

# 环境搭建

## 一、.Atom安装与使用

### 1. 下载安装

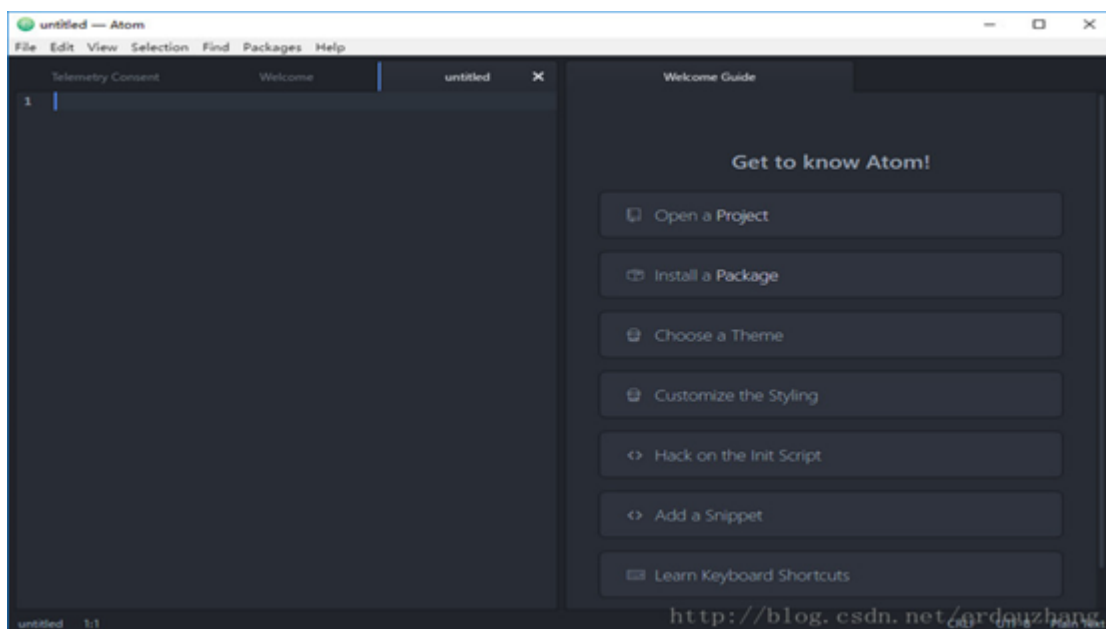
Atom官网: <https://atom.io>



下载之后双击文件AtomSetup-x64.exe



安装运行过程有点慢。之后显示这个界面：



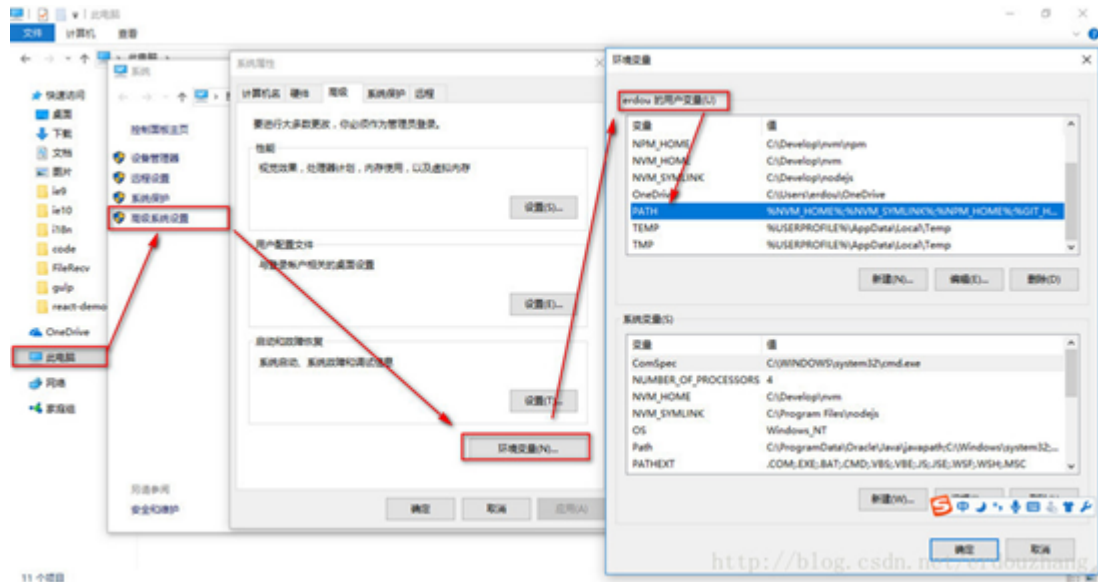
## 2. 配置环境变量

由于我们安装插件需要用到apm这里命令，这里需要将其添加到环境变量里面，如图。

