

左手代码右手写作-Jupyter Notebook入门

本文主要讲解Jupyter Notebook常用功能的使用方法。



概要

Jupyter Notebook是一个交互式**Python**开发工具，基于Web。有很多魔法 **Magic**，可以帮助你在使用Python进行科学计算的时候拥有更好的用户体验。

[Jupyter Notebook官方文档](#)

不同开发方式的对比

- **PyCharm**
比较智能的Python IDE, 对小白友好.
- **vscode + python3**
编辑好python源文件, 然后在终端里面执行.
- **python3的原生REPL**
交互式, 但是编辑多行代码不方便
- Jupyter Notebook 基于Web的交互式IDE(曾用名 IPython)
交互式, 适合做实验, 交互式查看结果. 或者学习的时候, 试验各种api的效果. 科学计算必备IDE

运行Jupyter Notebook服务器

本质上Jupyter Notebook是在本地搭建了一个服务器，Jupyter是基于Web的一款IDE。我们在前端编辑代码，然后把代码块提交给本地的**Python**解释器解释执行，然后再把输出的信息回传给前端，显示出来。所以为了能够使用Jupyter Notebook，我们必须运行服务器。

在终端输入：

```
$ jupyter notebook
```

输出结果类似这样:

```
→ Python科学计算 jupyter notebook
[I 12:15:43.319 NotebookApp] Writing notebook server cookie secret to
/run/user/1000/jupyter/notebook_cookie_secret
[I 12:15:45.006 NotebookApp] Serving notebooks from local directory: /home/zr/Python科学计算
[I 12:15:45.006 NotebookApp] The Jupyter Notebook is running at:
[I 12:15:45.006 NotebookApp] http://localhost:8888/?
token=fe8ea91edccb2a1c693550b0c5a49b0bf0f7b89052adb8c1
```

点击url `http://localhost:8888/?token=fe8ea91edccb2a1c693550b0c5a49b0bf0f7b89052adb8c1`，或者在浏览器里面输入这段url，你就可以进入到Jupyter Notebook了。另外 `token=fe8ea91edccb2a1c693550b0c5a49b0bf0f7b89052adb8c1` 是web的一种加密技术，确保同一个局域网下的人不会随意进入到你的jupyter后台。

注意：运行之后你所看到的目录，就是你在终端运行 `jupyter notebook` 指令的时候的路径下的所有文件/文件夹列表。

创建一个文件

在 `New` 里面选择 `Python3`，这个时候Jupyter Notebook会创建一个名为 `Untitled.ipynb` 文件。

`Untitled` 是说，你新创建的这个文件还没有命名，你可以点击上方的文件名进行修改。

问题：那么 `.ipynb` 是什么，跟 `.py` 的区别是什么？`ipynb`的英文全称是 `interactive python block`，交互式的python语句块。里面不光保存了python代码，还记录了代码块的执行过程，每个代码块的输出（文本/图像）。它是以 `json` 的格式存储的，我们可以用文本编辑器打开看一下。

Cell-小窗格

Cell在英文里面是细胞的意思，在Jupyter Notebook里面代表小窗格，你可以在里面填写文本 (**Markdown**) 或者是代码(**Code**)，当你把鼠标点击到cell上面，就可以在下拉框看到cell的类别，同时也可以进行转换。

一个IDE当然是用来写代码，所以我们先来介绍代码块。

Code代码块

Run-运行代码块

代码可以是一个块一个块的去执行 在代码块里面填写代码，完成后可以点击工具栏上面的 `Run` 按钮

```
a = 9
b = 2
a + b
```

符号 `In [n]` 代表是第n次输入, 当你把鼠标放在 `In[n]` 标志上面会出现一个运行的按钮, 点击即可运行代码, 另外有没有发现当你第二次运行的时候这个n也会递增。

符号 `Out [n]` 代表是第n次输出, 相当于 `print` 整个语句块的最后一行, 等效为:

```
a = 9
b = 2
print(a + b)
```

```
11
```

Jupyter Notebook有很多快捷键, 你可以点击工具栏上面的键盘**icon**查看所有的快捷键。

使用快捷键 `Ctrl + Enter` 你就可以直接运行代码了。

在后面的课程里面会教大家几个必要的快捷键, 也不需要强迫自己记住, 多用就好。

新建代码块

点击工具栏的 `+` 号, 创建一个新的代码块。快捷键 `Alt + Enter`

```
# new code cell
```

而且你可以通过工具栏的 `上下` 按键调整cell的顺序。

删除代码块

通过工具栏的 `剪刀` 按钮, 你可以删除不用的代码块。快捷键 `Ctrl+X`

测试删除代码块

```
# trash1
```

```
# trash2
```

另外, 如果你一不小心手欠, 还可以通过 `Edit/Undo Delete Cell` 来撤销刚才的操作。

中断代码块

有时候, 你写了一个非常耗时的操作, 这个时候你需要中断代码, 再或者是你手欠写了一个死循环。这个代码块没办法结束, 这个时候, 你就需要中断: 工具栏上面的: 正方形按钮

