

# Python工具入门：配置VScode、Anaconda | V1.0

制作者：人工智能基础董豪老师班助教 王鸿铖

## 写教程的起因

从第一次上机课的结果看，大家主要使用的语言依然是C/C++，然而在没有程设的基础下，对于这些C++的类、继承、虚拟函数、模板等概念依然是一头雾水，而且由于课程中后期会涉及到深度学习的内容，python是大家需要学习的必不可少的语言。在折腾python和科研一整天之后，助教觉得应该换一下脑子，把一团浆糊的大脑清空一下，写一份教程帮助大家尽快入门python。

这里就介绍一下如何从安装python环境开始最终写一段python代码。助教将从一个全新的windows10虚拟机入手，一步一步讲解安装过程（MacOS类似）

## 简介

python在我们的学习生活中，特别在人工智能领域是一个非常重要的语言，更以其入门简单、语法简单著称（在这里先请大家以C++的视角来学习python）。

VSCode全名 Visual Studio Code（而非 Visual Studio）是一款轻量级的代码编辑器，可以安装各式插件以满足需求，因其优秀的远程连接服务器和丰富的插件被我个人所喜欢。

Anaconda是一个开源的Python发行版本（反正就是能用来写Python）。我们主要利用其中的conda来管理虚拟环境（会在之后介绍什么是虚拟环境），用jupyter notebook进行一些python交互（jupyter用于学习某些python语法、简易的实现一些功能）。

## 配置VScode

从官网下载安装包，点下图中的Download或者进入官网点Download。


国内版 国际版

Microsoft Bing

vscode

网页 图片 视频 学术 更多

约 56,800,000 个结果 时间不限 在新选项卡中打开链接



打开网站

### Visual Studio Code - Code Editing. Redefined

<https://code.visualstudio.com>

网页 2022年11月16日 · Visual Studio Code is a code editor redefined and optimized for built and debugging modern web and cloud applications. Visual Studio Code is free and ...

Download

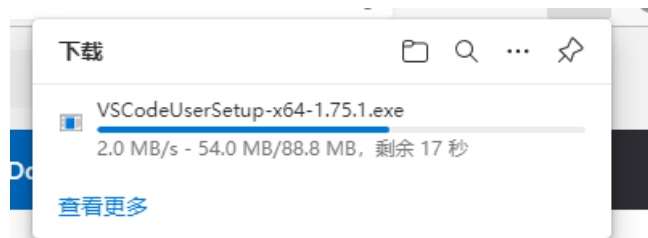
Visual Studio Code is free and available on your favorite platform - Linux, macOS, ...

Docs

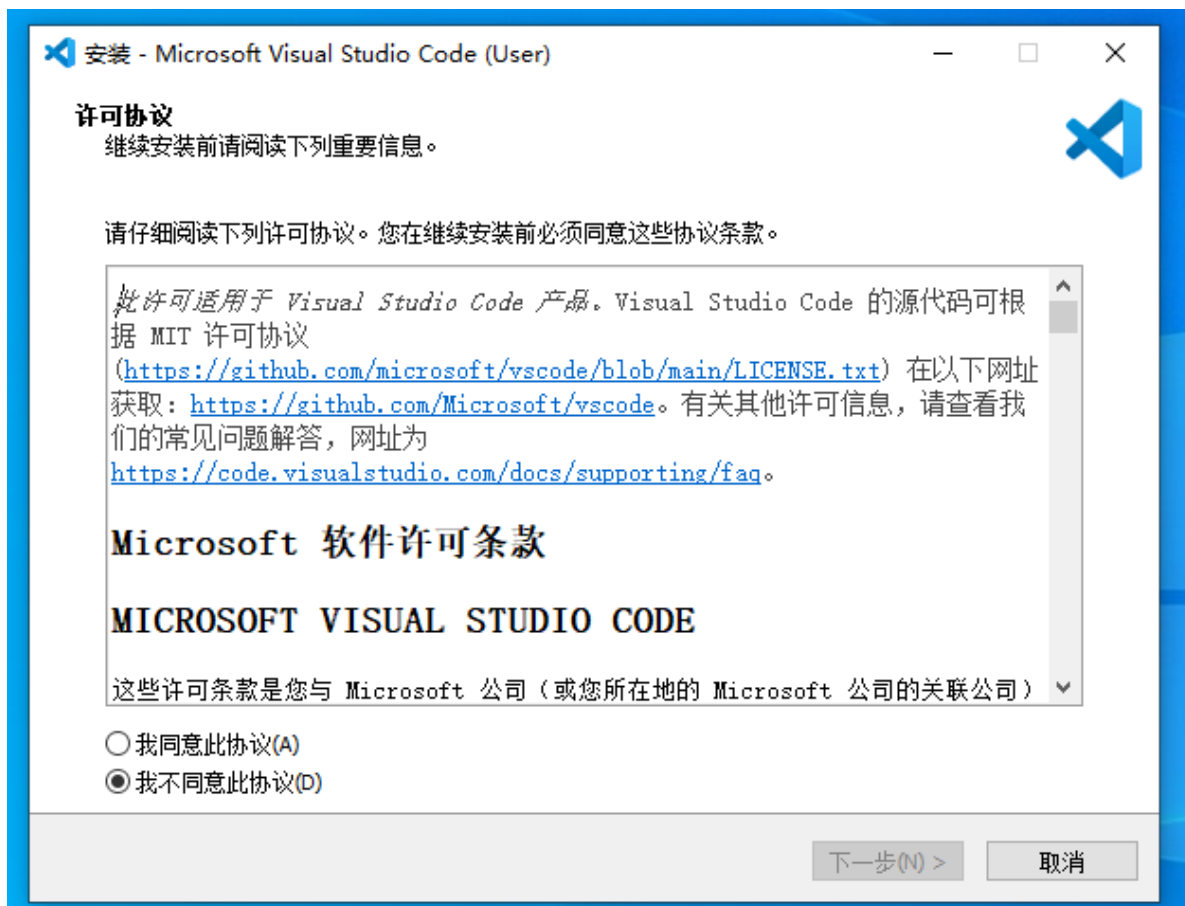
Blog

In a client of your choice, you can open 1 vscode.dev link from the notification ...

