

Python + VSCode 快速配置

AMoment*

2024 年 9 月 14 日

目录

1 简介	1
2 Python 解释器的下载	2
3 VSCode 的安装与配置	3
4 Python脚本试运行	4
5 Python Debug, Pip, Jupyter Notebook	4
5.1 调试 (Debug)	5
5.2 Pip	5
5.3 Jupyter Notebook	6
6 后记	7

1 简介

Python, 一门伟大的语言. 简易的, 贴合人类语言的语法, 丰富的生态, 强大的功能让Python近几年来几乎稳坐最受欢迎编程语言的宝座. 然而, 对于

*同构意义下也可称为F(). E-mail: amoment096@gmail.com

刚开始接触编程语言的初学者而言,最麻烦的可能并非学习语法或者处理报错,而是搭建一个简单易用的开发环境.本文将尽笔者所能,介绍如何配置出一套使用 VSCode+Python 的新手或轻度使用者适用的编程环境,以供新手平稳度过前期繁琐的边角料过程,尽快开始主菜.

然而需要提醒各位读者的是,笔者本人并非Python主力用户,Python于笔者而言仅为日常处理数据之用.因此如有不正之处,请与笔者联系修改,如有遗漏或不妥之处,欢迎联系笔者.在本文写作过程中,笔者并没有将自己搭建的环境删除后重新搭建以完成本文,因此可能会有很多与实际不相符之处.笔者的新电脑很快就到了,届时会根据本文对照搭建对应环境以检测本文内容是否合适,还请读者朋友包容.本文也假设读者您使用的是Windows10或以上的系统.如果您是Linux或其他系统的用户,我相信您不需要本文也可以快速搭建好环境.

2 Python 解释器的下载

Python语言的运行依靠Python解释器对编写好的Python脚本进行逐行解释,也可以通过交互式的方法,在解释器读取到输入内容后立即执行.因此,Python的语言解释器对学习Python是必要的.下载Python解释器请前往[Python官网](#).初学者可以不用太过在意语言版本的问题(语言版本过新可能会导致某些未进行适配的库无法正常使用),只需要保证您下载的版本是Python3即可(版本号以3开头).Python3与Python2有着很多基础语法上的区别,且很多库目前不怎么支持Python2.当然,为了省事,直接下载最新版也是没有任何问题的,在后续遇到实际需要时再进行版本修改也是没有问题的.

在下载好Python解释器后,便可以进行安装. **请注意勾选添加到 PATH(ADD TO PATH)以避免后续复杂的手动添加环境变量的过程!**当然,如果不幸,您已经在没有勾选此选项的情况下安装了Python解释器,您可以考虑卸载后重装或者考虑手动添加Python路径到环境变量中.这里不再赘述.

其余选项都可以一路默认.有个选项会提示您是否为所有用户安装,如果读者您使用的计算机内仅有一个账户(或者通俗而言,仅有您一人使用该计算机),那么是否选择此选项一般而言是无关紧要的.如果您使用公共电脑或者服务器,请不要勾选此选项,亦即仅为自己安装.

最后, 请检查您是否成功安装了Python解释器. 您可以在键盘上按下 **Win+R** 键打开运行对话框, 在对话框中输入 **cmd** 后确认, 您将会进入一个“黑框”(命令提示符)中. 此时在其中输入 **python -V** (请注意是大写的V)或者 **python --version**, 如果成功安装了Python并添加到了环境变量中, 则界面中将会出现您所安装的Python解释器的版本. 否则, 如果您看到类似于找不到Python定义之类的报错, 那么有可能您的安装失败或者安装过程中没有将Python添加到环境变量中.

以上便是安装Python解释器的过程.

3 VSCode 的安装与配置

这一步与上一步是平行的, 没有先后顺序一说, 您可以自由选择先进行哪个部分. 但建议您先进行上一部分, 在本部分结束后您将可以直接在VSCode中开始Python编程.

VSCode是一个强大的文本编辑器. 其最大的特点是其优秀的插件生态以及众多的语言支持(也是通过插件实现的). 通过VSCode与插件之间的配合, 可以实现媲美IDE的开发环境搭建. 笔者推荐由VSCode官方出品的在VSCode中使用Python的引导文档: [Python in Visual Studio Code](#), 该文档详细介绍了如何从0开始在VSCode上使用Python, 除了是英文内容外几乎没有缺点(当然, 您可以选择网页翻译). 下面笔者将自行介绍如何安装VSCode与相关插件.

