《Jupyter教程》_02_Jupyter介绍

Author:{时海涛:Thomas.Shih}
Email:147080896@qq.com
Website:https://www.cnblogs.com/noah0532/
Github:https://github.com/Thomas-Shih/JupyterLesson

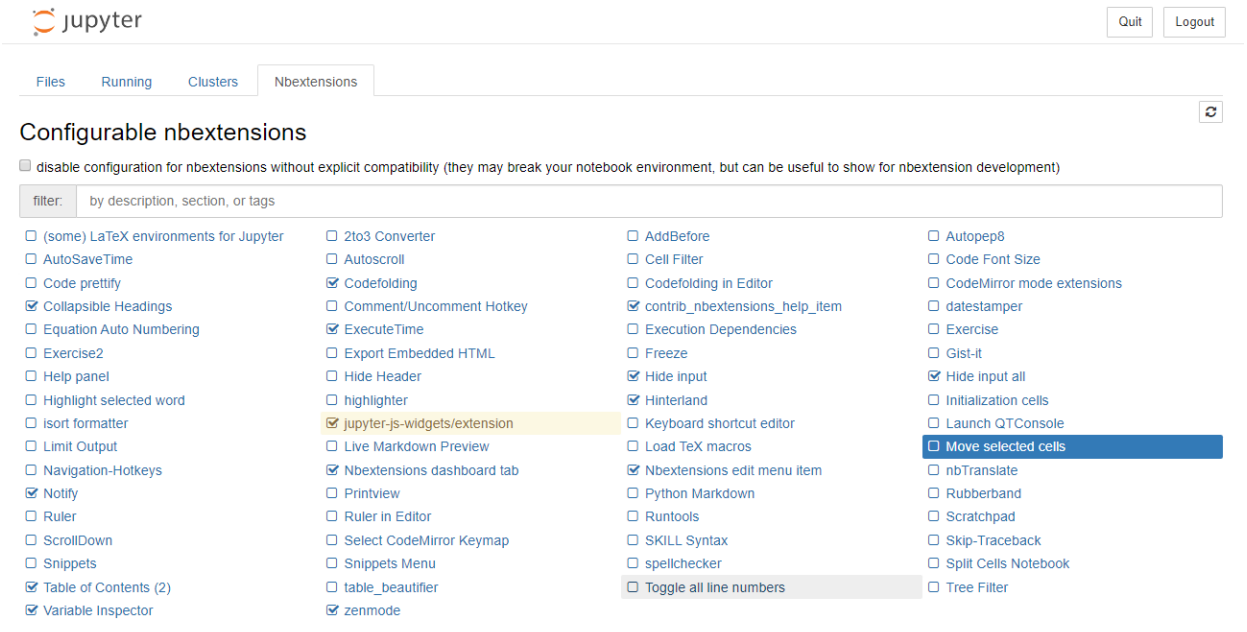
3. Jupyter Notebook的扩展功能:

有些人在使用Jupyter Notebook的时候发现没有相关的提示功能，并不像其他的Python的IDE环境一样辅助功能比较便捷。在这里Jupyter Notebook提供了一个扩展功能，方便我们实现想要的辅助。

这里我们需要安装Jupyter Notebook的扩展插件。如果你是通过Anaconda方式安装的Jupyter Notebook，需要在Anaconda的conda prompt命令创建进行安装，命令也只需将pip替换成conda即可。

```
安装扩展包:
pip install juypter_contrib_nbextensions
安装js和cssfiles：目的是按照后有些内容可能无法下拉
jupyter contrib nbextension install --user
安装配置文件:
pip install jupyter_nbextensions_configurator
```

安装完成后并重启Jupyter Notebook会出现第四个Nbextensions选项:



在这里包括代码的提示、注释内容等等非常全面。上图我已挑选了常用的内容。另外，选择每一项后在下方会有相关使用帮助可以观看。

4. Jupyter Notebook的配置：

之前我们提到过，当前Jupyter Notebook是默认在安装的路径目录下面的，也就是说你编写的任何文件是存储在这里的。一般如果是Linux的话会在家目录下，Windows的话会默认在个人C盘的某一个目录。另外，电脑上如果安装了多个浏览器，比如360浏览器、chrome、Firefox等等，我们想指定的浏览器打开都是通过配置文件进行修改的。常用的配置文件修改主要有创建默认目录和默认浏览器我们分别可以对它们进行修改。