在桌面Atom图标→右击→属性→查看安装位置。默认安装位置是在：C:\Users\erdou\AppData\Local\atom。

现在配置环境变量：在PATH后面添加;C:\Users\erdou\AppData\Local\atom\bin

注意前面有一个英文状态分号。这样就可以使用apm命令了。



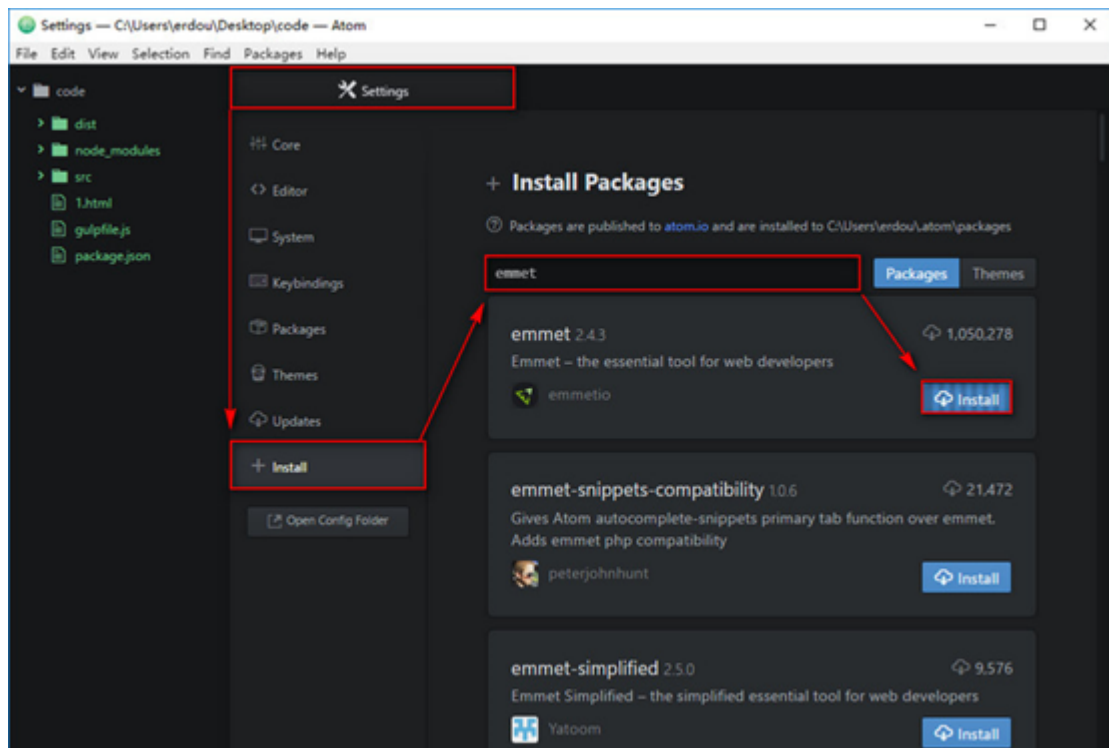
打开命令窗口看是否可以使用apm Win+R → cmd → apm ls 查看安装了哪些包：apm ls；