点击 [此处](#) 即可打开 VSCode 官网. VSCode的安装可以全部选择默认安装, 如此便可使用VSCode的基础功能. 安装插件可以在侧边栏选择或者使用快捷键 **Ctrl+Shift+X** 打开插件市场, 在页面上方框中输入相应关键词即可检索相关插件. 如要进行Python开发, 请安装如下插件.

- Chinese (Simplified)(简体中文) Language Pack for Visual Studio Code: 可以使VSCode的语言显示变为中文显示.
- Python Extension Pack: VSCode上的Python插件全家桶. 安装这个比较省事.

4 Python脚本试运行

在以上所述的步骤完成后, 您便可以开始编写您的第一个Python脚本以检测您的环境是否搭建完成. 下面是一些简单的步骤:

1. 新建一个文件, 将之按照自己喜欢的名字命名, 并修改其后缀为 `.py`.
2. 右键该文件, 选择用VSCode打开. 打开VSCode界面后, 此时VSCode可能会询问您是否信任该文件夹. 请选择“信任”以使您安装的插件正常运行, 否则插件可能会被VSCode所屏蔽.
3. 现在您可以编辑该文件了. 输入一些Python代码, 下面是一个简单的测试代码:

```
print("Hello Python!")
```

请写好并保存该文件.

4. 现在请尝试运行该脚本. 如果您成功安装Python插件, 该文件界面的右上角应该会出现一个小的向右的箭头. 点击该箭头即可开始运行. 由于您很有可能是第一次在VSCode中运行Python脚本, 因此右下角会弹出一个通知框, 通知您还未选择Python解释器. 此时界面上方会出现一个对话框, 让您选择您需要的Python解释器. 您可能会看到多个解释器(如您下载了多个解释器版本)或者 **创建虚拟环境(Create Virtual Environment)**. 您可以先暂时不考虑设置虚拟环境, 先使用已有的全局生效的Python解释器.
5. 选择好后, 请再次重复上一步, 按下小箭头. 这是, VSCode界面下方会出现一个新的窗口界面, 显示的便是您程序运行的结果. 此时您便已经成功运行了该Python脚本, 也说明您的Python运行环境已经搭建成功了.

5 Python Debug, Pip, Jupyter Notebook

本节将简要介绍有关Python与VSCode的其他方面.

5.1 调试 (Debug)

调试是用以排查程序运行错漏的操作。代码一次写成, 运行良好固然很好, 但这种情况在实际开发中很难遇到。实际开发中常会遇到各种各样的问题阻碍开发进展。这些程序中或逻辑或语法的错误就被称为Bug, 在程序中排查Bug并修正以使程序得以正常运行的过程即是调试, 亦即所谓的Debug。

最简单的调试方法即将程序在某一步的数据通过 `cout` 或 `printf` 输出到控制台上。但这种方法毕竟还是比较繁琐, 特别是遇到难以输出到屏幕上的数据, 此时输出的方式便会失灵。现代程序开发过程中, 经常使用调试器(Debugger)来逐步运行程序, 以此尝试发现程序中隐藏的问题。虽然Python本身已经是解释型语言, 逐行运行已有的程序, 但是通过调试器的诸多功能, 仍可以为寻找程序漏洞问题提供帮助。

首先介绍断点, 程序在运行至断点后将会停在该处之前, 等待用户的下一步命令。断点的插入在代码编辑器中一般处于左侧的行号附近 (VSCode在行号的左侧), 插入成功后会出现一个小红点。当程序停在断点处时, 您可以查看变量的值, 函数调用栈等多种信息, 随后您可以逐步向下运行程序, 中断调试或者向步入函数内部(VSCode通过右上角小框控制)。

要进入VSCode的Debug模式, 请在运行Python脚本时, 在右上角的代表运行的箭头旁找到一个向下的箭头, 点击展开菜单后选择 **Python调试器: 调试Python文件(Python Debugger: Debug Python File)**, 随后便可进入调试模式。请注意此选项不仅会启动调试, 也会改变右上角的默认启动模式为调试。调试模式下, 该三角旁会出现一个小虫子, 代表此时处于调试模式。

请善用调试模式与调试器。

5.2 Pip

Pip(Package Installer for Python)是Python的包管理器。所谓的“包”指的是Python运行过程中需要调用的函数库, 类库等等。Python的优点很大一部分来自于Python活跃的生态, 指的便是丰富的第三方库, 或者, 第三方包。甚至于有人说, Python是一门胶水语言, 其就是用来将各种库粘合在一起以发挥作用。无论如何, 包对于Python的意义是毋庸置疑的, 而作为Python自带的默认包管理器, Pip的基础操作也是值得简单学习的。下面介绍Pip的一些简单命令,

并以安装Python下著名的科学运算库Numpy为例演示Pip的使用方法.