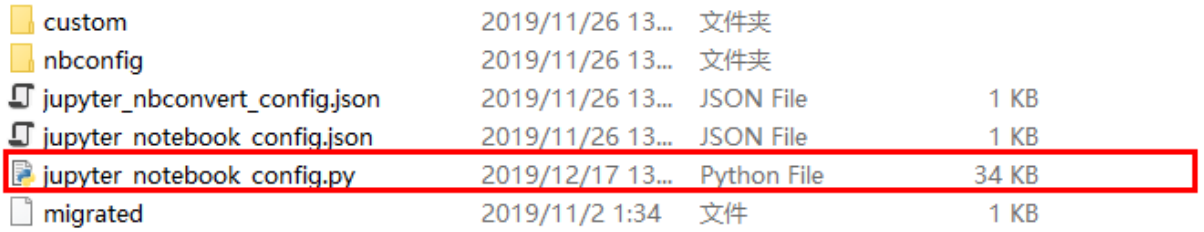
第一：目录修改

方式1：

通过终端输入命令，打开配置文件或者找到Jupyter的配置文件用记事本打开它。

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

或者



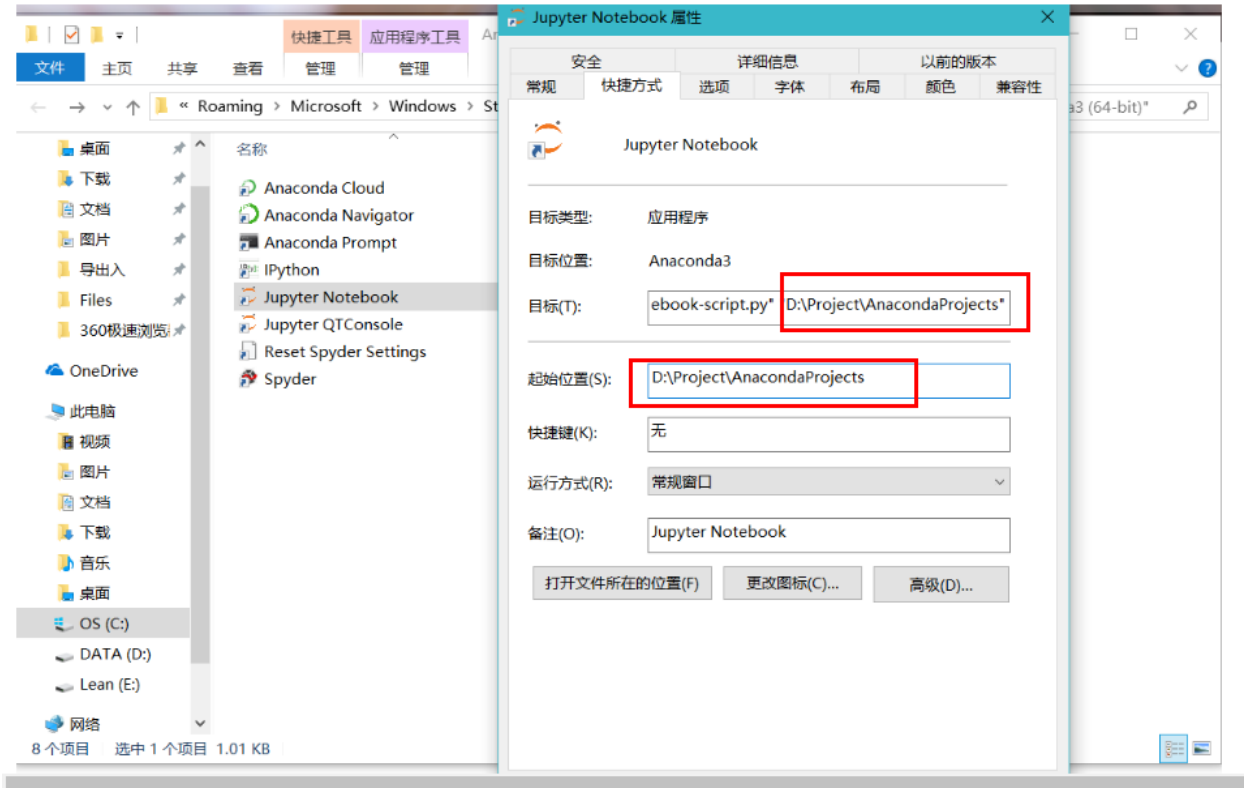
查看内容找到notebook的目录地址

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

注意：要修改目录首先要在你想要创建目录的硬盘位置创建一个文件夹。去掉前面的"#"注释，填写创建文件夹的路径。一般在Windows下面用双斜杠"\\"表示路径，在Linux下面一般用反斜杠"/"表示路径。

方式2：

我们还可以通过快捷方式属性的的方式进行修改。一般是把默的%USERPROFILE%修改掉。对应创建文件的目录地址：



第二：默认浏览器修改

默认浏览器的修改也是在配置文件中进行修改。找到文件中的这个位置：

```
# c.NotebookApp.browser
```

然后输入如下内容（在这类是修改为chrome浏览器），同样一般在Windows下面用双斜杠"\\"表示路径，在Linux下面一般用反斜杠"/"表示路径。

```
import webbrowser
webbrowser.register(''chrome'', None,
webbrowser.GenericBrowser(u''C:\Program Files
(x86)\Google\Chrome\Application\chrome.exe''))
c.NotebookApp.browser = ''chrome''
```

关闭Jupyter Notebook再重启就可以实现浏览器的修改了。

5. Jupyter Notebook主题修改：

安装完毕Jupyter Notebook之后，默认的主题是黑白配的样式，有些人感到比较的单调。我们可以通过安装相关的主题进行修改，同样如果是安装的Anaconda的话，把pip命令修改为conda，并在conda prompt中打开即可。

```
pip install --upgrade jupyterthemes
恢复
jt -r
```

查看可以用
jt -l
更改：

jt XXX

下面了列举了一些字体、行的修改命令，方便查找。

Command Line Usage

```
