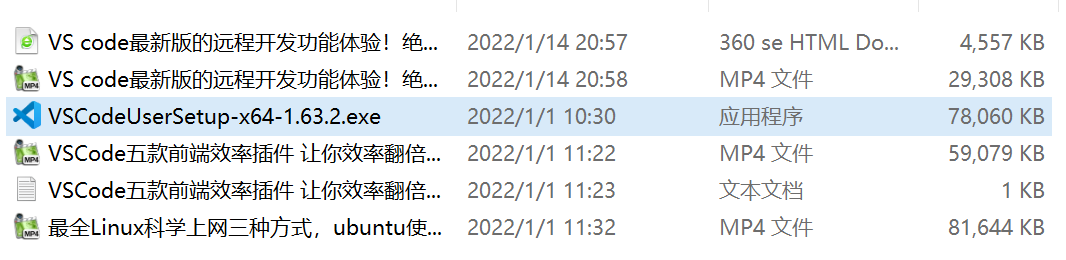
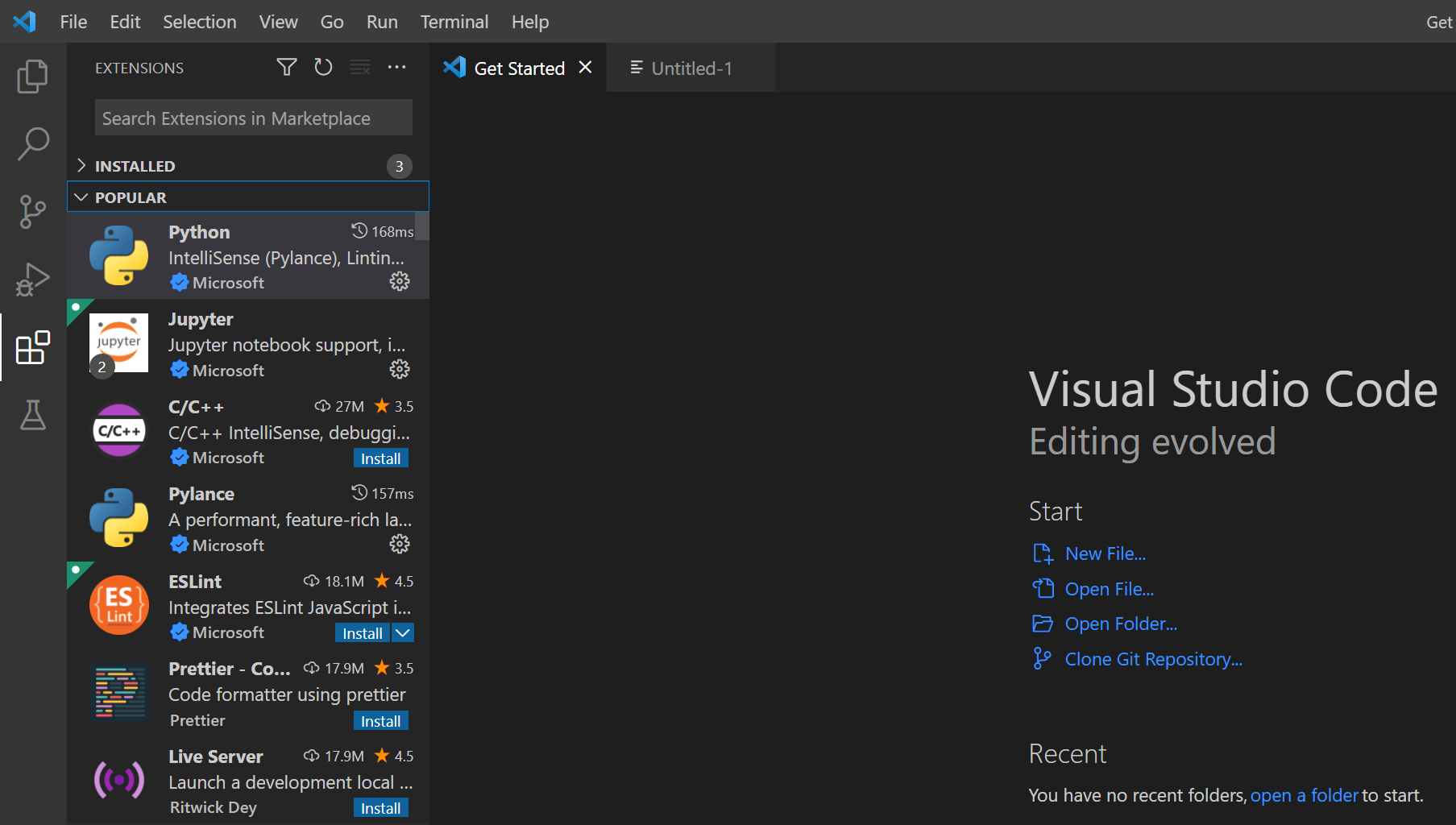
**如何安装VS Code实现远程python调试**

