

Python环境搭建 (Windows)

2022年12月2日, ©Brice(datawm.cn)

解释器(python)

1. 下载python解释器安装包

在python官网 <https://python.org> 下载合适版本安装包, 注意选择最新 (目前3.9.5) 或特定版本 (3.6, 3.8.2), 选择适合的操作系统 (如 windows 64 位)

或者到数域桃源资源 <https://datawm.cn/ziyuan/> 下载

2. 执行安装包

建议用customize模式, 安装时注意勾选如下选项

- add path
- pip
- add python to envircoment...

3. 查看版本, 运行 hello world

安装完成后打开CMD窗口 (可以通过 **开始菜单--windows系统--命令提示符** 找到, 或者Win+R后输入cmd直接打开),

输入"python -V"可查看版本号;

输入python可进入>>>提示符, 在提示符下输入exit()可退出python命令行模式, 通过函数可以一次执行多条指令。

也可以通过**开始菜单--python3.9--IDLE**打开解释器自带的IDLE编辑器, 可以新建、执行 .py 文件。

包管理(pip)

Python一般通过 PIP 进行包管理, 可使用 **pip install** 命令插入相关的包; 也可使用Anaconda或Pycharm等的包管理模块来进行包管理。

```
1 pip install pandas #插入pandas包的最新版本
2
3 ### 建议在网络较好的时候预先安装一些常用的包
4 #数据处理常用的包
5 pip install numpy pandas #数据分析基础包
6 pip install scipy scikit-learn factor_analyzer statsmodels #几个数据分析模型方面的包
7 pip install pandas_datareader tushare quandl #可用于在线获取经济、金融数据
8
9 #画图和数据可视化常用的包
10 pip install matplotlib seaborn pyecharts
11
12 #数据库访问常用的包
13 pip install pymysql redis elasticsearch pymilvus pyneo4j
14 pip install pymongo==3.11.0 #pymongo3和pymongo4差别较大, 如果需要兼容old代码, 建议指定安装3.11.0版。
15 pip install pymilvus_orm #milvus数据库支持有时用pymilvus_orm
16
```

```

17 #web开发常用的包, flask跟fastapi需要的包有些不同, 建议先安装fastapi[all], 再使用conda
    install flask
18 pip install flask
19 pip install pyinstaller
20 #API接口开发常用的包
21 pip install fastapi[all]
22 pip install websocket websocket-client
23 #图形视频处理方面的包
24 pip install opencv-python #若提示没有包skbuild, 可指定版本==4.1.1.26
25 pip install yolo #yolo包
26

```

pip命令进阶

```

1 pip install pandas==1.1.3 #指定包的版本
2 pip install numpy pandas #一次安装多个包, 用空格隔开
3 pip install -r requirements.txt # 通过安装列表来一次性安装多个包, 需要先用记事本生成
    一个列表, 格式为 pandas==1.1.3
4 pip install tensorflow -i https://pypi.tuna.tsinghua.edu.cn/simple #通过-i参
    数指定使用国内的源来安装相应的包, 可以百度下将外部源添加到系统的方法。外部仓库地址可以换成
    腾讯镜像 http://mirrors.tencentyun.com/pypi/simple 或者 阿里云镜像
    http://mirrors.cloud.aliyuncs.com/pypi/simple/。也可以通过配置pip的参数来设置缺省
    调用的镜像源(参下节创建pip配置文件)
5 pip install --no-cache-dir --force-reinstall -Iv grpcio==1.22.0 #强力安装
    (一般情况下不建议使用)
6 pip uninstall panda #删除某个已安装的包
7
8 # 若安装了anaconda, 可使用conda install命令代替pip install来安装包, 使用conda安装包
    会自动考虑包的兼容性且自动安装关联的包
9 conda install pandas==1.1.3
10 conda update pandas # conda update --all 更新所有库
11 conda remove pandas #删除某个已安装的包
12
13 #若警告pip版本较低, 可使用upgrade参数升级pip
14 pip install --upgrade pip #升级pip
15 pip install --upgrade setuptools #有时需要升级setuptools
16 #忘记安装PIP, 或者更新pip失败导致ModuleNotFoundError: No module named 'pip', 可以
    试试如下操作
17 python -m ensurepip
18 python -m pip install --upgrade pip
19 #pip安装时出现 WARNING: Ignoring invalid distribution - ***
20 #找到警告信息中报错的目录, 然后删掉~开头的文件夹, 是安装插件失败/中途退出, 导致插件安装出现
    异常导致。

```

创建pip配置文件

Windows下修改%APPDATA%/pip/目录下的配置文件(pip.ini), 先按下win+R键, 输入%APPDATA%。一般是C:\Users\.....\Appdata\roaming 文件夹, 若没有pip/pip.ini, 则新建一个。

Linux下修改 `vim ~/.pip/pip.conf` (没有就创建一个文件夹及文件);

输入如下内容:

```

1 [global]
2 time-out=60
3 index-url = https://pypi.tuna.tsinghua.edu.cn/simple
4