jt  [-h] [-l] [-t THEME] [-f MONOFONT] [-fs MONOSIZE] [-nf NBFONT]
    [-nfs NBFONTSIZE] [-tf TCFONT] [-tfs TCFONTSIZE] [-dfs DFFONTSIZE]
    [-m MARGINS] [-cursw CURSORWIDTH] [-cursc CURSORCOLOR] [-vim]
    [-cellw CELLWIDTH] [-lineh LINEHEIGHT] [-altp] [-altmd] [-altout]
    [-P] [-T] [-N] [-r] [-dfonts]
```

Description of Command Line options

cl options	arg	default
Usage help	-h	--
List Themes	-l	--
Theme Name to Install	-t	--
Code Font	-f	--
Code Font-Size	-fs	11
Notebook Font	-nf	--
Notebook Font Size	-nfs	13
Text/MD Cell Font	-tf	--
Text/MD Cell Fontsize	-tfs	13

6. Jupyter Notebook快捷键：

Ctrl + a	将光标移动到本行的开始处
Ctrl + e	将光标移动到本行的结尾处
Ctrl + b（或左箭头键）	将光标退一个字符
Ctrl + f（或右箭头键）	将光标前进一个字符

Backspace键	删除前一个字符
Ctrl + d	删除后一个字符
Ctrl + k	从光标开始剪切至行的末尾
Ctrl + u	从行的开头剪切至光标
Ctrl + y	yank（既粘贴）之前剪切的文本
Ctrl + t	transpose（既交换）前两个字符
Ctrl + p（或向上箭头）	获取前一个历史命令
Ctrl + n（或向下箭头）	获取后一个历史命令
Ctrl + r	对历史命令的反向搜索
Ctrl + l	清除终端屏幕的内容
Ctrl + c	中断当前的Python命令
Ctrl + d	退出IPython会话

另外，如果像定义自己的快捷键的话，可以在菜单栏的Help位置点击"Keyboard Shortcuts"进入并编辑自己的快捷键

7. Jupyter Notebook两种模式:

7.1 命令模式:

命令模式将键盘命令与Jupyter Notebook笔记本命令相结合，可以通过不通键的组合运行笔记本的命令。命令模式下，单元边框为灰色，且左侧边框线为蓝色租线条，按Esc键进入命令模式。