**第一步，安装VS Code**，在提供的文件家中选择安装包进行安装（选择合适路径）

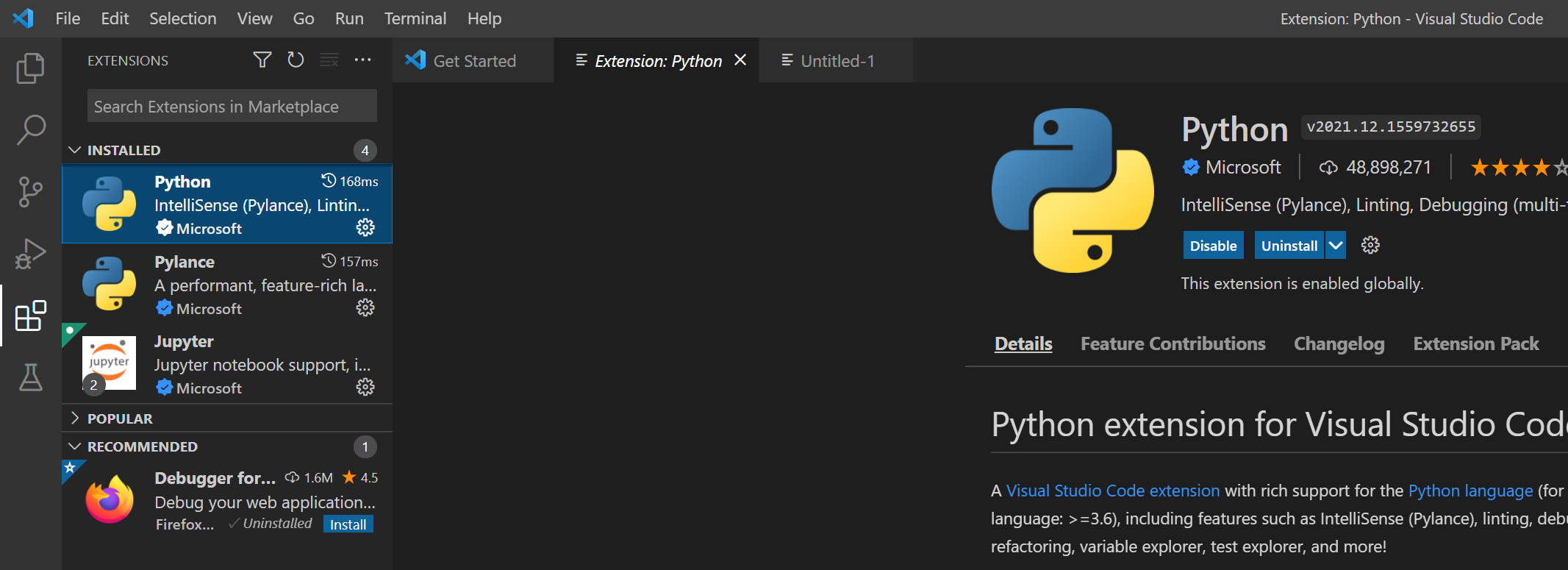


**第二步，安装python相关插件**

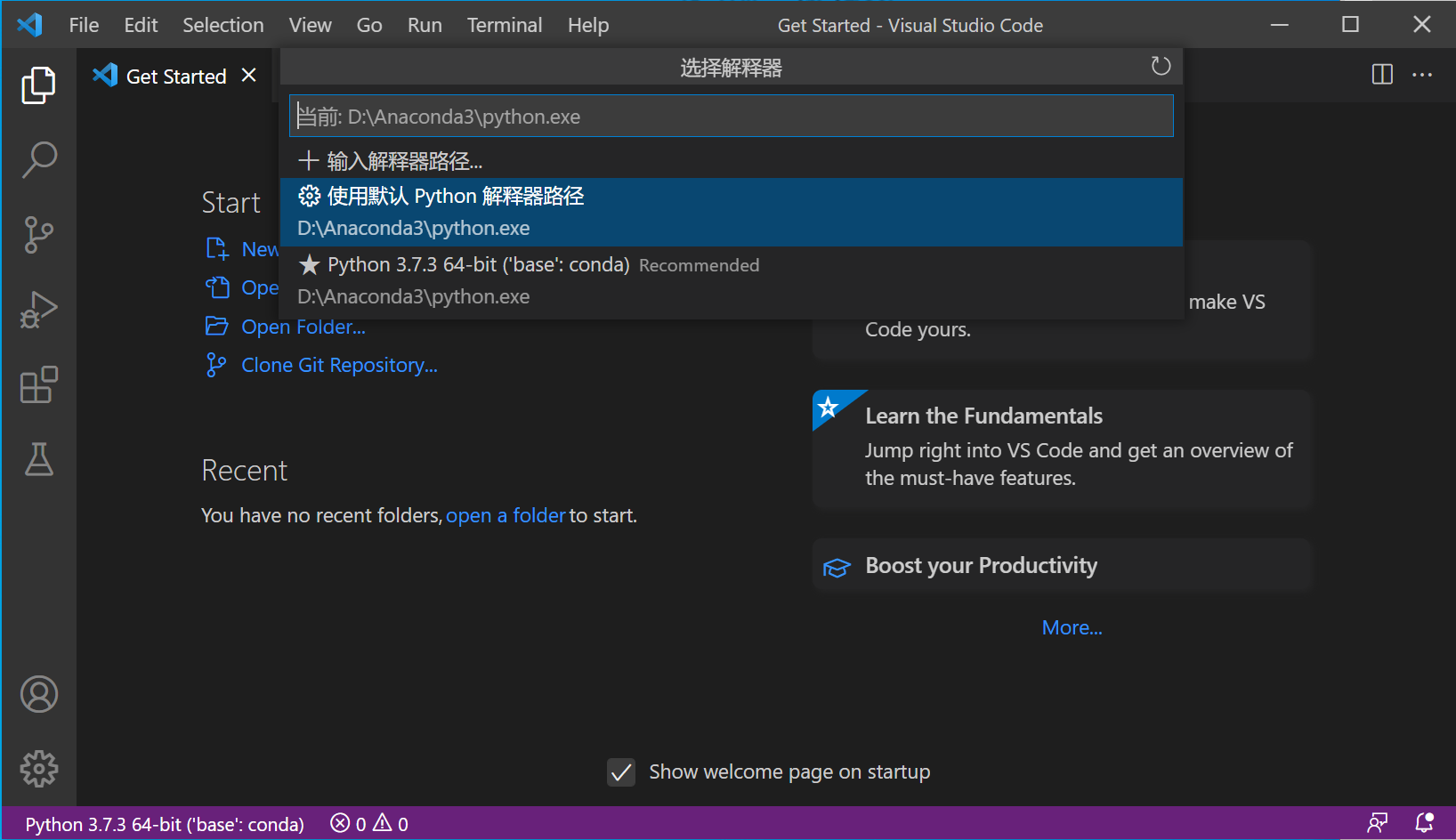
(1) 在**右侧工具栏**中选择扩展（Extensions），点击之后推荐**第一项就是python**，安装即可。如果推荐第一项不是python，在搜索栏中搜索“python”。



安装之后，可以发现在INSTALLED成功安装了3个插件(Python、Pylance、Jupyter)，如下：



需要注意的是这里**安装的是VS Code的python插件**，**python环境还需要另外安装**，比较推荐的是Anaconda环境。另外一个重要的问题是如何切换解释器？如下图所示，通过点击左下角的“python 3.7.3 64bit”，在正上方激活**“选择解释器”**功能。



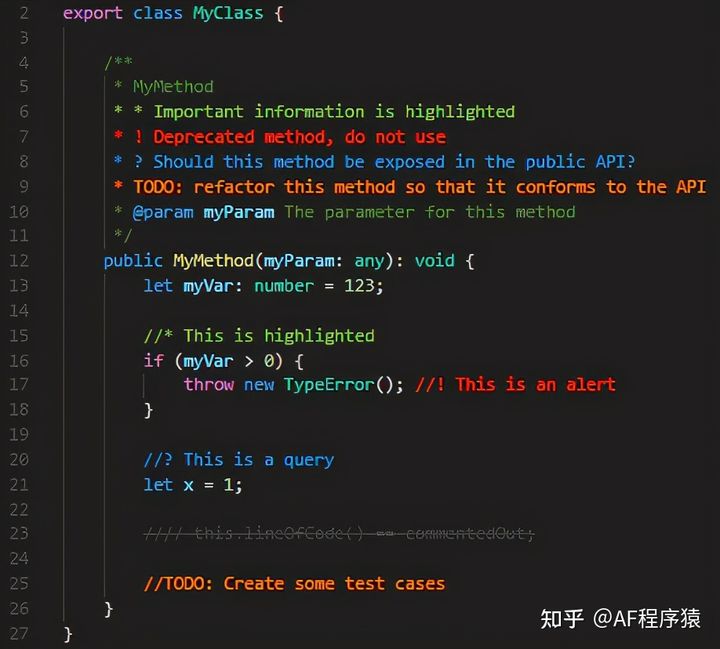
(2) 在**右侧工具栏**中选择扩展（Extensions），搜索**“Code Runner”**

