运行 cmd 命令窗口, 执行 `pip3 install Jupyter`。



```
管理员: C:\windows\system32\cmd.exe
er>
  Downloading path.py-8.1.2-py2.py3-none-any.whl
Installing collected packages: decorator, ipython-genutils, traitlets, backports
.ssl-match-hostname, singledispatch, certifi, backports-abc, tornado, jupyter-co
re, pyzmq, jupyter-client, funtools32, jsonschema, nbformat, pygments, nistune,
MarkupSafe, Jinja2, nbconvert, simplegeneric, setuptools, path.py, pickleshare,
ipython, ipykernel, notebook, ipywidgets, pyreadline, jupyter-console, qtconsole
e, jupyter
  Running setup.py install for backports.ssl-match-hostname
  Running setup.py install for tornado
  Running setup.py install for funtools32
  Running setup.py install for MarkupSafe
  Running setup.py install for simplegeneric
  Found existing installation: setuptools 18.2
  Uninstalling setuptools-18.2:
    Successfully uninstalled setuptools-18.2
  Running setup.py install for pyreadline
Successfully installed MarkupSafe-0.23 backports-abc-0.4 backports.ssl-match-hos
tname-3.5.0.1 certifi-2015.11.20.1 decorator-4.0.6 funtools32-3.2.3.post2 ipyke
rnel-4.2.2 ipython-4.1.0rc1 ipython-genutils-0.1.0 ipywidgets-4.1.1 Jinja2-2.8 j
sonschema-2.5.1 jupyter-1.0.0 jupyter-client-4.1.1 jupyter-console-4.1.0 jupyter
```

图 3: 'Jupyter Notebook 安装过程'



## 安装科学计算的基础库 Numpy

下载地址: <http://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy>,  
将文件下载至 D 盘DataAnalysisPackage文件夹中。

Windows 平台: 运行 cmd 命令窗口, 执行

`pip3 install numpy-1.13.1+mkl-cp36-cp36m-win_amd64.whl` 本地安装。

```
1 D:\>cd DataAnalysisPackage
2 D:\DataAnalysisPackage>pip3 install
   numpy-1.13.1+mkl-cp36-cp36m-win_amd64.whl
3 Processing
   d:\dataanalysispackage\numpy-1.13.1+mkl-cp36-cp36m-win_amd64.whl
4 ...
5 Installing collected packages: numpy
6 Successfully installed numpy-1.13.1+mkl
```

## 安装科学计算标准工具集合库 Scipy

下载地址: <http://www.lfd.uci.edu/~gohlke/pythonlibs/#scipy>, 将文件下载至 D 盘DataAnalysisPackage文件夹中。

Windows 平台: 运行 cmd 命令窗口, 执行

```
pip3 install scipy-0.19.1-cp36-cp36m-win_amd64.whl
```

```
1 D:\DataAnalysisPackage>pip3 install
    scipy-0.19.1-cp36-cp36m-win_amd64.whl
2 Processing
    d:\dataanalysispackage\scipy-0.19.1-cp36-cp36m-win_amd64.whl
3 ...
4 Installing collected packages: scipy
5 Successfully installed scipy-0.19.1
```

# 安装高级数据结构和操作类库 Pandas

Windows 平台: 运行 cmd 命令窗口, 执行 `pip3 install pandas`

```
1 C:\Users\强>pip3 install pandas
2 Collecting pandas
3   Downloading pandas-0.20.3-cp36-cp36m-win_amd64.whl
   (8.3MB)
4 ...
5   Downloading pytz-2017.2-py2.py3-none-any.whl (484kB)
6 ...
7 Installing collected packages: pytz, pandas
8 Successfully installed pandas-0.20.3 pytz-2017.2
```

## 安装可视化图表类库 Matplotlib

下载地址:

<http://www.lfd.uci.edu/~gohlke/pythonlibs/#matplotlib>, 将文件下载至 D 盘DataAnalysisPackage文件夹中。

