

# Jupyter Notebook的安装与使用

计算导论 WWD

## Jupyter Notebook是什么

Jupyter Notebook的使用界面：



从图片可以看到，Jupyter Notebook是Python的一种集成开发环境（IDE），通过它可以编译与运行Python代码。与其他IDE不同的是，Jupyter Notebook使用文本和代码混合的方式编辑文件。

## 准备工作

阅读文章前你需要掌握：

1. 知道将路径添加至环境变量的方法，并将Python路径添加至环境变量。
2. 对使用pip安装库的方法有简单的了解。

Windows 10的具体操作方法可参阅之前的“Python3安装指引”文档。

验证是否将Python添加至环境变量的方法：

打开cmd，输入 `python`，如果在命令提示符中能够正常打开Python并显示 `>>>` 符号，则说明Python路径已添加至环境变量。若出现 `"python' 不是内部或外部命令，也不是可运行的程序或批处理文件"` 等类似提示，则说明未添加。

# Jupyter Notebook的安装

1. 我们使用pip安装Jupyter Notebook。用管理员身份打开cmd，输入以下命令：

```
python -m pip install Jupyter
```

或者：

```
py -3 -m pip install Jupyter
```

如果由于权限原因导致安装失败，则尝试用管理员身份运行cmd再重新输入。

2. 安装完成时，在cmd中输入 `jupyter notebook`。如果出现“*jupyter' 不是内部或外部命令，也不是可运行的程序或批处理文件*”字样，就说明Jupyter所在路径尚未添加至环境变量，无法正常使用。这时我们需要将Jupyter Notebook的路径添加至环境变量。如果能够正常打开，则说明环境变量已配置好。

Jupyter Notebook程序所在位置位于Python路径下的Scripts文件夹中，将这一路径添加进环境变量即可。（如果你的Python路径是“C:\Program Files\Python36”，则你的Jupyter Notebook路径应为“C:\Program Files\Python36\Scripts”）

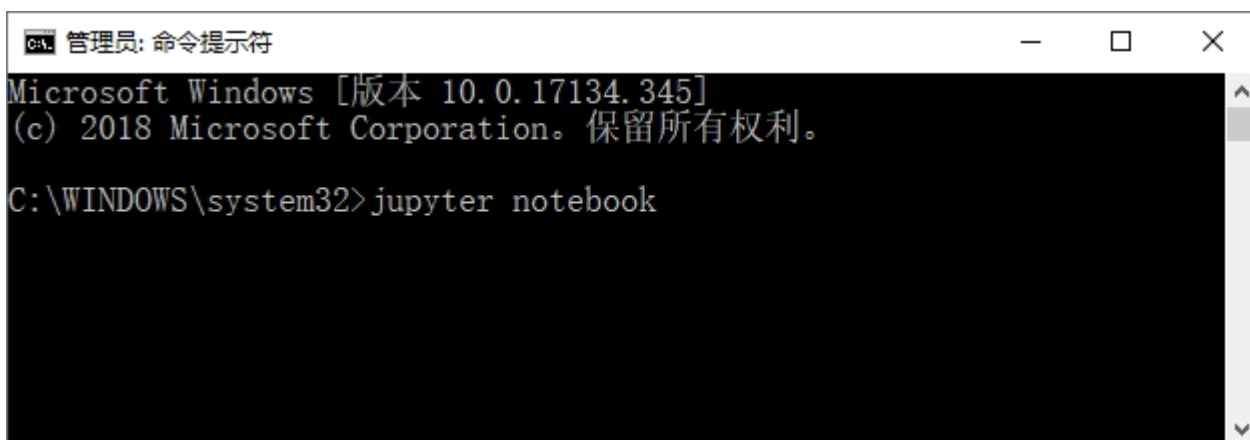
3. 环境变量配置完成后，在cmd中输入 `jupyter notebook`，将会在浏览器中打开Jupyter Notebook（Jupyter Notebook界面依赖于浏览器）。