**可以在 VS Code 中运行代码片段和代码文件，支持绝大部分编程语言。**



(3) 在**右侧工具栏**中选择扩展（Extensions），搜索**“Better Comments”**

* 感叹号 “!” 代码警告。
* 问号“?”代表存留疑问。
* TODO 代码未来将要进行的操作。
* @param 参数



(4) 在**右侧工具栏**中选择扩展（Extensions），搜索**“Python Indent”** ，纠正VSCode缩进错误的扩展

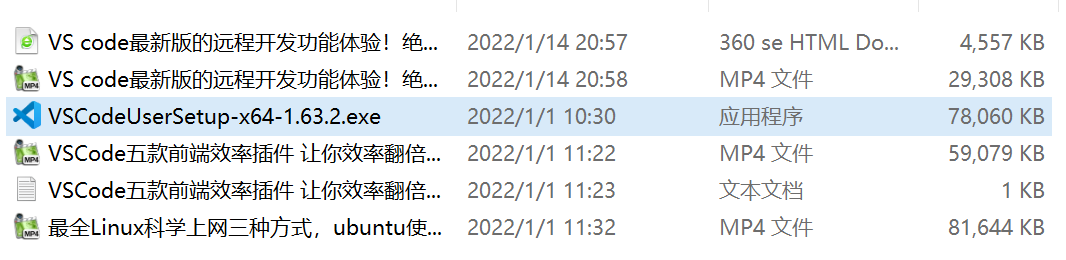


(5) 在**右侧工具栏**中选择扩展（Extensions），搜索**“Python Preview” （一般推荐）**



Python Preview可展现可视化调试的过程，并添加到我们的Python代码中。它将调试代码转换为带有动画和图形元素的交互式会话，以表示应用程序状态，对于代码调试非常直观。

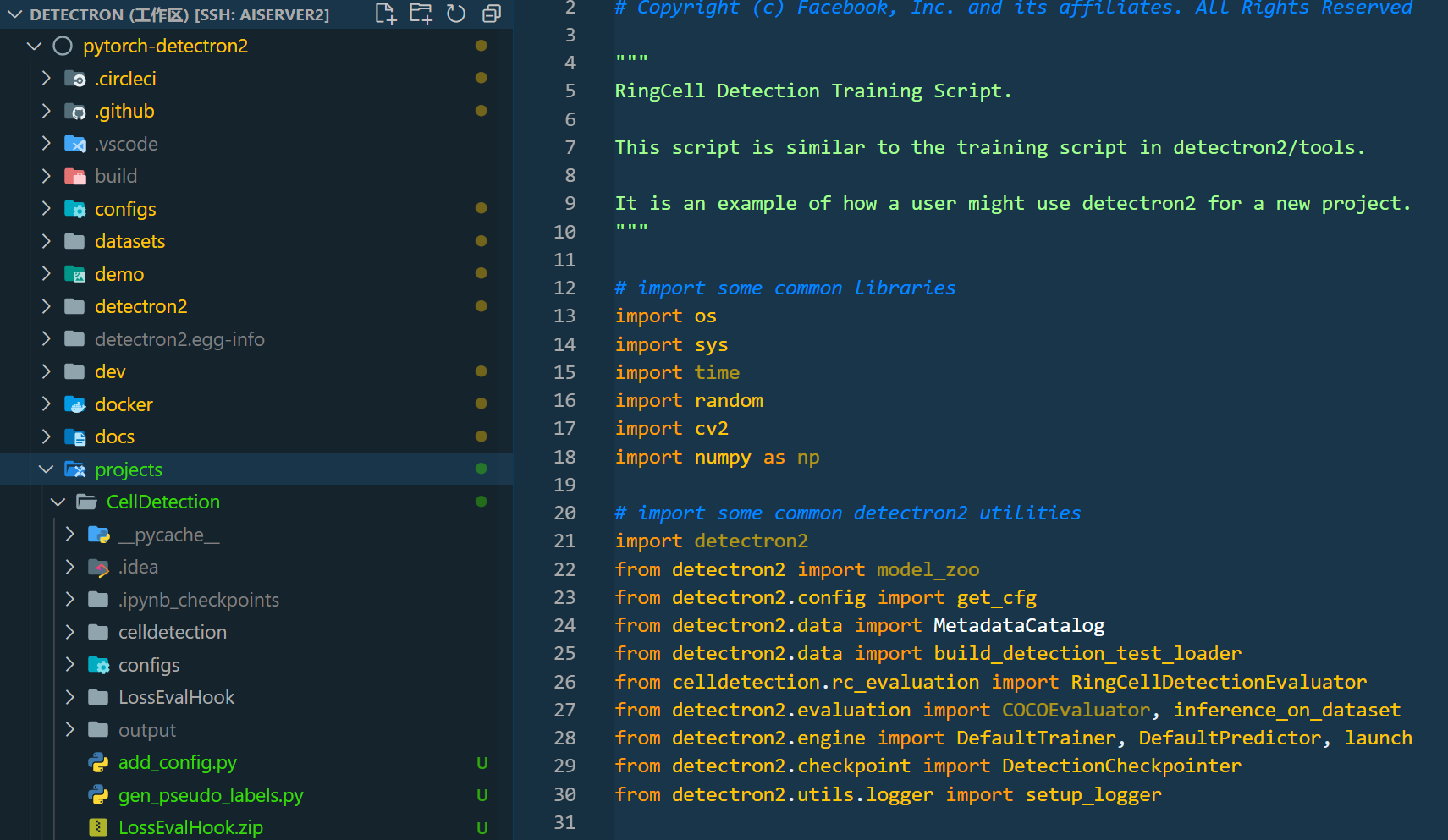
**第三步，安装一些常用插件，**按照文件夹所提供的“VSCode五款前端效率插件 让你效率翻倍”视频进行安装（包含使用介绍）。



(1) 在**右侧工具栏**中选择扩展（Extensions），搜索**“Material Icon Theme”**，安装之后通过右侧**“设置文件图标主题”**可以改变VS Code图标。



设置后工作区中图标变为：



此外，漂亮的主题也会让代码更加醒目，同时心情更加愉悦！！这里推荐主题有“**Cobalt2 Theme Official**”，**”Ayu”(Mirgage Bordered)**，” [**Monokai Pro**](https://www.oschina.net/action/GoToLink?url=https%3A%2F%2Fmarketplace.visualstudio.com%2Fitems%3FitemName%3Dmonokai.theme-monokai-pro-vscode)”