Windows 平台: 运行 cmd 命令窗口, 执行

```
pip3 install matplotlib-2.0.2-cp36-cp36m-win_amd64.whl
```

```
1 D:\DataAnalysisPackage>pip3 install
    matplotlib-2.0.2-cp36-cp36m-win_amd64.whl
2 Processing
    d:\dataanalysispackage\matplotlib-2.0.2-cp36-cp36m-win_amd64.whl
3 Requirement already satisfied: pytz in c:\users\强
4 ....
5 Installing collected packages: cycler, matplotlib
6 Successfully installed cycler-0.10.0 matplotlib-2.0.2
```

## 交互式计算和开发环境

交互式计算和开发环境安装

IPython 基础

内省

使用命令历史

jupyter notebook

结论

# 如何启动 IPython

Windows 平台: 运行 cmd 命令窗口, 执行 `ipython`。



```
Microsoft Windows [版本 10.0.14393]  
(c) 2016 Microsoft Corporation. 保留所有权利。  
  
C:\Users\强>ipython  
Python 3.6.1 (v3.6.1:09c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]  
Type 'copyright', 'credits' or 'license' for more information  
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.  
  
In [1]: num1 = 5  
In [2]: num1  
Out[2]: 5  
In [3]:
```

图 4: '启动 IPython'

## IPython 与 Python 进行比较

```
ipython In [4]: from numpy.random import randn In [5]: data =  
{i : randn() for i in range(3)} In [6]: data Out[6]: {0:  
1.6977986467936432, 1: 0.22356532563683298, 2:  
-0.7346524723656888}
```

**注意：** 执行之前需要通过 *pip3* 安装 *NumPy* 类库。

```
python >>> from numpy.random import randn >>> data = {i :  
randn() for i in range(3)} >>> print(data){0:  
1.6977986467936432, 1: 0.22356532563683298, 2:  
-0.7346524723656888}
```

## Tab 键自动完成功能 (1)

IPython 的 Tab 键自动完成功能是对标准 Python Shell 的主要改进之一。只要按下 Tab 键, 当前命名空间中任何与已输入的字符串想匹配的变量(对象、函数等) 就会被找出来。



The screenshot shows a Windows command prompt window titled "IPython: C:\Users\强". The prompt is "C:\Users\强>ipython". The IPython shell version is 6.1.0. The user has entered three lines of Python code: `an_num1 = 20`, `an_num2 = 30`, and `an_`. The IPython shell has automatically completed the third line to `an_num1 any()` and `an_num2 and`. The window title bar shows standard Windows window controls (minimize, maximize, close).

```
IPython: C:\Users\强
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

C:\Users\强>ipython
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: an_num1 = 20
In [2]: an_num2 = 30
In [3]: an_
         an_num1 any()
         an_num2 and
```

图 5: 'Tab 键自动完成方式 (1)'



## Tab 键自动完成功能 (2)

也可以在任何对象后面输入句号 (.) 以便自动完成方法和属性的输入。



```
Python: C:\Users\强
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 保留所有权利。

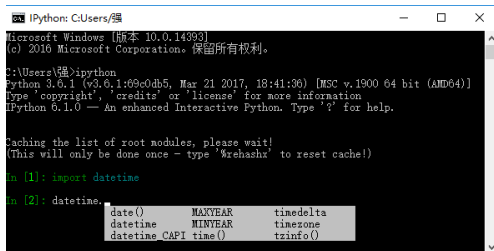
C:\Users\强>ipython
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: list1 = [0,7,8,9]
In [2]: list1.
append()  count()  insert()  reverse()
clear()   extend()  pop()     sort()
copy()    index()   remove()
```

图 6: 'Tab 键自动完成方式 (2)'