如果没有安装插件会显示：Community Packages (0)

## 3. atom插件安装

方法一

这是最简单的插件安装，但可能会安装很慢。（如果安装特别慢，就直接用方法二）



## 方法二

以安装emmet插件为例子：

①Ctrl+Shift+P: 打开面板

②输入：install packages选择install Packages Themes

③输入要安装的插件名字emmet

④点入插件介绍 找到github地址

⑤打开git切换到 C:\Users\erdou.atom\packages

⑥git clone <https://github.com/emmetio/emmet-atom>，切换到安装的插件文件夹：

cd emmet-atom

⑦npm install

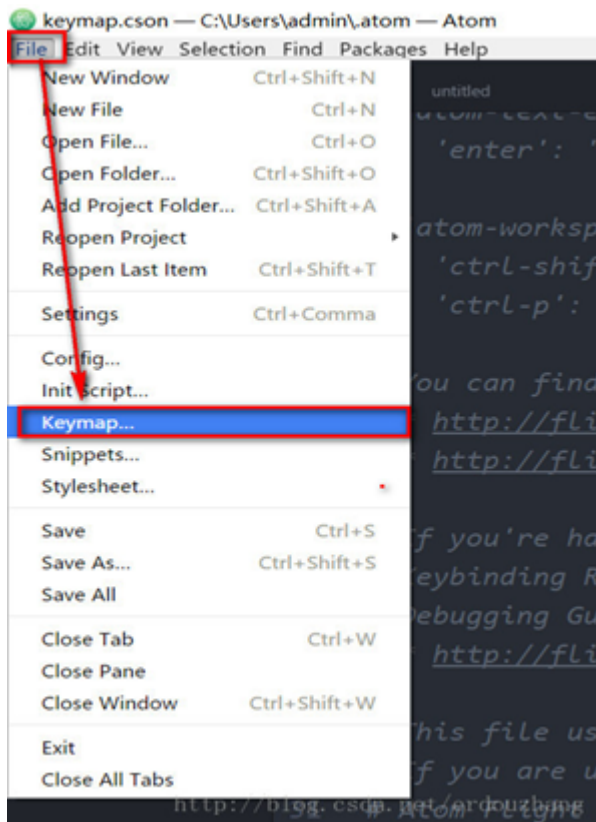
⑧重启Atom

下面是参考：

```
1 cd ~/.atom/packages
2 git clone https://github.com/emmetio/emmet-atom
3 cd emmet-atom
4 npm install
```

## 4.如何更改Atom快捷键

打开keymap.cson文件：



把要改的快捷键写在最下面，依次添加：

```
31 # Atom Flight Manual:
32 # http://flight-manual.atom.io/using-atom/sections/basic-customization/#
33
34 'atom-text-editor':
35 # F12在浏览器预览html页面
36 'f12': 'rib:run-in-browser'
37 #右边栏实时浏览器调用快捷键
38 'ctrl-F12': 'atom-html-preview:toggle'
```

```
1 'atom-text-editor':
2 # F12在浏览器预览html页面
3 'f12': 'rib:run-in-browser'
4 #右边栏实时浏览器调用快捷键
5 'ctrl-F12': 'atom-html-preview:toggle'
```