当代码在执行的时候, 左侧的代码块会显示 `In[*]`。

ps: 教大家另外一个快捷键 选中你要注释的代码区域, 然后 `ctrl + /` 批量添加注释/取消注释

```
# a = 1
# while True:
#     a = a + 1
```

`KeyboardInterrupt` 代表按键中断异常，刚才中断代码，产生了这个按键中断。

关于Kernel

当我们运行一个ipyb的时候，在目录里面，这个文件图标会变成绿色，此时就有一个kernel在处理这个ipyb文件的内容，所有的变量都会被保存。

```
a = 0
```

上面创建了一个变量是 `a`，这个变量会常驻kernel，代码块的执行顺序还有执行次数，都会有影响。例如，多次运行下面的代码块

```
a = a + 1  
print(a)
```

```
1
```

有时候，当你在进行一些不可逆的操作的时候，重复运行代码块就会报错或者得到你不想要的结果。在运行的时候需要注意。

```
# 导入numpy的包  
import numpy as np
```

```
# 定义一个3×1的矩阵A  
A = np.array([  
    [1],  
    [2],  
    [3]])  
  
# 定义一个 3×1的矩阵B  
B = np.array([  
    [1],  
    [1],  
    [1],  
])
```

让A的转置与B点乘 **Dot Product**

```
A = A.T # 注意， 这里我把A的转置重新赋值给了A  
result = A.dot(B)  
print(result)
```

```
[[6]]
```

第一次可以正常运行，第二次就会报错 `ValueError`。上面只是举个例子，大家在使用的过程中，要注意到这些不可逆问题。

重启Kernel

重启kernel，类似重新执行程序，把当前的所有kernel里面的变量，产生的中间结果，输出，都清空。点击工具栏的 `restart the kernel (with dialog)` 就是重启kernel，但是在重启之前会有一个对话框确认，问你是不是真的要重启。

另外，你还可以直接点击 工具栏的 `>>` 按钮，重启kernel，并运行按照顺序，重新执行所有的Cell。

编辑文本 Markdown

Jupyter Notebook除了可以让你编辑代码之外，你也可以往里面插入图片, 文本列表，数学公式（**Latex**）。那我们编辑这些文本的语法称之为**Markdown**. 这个**Markdown**并不是Jupyter Notebook特有的，它是一种文本编辑规范。

双击就可以对Markdown块进行编辑，`CTRL+ENTER` 就是渲染该Markdown代码块为html。

下面列举一些简单的样例：

大标题

1. `#` 1级标题
2. `##` 2级标题
3. `###` 3级标题

文本效果

- `*斜体*` 斜体
- `**文本**` 加粗

插入代码

`a = 7` 代码片段

嵌入代码块

```
# python代码块
a = 1
b = 2
print(a + b)
```

嵌入**Latex**数学公式

Latex是专门用于写论文/数学公式的一种语法.

`$公式片段$` : θ_i

```
$$
这里写公式
$$
```

$$r = \sqrt{x^2 + y^2} \quad (1)$$

嵌入网页链接

[网页名称] (网页url)

[1Z实验室官网](#)

[typora.io](#)

注：Typora是一款非常好用的 Markdown编辑器。

嵌入图片

![图片名称] (图片路径)

 图片名称

文件导出

.ipynb 文件可以导出各种各样的格式。

- .py
- .md Markdown语法
- html
- pdf

导入pdf的话，我一般是先导出 .md，然后再用typora导出pdf。

Magic魔法指令

% 百分号开头的是Jupyter的魔法指令

segmentfault上有对魔法指令的详细介绍。

[官方文档-Jupyter魔术命令](#)