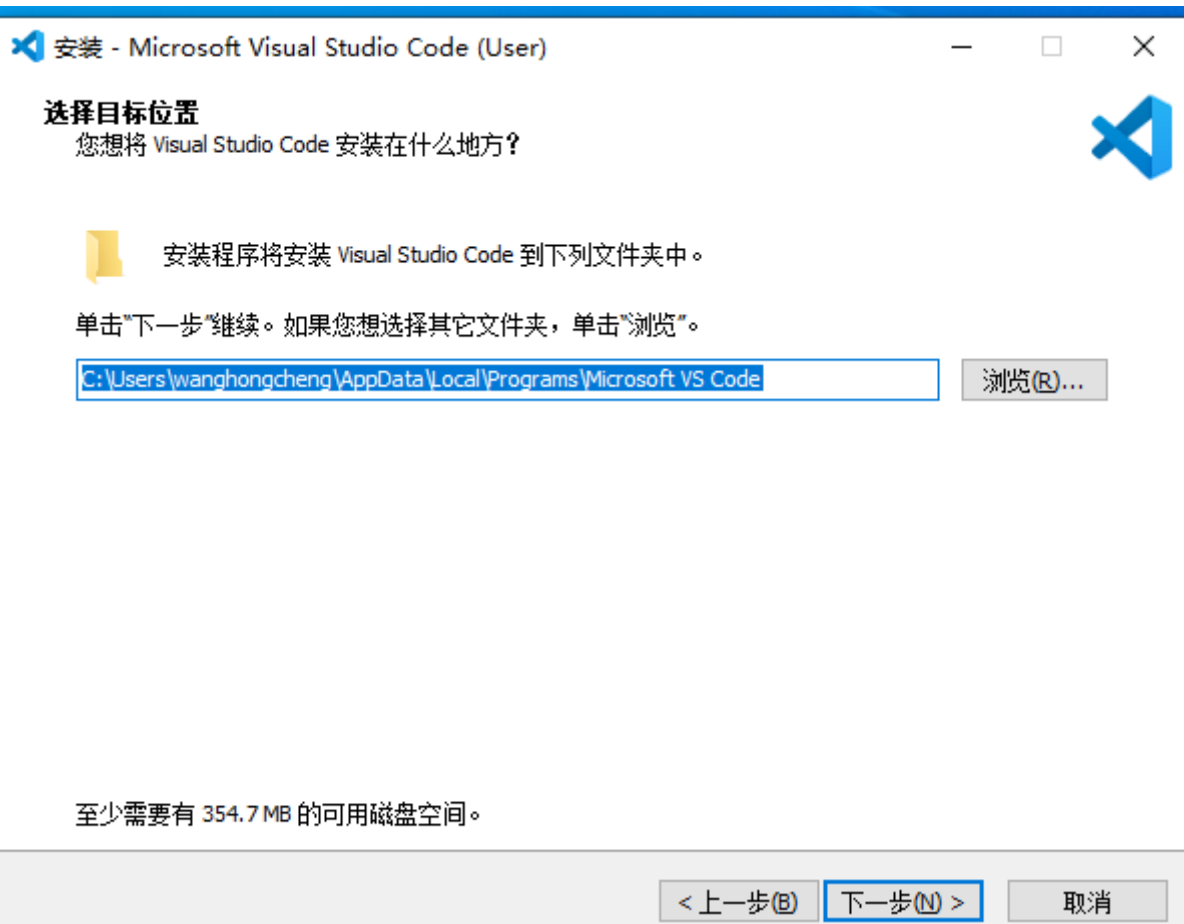
API



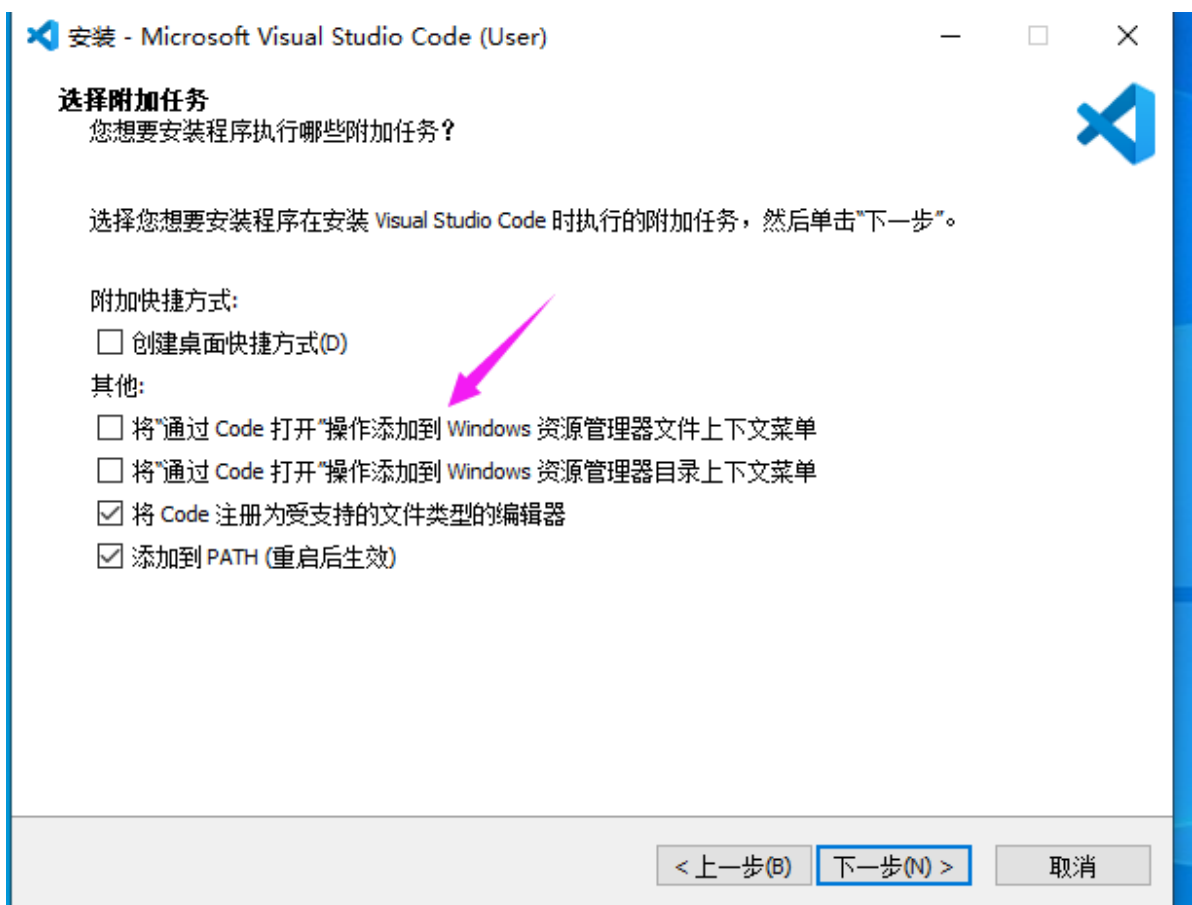
下载完成之后点开安装包。



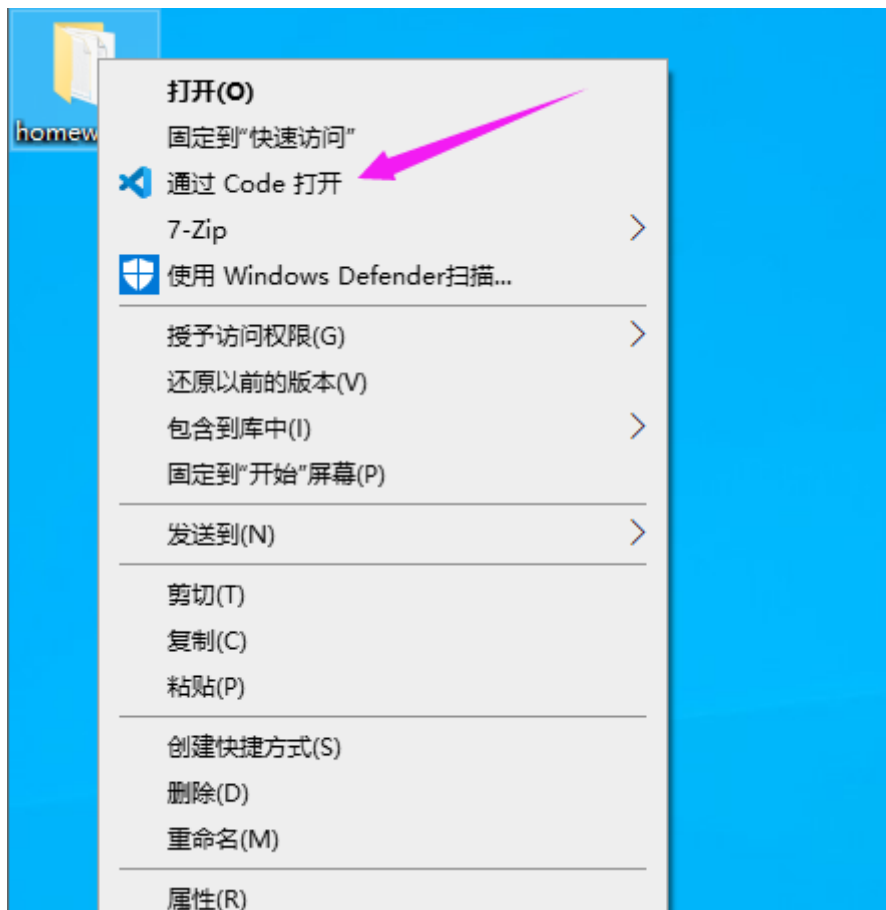
一路下一步（当然也可以根据自己情况更改安装路径）



根据自己的情况打钩。我个人选择5个全部打上（重度使用用户）。

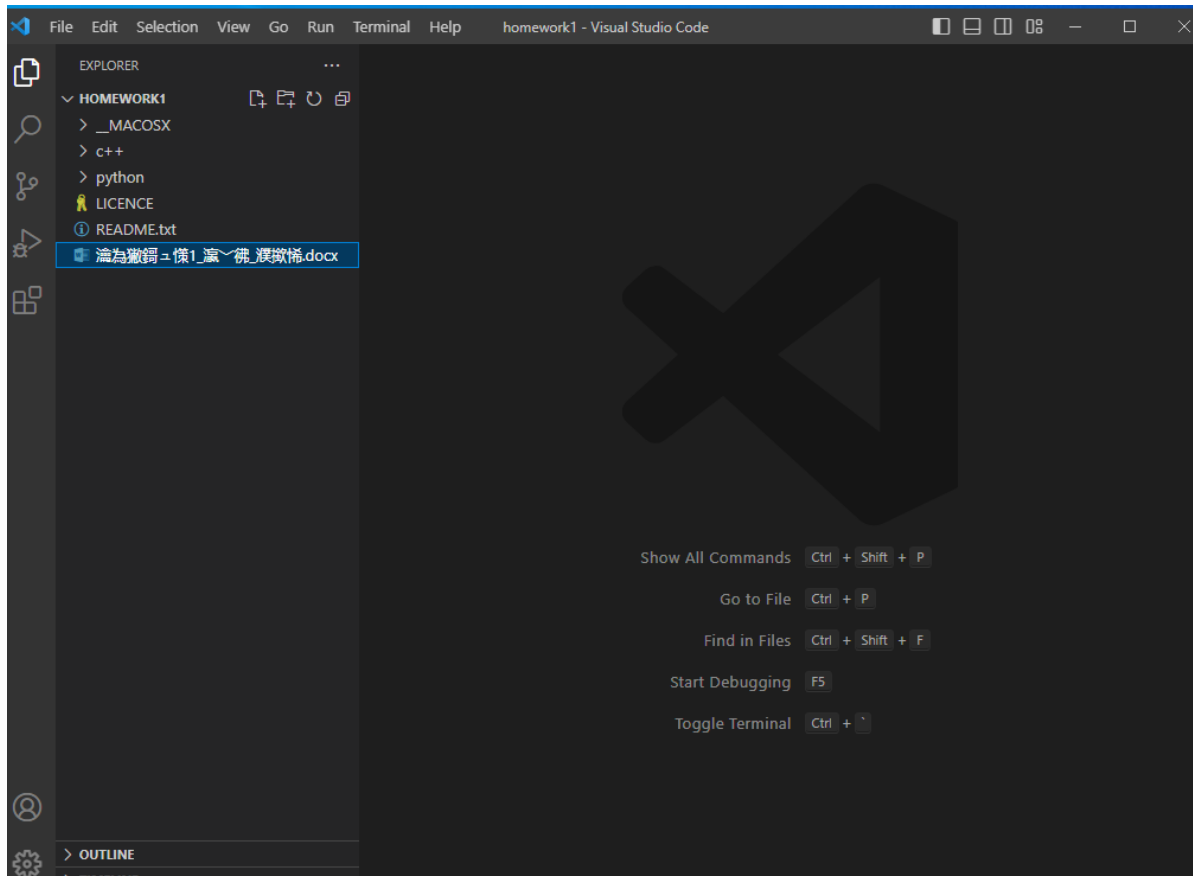


如果刚刚把2、3两个√都打上了，现在可以右键文件夹，选择vscode打开。

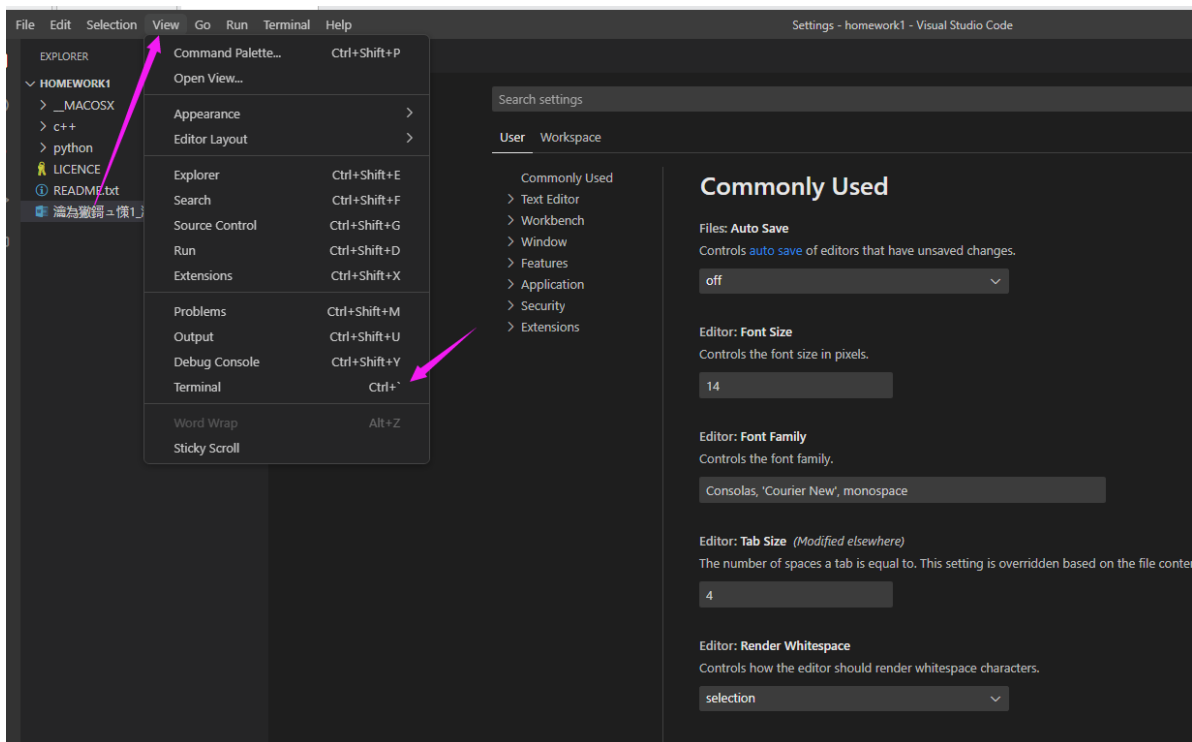


如果没有打上，那么就双击打开VScode，然后把文件夹拖入VScode。

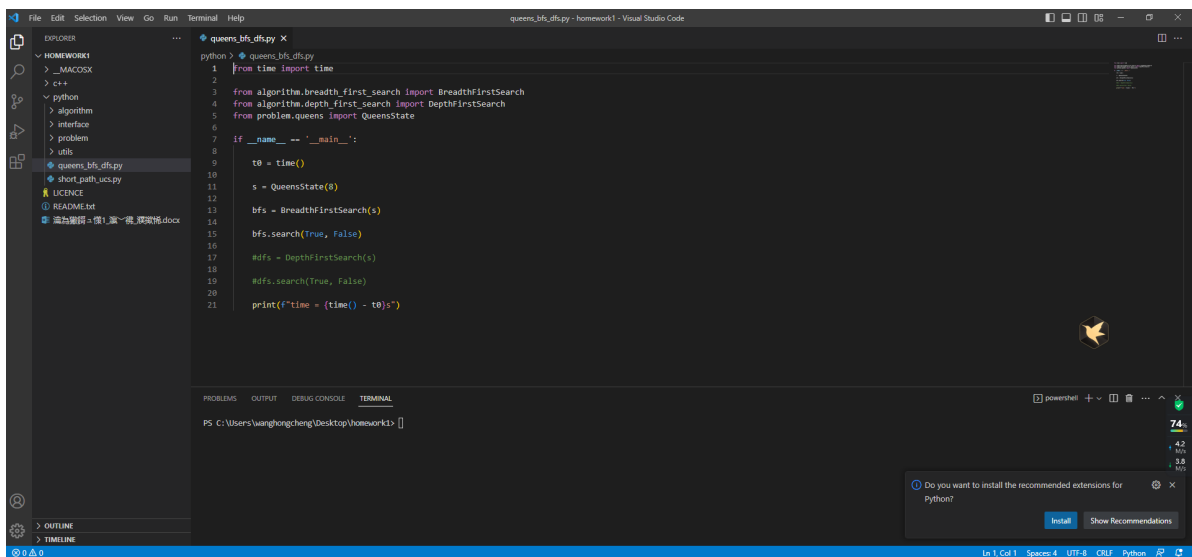
两种方法打开效果如下：



点击上方的View，再点terminal。



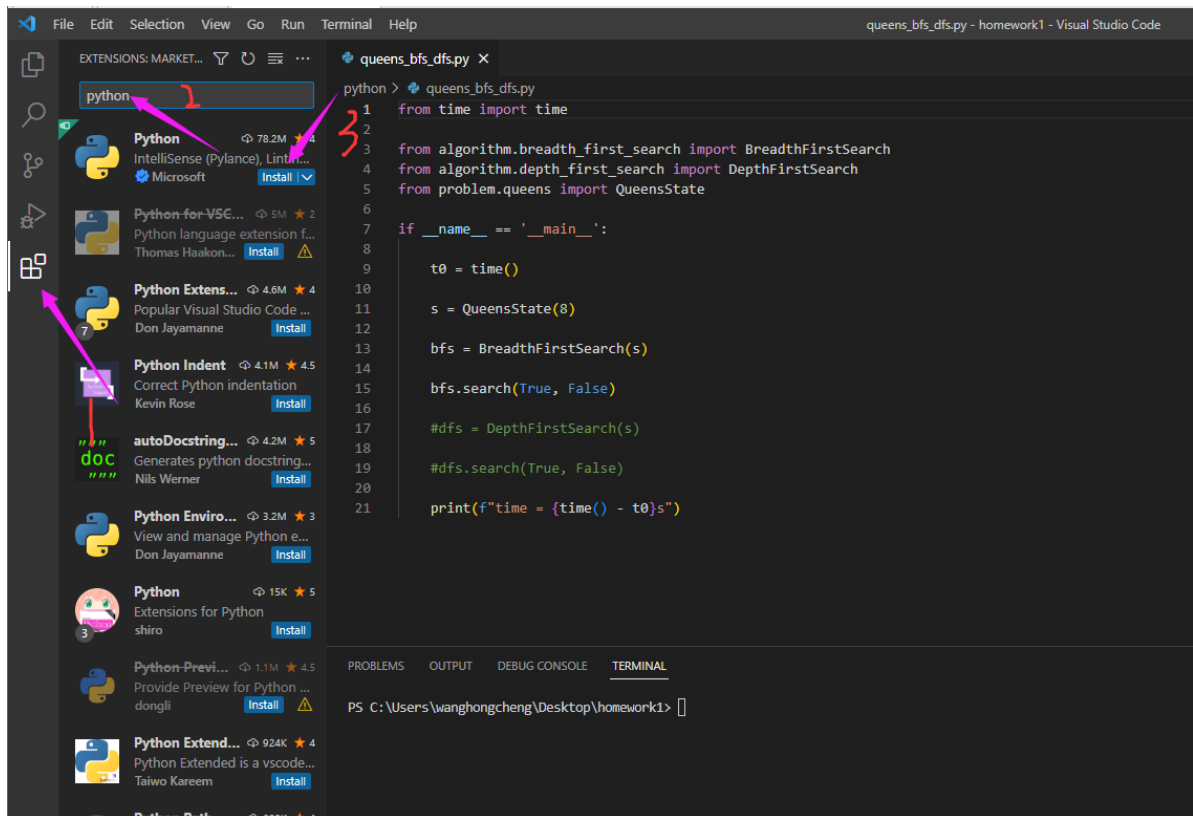