4. 点击右上角的“Quit”按钮可以退出Jupyter Notebook。如果依靠关闭浏览器的方式关闭Jupyter Notebook的界面，cmd仍然不会退出程序，如需退出Jupyter Notebook，需要在cmd中按下 `Ctrl + c`。

# Jupyter Notebook的路径

1. 从Jupyter Notebook的首页可以看出，Jupyter Notebook的使用目录是当前cmd下的目录：

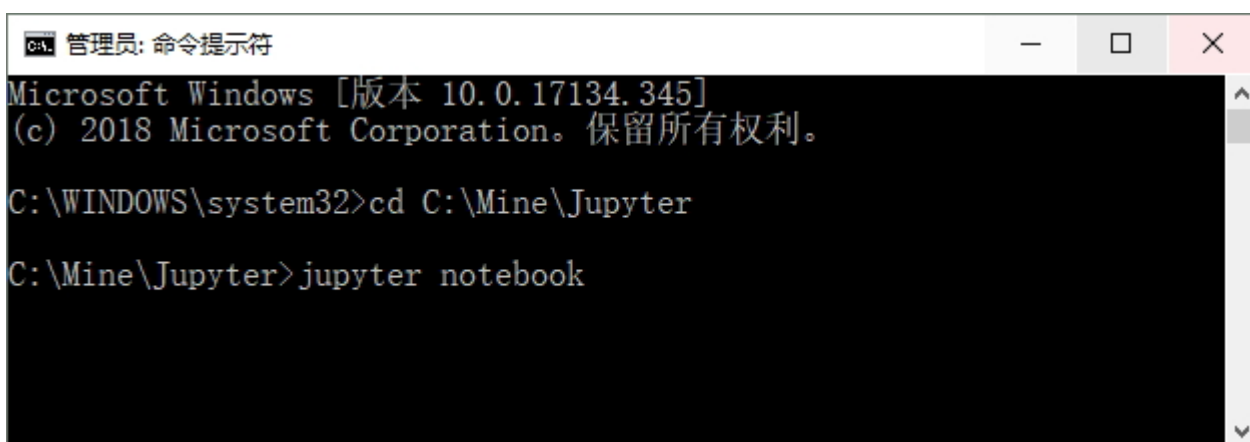


```
管理员: 命令提示符
Microsoft Windows [版本 10.0.17134.345]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>jupyter notebook
```

这时Jupyter Notebook的运行目录是“C:\WINDOWS\system32”，Jupyter Notebook的操作范围限于该目录以及其子文件夹。

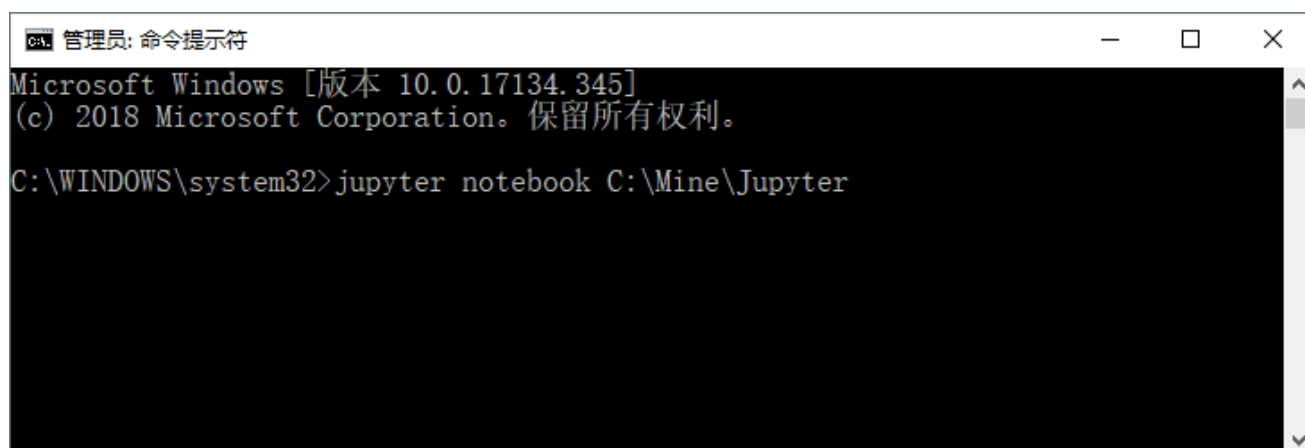
2. 如果我们需要在其他路径下使用Jupyter Notebook编辑文件，可以在cmd中使用 `cd 路径` 命令跳转至其他的文件夹，再输入 `jupyter notebook` 打开Jupyter Notebook。



```
管理员: 命令提示符
Microsoft Windows [版本 10.0.17134.345]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>cd C:\Mine\Jupyter
C:\Mine\Jupyter>jupyter notebook
```