设置字体：默认配置（Consolas, 'Courier New', monospace）

(2) **右侧工具栏**中选择扩展（Extensions），搜索**格式化排版“Prettier”** ，注意远程和本地需要单独安装。

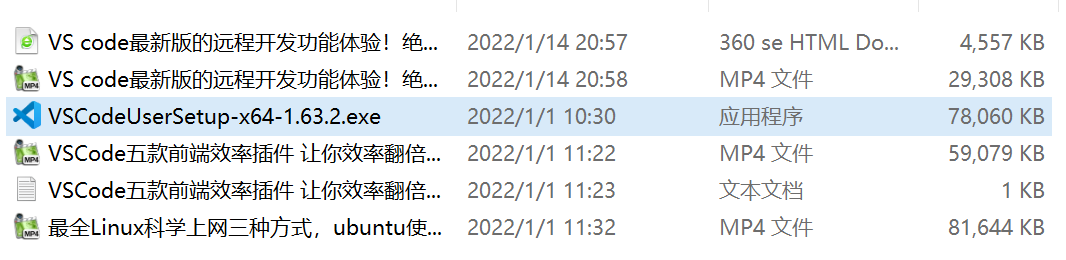


安装之后，需要在 **管理 –> 设置 –> 文本编辑器 –> 格式化 –> Format On Save**，勾选，每次保存时都会对代码进行格式化。



(3) **右侧工具栏**中选择扩展（Extensions），搜索并安装 显示大括号区块**“Bracket Pair Colorizer 2”** ，**自动完成对应的Tag“Auto Rename Tag”**，**快速显示CSS “CSS Peek”**

**第四步，安装远程调试插件“Remote-SSH”**，按照文件夹所提供的“VS code最新版的远程开发功能体验！绝对不能错过的开发利器！”视频进行安装（包含使用介绍）。

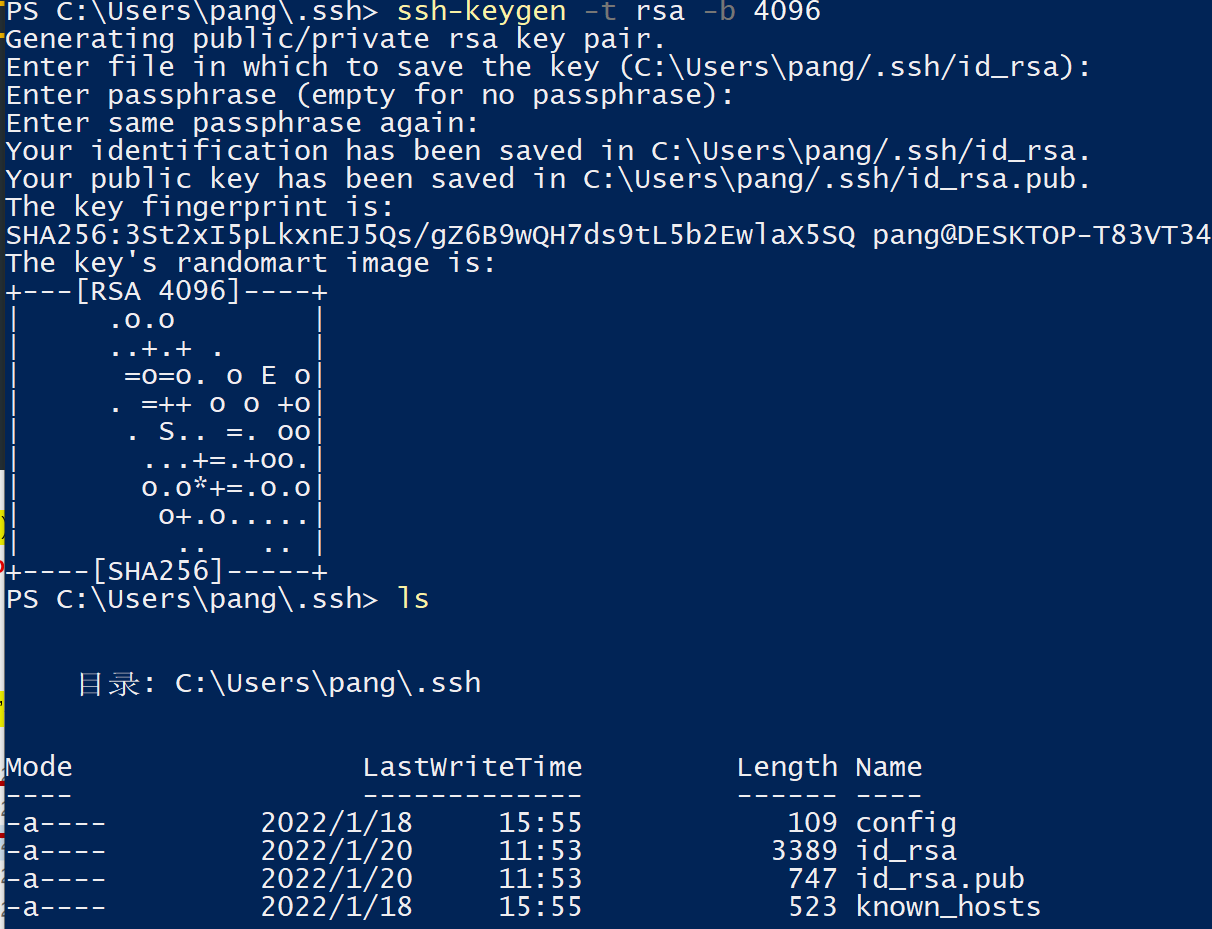


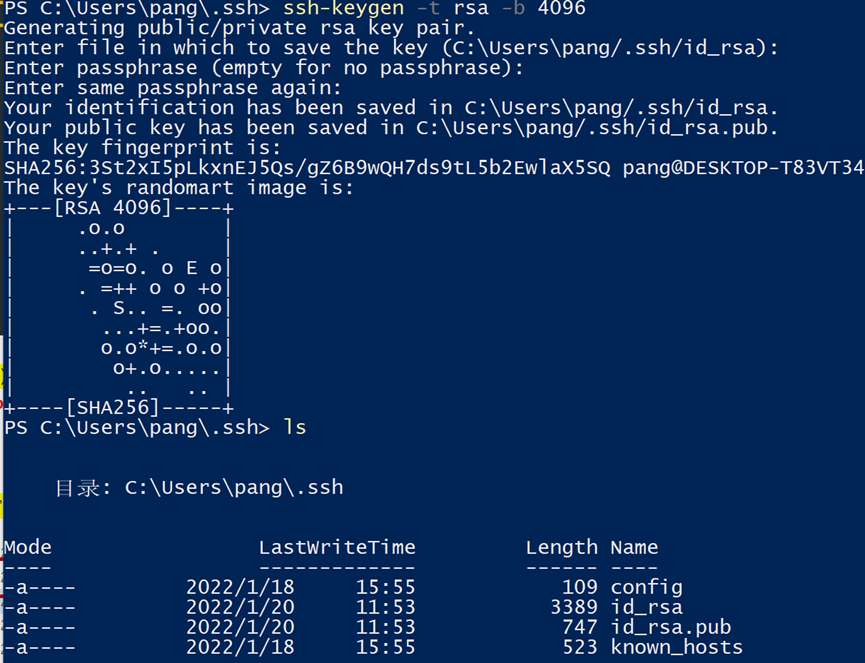
(1) **右侧工具栏**中选择扩展（Extensions），搜索**“remote”**，**安装“Remote - SSH”**