然后双击打开需要工作的python文件，下图就是我的工作状态。



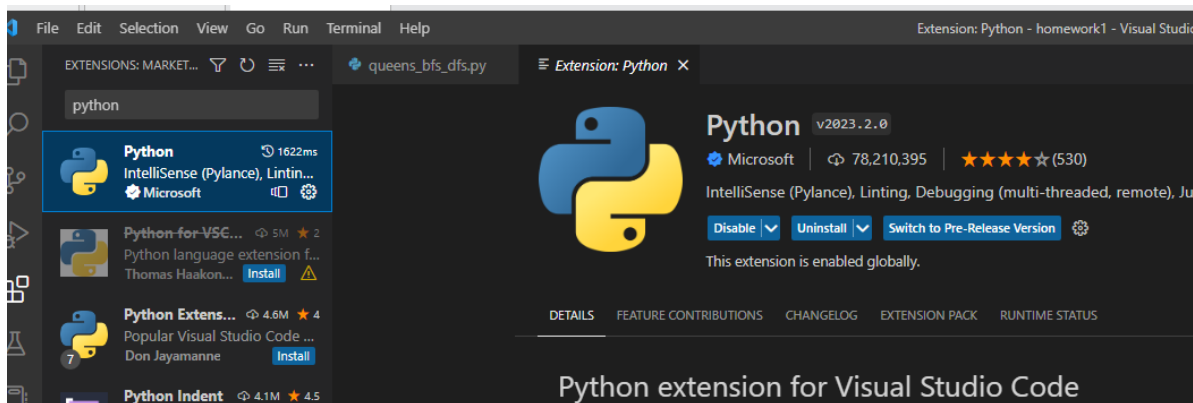
## 安装Python插件

请注意，这里的插件并非Anaconda，而是指VSCode当中的python插件，其用途对我来说是“单步调试”。如何单步调试将在之后讲解。

点击最左侧的四个方块，然后在上方的搜索栏中输入python，再点install。



安装完成：



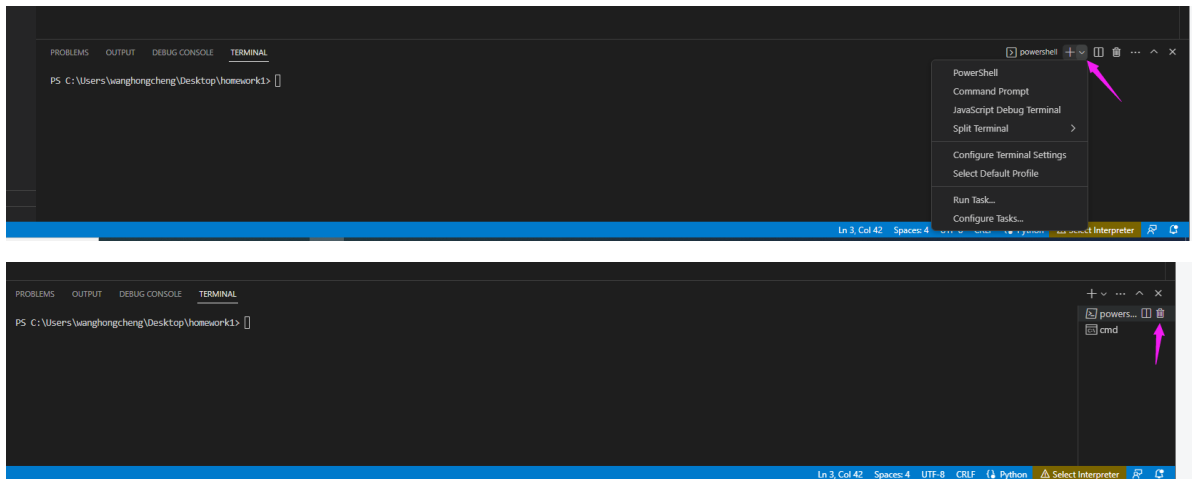
实际上，插件市场非常丰富，比如有让代码变颜色使得逻辑结构更加清晰的插件、代码补全插件等等，大家可以在熟悉了VScode之后自行百度探索那些神奇的插件

## 命令行介绍

在刚刚介绍的工作状态时，我们点击了View，并点了Terminal，这时候下面跳出来了一个窗口。这就是命令行。

命令行有不同的版本，比如powershell，bash，cmd，zsh，其实都是各种shell。大家可以看上图中的命令行的起始有一个"PS"，代表当前用的是powershell。