3. 另一种跳转路径的方法是在命令 `jupyter notebook` 后加上路径这一参数：



```
管理员: 命令提示符
Microsoft Windows [版本 10.0.17134.345]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>jupyter notebook C:\Mine\Jupyter
```

如果使用这一种方法，当路径中含有空格时，路径前后需加上英文双引号。

```
C:\WINDOWS\system32>jupyter notebook "C:\Mine\Jupyter\Hello World"
```

4. 我们可以为Jupyter Notebook设置一个默认路径，如果输入 `jupyter notebook` 时不添加参数，则工作路径自动设为用户指定的默认路径而非cmd当前路径。

使用cmd输入命令：

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

可看到cmd列出一个路径，这一路径是Jupyter Notebook的配置文件。

```
C:\WINDOWS\system32>jupyter notebook --generate-config
Writing default config to: C:\Users\...\.jupyter\jupyter_notebook_config.py
C:\WINDOWS\system32>
```

根据该路径找到并用文本编辑器打开这一配置文件。

使用文本编辑器的查找功能，找到这一行：

```
#c.NotebookApp.notebook_dir = ''
```

在单引号内加入需要使用的路径，同时去掉前面的井号，代表激活此项配置：

```
c.NotebookApp.notebook_dir = 'C:\Mine\Jupyter'
```

保存文本后，在cmd中输入不带路径参数的命令 `jupyter notebook`，打开后路径将会是设置的初始目录。

```
管理员: 命令提示符 - jupyter notebook
Microsoft Windows [版本 10.0.17134.345]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>jupyter notebook
[I 19:32:14.387 NotebookApp] Serving notebooks from local directory: C:/Mine/Jupyter
[I 19:32:14.387 NotebookApp] 0 active kernels
[I 19:32:14.387 NotebookApp] The Jupyter Notebook is running at:
[I 19:32:14.387 NotebookApp] http://localhost:8888/?token=8f30c035ead4b2d299cee4acb80b7be7a567e29590d0a2f2
[I 19:32:14.388 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 19:32:14.393 NotebookApp]
```

