

11.1 使用 Jupyter Notebook



学习目标

- 安装 Jupyter
- Jupyter Notebook 基本使用
- Jupyter 中魔法命令的使用

Jupyter 简介

Jupyter Notebook（此前被称为 IPython notebook）是一个交互式笔记本，支持运行 40 多种编程语言。

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。

安装

<http://jupyter.org>

```
$ sudo pip3 install jupyter
```

出错

```
Cannot uninstall 'ipython'. It is a distutils installed project and thus we cannot accurately determine which files belong to it which would lead to only a partial uninstall.
```

因为我之前通过 apt 安装了 ipython3，所以卸载后重新执行 pip 安装

```
$ sudo apt-get remove ipython3
```

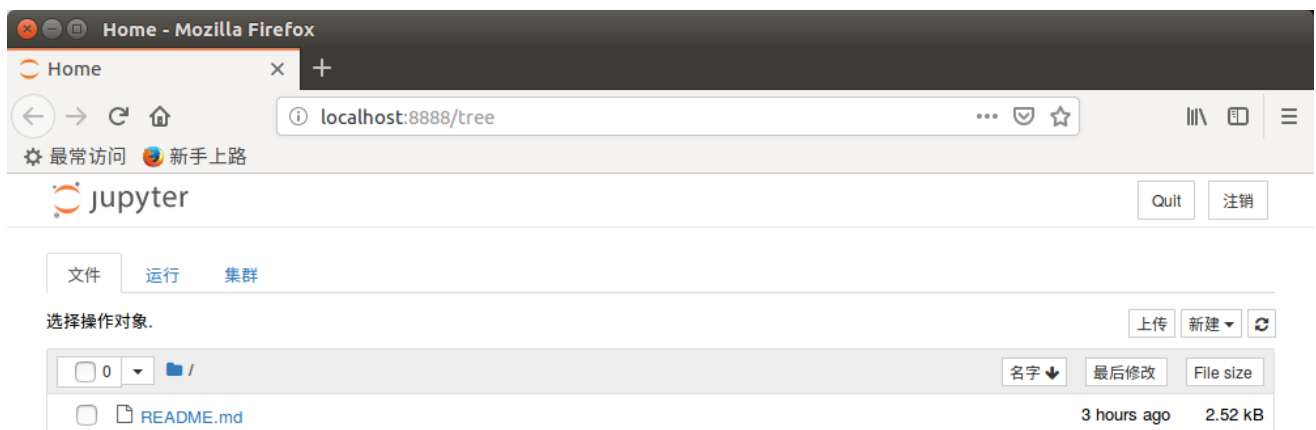
使用

执行 `jupyter notebook` 启动

```
$ jupyter notebook
[I 19:36:11.806 NotebookApp] 把notebook 服务cookie密码写入
/run/user/1000/jupyter/notebook_cookie_secret
[I 19:36:12.020 NotebookApp] 启动notebooks 在本地路径:
/home/rudy/workspace_hd/Python_Learning/14th_week
[I 19:36:12.020 NotebookApp] 本程序运行在: http://localhost:8888/?
token=e36440d9e0adb34adcfacba94f0637b3c59d204059685a78
[I 19:36:12.020 NotebookApp] 使用control-c停止此服务器并关闭所有内核(两次跳过确认).
[C 19:36:12.022 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://localhost:8888/?token=e36440d9e0adb34adcfacba94f0637b3c59d204059685a78
[I 19:36:13.661 NotebookApp] Accepting one-time-token-authenticated connection from
127.0.0.1
```

自动打开浏览器



点击“新建”，选择 Python3，在打开的单元格中就可以编写 Python 代码啦！

快捷键

- `Ctrl + Enter`：只运行单元格，后面不增加新的单元格

- `Shift` + `Enter` : 运行单元格, 运行介绍后跳到下一个单元格, 如果没有则增加一个单元格
- `Alt` + `Enter` : 运行单元格, 不管后面有没有单元格, 都会在后面增加一个单元格
- `Esc` : 进入 Vim 模式

Jupyter 魔法命令

Jupyter 为我们提供了很多特殊的高级用法, 我们把这些命令称为魔法命令。掌握这些魔法命令, 会使我们更好地使用 Jupyter Notebook。

- **%load**

- 作用: 导入文件内容, 并不会执行
- 示例:

```
%load my/hello.py
```

- **%run**

- 作用: 直接运行 Python 文件
- 示例:

```
%run my/hello.py
```

- **%timeit**

- 作用: 循环测试, 获取执行时间, 后面只能跟一句代码
- 示例:

```
%timeit li = [i**2 for i in range(1000)]
```

- 多行模式:

```
%%timeit  
li = []  
for i in range(1000):  
    li.append(i**2)
```

- **%time**

- 作业: 只测量一次, 获取执行时间
- 示例:

```
import random
li = [random.random() for i in range(100000)]
%%timeit li.sort()
%time li.sort()
```

- 多行模式：

```
%%time
```

- **%%html**

- 作用：执行前端代码
- 示例：

```
%%html
<div class='mytest' style='color:red'>
    html test
</div>
```

- **%%js**

- 作用：执行JavaScript代码
- 示例：

```
%%js
document.querySelector(".mytest").innerHTML = 'new content';
```

- **%%writefile**

- 作用：将单元格中的内容写入文件
- 示例：

```
%%writefile "test.py"
import random
li = [random.random() for i in range(100000)]
%time li.sort()
```

总结

魔法命令	作用
<code>%load</code>	导入文件内容，并不会执行
<code>%run</code>	直接运行 Python 文件
<code>%timeit</code>	获取执行时间，后面只能跟一句代码，如果测试多行语句则用 <code>%%timeit</code>
<code>%time</code>	获取执行时间（执行1次），如果测试多行语句则用 <code>%%time</code>
<code>%%html</code>	执行前端代码
<code>%%js</code>	执行JavaScript代码
<code>%%writefile</code>	将单元格中的内容写入文件

在 Jupyter 中，由于 `import` 导入一次后就不会重新导入，即便相应的 python 模块被修改了。这时候，使用 `%run` 就比较灵活。

此外还可以**执行系统命令**，并返回结果。只需在系统命令前面加一个感叹号即可

```
!ls -l
```