点击下方的紫色箭头所指的地方，可以选择开始不同的shell。接下来点击command prompt，可以发现右侧出现了两个不同的shell，我们点Powershell右侧的垃圾桶，关掉powershell，之后我们将使用command prompt作为shell进行conda系列的操作，写完命令之后，按下回车即开始执行。

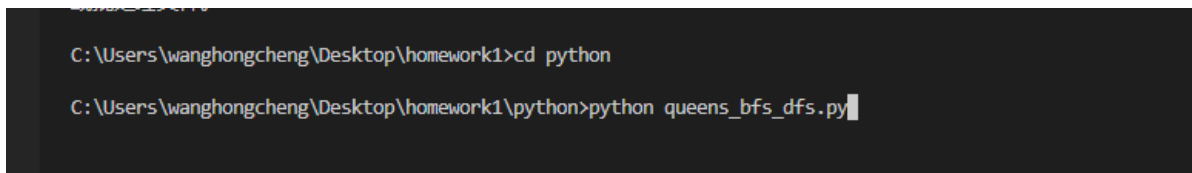


以下内容请在完成安装Anaconda之后再尝试。。。 （原因可以看上面那张图的右下角，有个Select Interpreter字样，但是点击这个字样发现中上方出现了一个提示窗，里面发现python is not installed。我们按ESC关掉这个提示窗。）

在作业的readme.txt当中，给出了以下提示。

- 1 | **【执行Python脚本】**
- 2 |       python [文件名].py

我们在命令行使用cd命令，进入python文件夹。然后执行python queens\_bfs\_dfs.py。



## Anaconda安装

搜索Anaconda，进入官网。点Download。

约 89,000,000 个结果    时间不限    在新选项卡中打开链接

来自 360 简体中文版，一键安装，永久使用！配套从入门到精通学习教程！无广告、安全下载、原版软件下载 广告

来自 360 简体中文版，一键安装，永久使用！配套从入门到精通学习教程！无广告、安全下载、原版软件下载 广告

正版激活: [Anaconda 2016](#) · [Anaconda 2015](#) · [Anaconda 2014](#)

网页 2023年2月14日 · **Anaconda** offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today. point solutions. Powerful solving. **Anaconda** was built by ...

Anaconda Distribution equips individuals to easily search and install thousands of ...

[Blog Anaconda Nucleus Join Our Online Community Open Source Technologies ...](#)

Anaconda Enterprise is a secure and

Anaconda is the birthplace of Python data

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

Download 

For Windows

Python 3.9 • 64-Bit Graphical Installer • 621 MB

Get Additional Installers



下载

Anaconda3-2022.10-Windows-x86\_64.exe

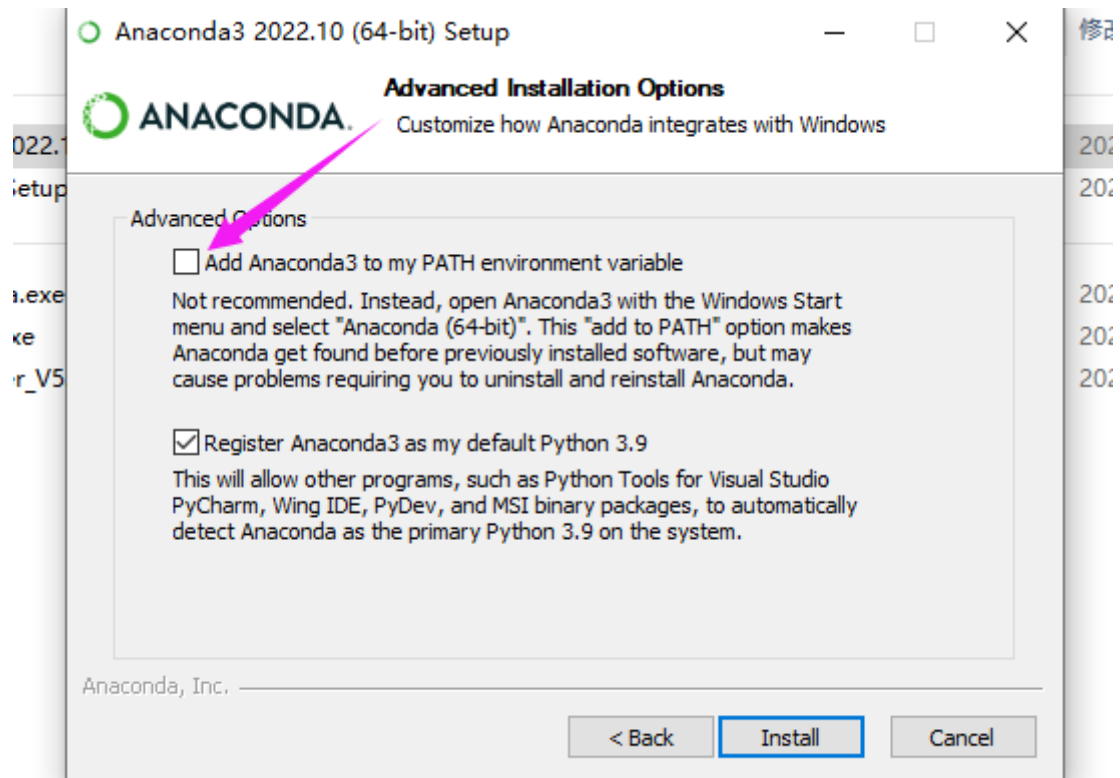
597 KB/s - 11.3 MB/621 MB, 剩余 17 分钟

VSCodeUserSetup-x64-1.75.1.exe



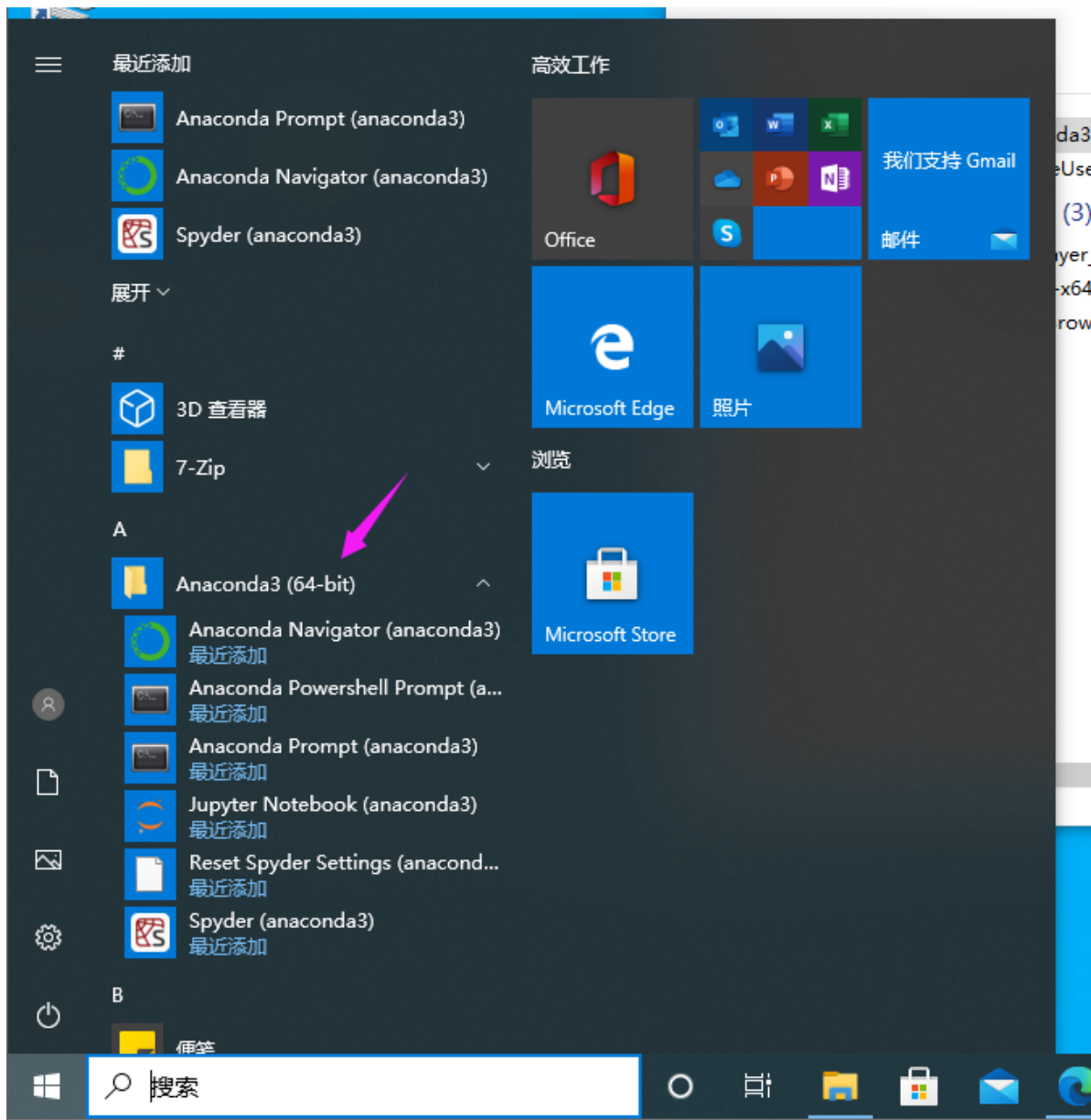
同样的，下载完成之后双击打开，在点下一步的时候，请注意，安装路径可以自己放到喜欢的地方（安装路径需要全英文+无空格）。

下面这个√我选择了打上。如果不打上，似乎需要手动把它加到系统路径下，**我个人强烈建议打上**√，后面会方便不少。如果打上了，**用户可以在任意命令行路径下启动Python或使用conda命令**



然后点击install。

安装完成之后，在windows的开始菜单中，会有一个anaconda的文件夹



我个人一般不使用里面的东西，最常用的是jupyter notebook，用来写一些小型的python代码。

## Anaconda 使用

我们回到VSCode。关闭之前打开的窗口，然后按照刚刚的流程打开cmd。

在cmd当中输入conda，如果咋咋咋输出了一堆东西，说明安装成功了。

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
History restored
Microsoft Windows [版本 10.0.18362.30]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Users\wanghongcheng\Desktop\homework1>conda
usage: conda-script.py [-h] [-V] command ...

conda is a tool for managing and deploying applications, environments and packages.