(2) 配置基于Key的授权（默认通过密码）

**首先，制作密钥对**，在CMD中输入：ssh-keygen -t rsa -b 4096，然后输入回车即可，在C:\Users\pang\.ssh 路径下。





**然后，在服务器上安装公钥**，具体步骤如下。 （需要问我）

授权本地Windows主机进行远程连接，在本地的**PowerShell命令窗**中运行下列命令，需要替换user-name和hostname。

* Connecting to a **macOS or Linux** SSH host:

$USER\_AT\_HOST= "your-user-name-on-host@hostname"

$PUBKEYPATH="$HOME\.ssh\id\_rsa.pub"

$pubKey=(Get-Content "$PUBKEYPATH" | Out-String); ssh "$USER\_AT\_HOST" "mkdir -p ~/.ssh && chmod 700 ~/.ssh && echo '${pubKey}' >> ~/.ssh/authorized\_keys && chmod 600 ~/.ssh/authorized\_keys"

在校园网环境下(主机名10.32.41.57，端口号：默认22，用户名pang) 上述命令修改如下，切换目录到C:\\Users\\\*\*\*\\.ssh,运行下面代码。

$USER\_AT\_HOST= "pang@10.32.41.57"

$PUBKEYPATH="$HOME\.ssh\id\_rsa.pub"

$pubKey=(Get-Content "$PUBKEYPATH" | Out-String); ssh "$USER\_AT\_HOST" "mkdir -p ~/.ssh && chmod 700 ~/.ssh && echo '${pubKey}' >> ~/.ssh/authorized\_keys && chmod 600 ~/.ssh/authorized\_keys"

外网环境,(主机名39.106.12.194，端口号：7100，用户名pang)上述命令修改如下：

$USER\_AT\_HOST= "pang@39.106.12.194"

$PUBKEYPATH="$HOME\.ssh\id\_rsa.pub"

$pubKey=(Get-Content "$PUBKEYPATH" | Out-String); ssh -p 7100 "$USER\_AT\_HOST" "mkdir -p ~/.ssh && chmod 700 ~/.ssh && echo '${pubKey}' >> ~/.ssh/authorized\_keys && chmod 600 ~/.ssh/authorized\_keys"

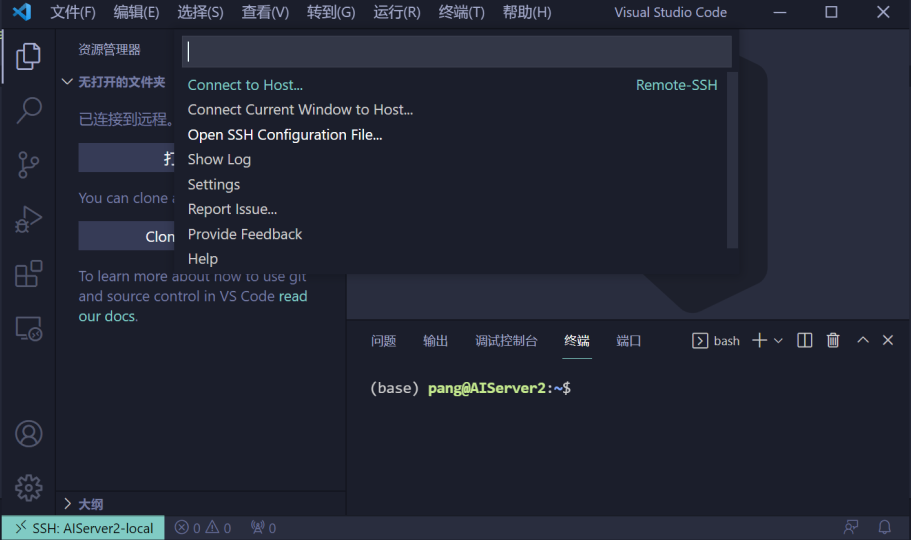
外网环境,(主机名39.106.12.194，端口号：7200，用户名pang)上述命令修改如下：

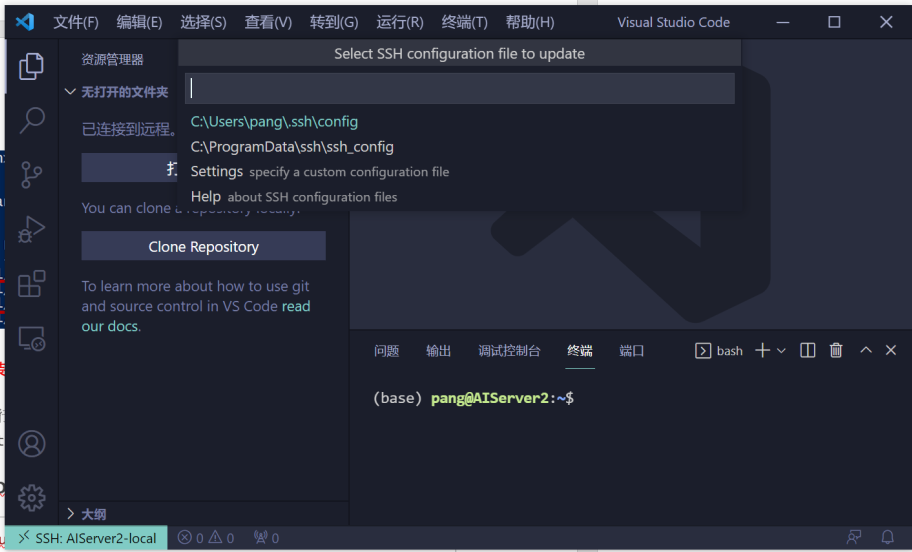
$USER\_AT\_HOST= "pang@39.106.12.194"

$PUBKEYPATH="$HOME\.ssh\id\_rsa.pub"

$pubKey=(Get-Content "$PUBKEYPATH" | Out-String); ssh -p 7200 "$USER\_AT\_HOST" "mkdir -p ~/.ssh && chmod 700 ~/.ssh && echo '${pubKey}' >> ~/.ssh/authorized\_keys && chmod 600 ~/.ssh/authorized\_keys"