```

```

5  ### 国内可用的源(上面清华源的地址可以换成腾讯或阿里的源)
6  #腾讯云
7  [global]
8  index-url = http://mirrors.tencentyun.com/pypi/simple
9  trusted-host = mirrors.tencentyun.com
10 #阿里云
11 [global]
12 index-url=http://mirrors.cloud.aliyuncs.com/pypi/simple/
13 [install]
14 trusted-host=mirrors.cloud.aliyuncs.com
15 #其他常见的源地址 (未测试)
16 中国科技大学 https://pypi.mirrors.ustc.edu.cn/simple/
17 华中理工大学: http://pypi.hustunique.com/
18 山东理工大学: http://pypi.sdutlinux.org/
19 豆瓣: http://pypi.douban.com/simple/

```

交互式编辑器

数据开发和小工作量的代码修改调试，使用 **jupyter notebook** 非常方便，已经成为python一族最喜欢的工具。与linux系列jupyter跟anaconda（一款数据分析人员比较常用的python多解释器环境配置工具软件）结合的非常紧密不同，windows下的jupyter也较多的单独使用。

1. 建议在非系统盘新建一个新文件夹，命名为mypython或mycodes等容易记的名称。如 D:\mypython
2. 打开CMD，在命令行模式下执行如下命令：（可以复制后在命令行窗口右击）

```

1  C:\Users\admin> d:
2  D:\> mkdir mypython #此步可跳过，直接在计算机文件夹中新建文件夹即可
3  D:\> cd mypython
4  D:\mypython> pip install jupyterlab -i
   https://pypi.tuna.tsinghua.edu.cn/simple #在网络顺畅的环境下安装，等候安装完
   成，大约需要5-10分钟。
5  D:\mypython> jupyter notebook password #设置密码
6  D:\mypython> jupyter lab # 可以使用jupyter notebook命令代替jupyter lab，会进
   入到传统的jupyter notebook界面。

```

注意细心查看cmd窗口中的输出提示，其中有访问地址，端口，token等，使用过程中切勿关闭cmd窗口，用完以后可以按 Ctrl+C 退出jupyter，也可以直接关闭窗口。

若出现错误，一般重新安装jupyter或者尝试换如下命令

```

1  D:\mypython> python -m pip install --upgrade pip
2  D:\mypython> python -m pip install jupyter #或jupyterlab, jupyterlab的界面
   要友好一些
3  D:\mypython> python -m notebook

```

3. 在浏览器（如IE，EDGE，谷歌浏览器，火狐等）地址栏输入 <http://localhost:8888> 输入设置的密码即可进入交互式开发环境。

默认根目录为执行jupyter启动命令时所在的目录，如 D:\mypython

若未设置密码，可复制命令行窗口的token码来进入jupyter。

4. 点notebook选项下的python3，新建一个交互式文件（扩展名为.ipynb），在单元框中输入 `print("hello world!")`，点编辑窗口顶部的执行图标。

左边可以新建文件夹和txt、markdown、py等各种类型文件，也可以上传和下载相关文件。

比如，可以在/目录下新建一个文件，点右键重命名为 start.bat，双击打开编辑窗口，在其中输入 `jupyter lab --ip='*' --port=8888 --no-browser --allow-root &`，点file-save，这样，下次在cmd中加入D:\mypython后，可以直接输入 start.bat，命令即可打开jupyter。

5. 常见错误提醒的处理

比如在代码块中输入如下代码，点执行

```
1 import pandas as ps
2 a=[1,1,2]
3 b=ps.datafram (a)
4 print(b)
```

错误提示如下

```
1 ModuleNotFoundError                                Traceback (most recent call
  last)
2 <ipython-input-79-d35c46f8d1a2> in <module>
3 ----> 1 import pandas as ps
4
5 ModuleNotFoundError: No module named 'pandas'
```

这表明pandas包 (module) 未安装，需要在cmd命令窗口中使用Pip安装缺失的包，如 `pip install pandas`，后续如果见到类似的错误提示，一般都是需要安装相应的包。

安装好pandas包，继续执行代码，错误提示如下

```
1 File "<ipython-input-2-fb04ef25a9e2>", line 3
2     b=ps.datafram (a)
3                   ^
4 SyntaxError: invalid character ' (' (U+FF08)
```

这表明在第3行箭头的位置的左括号可能用的是全角，不是英文的括号。注意Python中各种符号，空格等必须是英文半角方式输入，这是初学者容易碰到的问题。

修改括号，继续执行，

```
1 AttributeError                                Traceback (most recent call
  last)
2 <ipython-input-3-eb00d5ecace6> in <module>
3     1 import pandas as ps
4     2 a=[1,1,2]
5 ----> 3 b=ps.dataframe(a)
6     4 print(b)
7
8 d:\prog\python39\lib\site-packages\pandas\__init__.py in
  __getattr__(name)
9     242         return _SparseArray
10    243
11 --> 244         raise AttributeError(f"module 'pandas' has no attribute
  '{name}'")
12    245
13    246
14
15 AttributeError: module 'pandas' has no attribute 'dataframe'
```