Options:
positional arguments:
  command
  clean                Remove unused packages and caches.
  compare              Compare packages between conda environments.
  config               Modify configuration values in .condarc. This is modeled after the git config command. Writes to the user .condarc file (C:\Users\wanghongcheng\.condarc)
  create               Create a new conda environment from a list of specified packages.
  info                 Display information about current conda install.
  init                 Initialize conda for shell interaction.
  install              Install a list of packages into a specified conda environment.
  list                 List installed packages in a conda environment.
  package              Low-level conda package utility. (EXPERIMENTAL)
  remove               Remove a list of packages from a specified conda environment.
  rename               Renames an existing environment.
  run                  Run an executable in a conda environment.
  search               Search for packages and display associated information. The input is a MatchSpec, a query language for conda packages. See examples below.
  uninstall            Alias for conda remove.
  update               Updates conda packages to the latest compatible version.
  upgrade              Alias for conda update.
  notices              Retrieves latest channel notifications.

optional arguments:
  -h, --help            Show this help message and exit.
  -V, --version          Show the conda version number and exit.

conda commands available from other packages:
```

一大堆

如果说conda 命令没找到，多半是你刚刚没有把conda加入到系统路径当中，你可以百度一下如何把conda添加到系统路径。

## 虚拟环境是什么？

在说虚拟环境之前，我们先介绍什么是包；包其实也是python代码，是由别人写好，发布给大家使用的。我们在写python的时候，常常要引入一系列的包，这避免了我们自己写很多代码（比如你要写一个求平均数的，那就导入一个拥有求平均数功能的包），比较著名的包有：numpy（用于数据统计、计算的），pytorch（用来搭建神经网络、训练人工智能的）。我们homework1也导入了不少包，如下图所有import、from xxx import xx，都是在导入包。

```
python > queens_bfsDFS.py > ...
1 from time import time
2
3 from algorithm.breadth_first_search import BreadthFirstSearch
4 from algorithm.depth_first_search import DepthFirstSearch
5 from problem.queens import QueensState
```

其中，time是python内置的包，是别人写好的；algorithm是其他助教为大家布置作业的时候，自己写的包。

包是有版本的。一个作者在发布了一个包之后，发现有bug、想更新功能，就会不断迭代版本。这会导致一个问题，比如作者A在写他的包a时，利用了作者B写的包b的版本1.0（称为包b1.0），这时候作者B更新了他的包至b2.0去掉了一些函数（而这个函数正好是包a所需要的），然后你在使用作者A的包a时，也应该安装包b1.0，否则会用不了包a。

那么虚拟环境是什么？一个虚拟环境是一系列包的总和。那么为什么要虚拟环境？从我们刚刚的例子中可以看到，想用包a必须安装包b1.0；但是假如这时候你想用作者C写的包c，而包c又必须要求使用包b2.0，这就导致了冲突：你不能同时使用包a和包c！而这个时候，你选了人工智能基础和机器学习概论这两门课。在AI基础上你要用包a，在机器学习概论上你要用包c，然后你只能一次次在包b1.0和包b2.0之间来回安装卸载。**这时，虚拟环境来了！**你可以创建一个虚拟环境X，在虚拟环境X之下，你可以把包a和包b1.0安装这；然后再创建一个虚拟环境Y，在Y下，你可以安装包c和包b2.0。虚拟环境X和Y互不干扰，每次上课前，你只需要从你的众多虚拟环境当中，挑选出一个你需要的虚拟环境就行。

## 创建虚拟环境

在命令行当中输入以下命令，其中NameOfYourEnv是你对这个环境的命名，比如我想创建一个名叫aibasis，那么就是第二行的命令。后面跟着的python=3.9是指定的python版本，有的代码要求跑在3.8下，那么就是python=3.8，根据需求指定python版本即可。然后按回车。

```
1 conda create -n NameOfYourEnv python=3.9
2 conda create -n aibasis python=3.9
```

```
C:\Users\wanghongcheng\Desktop\homework1>conda create -n aibasis python=3.9
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

```
==> WARNING: A newer version of conda exists. <==
current version: 22.9.0
latest version: 23.1.0
```

Please update conda by running

```
$ conda update -n base -c defaults conda
```

```
## Package Plan ##
```

```
environment location: C:\Users\wanghongcheng\anaconda3\envs\aibasis
```

```
added / updated specs:
- python=3.9
```

The following packages will be downloaded:

package	build	
ca-certificates-2023.01.10	haa95532_0	121 KB
certifi-2022.12.7	py39haa95532_0	149 KB
openssl-1.1.1t	h2bbff1b_0	5.5 MB
pip-22.3.1	py39haa95532_0	2.7 MB
python-3.9.16	h6244533_0	19.4 MB
setuptools-65.6.3	py39haa95532_0	1.1 MB
sqlite-3.40.1	h2bbff1b_0	889 KB
tzdata-2022g	h04d1e81_0	114 KB
wheel-0.38.4	py39haa95532_0	83 KB
Total:		30.1 MB

The following NEW packages will be INSTALLED:

ca-certificates	pkgs/main/win-64::ca-certificates-2023.01.10-haa95532_0	None
certifi	pkgs/main/win-64::certifi-2022.12.7-py39haa95532_0	None
openssl	pkgs/main/win-64::openssl-1.1.1t-h2bbff1b_0	None
pip	pkgs/main/win-64::pip-22.3.1-py39haa95532_0	None
python	pkgs/main/win-64::python-3.9.16-h6244533_0	None

```
Total: 30.1 MB

The following NEW packages will be INSTALLED:

ca-certificates  pkgs/main/win-64::ca-certificates-2023.01.10-haa95532_0 None
certifi          pkgs/main/win-64::certifi-2022.12.7-py39haa95532_0 None
openssl         pkgs/main/win-64::openssl-1.1.1t-h2bbff1b_0 None
pip             pkgs/main/win-64::pip-22.3.1-py39haa95532_0 None
python          pkgs/main/win-64::python-3.9.16-h6244533_0 None
setuptools      pkgs/main/win-64::setuptools-65.6.3-py39haa95532_0 None
sqlite          pkgs/main/win-64::sqlite-3.40.1-h2bbff1b_0 None
tzdata          pkgs/main/noarch::tzdata-2022g-h04d1e81_0 None
vc              pkgs/main/win-64::vc-14.2-h21ff451_1 None
vs2015_runtime  pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2 None
wheel           pkgs/main/win-64::wheel-0.38.4-py39haa95532_0 None
wincertstore     pkgs/main/win-64::wincertstore-0.2-py39haa95532_2 None

Proceed ([y]/n)?
```

输入y，然后按回车。

## 激活/退出虚拟环境

正如我们刚刚所说，我们可以为不同的课程需求安装不同的虚拟环境，那么我们应该如何选择并启动虚拟环境呢？

输入以下命令即可。NameOfYourEnv就是你想要的环境的名称，比如我们刚刚创建了aibasis的环境，就是如第二行所示

```
1 | conda activate NameOfYourEnv
2 | conda activate aibasis
```

激活之后，会在路径前面有一个小括号，括号内就是当前正在使用的虚拟环境。

```
C:\Users\wanghongcheng\Desktop\homework1>conda activate aibasis

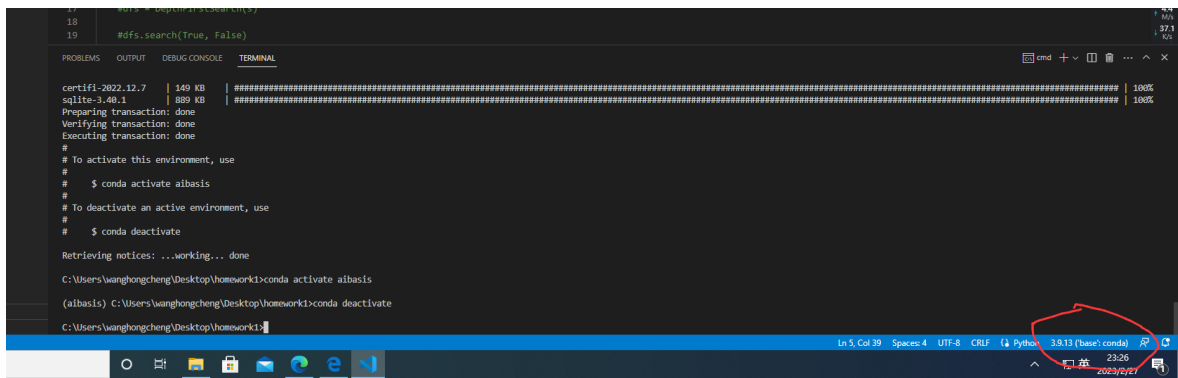
(aibasis) C:\Users\wanghongcheng\Desktop\homework1>
```