需要注意的是，如果配置了默认目录，则不能通过 `cd` 来改变Jupyter Notebook的工作目录，只能通过参数改变目录。

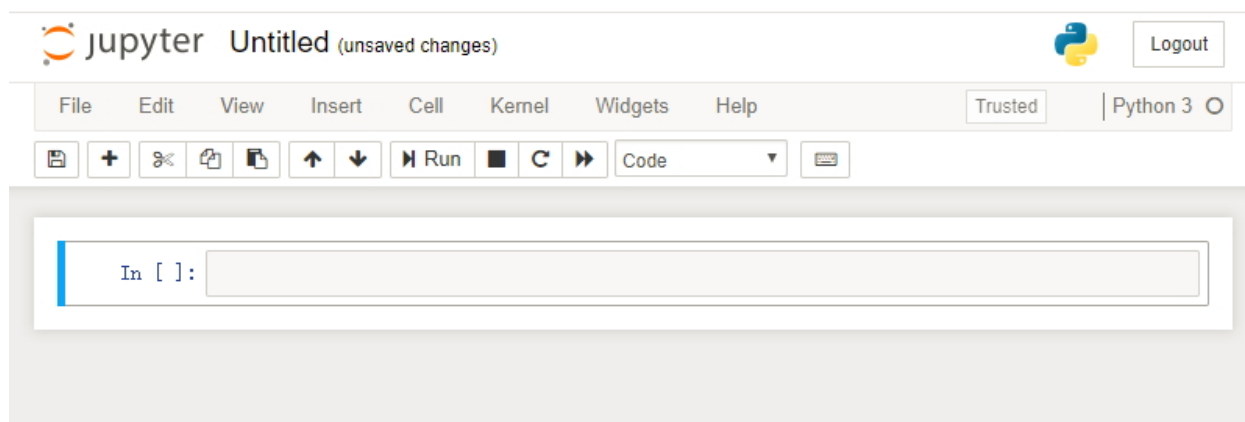
## Jupyter Notebook的使用

### Markdown文本编辑

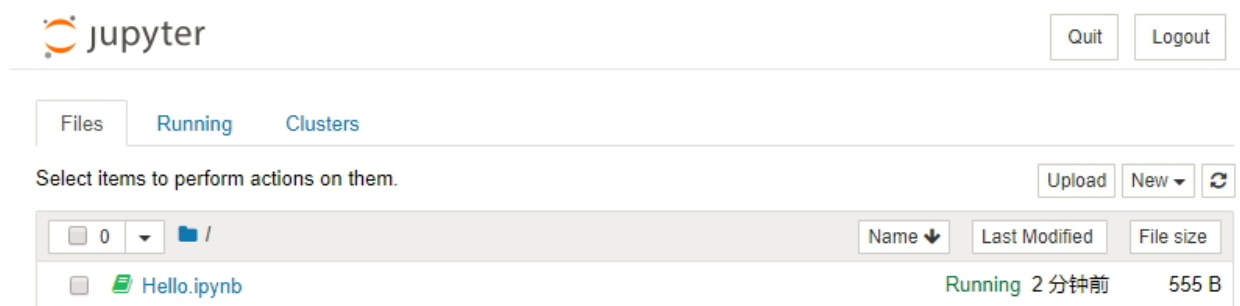
1. Jupyter Notebook = Python + Markdown
2. Markdown是一种特殊的文本编辑方式。有别于普通的文本编辑，Markdown使用了一些特殊符号（例如 `*`、`#`、`_`、`>`）用来规定文本的格式。如果只使用Markdown书写文本而不对文本格式进行编辑，需要注意一些特殊符号可能会被系统理解成其他功能（一般需要使用转义字符 `\` 来输出这些特殊符号）。  
Markdown中文教程详见：<http://www.markdown.cn/>
3. Jupyter Notebook中的Markdown支持HTML语法。