7.2 编辑模式:

编辑模式使用户可以在单元格内进行代码和文档的编辑，按Enter和Esc键进入编辑模型，编辑模式下，单元格边框和左侧边框线为绿色粗线条。

8. Jupyter Notebook 内核的切换:

Jupyter Notebook默认内核是电脑中已经安装好的Python版本或者Anaconda安装的Python版本，如果我们电脑安装了不同的Python版本，需要切换或者安装内核该如何操作，以Python2和Python3为例：

8.1 在Python 3中创建Python 2内核

pip安装：

首先安装Python2的ipykernel包

```
python2 -m pip install ipykernel
```

再为当前用户安装Python2的内核

```
python2 -m ipykernel install --user
```

注意："--user"参数的意思是针对当前用户安装，而非系统范围安装。

conda安装：

首先创建Python版本为2.X具有ipykernel的新环境，其中'<env_name>'为自定义环境名，环境名不加尖括号。

```
conda create -n <env_name> python=2 ipykernel
```

然后切换至新创建的环境。

```
Windows: activate <env_name>
Linux/macOS: source activate <env_name>
```

为**当前用户**安装Python 2的内核（ipykernel）

```
python2 -m ipykernel install --user
```

8.2 在Python 2中创建Python 3内核

pip安装：

首先安装Python3的ipykernel包

```
python3 -m pip install ipykernel
```

再为**当前用户**安装Python 2的内核（ipykernel）。

```
python3 -m ipykernel install --user
```

注意："--user"参数的意思是针对当前用户安装，而非系统范围安装。

conda安装：

首先创建Python版本为3.x且具有ipykernel的新环境，其中“<env_name>”为自定义环境名，环境名两边不加尖括号“<>”。

```
conda create -n <env_name> python=3 ipykernel
```

然后切换至新创建的环境。

```
Windows: activate <env_name>
Linux/macOS: source activate <env_name>
```

为**当前用户**安装Python 3的内核（ipykernel）。

```
python3 -m ipykernel install --user
```

8.3 为不同环境创建内核

切换至安装内核的环境：

```
Windows: activate <env_name>
Linux/macOS: source activate <env_name>
```

注意："<env_name>"是需要安装内核的环境名称，环境名称两步不加尖括号。

检查该环境是否安装了ipykernel包

```
conda list
```

执行上述命令查看当前环境下安装的包，若没有安装ipykernel包，则执行安装命令；否则进行下一步。

```
conda install ipykernel
```

为当前环境下的当前用户安装Python内核：

若该环境是Python版本2.X，则执行命令：

```
python2 -m ipykernel install --user --name <env_name> --display-name "
<notebook_name>"
```

若该环境的Python版本为3.X，则执行名：

```
python3 -m ipykernel install --user --name <env_name> --display-name "
<notebook_name>"
```

注意:

1. “<env_name>”为当前环境的环境名称。环境名两边不加尖括号“<>”。
2. “<notebook_name>”为自定义显示在Jupyter Notebook中的名称。名称两边不加尖括号“<>”，但双引号必须加。

3. “--name”参数的值，即“<env_name>”是Jupyter内部使用的，其目录的存放路径为~/Library/Jupyter/kernels/。如果定义的名称在该路径已经存在，那么将自动覆盖该名称目录的内容。

4. “--display-name”参数的值是显示在Jupyter Notebook的菜单中的名称。

8.4 检查

使用命令jupyter notebook启动Jupyter Notebook；在“Files”下的“New”下拉框中即可找到你在第(3)步中的自定义名称，此时，你便可以尽情地在Jupyter Notebook中切换环境，在不同的环境中创建笔记本进行工作。

9. Jupyter Notebok内核切换常见问题：

你创建了一个新环境，且返现在Jupyter Notebook的New中找不到这个环境，无法在该环境中创建笔记本。

进入Jupyter Notebook -- conda -- 在conda environment中点击你要添加的ipykernel包的环境 -- 左下方搜索框输入 ipykernel -- 勾选 ipykernel -- 点击搜索框旁的 -- 箭头 -- 安装完毕 -- 右下方框内找到ipykernel说明已经安装成功，重启Jupyter Notebook