## Tab 键自动完成功能 (3)

还可以应用在模块上。



```
IPython: C:\Users\强
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 保留所有权利。

C:\Users\强>ipython
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

Caching the list of root modules, please wait!
(This will only be done once - type '%rehashx' to reset cache!)

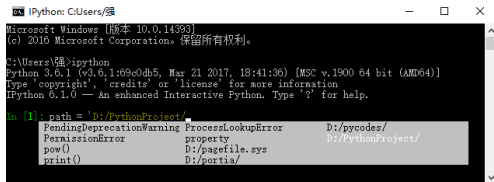
In [1]: import datetime

In [2]: datetime.
date()      MAXYEAR      timedelta
datetime    MINYEAR      timezone
datetime.CAPI time()      tzinfo()
```

图 7: 'Tab 键自动完成方式 (3)'

## Tab 键自动完成功能 (4)

除了上述功能以外，还可以找出电脑文件系统中与之匹配的东西。



```
IPython: C:\Users\强
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 保留所有权利。

C:\Users\强>ipython
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: path = 'D:/PythonProject/'
PendingDeprecationWarning ProcessLookupError D:/pycodes/
PermissionError property D:/PythonProject/
pow() D:/pagefile.sys
print() D:/portis/
```

图 8: 'Tab 键自动完成方式 (4)'

## 交互式计算和开发环境

交互式计算和开发环境安装

IPython 基础

内省

使用命令历史

jupyter notebook

结论

## 显示对象的通用信息 (1)

在变量的前面或者后面加上一个问号 (?) 就可以将有关该对象的一些通用信息进行显示。这称为对象内省。

```
1 In [1]: list1 = [6,7,8,9]
2 In [2]: list1?
3 Type:          list
4 String form: [6, 7, 8, 9]
5 Length:        4
6 Docstring:
7 list() -> new empty list
8 list(iterable) -> new list initialized from iterable's
    items
```

## 显示对象的通用信息 (2)

如果该对象是一个函数或实例方法，则其 docstring 也会被显示出来。

```
1 In [1]: def addition(a,b):
2     ...:     '''
3     ...:     返回执行加法的运算
4     ...:     '''
5     ...:     return a + b
6
7 In [2]: addition?
8 Signature: addition(a, b)
9 Docstring: 返回执行加法的运算
10 File:      c:\users\强\<ipython-input-2-29cdb915fb8a>
11 Type:      function
```

## 显示对象的通用信息 (3)

使用??还可以将该函数的源代码进行显示。

```
1 In [1]: def addition(a,b):
2     ...:     '''
3     ...:     返回执行加法的运算
4     ...:     '''
5     ...:     return a + b
6 In [2]: addition??
7 Signature: addition(a, b)
8 Source:
9 def addition(a,b):
10     '''
11     返回执行加法的运算
12     '''
13     return a + b
14 File:      c:\users\强\<ipython-input-1-29cdb915fb8a>
15 Type:      function
```

## %run 命令

在 IPython 会话环境中，所有文件都可以通过 %run 命令当做 Python 程序来执行。例如：在 D 盘有 test.py 文件中存放了比较简单的脚本。

```
1 In [2]: %run D:\test.py
2 Hello World
3 Hello Python
4 这是一个段落
5 包含了多个语句
6 你好 您好 很好
```



## 键盘常用快捷键

命令	说明
Ctrl + P或上箭头键	后向搜索命令历史中以当前 is 入的文本开头的命令
Ctrl + N或下箭头键	前向搜索命令历史中以当前输入的文本开头的命令
Ctrl + R	按行读取的反向历史搜索 (部分匹配)
Ctrl + Shift + V	从剪贴板粘贴文本
Ctrl + C	中止当前正在执行的代码
Ctrl + A	将光标移动到行首
Ctrl + E	将光标移动到行尾
Ctrl + K	删除从光标开始到行尾的文本
Ctrl + U	清除当前行的所有文本
Ctrl + F	将光标句前移动一个字符
Ctrl + B	将光标句后移动一个字符
Ctrl + I	清屏