### 新建与关闭文件

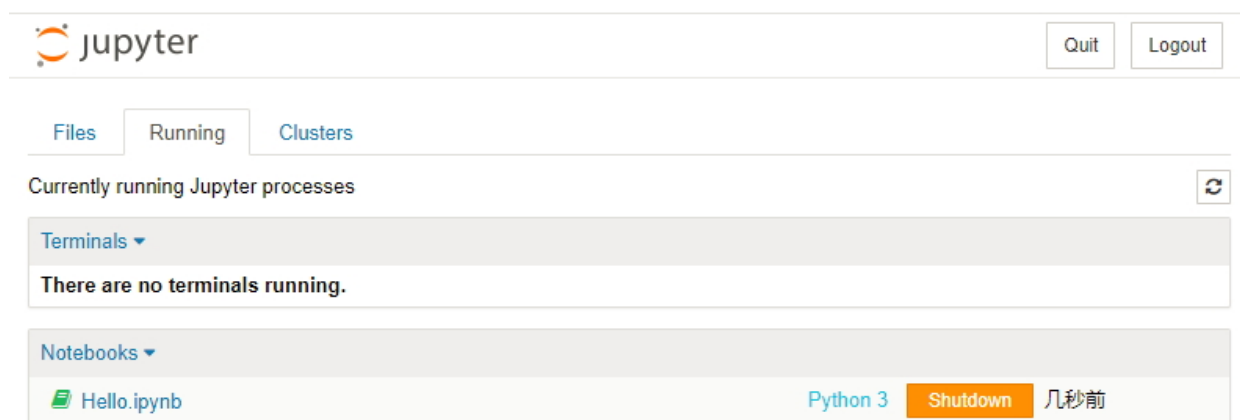
1. 使用cmd在某一目录（不要使用“C:\WINDOWS\system32”这一目录）下运行Jupyter Notebook，点击浏览器中的右上角按钮“New”，在下拉菜单中选择“Python 3”，即可新建文件。此时浏览器弹出新标签，进入文本编辑界面。



2. 点击文本编辑界面上方的File菜单，选择“Rename...”，即可为文件重命名（初始文件名为“Untitled”）。
3. Jupyter Notebook的文件格式为\*.ipynb，本质上是一个文本文件。
4. 不关闭当前标签，将浏览器切换至Jupyter Notebook的首页，可以看见当前文件处于Running状态。

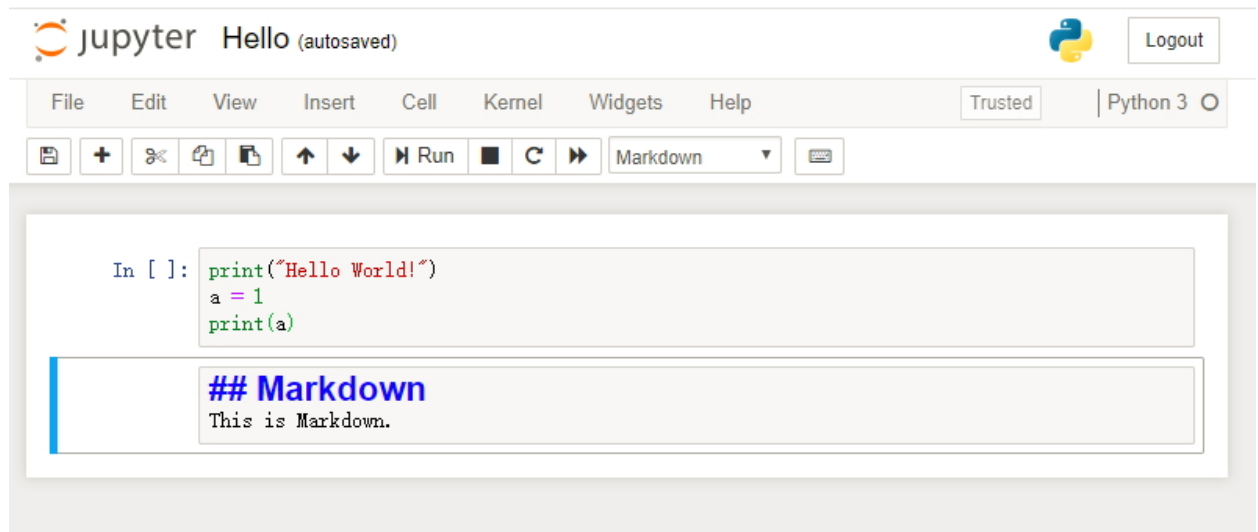


5. 如果我们需要关闭当前文件，点击文本编辑界面上方的File菜单，选择“Close and Halt”，即可结束并退出当前文件。
6. 如果采用浏览器关闭标签的方式关闭当前文件窗口，该文件将不会结束Running状态（说明这个文件没有完全关闭），这时需要在首页上方的Running选项卡中，手动关闭该文件（点击“Shutdown”按钮）。