**按照视频介绍利用密码连接远程主机**，成功之后点击左下角SSH：10.32.41.57，中间选**Open SSH Configuration File…**，接着选ssh的config文件位置，在需要配置的主机上添加密钥的位置。







(3) 如何通过内网穿透登陆远程主机

公网主机IP：39.106.12.194 转发端口：7100

在配置主机时，原本命令 ssh 用户名@主机IP(例：ssh pang@10.32.41.57)，需要使用内网穿透的主机IP和端口号，格式如下：

ssh –oPort=公网主机转发端口 用户名@公网主机IP

例如：ssh –oPort=7100 pang@39.106.12.194

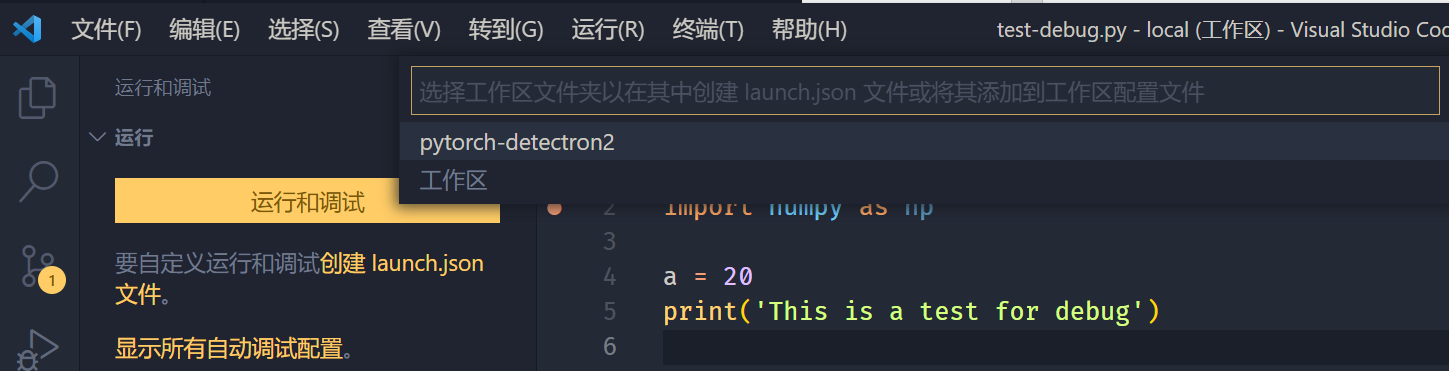


**第五步，如何进行程序的Debug**，按照这里主要介绍在**本地调试**和**远程调试**（校园网内和内网穿透两种情况下）。**目前测试发现远程服务器利用本地调试也可用**

**(1) 本地调试**：**右侧工具栏**中选择调试，在“运行和调试”下面点击**创建launch.json文件**。



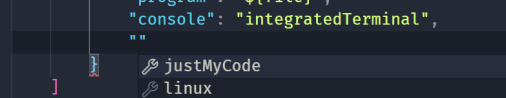
点击后出现配置选择（图中红框），这里选项是**当前工程（pytorch-detectron2）**或者**工作区**，可以单独为当前工程或整个工作区配置Debug选项。



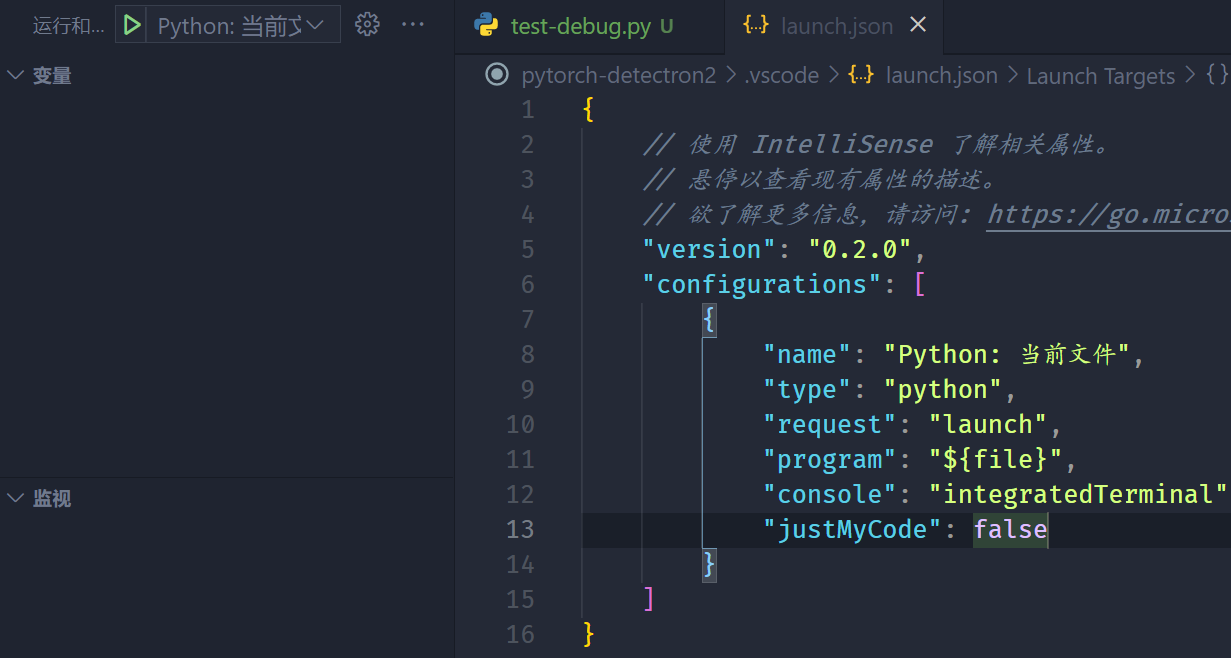
上述选项之后，进一步出现如下选项，本地调试选择第一项**“Python文件”**，远程调试选择**“远程连接”**。这里选择第一项。



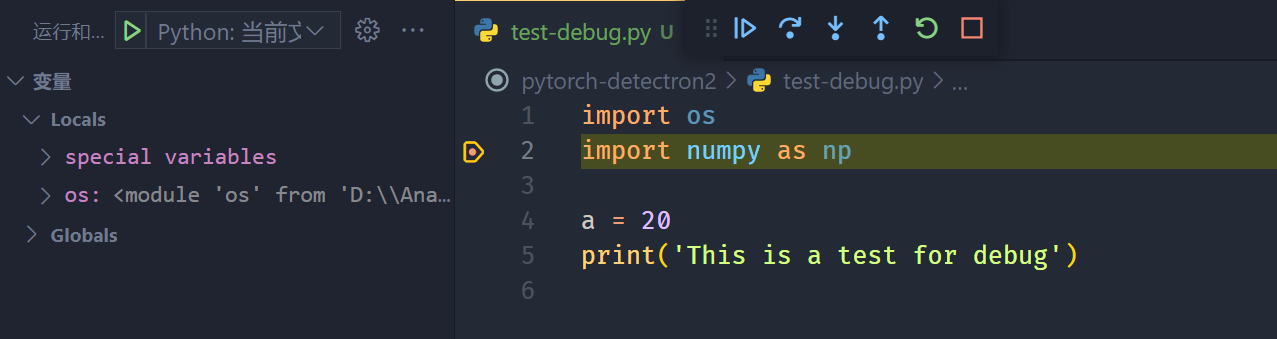
选择之后会出现相应的配置文件launch.json，后续可以通过**左侧的“小齿轮”**打开配置文件，这个配置文件中一些参数需要修改（特别是在远程调试过程中），在本地调试中介绍一个重要参数**“justMyCode”（，+回车+“”会出现提示）**。



