下载miniconda

<https://mirror.tuna.tsinghua.edu.cn/help/anaconda/>

------------------------------------------------------ 安装jupyter

临时使用

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple some-package jupyter

设为默认

升级 pip 到最新的版本

pip install pip -U

pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple

如果您到 pip 默认源的网络连接较差，临时使用本镜像站来升级 pip：

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pip -U

--------------------------------

jupyter notebook界面汉化

我的电脑-高级系统设置-环境变量-新建

在环境变量中的用户变量中新建一个变量。

变量名：LANG

值：zh\_CN.UTF8

-------------------------------

anaconda conda 切换为清华源

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/

conda config --append channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/fastai/

conda config --append channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/

conda config --append channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/

# 搜索时显示通道地址

conda config --set show\_channel\_urls yes

**移除清华源**

输入：conda config --remove channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/

这个命令是为了移除之前conda config --show channels显示的清华源

--------------------------------

[Github镜像网站访问 v1.1.1 油猴脚本 下载速度太慢提速](http://qianfangzy.com/44820.html)

--------------------------------

jupyter安装nbextensions扩展

安装Nbextensions

以管理员的方式打开Anaconda Prompt

pip install jupyter\_contrib\_nbextensions

jupyter contrib nbextension install --user

安装nbextensions\_configurator的命令

pip install jupyter\_nbextensions\_configurator

jupyter nbextensions\_configurator enable --user

在指定文件夹下启动Jupyter Notebook

首先，切换盘符，直接输入“d:”；

然后，使用“cd”指令进入指定的文件夹；

最后，输入“jupyter notebook”启动程序。

配置自动代码自动补全功能

去掉勾disable

勾选 Hinlerland

-------------------------------------------------------

**安装 nbextensions 插件**

一、打开Anaconda Prompt窗口，执行第一个命令，用于安装nbextensions：

pip install jupyter\_contrib\_nbextensions

二、再执行第二个命令，用于安装 javascript and css files

jupyter contrib nbextension install --user

三、最后执行，用于安装configurator

pip install jupyter\_nbextensions\_configurator

四、如果出错的话，先执行卸载命令，再执行上面命令，再试一次：

pip uninstall jupyter\_contrib\_nbextensions
pip uninstall jupyter\_nbextensions\_configurator

1

2

若是pip连接超时，采用清华镜像安装

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple jupyter\_contrib\_nbextensions
*# 超时可以加--default-timeout=10000*

1

然后重新启动Jupyter Notebook后，就会发现已经有Nbextensions标签了。

----------------------------------------------------------------

Jupyter notebook改变默认的工作目录

查看配置文件位置

jupyter notebook --generate-config

Writing default config to: C:\Users\jiangshan\.jupyter\jupyter\_notebook\_config.py

打开jupyter\_notebook\_config.py找到如下

## The directory to use for notebooks and kernels.

#c.NotebookApp.notebook\_dir = ''这一行修改如下：

windows是双斜杠（\\），linux是反斜杠（/）

c.NotebookApp.notebook\_dir = 'F:\\workspace'

----------------------------------------------------------------

Jupyter修改主题
在conda环境运行如下命令，安装主题并列出可选主题：
pip install jupyterthemes
jt -l
选择自己想要的主题，并修改字体间距等（详细命令可参考序中提到的github大佬）
懒人可直接尝试我的配置，看是否和你的胃口：

黑色主题
jt -t onedork -f hack -fs 13 -ofs 11 -cellw 90% -dfs 11 -lineh 140 -T -N

jt -t **gruvboxl** -f hack -fs 13 -ofs 11 -cellw 90% -dfs 11 -lineh 140 -T -N

恢复默认，则jt -t

重启jupyter

---------------------------------------------------------------------------

**Jupyter NBextensions 插件介绍**

<https://www.jianshu.com/p/f2adb4f64051>

常用插件

1111111111111111111111111111111

Code Prettifier ---- 重新格式化，美化代码

Collapsible Headings ---- 可折叠标题，可以根据headings折叠区域；

Notify 通知机制，跑一些耗时较久的任务，完成后通知

ScrollDown，当代码输出内容过长，自动下拉滚动条；

Snippets---- 折叠一个标题下的全部内容+快速插入代码段+拆分单元格

插入代码段，需要自己修改配置文件，路径为..\Anaconda3\share\jupyter\nbextensions\snippets\snippets.json，配置文件里有例子可以参考，添加了代码段后，在notebook界面有Snippets下拉菜单选择自己的代码段，自动插入notebook中

Table of Contents (2) ----显示目录,找到所有标题并生成目录，且该目录可以拖拽移动或放在侧边栏

Variable Inspector
所有变量名、类型，大小，形状和值。其中变量的 shape (形状)

222222222222222222222222

Codefolding ---- 代码折叠，将一个循环、条件、函数、类甚至一个代码块的代码折叠起来，节约页面，看起来比较简洁

ExcecuteTime ---- 计算每一个模块的时间，和运行结束时间，显示运行的时间

Highlighter ---- 荧光笔

Zenmode ---- 隐藏活动状态栏，方便注意代码

Snippets Menu ---- 设置自定义代码片段

勾选  Insert snippets as new cells, rather than at cursor inside current cell

勾选 Include custom menu content parsed from JSON string below

333333333333333333333

AddBefore---- 在按钮栏加入两个按钮, 分别在当前cell前和后插入新Cell.，补充了只能在后面插入的缺点。

Hinterland — 代码自动提示，无需Tab键

Runtools------运行工具栏

Toggle all line numbers ----
可以加入状态栏按钮或者快捷键Option + N 来显示Cell里面的代码行编号，用途一般。
4444444444444444