## 5.linux下安装

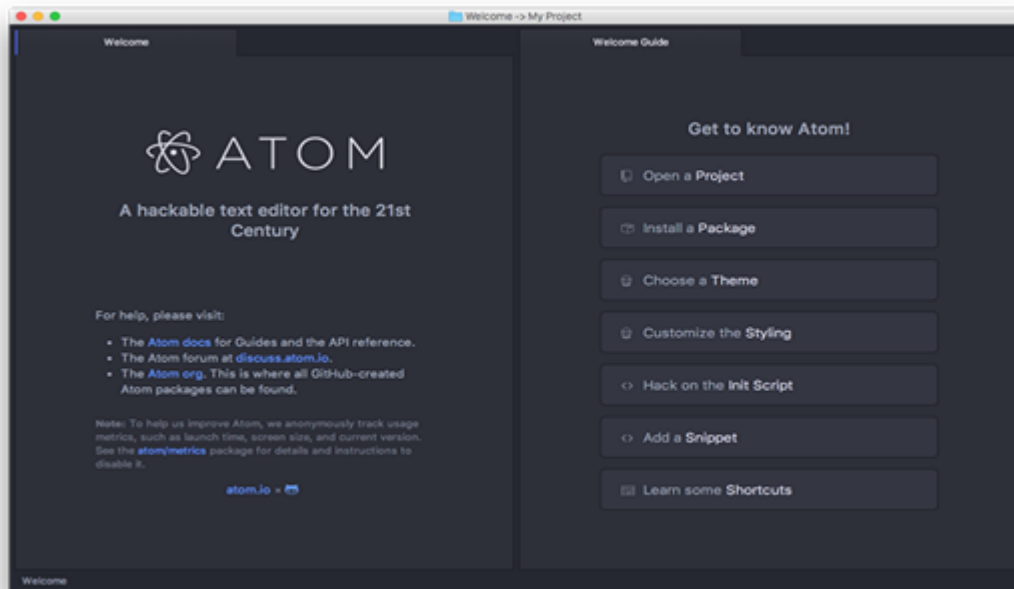
如果是基于Debian的发行版, 执行命令:

```
sudo dpkg -i atom-amd64.deb1
```