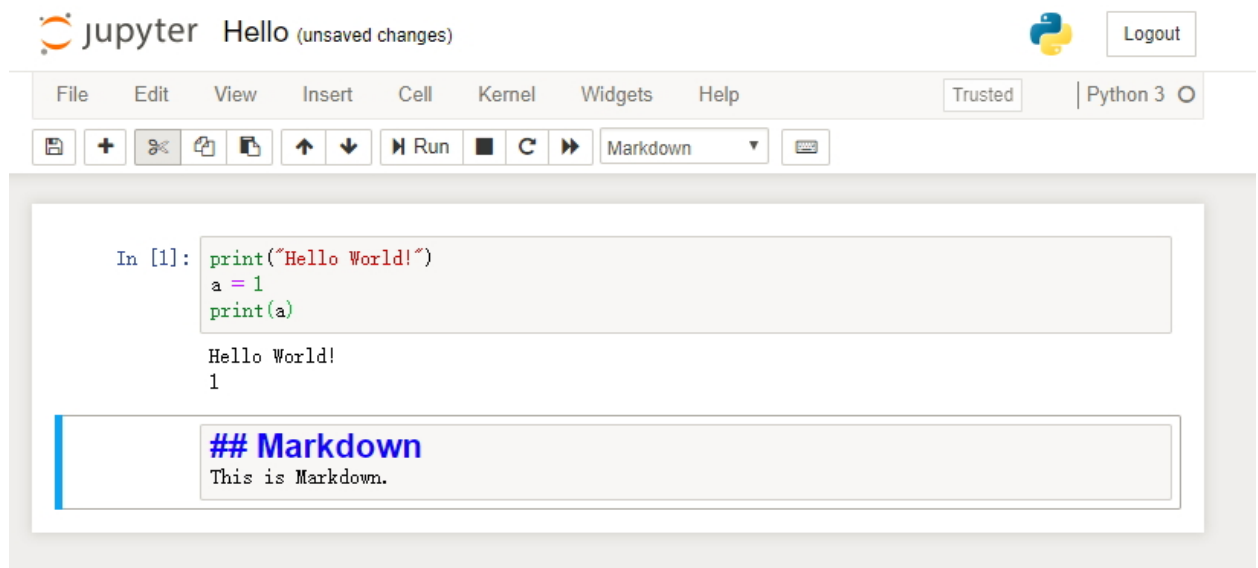
## 编辑文件

1. Jupyter Notebook采用了文本分区的形式，不同的区块可以选择“Code”、“Markdown”模式，分别代表Python代码、Markdown文本。
2. 使用控件栏上的 + 号新增一个区块。新增后可在右侧的下拉框中选择使用Code还是Markdown模式。使用Code模式时，区块左侧会出现“In [ ]”标签。  
下拉框内还有另外两种模式：“Raw NBConvert”以及“Heading”。“Heading”的功能已被包含于“Markdown”中，可以忽略。而“Raw NBConvert”模式下编辑的文本不会被运行（之后会涉及到），因此可以替代“Markdown”进行文本编辑（如果用户不熟悉Markdown的形式，可以采用此模式输入文本）。
3. 现在我们拥有两个区块，一个为Code模式，另一个为Markdown模式。对这两个区块进行文本编辑。