默认是true,表示只调试自己写的代码，调试不会进入安装的库文件，例如，torch包中自带的函数无法调试[[1]](#footnote-1)。因此，有些时候需要将其设为false



配置完成之后，点击运行（绿色三角），即可进行调试。代码的左侧点击可以加断点，正上方有控制面板（黄色框所示），利用控制面板可以调试程序。



**目前测试发现远程服务器利用本地调试也可用，后面暂时可以不用**

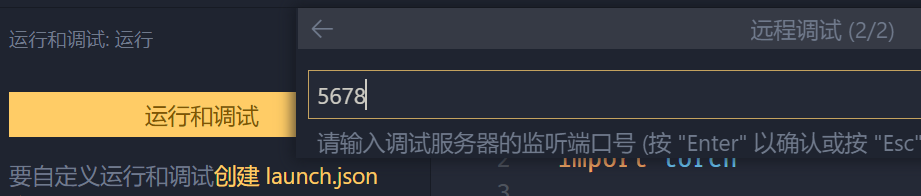
**(2) 远程调试**：在**“Python文件”**和**“远程连接”**步骤中选择“远程调试”。



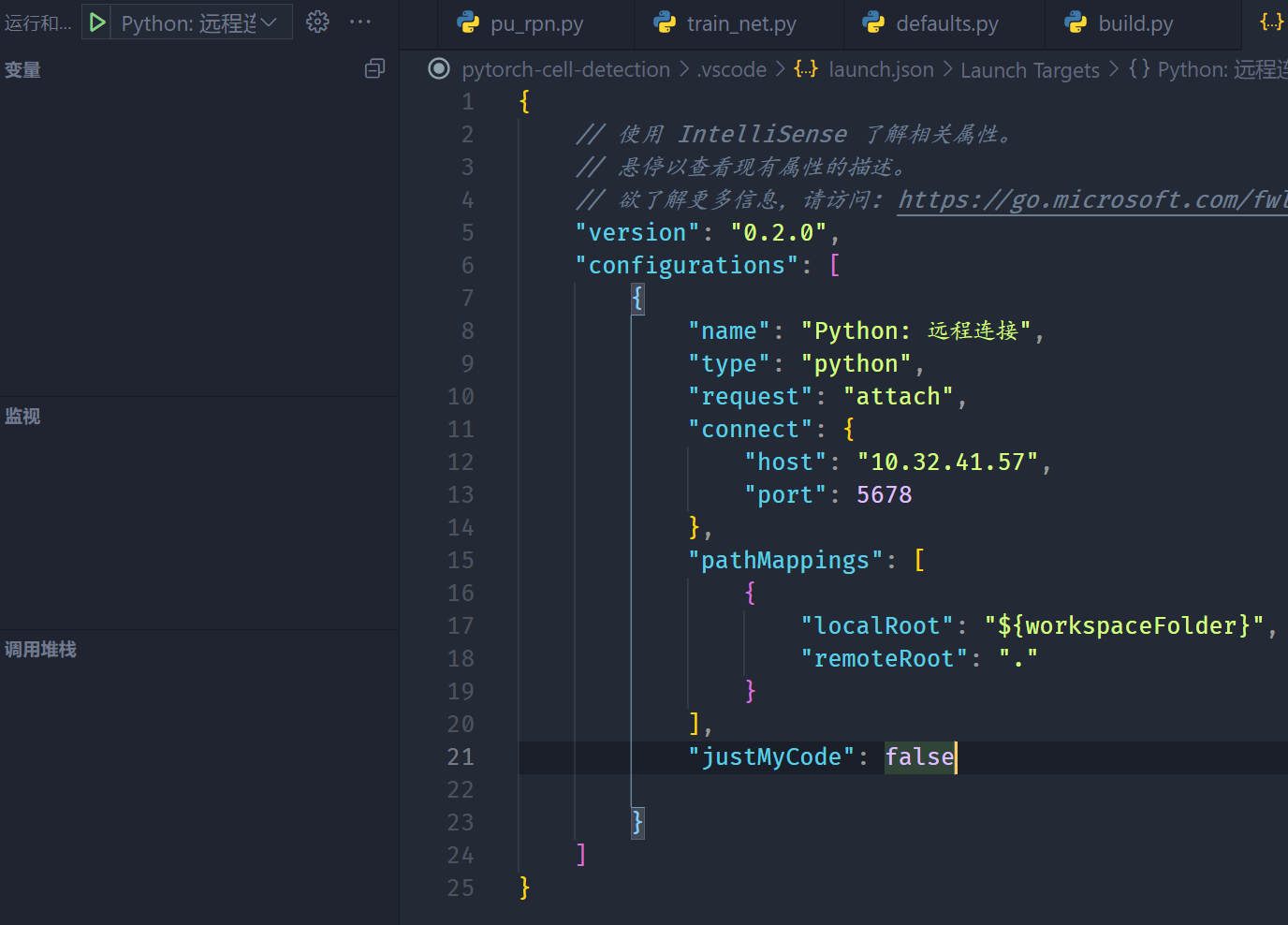
下一步输入远程主机IP，**校园网（10.32.41.57）**或**内网穿透（39.106.12.194）**