## 交互式计算和开发环境

交互式计算和开发环境安装

IPython 基础

内省

使用命令历史

jupyter notebook

结论

## 使用命令历史的作用

---

IPython维护着一个位于硬盘上的小型数据库，其中包含有我们执行过的每条命令的文本。这样做有几个目的。

- 减少按键次数、自动完成并执行之前已经执行过的命令。
- 在会话键持久化命令历史。
- 将输入/输出历史记录到日志文件。

## 输入和输出变量 (1)

在实际开发过程中，如果忘记把函数结果赋值给变量是一件很郁闷的事情。而 IPython 会将输入和输出的引用保存在一些特殊变量中。

```
1 In [1]: 2 ** 6
2 Out[1]: 64
3 In [2]: 5 ** 4
4 Out[2]: 625
5 In [3]: _          # 最近的输出结果保存在_ (一个下划线)
6 Out[3]: 625
7 In [4]: __         # 最近的输出结果保存在__ (两个下划线)
8 Out[4]: 625
```

## 输入和输出变量 (2)

输入的变量保存在名为 `_ix` 的变量中，输出的变量保存在名为 `_x` 的变量中。其中 `x` 是输入或输出行的行号。

```
1 In [5]: username = 'freeman'
2 In [6]: username
3 Out[6]: 'freeman'
4 In [7]: _i6
5 Out[7]: 'username'
6 In [8]: _6
7 Out[8]: 'freeman'
```

## 记录输入和输出

IPython 能够记录整个控制台的会话，包括输入和输出。执行 `%logstart` 即可开始记录日志。

```
1 In [9]: %logstart
2 Activating auto-logging. Current session state plus
   future input saved.
3 Filename      : ipython_log.py
4 Mode          : rotate
5 Output logging : False
6 Raw input log  : False
7 Timestamping  : False
8 State         : active
```

## 交互式计算和开发环境

交互式计算和开发环境安装

IPython 基础

内省

使用命令历史

jupyter notebook

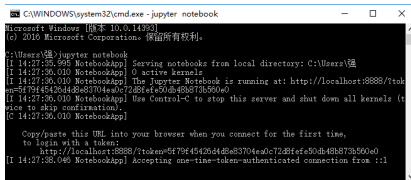
结论

# jupyter notebook 简介

Jupyter Notebook 的本质是一个 Web 应用程序，便于创建和共享文学化程序文档，支持实时代码，数学方程，可视化和 markdown。用途包括：数据清理和转换，数值模拟，统计建模，机器学习等等。

## 启动 jupyter notebook

Windows 平台: 运行 cmd 命令窗口，执行 `jupyter notebook`。



```
C:\WINDOWS\system32\cmd.exe - jupyter notebook
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 保留所有权利。

C:\Users\强>jupyter notebook
[I 14:27:35.595 NotebookApp] Serving notebooks from local directory: C:\Users\强
[I 14:27:36.010 NotebookApp] 0 active kernels
[I 14:27:36.010 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=5179f45426d443e83704e0c7238fefe50db48b873b560e0
[I 14:27:36.010 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 14:27:36.010 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=5179f45426d443e83704e0c7238fefe50db48b873b560e0
[I 14:27:38.046 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

图 9: 'JupyterNotebook 启动命令'



# jupyter notebook 启动后 Web 应用程序

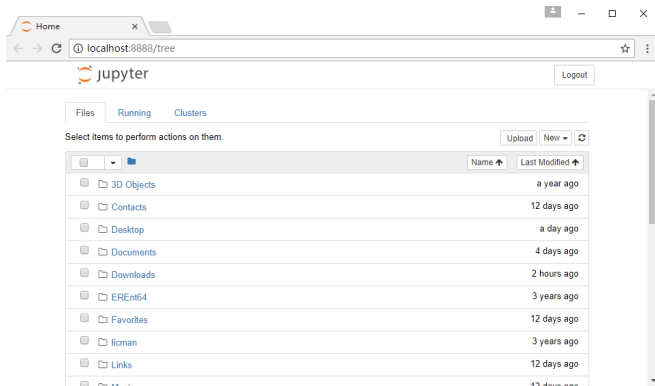
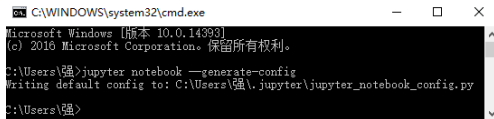


图 10: 'Web 应用程序'

## 配置 Jupyter notebook 路径

Windows 平台: 运行 cmd 命令窗口, 执行