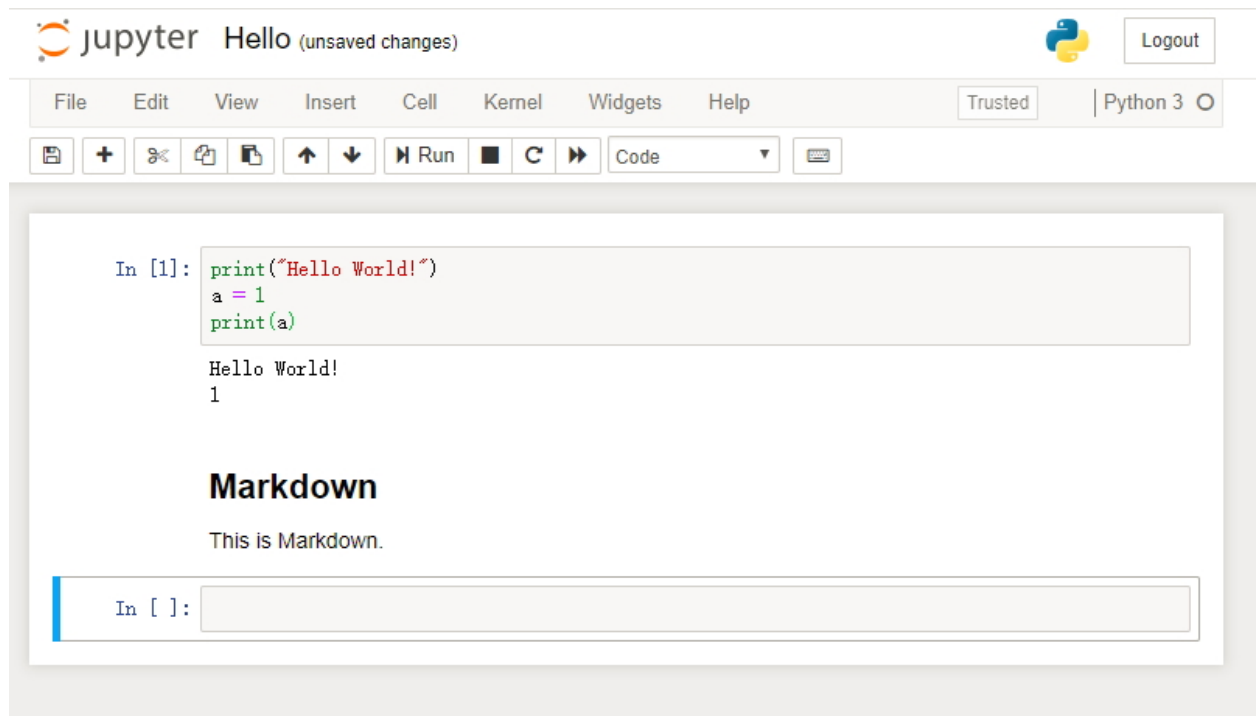
## 运行文件

1. 我们选中Code区块，点击控件栏上的“Run”按钮，即可运行区块内的Python代码。结果如下：



可看到文本输出在代码下方，区块左侧标签中括号内变为1。

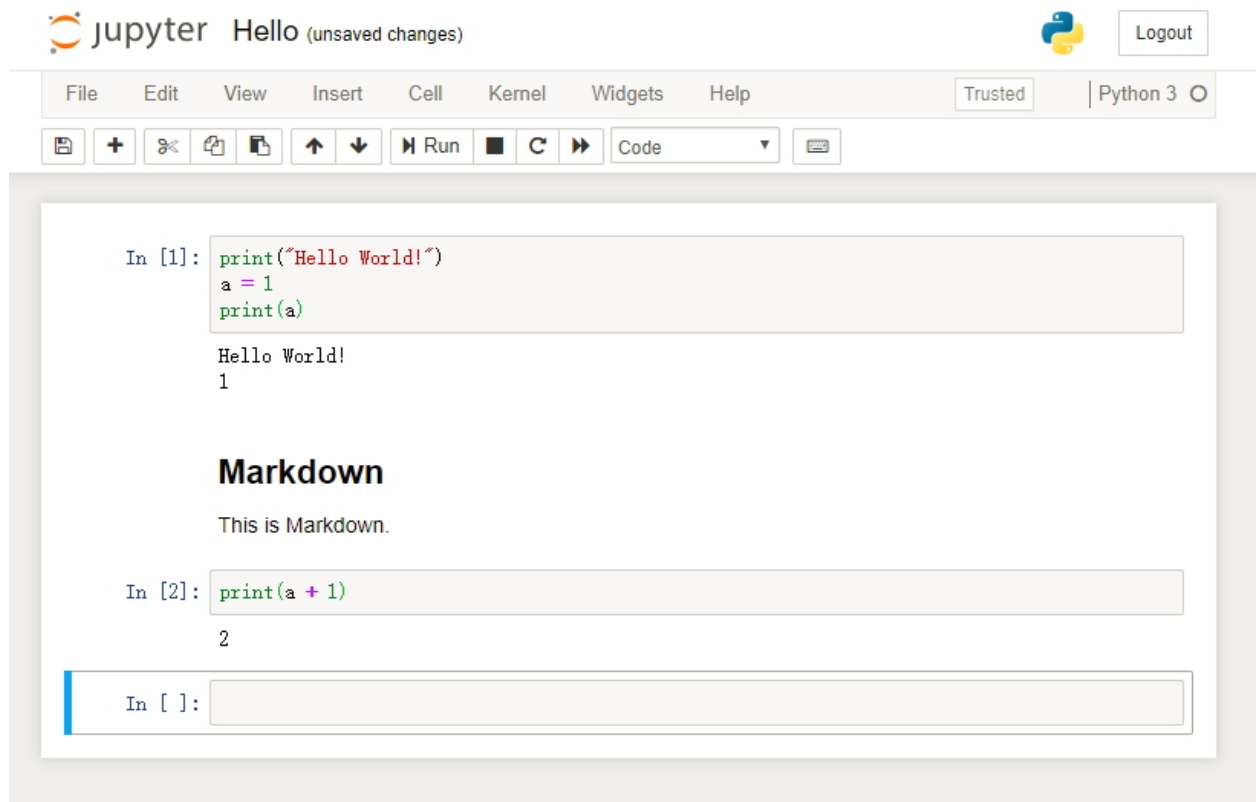
2. 我们运行Markdown模式下的区块：



可以看到文本根据Markdown的标签被转化为相应形式。

3. Raw NBConvert区块无法运行。

4. 不同Code区块可有联系（只要保证不同代码区块间的运行次序即可）。在新增的代码区块调用第一个Code区块内的变量a，程序不会报错：



5. 如果我们需要删除某一区块，可选中该区块，点击菜单栏中Edit的“Delete Cells”选项。

6. 选择控件栏或菜单栏File的相关按钮对文件进行保存。也可使用相应按钮对区块进行剪切复制粘贴等操作。



## 快捷键

使用菜单栏进行一些命令的使用可能十分烦琐，可以使用快捷键来替代这些操作。

使用快捷键的前提是选中一个区块但不让其进入输入状态（即不出现文本编辑光标）。可以通过鼠标点击区块左侧实现。如果正处于该区块的输入状态，按下键盘上的 `ESC` 键退出输入状态。

退出输入状态后，可通过键盘上的快捷键对区块进行操作。快捷键可查看菜单栏Help下的“Keyboard Shortcuts”。下方列出部分快捷键：

- **Enter** : 转入编辑模式
- **Shift-Enter** : 运行本单元，选中下个单元
- **Ctrl-Enter** : 运行本单元
- **Alt-Enter** : 运行本单元，在其下插入新单元
- **Y** : 单元转入Code状态
- **M** : 单元转入Markdown状态
- **R** : 单元转入Raw状态
- **Up / K** : 选中上方单元
- **Down / J** : 选中下方单元
- **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
- **Shift-V** : 粘贴到上方单元
- **V** : 粘贴到下方单元
- **Z** : 恢复删除的最后一个单元
- **D,D** : 删除选中的单元
- **S** : 保存文件
- **L** : 显示或隐藏行号