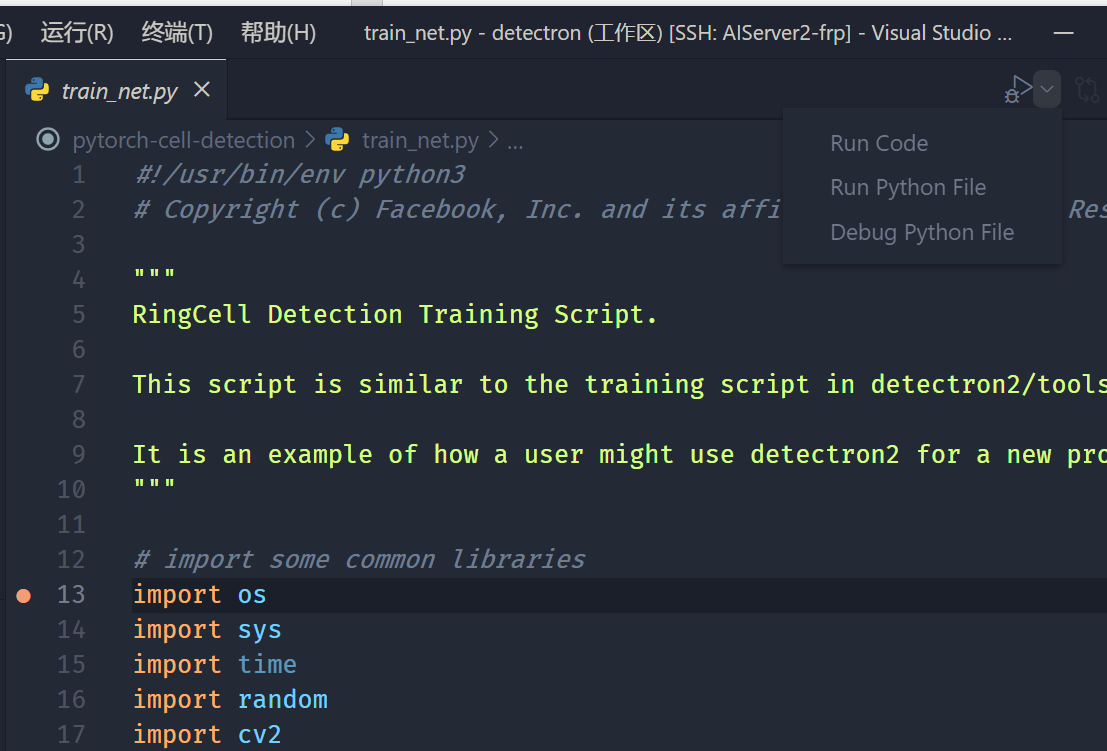
下一步输入端口号，**校园网（使用默认5678）**，而**内网穿透（使用ssh端口号7100）**



然后，会生成如下配置文件，为了方便调试，这里添加**"justMyCode": false**.



注意：远程调试无法通过点击运行（绿色三角）进入，点击**右上角运行（三角），选择最后一项“Debug Python File”**,需要提前设置断点，否则会整体执行。



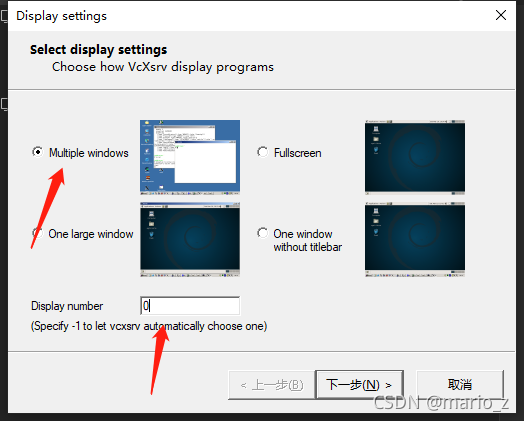
**第六步，如何在VSCode本地显示程序中的图像**，按照这里主要介绍基于X server的方法，利用jupyter notebook的方法很简单，但jupyter调试不是很方便。但基于X server的方法只能在局域网实现，远程的应用还是使用jupyter吧。

首先，需要装MobaXterm，这是一个功能强大的ssh工具，然后，需要安装vcXsrv。

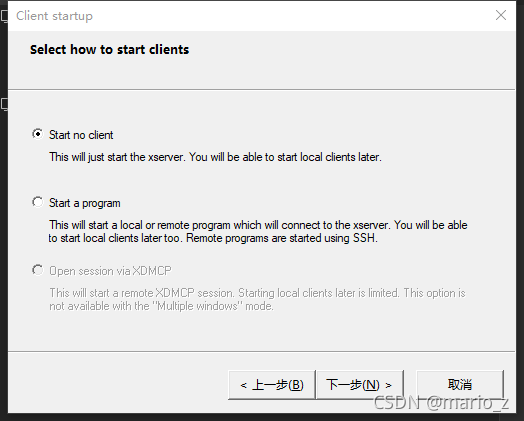
在**配置vcXsrv之前**，先启动MobaXterm，然后，启动并配置vcXsrv，过程如下：



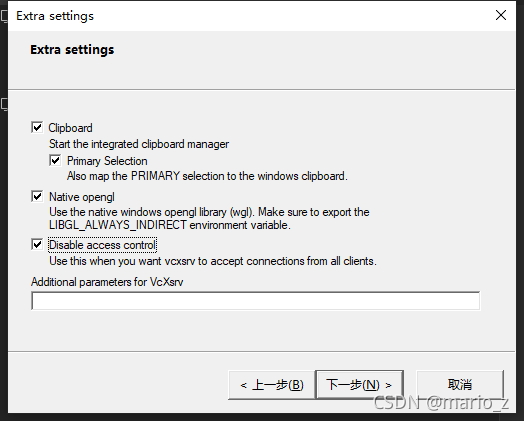
启动，现在Multiple windows, Display number 选中0.



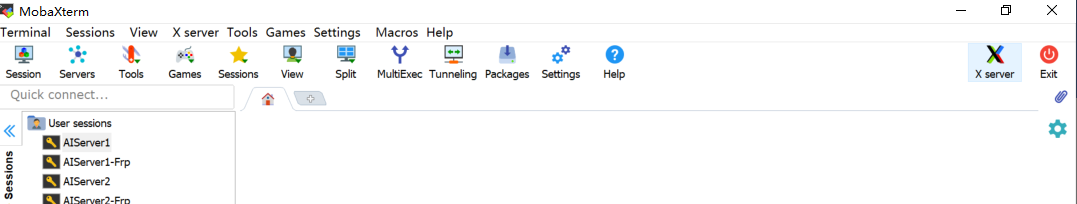
下一步， start no client



下一步，三个都选，完成



设置成功后，鼠标停靠在MobaXterm右上角的X Server上，会出现“Current DISPLAY=**10.32.28.40**:0.0”的提示，10.32.28.40是本机IP地址。

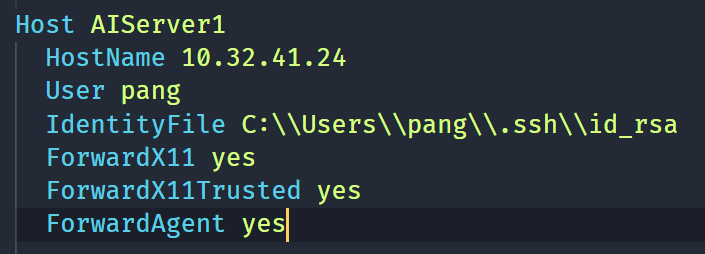


然后，配置VSCode, 点击左下角设置，输入open ssh configration file.后面添加

ForwardX11 yes

ForwardX11Trusted yes

ForwardAgent yes



最后，查找本地Windows的IP地址(如：**10.32.28.40**)，在vscode运行代码的终端(运行模式或调试模式)，输入如下语句并执行：

export DISPLAY="**10.32.28.40**:0.0"

之后，运行画图程序就本地就会跳出Figure。

1. Detectron2的函数调用是“注册”机制，调用关系不是很明显，需要设为false [↑](#footnote-ref-1)