如果我们想要在不同的环境之间跳转，则直接用上面的方式激活另一个环境即可。

如果想要退出虚拟环境，则输入

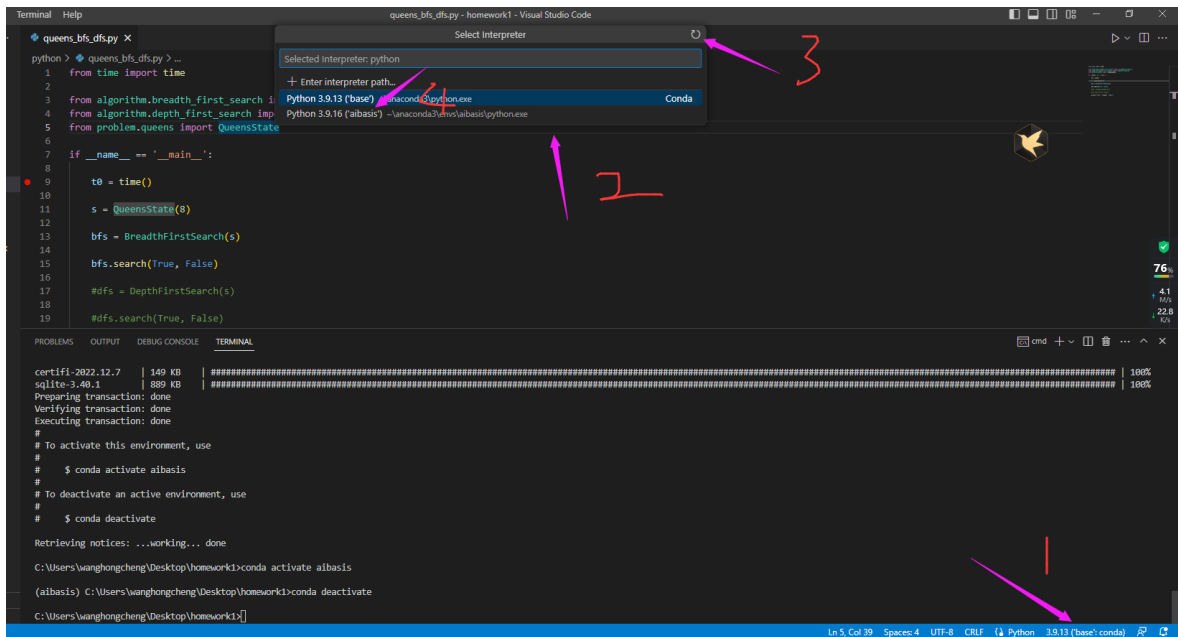
```
1 | conda deactivate
```

有眼尖的同学会发现，在命令行的右下角，有一个base:conda

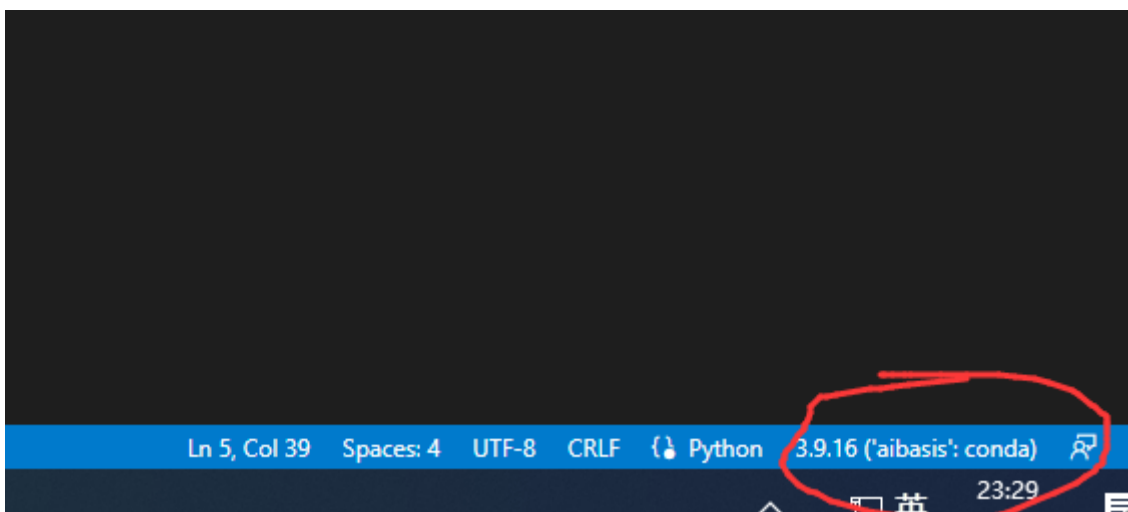


这个base，也是虚拟环境，它的名字就叫做base，是最基础的虚拟环境（一般不在这个虚拟环境内安装东西），算是安装anaconda附带的，可以不用管它。

然后你点击上方红色圈圈内，会在中上方出现一个提示窗。点击提示窗右上角的圆圈刷新一下，就会发现你刚刚创建的虚拟环境就出现了。



然后再点击你刚刚创建的虚拟环境，就会发现，右下角切换成了aibasis:conda。



那么右下角的虚拟环境名称和刚刚路径旁边的小括号内的名称有什么区别呢？

从我的使用来看，如果你直接在命令行内运行python，那么会使用路径旁边的小括号中的虚拟环境：



```
(aibasis) C:\Users\wanghongcheng\Desktop\homework1\python\python queens_bfsDFS.py
Q+++++
+++Q+++
+++++Q
+++++Q+++
++Q++++
+++++Q+
+Q++++
+++Q+++
04752613
Q+++++
+++++Q+
+++++Q
++Q++++
Ln 5, Col 39  Spaces: 4  UTF-8  CRLF  Python  3.9.16 (aibasis\conda)
```

如果你是使用了单步调试，那么会使用右下角的虚拟环境。

## 安装包

刚刚提到了作者A、B、C之间的爱恨情仇，那么问题来了，我该怎么安装包呢？在激活虚拟环境A的前提下（也就是路径左边的小括号内是虚拟环境A的名字）

其实很简单，在命令行当中输入以下两者之一都行，请注意，优先选择第一种。比如要在aibasis中安装numpy这个包，就可以参考三四行。

```
1 | conda install NameOfPackage
2 | pip install NameOfPackage
3 | conda install numpy
4 | pip install numpy
```

```
(aibasis) C:\Users\wanghongcheng\Desktop\homework1\python>pip install numpy
```

如果你在之后跑代码的时候，出现了以下报错（“No module named xxxx”），则一般是你这个包没安装到位，你需要百度以下这个xxx应该如何安装（通常情况就是conda install xxx，偶尔会有一些不一样比如cv2这个包要使用pip install opencv-python才能安装）

```
(aibasis) C:\Users\wanghongcheng\Desktop\homework1>python test.py
Traceback (most recent call last):
  File "C:\Users\wanghongcheng\Desktop\homework1\test.py", line 1, in <module>
    import torch
ModuleNotFoundError: No module named 'torch'

(aibasis) C:\Users\wanghongcheng\Desktop\homework1>
```

有时候你会发现安装很慢，是因为这些库很多都是国外的，那么你可以使用国内的镜像，比如我们亲爱的清华源。只需要在上面的安装命令后面以下指令即可。完整的见下面第二行。（当然本课程homework1不需要opencv-python，仅仅作例子）

```
1 | -i https://pypi.tuna.tsinghua.edu.cn/simple
2 | pip install opencv-python -i https://pypi.tuna.tsinghua.edu.cn/simple
```

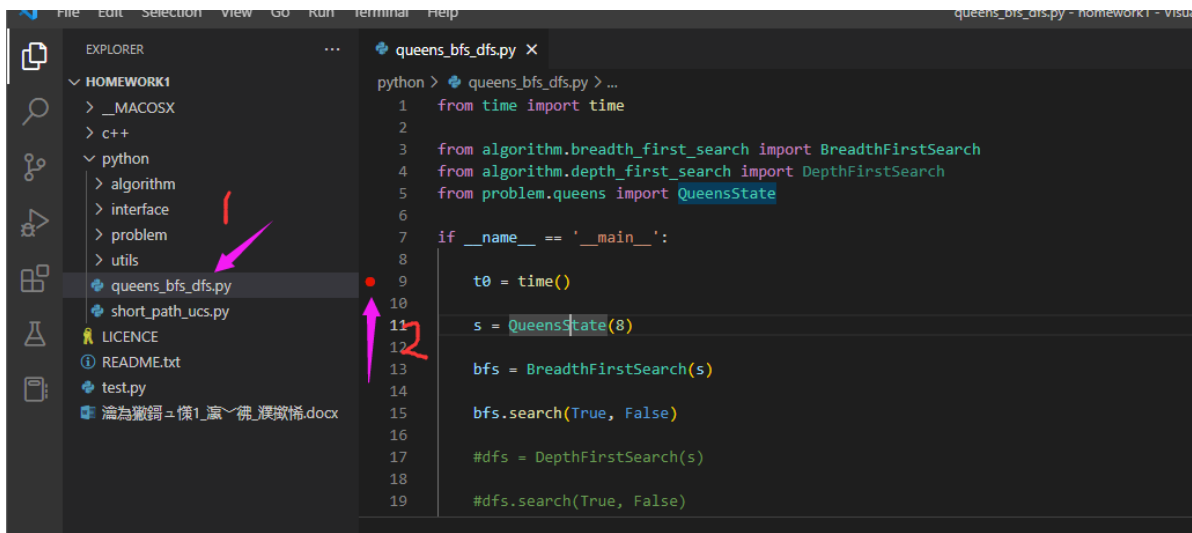
```
(aibasis) C:\Users\wanghongcheng\Desktop\homework1>pip install opencv-python -i https://pypi.tuna.tsinghua.edu.cn/simple
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting opencv-python
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/36/98/fab8d982e2e2b57bdebcad64c7e5b5a14ac91c657cac589b9cf3f8ea49d2/opencv_python-4.7.0.72-cp37-ab13-win_amd64.whl (38.2 MB)
----- 13.8/38.2 MB 2.8 MB/s eta 0:00:09
```

## 单步调试

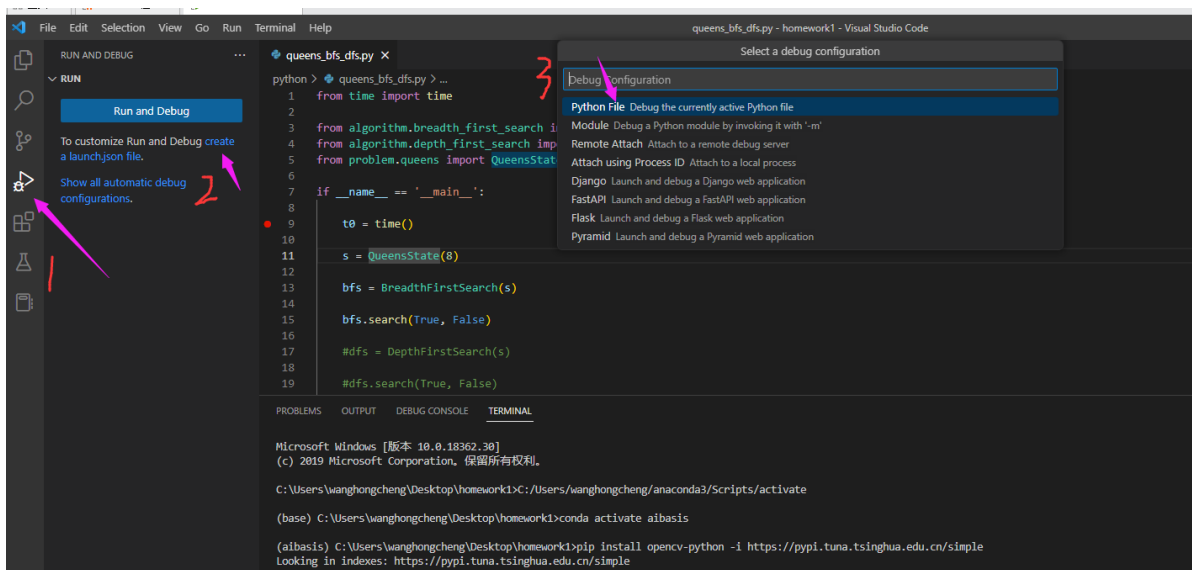
有时候程序有了bug，那么我们应该怎么办呢？一步一步的执行，然后看每个变量的值是什么样。

于是就有了单步调试。我们刚刚在VSCode中安装了python插件。现在得用起来了。

首先我们打开一个python文件，在行号的左侧按一下，会出现一个红点，这个就是“断点”，程序在运行到断点处会暂停，等你的下一步指令。



我们点击左侧的三角形，再点create a launch.json file，再点Python file.