Autopep8---- 格式化代码，需要安装Autopep8库，pip install autopep8 yapf

出现一个小锤头工具。点击它就能把当前cell的代码自动格式化变整齐

使用前要pip install autopep8 --user

Code Font Size---- 代码字体大小调节
Scratchpad ---- 可以创建一个临时的cell来执行你的代码，就像草稿纸一样，不用每次试验都使用好多正式的cell,弄的特别乱。

**4.1 Jupyter-notify 网页通知提示插件**

这个插件会在一个cell执行完成的时候，通过网页通知的方式告知用户这一完成的事件。当激活了这一插件之后，我们就可以不用时时盯着Jupyter Notebook的状态了。

要使用这个插件, 需要先安装pip install jupyternotify

%load\_ext jupyternotify *# 载入插件*

%autonotify -a 30

上述代码会对所有执行超过30s的cells在运行结束后进行通知。

如果只关注特定的cell，那么可以将%autonotify -a 30去掉，然后在对应的格子中加入%%notify。

如果想在Jupyter运行的时候自动加载该插件，在需要向Notebook的config文件中加入下列代码：

c.InteractiveShellApp.extensions = [

'jupyternotify'

]

**4.2  autoreload 重新导入模块**

这个插件使用以后, 使用import时会重新加载模块, 不需要使用reload方法.

%load\_ext autoreload

%autoreload

**4.3  tqdm 循环进度**

tqdm是一个python中对循环进程提供可视化支持的模块，目前该模块已经提供了对Jupyter的支持，我们可以通过下图体验一下可视化效果：

该插件可以安装pip install tpdm.

from tqdm import tnrange, tqdm\_notebook

from time import sleep

*## 原来的循环*

for i in range(10):

for j in tqdm\_notebook(range(100), desc='2nd loop'):

sleep(0.01)

*## 使用tqdm显示进度的循环*

for i in tnrange(10, desc='1st loop'):

for j in tqdm\_notebook(range(100), desc='2nd loop'):

sleep(0.01)

**总结：**以上就是小编整理

---------------------------------------------------------------

----------------------------------------------------------------

Jupyter Notebook 的快捷键
1.命令模式 (按键 Esc 开启)
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 : 单元转入代码状态
M :单元转入markdown状态
R : 单元转入raw状态
1 : 设定 1 级标题
2 : 设定 2 级标题
3 : 设定 3 级标题
4 : 设定 4 级标题
5 : 设定 5 级标题
6 : 设定 6 级标题
Up : 选中上方单元
K : 选中上方单元
Down : 选中下方单元
J : 选中下方单元
Shift-K : 扩大选中上方单元
Shift-J : 扩大选中下方单元
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
Shift-M : 合并选中的单元
Ctrl-S : 文件存盘
S : 文件存盘
L : 转换行号
O : 转换输出
Shift-O : 转换输出滚动
Esc : 关闭页面
Q : 关闭页面
H : 显示快捷键帮助
I,I : 中断Notebook内核
0,0 : 重启Notebook内核
Shift : 忽略
Shift-Space : 向上滚动
Space : 向下滚动
2.编辑模式 ( Enter 键启动)
Tab : 代码补全或缩进
Shift-Tab : 提示
Ctrl-] : 缩进
Ctrl-[ : 解除缩进
Ctrl-A : 全选
Ctrl-Z : 复原
Ctrl-Shift-Z : 再做
Ctrl-Y : 再做
Ctrl-Home : 跳到单元开头
Ctrl-Up : 跳到单元开头
Ctrl-End : 跳到单元末尾
Ctrl-Down : 跳到单元末尾
Ctrl-Left : 跳到左边一个字首
Ctrl-Right : 跳到右边一个字首
Ctrl-Backspace : 删除前面一个字
Ctrl-Delete : 删除后面一个字
Esc : 进入命令模式
Ctrl-M : 进入命令模式
Shift-Enter : 运行本单元，选中下一单元
Ctrl-Enter : 运行本单元
Alt-Enter : 运行本单元，在下面插入一单元
Ctrl-Shift-- : 分割单元
Ctrl-Shift-Subtract : 分割单元
Ctrl-S : 文件存盘
Shift : 忽略
Up : 光标上移或转入上一单元
Down :光标下移或转入下一单元

----------------------------------------------------------------

采用Github资源进行安装总是失败，改由国内的douban镜像进行安装成功
pip install jupyter\_contrib\_nbextensions -i https://pypi.douban.com/simple
jupyter contrib nbextension install --user --skip-running-check
重新打开Jupyter，在出现的Nbextensions中选择Hinterland

----------------------------------------------------------------

jupyter 里面的Autopep8用不了，

现实未定义Autopep8，但是又有安装了
在这里插入图片描述
解决方法：
电脑打开cmd
执行以下代码可以安装jupyter notebook扩展：
pip install jupyter ipywidgets autopep8 jupyter\_contrib\_nbextensions jupyter\_nbextensions\_configurator
jupyter contrib nbextension install --sys-prefix --skip-running-check
jupyter nbextensions\_configurator enable --sys-prefix
jupyter nbextension enable --py --sys-prefix widgetsnbextension
这里,有 --user 和 --sys-prefix, 前者配置到用户目录下, --sys-prefix会配置sys.prefix, 后者更适合用于虚拟环境, 前者适合无虚拟环境的情况.
然后重启jupyter即可。

----------------------------------------------------

Jupyter Notebook之提高可读性——插件安装及使用（包括插入目录，目录索引及缩进）

开启了Collapsible Headings

Table of Contents (2)(添加目录)

-------------------------------------------