```
jupyter notebook --generate-config
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

C:\Users\强>jupyter notebook --generate-config
Writing default config to: C:\Users\强\.jupyter\jupyter_notebook_config.py

C:\Users\强>
```

图 11: ‘执行命令生成配置文件’

## 查看 Jupyter notebook 生成配置文件

打开“.jupyter”文件夹，可以看到里面有个jupyter\_notebook\_config.py配置文件。

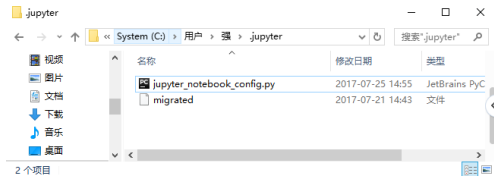


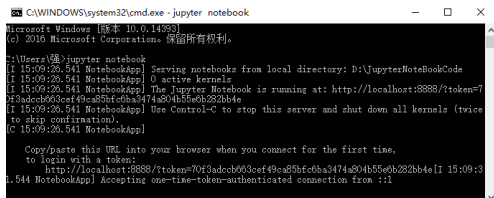
图 12: ‘jupyter notebook 的配置文件’

## 修改 jupyter\_notebook\_config.py 配置文件

1. 打开这个配置文件，找到"#c.NotebookApp.notebook\_dir=……"。
2. 路径改成自己的工作目录为：

```
c.NotebookApp.notebook_dir = 'D:\JupyterNoteBookCode'。
```

3. 重新通过 cmd 命令窗口启动 `jupyter notebook`。



```
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 保留所有权利。

C:\Users\强>jupyter notebook
[I 15:09:20.541 NotebookApp] Serving notebooks from local directory: D:\JupyterNoteBookCode
[I 15:09:20.541 NotebookApp] 0 active kernels
[I 15:09:20.541 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=70f3adccb03cef49ca85bfc0ba3474a804b55e6b282bb4e
[I 15:09:20.541 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://localhost:8888/?token=70f3adccb03cef49ca85bfc0ba3474a804b55e6b282bb4e[I 15:09:31.544 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

图 13: ‘修改配置文件后启动’

# 修改配置文件后 jupyter notebook 的 Web 应用程序

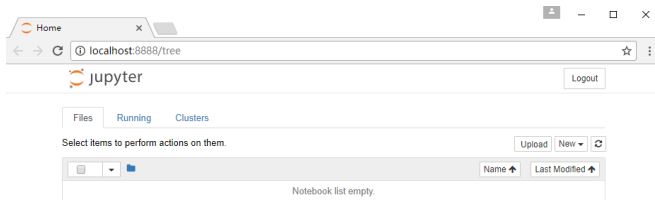


图 14: ‘修改后的 Web 应用程序’

# jupyter notebook 如何新建文件

在主页面的右上角点new下列列表，即可新建想要的文件类型。

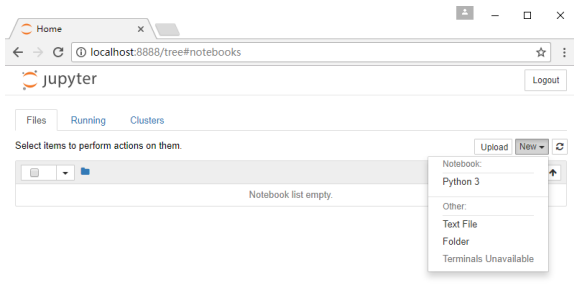


图 15: ‘文件类型’