然后就会出现一个launch.json文件。

“name”是名称，你可以任意修改。你会发现修改之后保存，左上角绿色三角形右侧的名称会随之改变

“type”不需要修改。

“request”不需要修改。

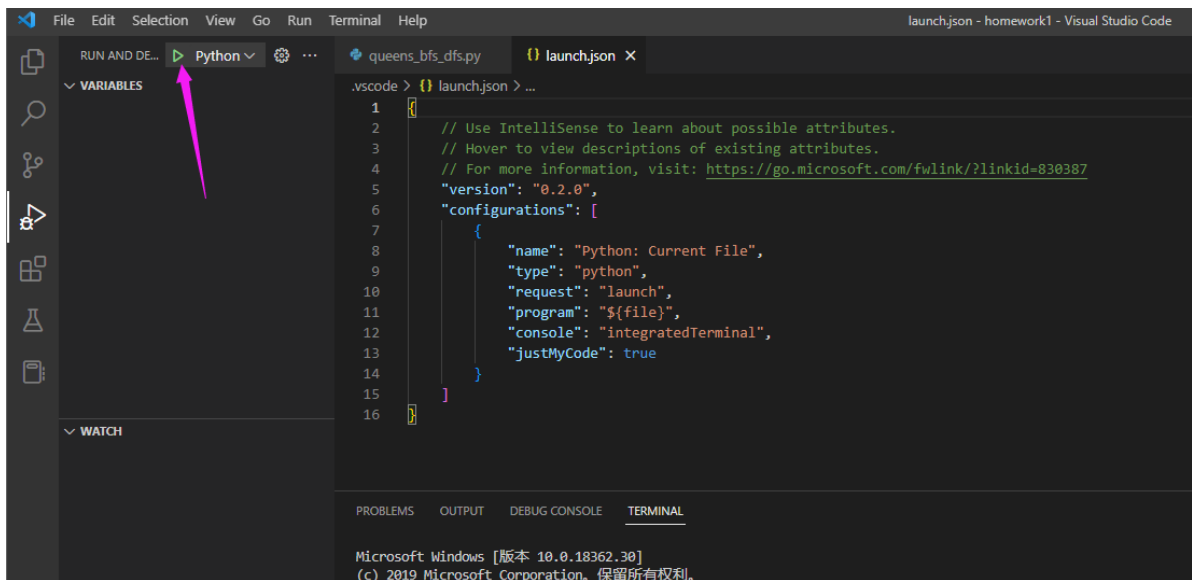
“program”，这个是当前这个configurations要跑的程序，如果是\${file}则说明是当前呈现的窗口中的代码。