Pip的常用命令和参数有:

- **help**: 弹出帮助信息, 会提示您命令的功能.
- **install**: 指示Pip进入下载模式. 在Pip后附加包名称即可下载该包. 如果需要更新某个包, 请在包名前加上 **--upgrade** 以提示Pip更新此包而非安装.
- **uninstall**: 卸载某个Python包. 在该命令后附加包名即可.
- **list**: 列出所有您已安装的Python包.

接下来介绍如何安装Numpy:

1. 请打开命令提示符, 并输入 **pip** 以检查Pip是否正常可用. 如果可用则会弹出部分帮助文本, 不会有报错信息,
2. 您可能会看到您的Pip有可用的更新. 若在使用 **pip** 命令后, Pip提示您可以更新到最新的版本, 您可以选择根据提示输入命令进行更新.
3. 输入 **pip install numpy** 以安装Numpy. 稍等片刻您便可以安装好Numpy以供全局使用. 注意, 您在全局环境下下载的Numpy将对全局生效.

通过如上非常简单的步骤, 您便可以安装好Numpy.

5.3 Jupyter Notebook

Python脚本经常需要写好后一次性从头执行到尾, 而使用交互模式(在命令提示符中打开Python(**python**)将会进入交互模式)时Python会执行每次用户所输入的命令. 前者不够灵活, 而后者容易丧失上下文. 是否有一种更加具有交互性的, 但又不丧失上下文环境的Python使用方法呢? Jupyter Notebook提供了这样的方法.

Jupyter Notebook集成了Python环境和Markdown, 可以使您在代码框中使用并运行Python脚本, 并在Markdown框中使用Markdown语法编辑文字. 两种框的位置十分灵活, 且Notebook可以打开在浏览器中直接使用, 省去专门

的编辑器的麻烦,也可以选择在VSCode中使用.下面将介绍如何安装和使用Jupyter Notebook.

1. 请使用Pip安装Jupyter: `pip install jupyter` 并等待安装完成.
2. 输入 `jupyter notebook` 并回车. 请注意不要关闭该窗口,该窗口将作为服务器运行,若关闭将会导致Jupyter Notebook无法使用.
3. 稍等片刻,此时您的默认浏览器将会弹出一个窗口,左上角显示着Jupyter,而下方主页面则是您的用户文件夹.您可以双击已有的以 `.ipynb` 后缀结尾的文件以打开一个已有的Jupyter Notebook文件,或者请点击右侧的**新建(New)→Notebook**,便会在当前文件夹下新建一个Jupyter Notebook 并打开在您的浏览器的新页面中.
4. 此时新页面会请求选择一个Python内核.采用默认设置即可,此时您便已经新建了一个Jupyter Notebook了.默认的第一个框将是程序输入框,点击页面中央的框进入输入模式,输入Python代码后 `Ctrl + Enter` 以运行代码,结果将展现在该代码框的下方.
5. 您可以通过上方的工具栏新建,插入,删除,运行代码框或者Markdown框.更多功能请自行探索.

以上,您便成功安装并试运行了Jupyter Notebook.

除了在浏览器中使用原生的Jupyter Notebook以外,您还可以在安装好Jupyter后在VSCode中启动.请安装好Jupyter的插件后,在VSCode中使用快捷键 `Ctrl + Shift + P`,或点击VSCode最上侧的搜索框后输入 `>` 以进入命令模式,然后输入 `jupyter`,此时对话框会提示您所有的可用命令,点击**创建: 新Jupyter Notebook(Create: New Jupyter Notebook)**即可创建新的Jupyter Notebook.后续操作类似于网页端操作.该方法不需要自行打开一个Jupyter服务器,VSCode中安装的Jupyter插件将在VSCode的后台自行启动一个Jupyter 服务器.

6 后记

笔者希望通过该文章将笔者自认为好的且简单方便的Python使用开发环境介绍给本文的读者.然而作为一个非Python主力的用户,本文的内容纰漏

在所难免,且 \LaTeX 的插图体验并不优秀,笔者没有向文章中插入图片而是采用语言描述的方法,希望读者能谅解.

感谢您能读到这里. 如果您对本文的内容有何看法或意见, 欢迎联系笔者. 最后, 希望本文能真的实现, 并帮助您实现Python的那句名言:

人生苦短, 我用Python.

Life is short, I use Python.

祝您生活愉快.