## 新建 Python3 文件

点击Python3后会在浏览器的新选项卡出现一下界面。

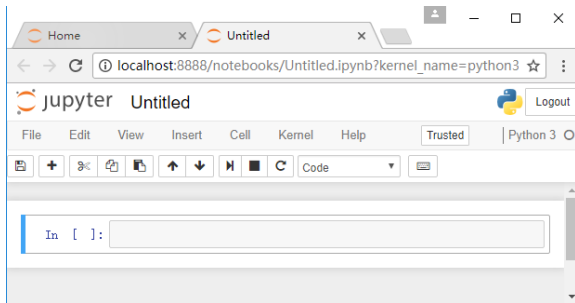


图 16: ‘新建 Python3 文件’

在 *jupyter notebook* 新建的 *Python* 文件后缀为 *.ipynb*。

# 修改新建文件名称

方法一： 点击 Jupyter 图标旁的Untitled

方法二： 点击File >> rename

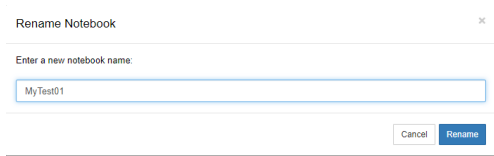


图 17: ‘修改名称对话框’



## jupyter notebook 常用快捷键

- Enter : 转入编辑模式。
- Ctrl + Enter: 执行单元格代码。
- Shift + Enter: 执行单元格代码并且移动到下一个单元格。
- Alt + Enter: 执行单元格代码，新建并移动到下一个单元格。
- A : 在上方插入新单元
- B : 在下方插入新单元
- X : 剪切选中的单元
- C : 复制选中的单元
- Y : 单元转入代码状态
- M : 单元转入 markdown 状态

*jupyter notebook* 其它相关快捷键可以进行百度搜索。

## 单元格格式

---

注意到快捷键栏中有一个 code 的下拉框，点击开发现有几个选项：

- Code 格式就是正常的 python 代码格式。
- Markdown 的一个 text 文档编辑格式，就像在 word 里编写一样。
- Heading 就是给 Markdown 的句子设置标题等级。

## 读取 CSV 文件

```
1 import pandas as pd
2 data = pd.read_csv('commodity.csv', encoding='gbk')
3 data
```

	产品名称	销售时间	销售数量	产品单价
0	手机	2017-3-4	7	1199.0
1	电脑	2017-3-4	3	3238.0
2	冰箱	2017-3-4	6	1599.0
3	洗衣机	2017-3-4	2	1088.0

图 18: 'pandas 读取文件'

## 绘制散点图

```
1 import numpy as np
2 from matplotlib import pyplot as plt
3 plt.figure(figsize=(10,7))
4 n=500
5 x=np.random.randn(1,n)
6 y=np.random.randn(1,n)
7 T=np.arctan2(x,y)
8 plt.scatter(x,y,c=T,s=50,alpha=0.5,marker='o')
9 plt.show()
```

# jupyter notebook 编写 Markdown

```
1 # 一级标题
2 ...
3 ##### 六级标题
4
5 - 无序列表1
6 - 无序列表2
7 - 无序列表3
8
9 1. 有序列表1
10 2. 有序列表2
11 3. 有序列表3
12
13 [百度](http://www.baidu.com)
14 ...
```

# 本章主题

---

## 交互式计算和开发环境

交互式计算和开发环境安装

IPython 基础

内省

使用命令历史

jupyter notebook

结论

## 基本要点

---

- 交互式计算和开发环境的安装
- 数据分析常用类库
- 科学计算的基础库 Numpy
- 科学计算标准工具集合库 Scipy
- 高级数据结构和操作类库 Pandas
- 可视化图表类库 Matplotlib
- IPython 基础运用
- jupyter notebook web 应用程序的运用
- jupyter notebook 启动方式
- Jupyter notebook 路径的配置
- Jupyter notebook 文件及文件夹创建、重命名、删除等
- Jupyter notebook 常用快捷键
- 文件读取和图表绘制