"console"不需要修改

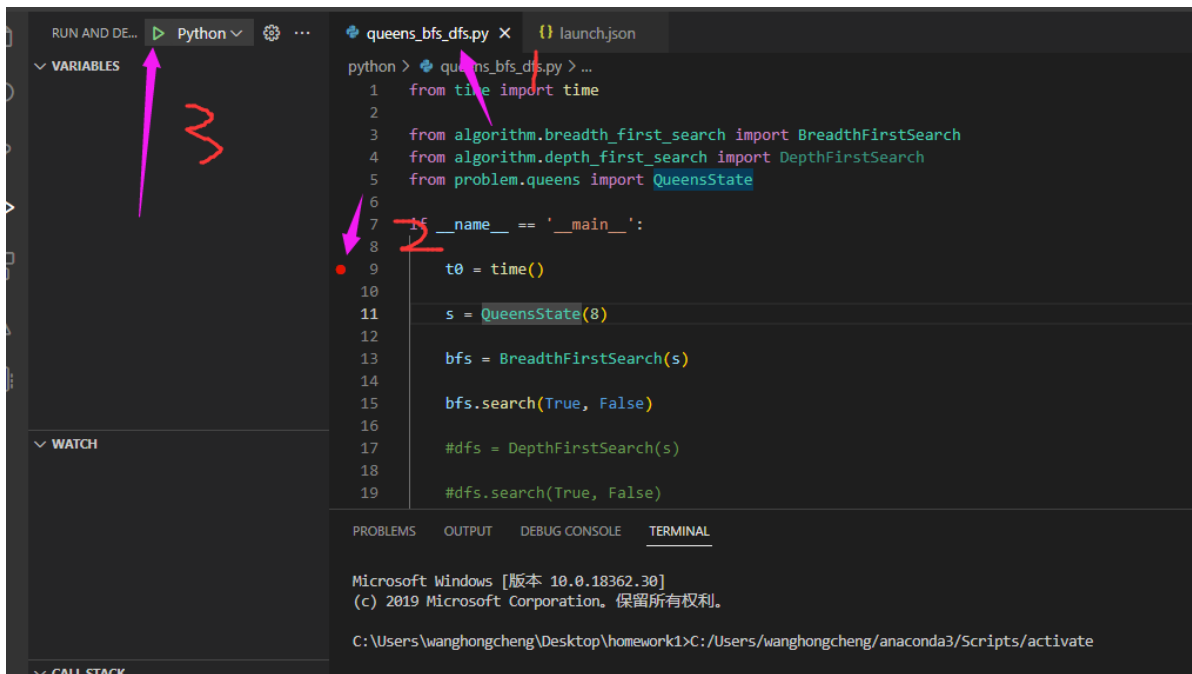
"justMyCode"不需要修改。

其实一开始学的时候，都不用修改。。大家可以自行了解其他的参数，比如"args"，"cwd"等。





我们把窗口切换回代码，然后打好断点（比如图中是第9行），然后点击左上角的绿色三角形，开始单步调试。



这时程序会直接运行到第9行，以黄色的作为该行配色。右上角出现了一个小窗，从左到右，第一个表示继续执行直到遇到下一个断点。

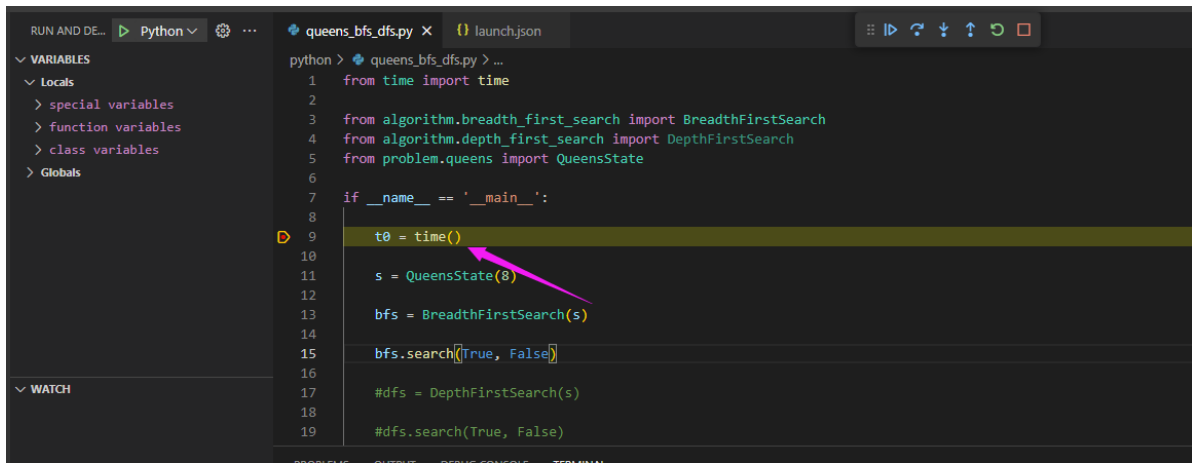
第二个表示往下执行(xíng)一行(háng)，忽略下一行的函数内部

第三个代表往下执行一行，并进入函数内部。

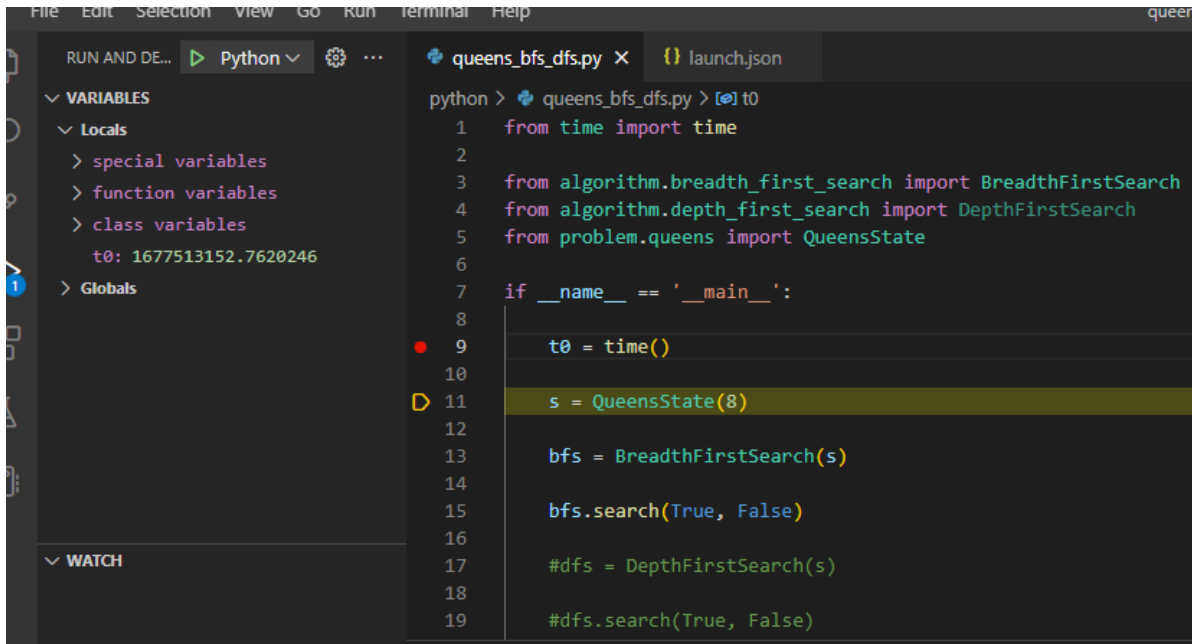
第四个和第五个我没用过，大家可以看说明书...（看样子也用不上）

第六个是终止所有执行，程序停止。

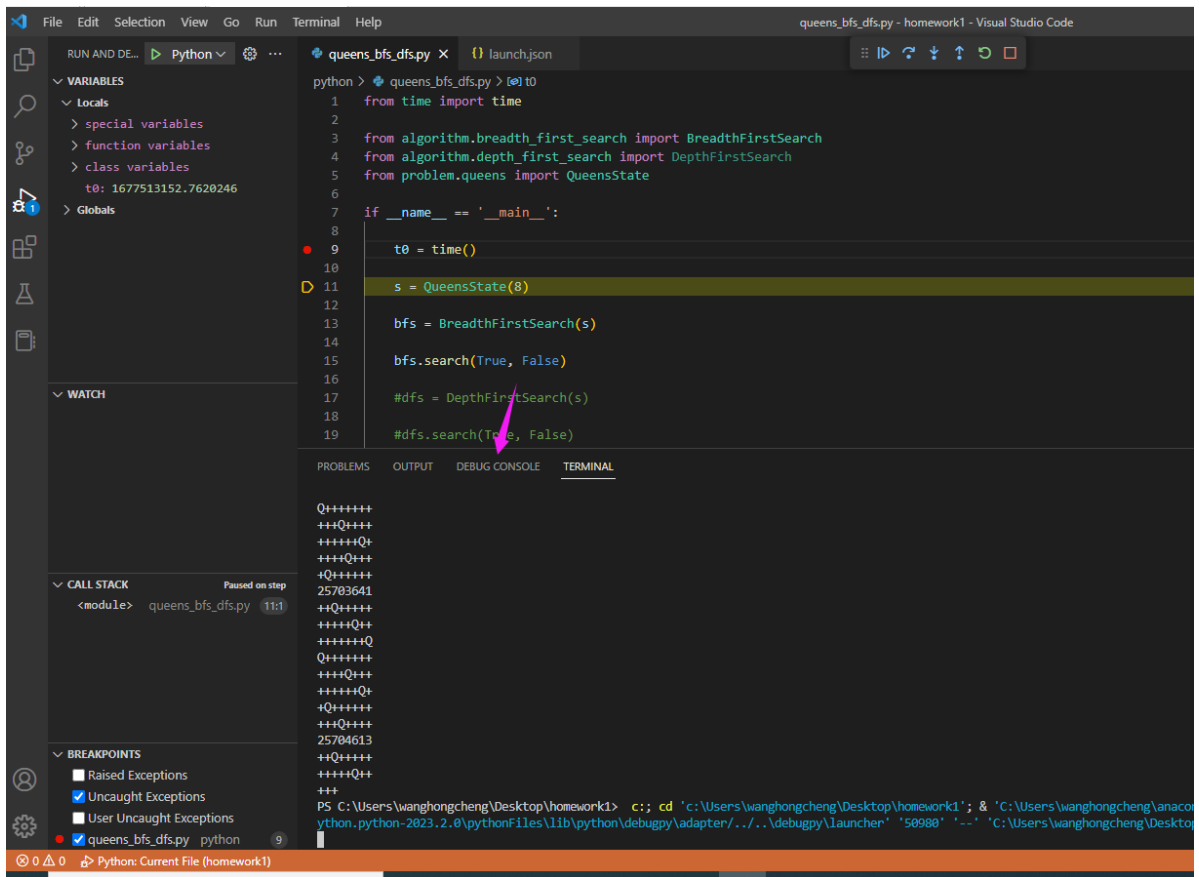
请大家自行探索这几个按钮的功能，特别是第二个和第三个的区别。



单步调试的时候，每一个执行完的变量都会出现在左上角，当然你也可以把鼠标悬浮到变量上面，然后会自然的浮现出这个变量的内容。

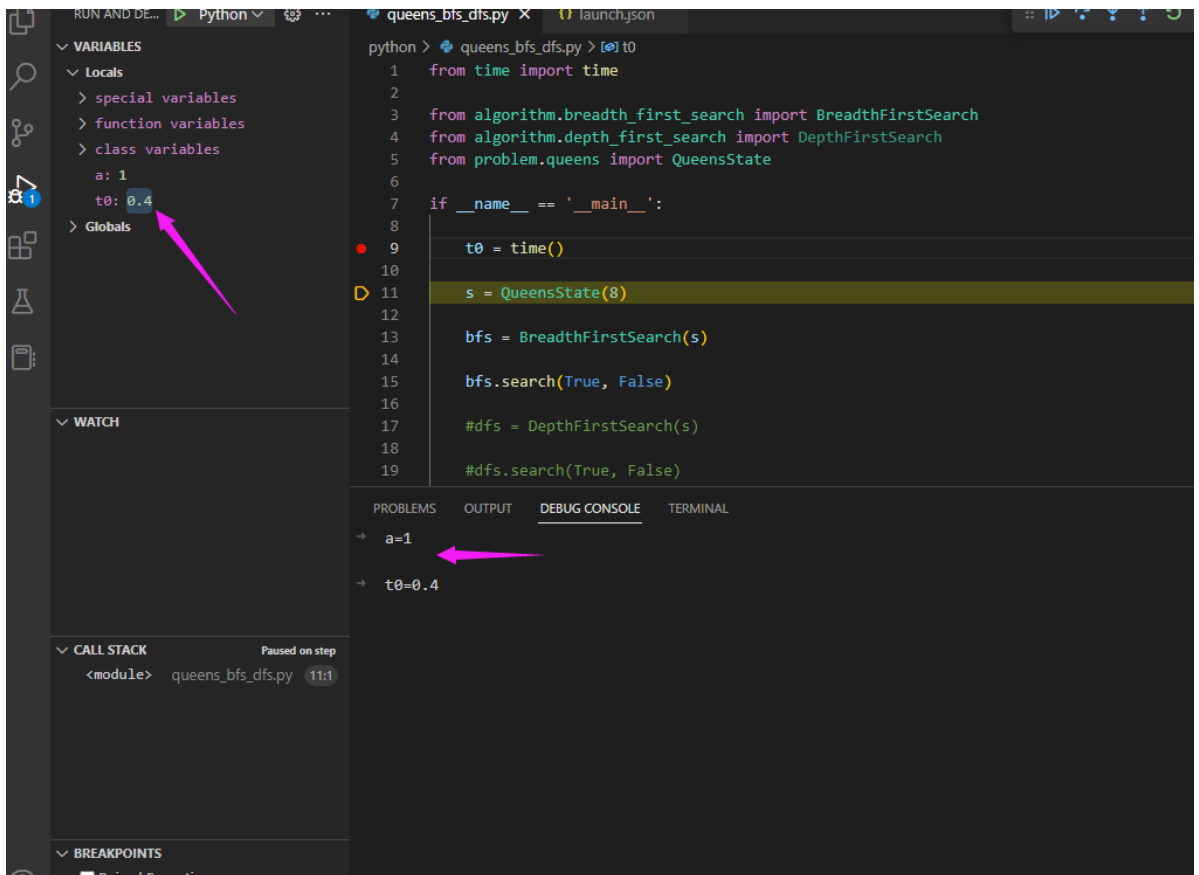


在下面的命令行中，还有一个神奇的功能，叫做debug console。

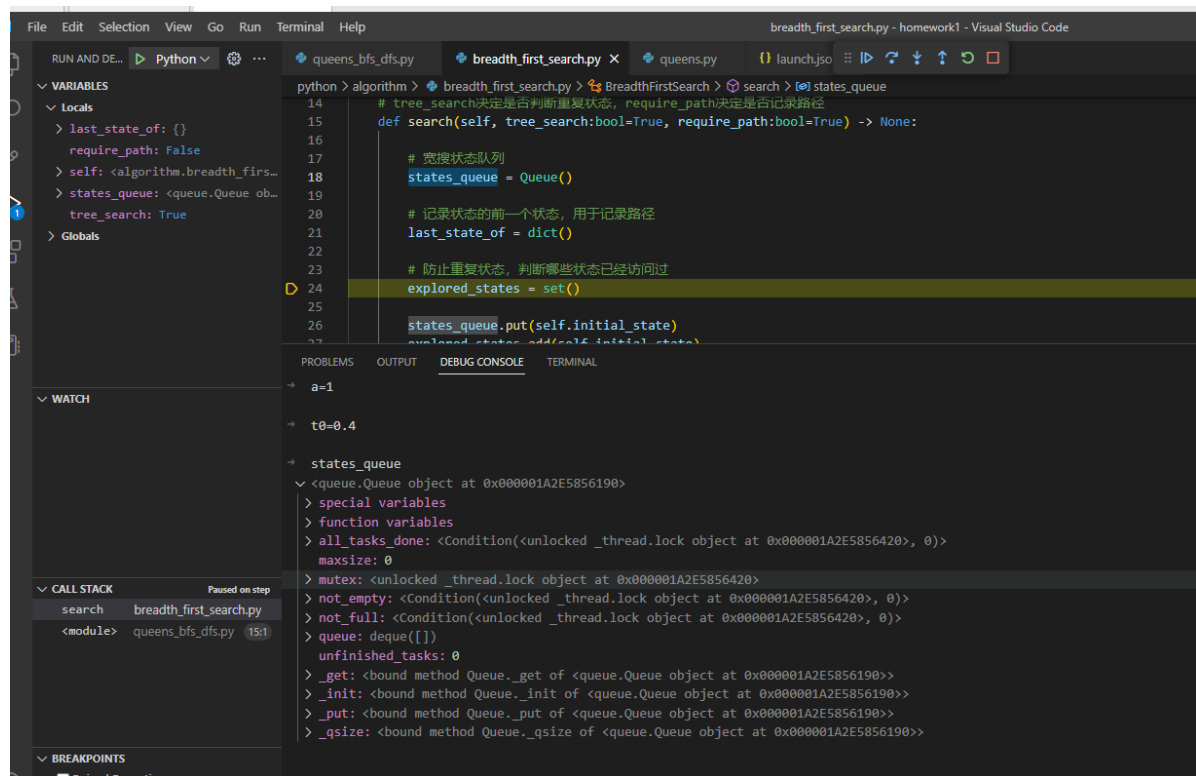


点一下它，会出现一个空白的窗口，最下面有一行可以输入的框。

这个的作用是实时与正在运行的程序交互。比如我可以设置一个变量a=1，也可以修改刚刚第9行的变量t0，在这里输入的内容，是一条python代码，按回车之后会被即刻执行。



我们也可以在这里输出某个变量，然后会呈现这个变量的所有内容。下图是我们进入到了search函数9（第15行）内部，然后输出了states\_queue的情况。



## 后记

这里的内容只是我临时起意写的，我并没有讲解任何如何写Python代码的知识，只是讲解了一些工具的使用，尽可能以简单的方式讲了一个非常重要的东西：虚拟环境。如果任何地方有名词看不懂，请在微信群内提出。

在我看来，python的逻辑和C/C++一样，语法上差别并不大，大家看过一遍python菜鸟教程（<http://www.runoob.com/python/python-tutorial.html>）或者廖雪峰python教程（<https://www.liaoxuefeng.com/wiki/1016959663602400>）中最基础的内容就可以。

我大二选了机器学习概论，第一次需要用到python，但是我还不会python，只能边看边学，网站的样例代码入手，跑一边看看结果，磕磕绊绊在百度的帮助下，我还是完成了大作业。

我非常建议大家做作业给的代码进行单步调试，看一看整个代码是怎样运行起来的。

整份入门教程写的很仓促，难免会有漏洞和错误，希望同学们指出。安装过程在一台全新的win10上进行，各位的可能会有之前各种疑难杂症导致安装不正确，我建议是把错误信息复制下来，百度搜索一下，如果没结果，再来寻求助教的帮助。自己解决报错是一项非常重要的能力，因为之后的科研中可能并没有人能帮你debug...（来自一个debug一下午的客户端）