这里讲两个跟Matplotlib有关的魔法指令。

将matplotlib的图片在行内输出，没有工具栏，不可以缩放。

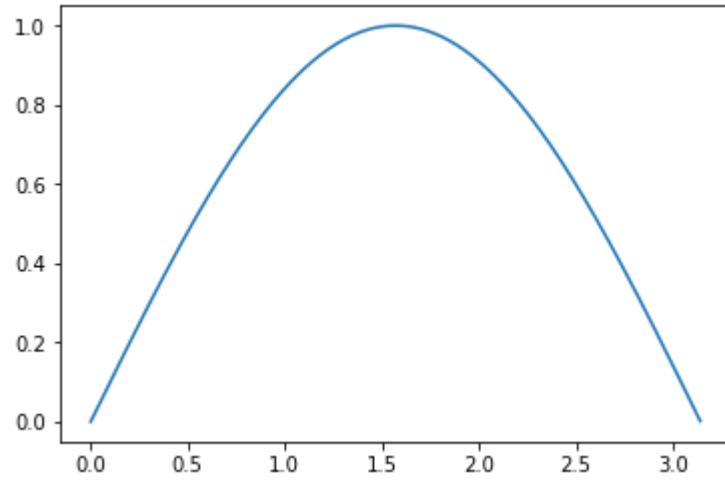
```
%matplotlib inline
```

```
%matplotlib inline
import matplotlib.pyplot as plt
import math
```

```
x = np.arange(0, math.pi, 0.01) # 在0-pi之间 每隔0.01取一个点
y = np.sin(x) # 矩阵y的每一个元素的值都是x元素sin运算的结果
```

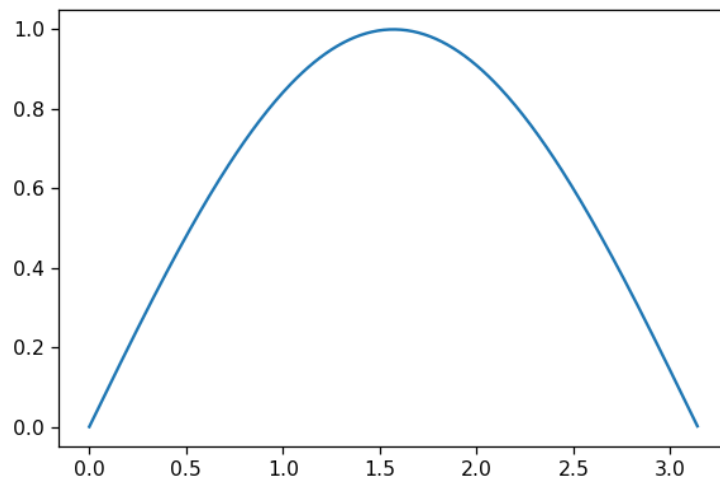
```
# 绘制图像
plt.plot(x, y)
```

```
[<matplotlib.lines.Line2D at 0x7f9ea46ab8d0>]
```



```
%matplotlib notebook  
# 交互式绘图  
plt.plot(x, y)
```

```
<IPython.core.display.Javascript object>
```



```
[<matplotlib.lines.Line2D at 0x7f9ea45173c8>]
```

Widgets-GUI组件

有时候我们需要可视化不同的参数对应的函数输出结果，例如进行调参的时候。`Wigets` 是Jupyter Notebook里面的GUI组件，可以让你短短几行就写出来一个上位机。

[官方文档-ipywidgets入门](#)

```
# 导入交互组件
from ipywidgets import interact, interactive, fixed, interact_manual
import ipywidgets as widgets
```

一个简单的滑动条的例子

```
def f(x):
    '''返回x的平方'''
    result = x**2
    return "x^2 = {}".format(result)

interact(f, x=10);
```

```
interactive(children=(IntSlider(value=10, description='x', max=30, min=-10), Output()),
            _dom_classes=('widget-...
```

一个简单的单选框的例子

```
def student(is_like_study=True):
    if is_like_study:
        print('你是一个好学生')
    else:
        print('没事，有别的爱好也挺好')

interact(student, x=True);
```

```
interactive(children=(Checkbox(value=True, description='is_like_study'), Output()),
            _dom_classes=('widget-inte...
```

这里我们给出一个使用Widgets组件制作RGB彩灯样例. 更多样例，你可以在官网查看：[Jupyter官网文档-Wigets样例](#)

```
%matplotlib inline

def rgb(r, g, b):
    image = np.ones((20, 20, 3))
    image[:] = (r, g, b)
    plt.imshow(image)
    plt.show()
```



```
interact(rgb,  
        r = widgets.FloatSlider(min=0, max=1, step=0.1, value=0, description='红色'),  
        g = widgets.FloatSlider(min=0, max=1, step=0.1, value=0, description='绿色'),  
        b = widgets.FloatSlider(min=0, max=1, step=0.1, value=0, description='蓝色'));
```

```
interactive(children=(FloatSlider(value=0.0, description='红色', max=1.0),  
FloatSlider(value=0.0, description='绿...
```

```
# 使用help方法, 查看API文档  
# help(widgets.FloatSlider)
```