这实际上datafram应该为DataFrame，python区分大小写。这种错误一般比较隐晦，需要查阅相关书籍或者去百度检索答案。

集成开发平台

目前比较流行的python的集成开发平台有pycharm和vscode两种，vscode为免费软件，pycharm有社区版（专业版破解比较麻烦）。

VSCode

官网下载相应的安装包，安装即可（可以选择当前用户或所有用户，如果选择当前用户一般会被安装到%AppData%文件夹下）。

Code的主窗口的左侧有资源管理器、搜索、git、调试、扩展等图标选项。

下载安装包安装完成后，一般需要添加扩展包（如中文支持、各种语言的支持等），需要进行相应的配置。点左侧扩展图标，输入关键字搜索相应插件安装即可。

- **Chinese (Simplified) Language Pack for Visual Studio Code:** 中文支持
- **Python Extension Pack:** python支持：包含7个相关的包
- **IPython for VSCode:** Integration with IPython, including useful keybindings
- **Jupyter:** iPython支持 Jupyter notebook support, interactive programming and computing that supports Intellisense, debugging and more.
- **Auto Close Tag :** 自动闭合HTML/XML标签
- **Auto Rename Tag :** 自动完成另一侧标签的同步修改
- **indent-rainbow:** 不同缩进使用不同颜色显示，彩虹，缩进不规范的显示红色
- **Power Mode:** 打字特效，需要在扩展设置里把Powermode:Enabled给勾上，Powermode:particles星星特效/flames火焰特效/enableShake代码抖动效果，CounterEnabled设置为hide关闭右上角计数

使用文件菜单或资源管理器图标，打开工作区或文件夹，新建或打开一个.py文件，在文件中适当位置输入 #%%，就可以实现类似 jupyter note的代码块功能，实现交互式调试。

使用查看菜单，可以设置外观和编辑器布局，这里可以打开终端面板。

通过终端面板，可以执行CMD命令，就不用再另外打开CMD窗口了。

在窗口左下角，可以看到当前的python解释器，左键点击可以选择解释器。

vscode短时间内打开多个文件会覆盖原先打开的文件，在右方编辑区只显示一个。若想每次打开，都新建一个编辑，可以在右侧打开的文件上，按Ctrl + S保存一次，或者在文件名上鼠标左键双击一次，【可以发现，右侧编辑区文件名的斜体变正】，再去打开即可。

常用快捷键：

Ctrl+Shift+P(F1) 打开命令面板(交互搜索框)

Ctrl+1/Ctrl+2/Ctrl+3 切分、切换窗口

Ctrl+B 显示/隐藏侧边栏

Ctrl+F 查找

Ctrl+H 替换

Ctrl+P 按名称查找文件

Ctrl+'+/' 放大/缩小界面

shift+enter 进入交互模式执行cell。一般在每块cell前加上 "#%%"，就可以使用交互模式按cell执行代码

Ctrl+K Z 进入禅模式，按两次ESC退出，或按F11退出

选中内容:

Ctrl+'/' 增加行注释 (可一次多行)

Tab 右移

Shift+Tab 左移

Ctrl+B 加粗 ()

Pycharm

下载安装包安装即可, 官网可下载社区版。破解版按破解流程来。汉化。插件安装。

pycharm是一款很受欢迎的python代码编辑器, 与vscode支持多语言不同, 他专门针对python, 但他有其他系列产品, 如IDEA针对java。

pycharm的使用与vscode类似,

通过“视图-工具窗口”菜单可以选择呈现哪些面板, 包括项目 (类似资源管理器)、Git、终端、日志、数据库等,

在窗口右下角, 可以看到当前的python解释器, 点击可以选择解释器, 会自动搜索添加本机设置的jupyter (同样可以通过#%%来实现.py文件的分段调试)

pycharm的插件 (扩展) 安装是在“文件--设置”菜单下, 在弹出的对话框中选“插件”选项卡, 可以搜索安装协作插件。但一般是在网上下载插件安装包, 直接拖放到pycharm主窗口中即可自动安装。

中文支持插件: Chinese(simplifies) Language pack / 中文语言包

多解释器

python不同版本之间存在较大差异, 如python2.x和3.x之间是无法兼容的, 然后各种第三方包互相之间又有各种兼容问题。解决这一问题的思路是构建多个不同的python环境, 不同的环境设置不同的python版本, 不同的包配置, 不同的包的版本。

所以一般的python开发人员会自行维护几个不同的常见的解释器 (环境)。

Anaconda是一款自动帮我们配置不同环境的软件, 可以去其官方网站

(<https://www.anaconda.com/>) 下载windows版本的安装包。安装后即可安装提示设置不同的环境 (python版本, 包和包的版本), anaconda会自动帮您消除包版本之间的冲突。

但anaconda的windows版本有一些不够友好的地方, 比如其体积会很容易达到20G甚至50G的程度, 如果按默认的安装安装在C:\anaconda3文件夹, 会很快将系统盘撑爆。可以人工去C:\anaconda3\envs文件夹下把一些不用的环境文件夹删除。

不同版本的python解释器, anaconda提供的各种环境, 在vscode和pycharm中都被定义为**解释器路径**, 可以自由切换, pycharm中还可通过菜单进行配置。