如果是RedHat, 则执行:

```
rpm -i atom.x86_64.rpm1
```

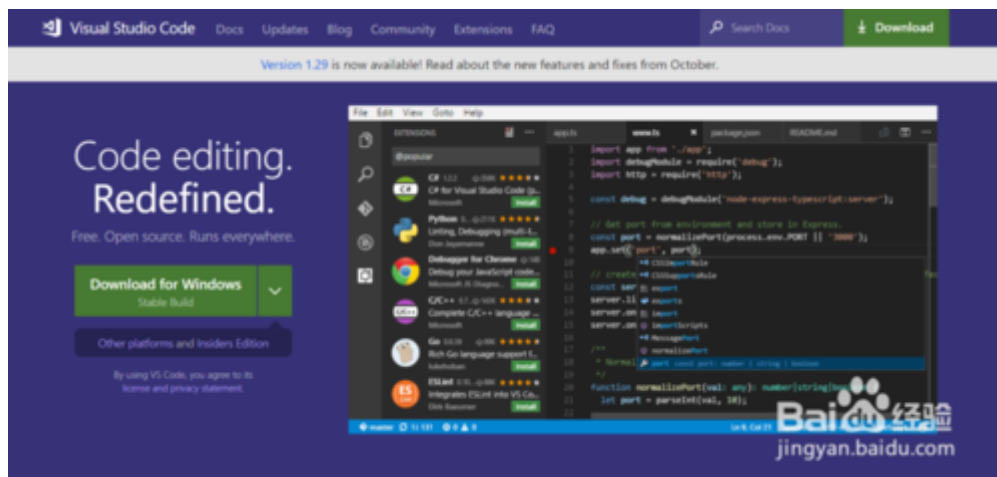
安装完成后运行Atom, 将会看到这样的界面, 说明我们已经安装成功了。



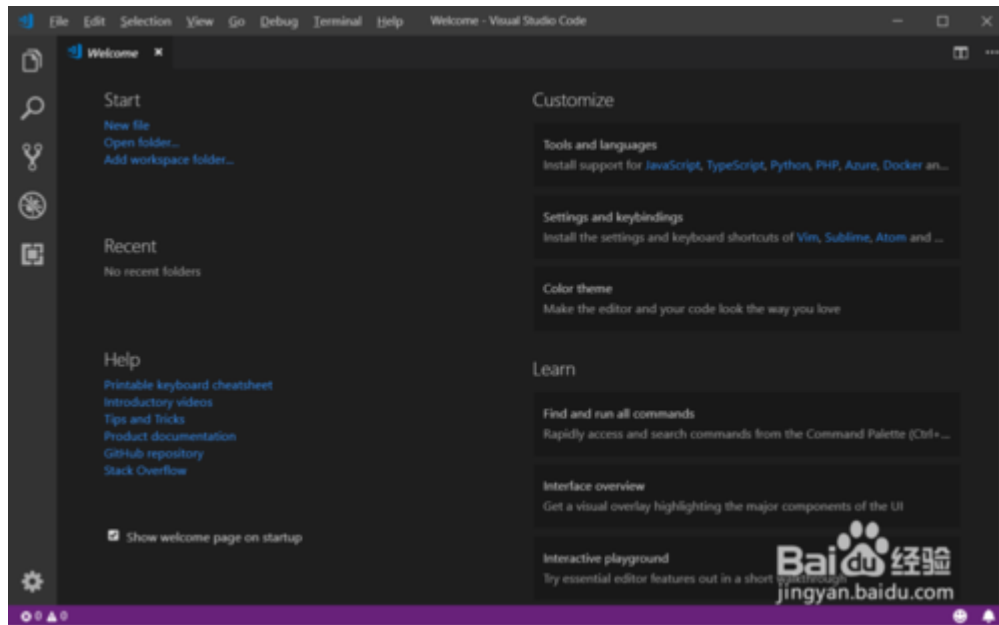
## 二、VS Code

### 1.windows

在Vscode官网下载安装包：<https://code.visualstudio.com/>

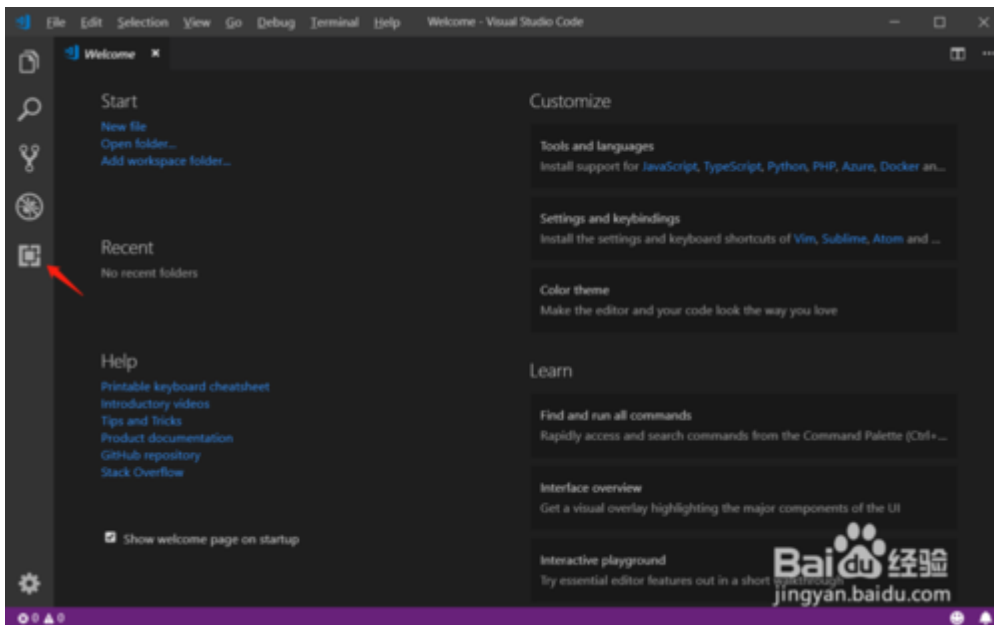


安装完成后打开Vscode，主界面如下：

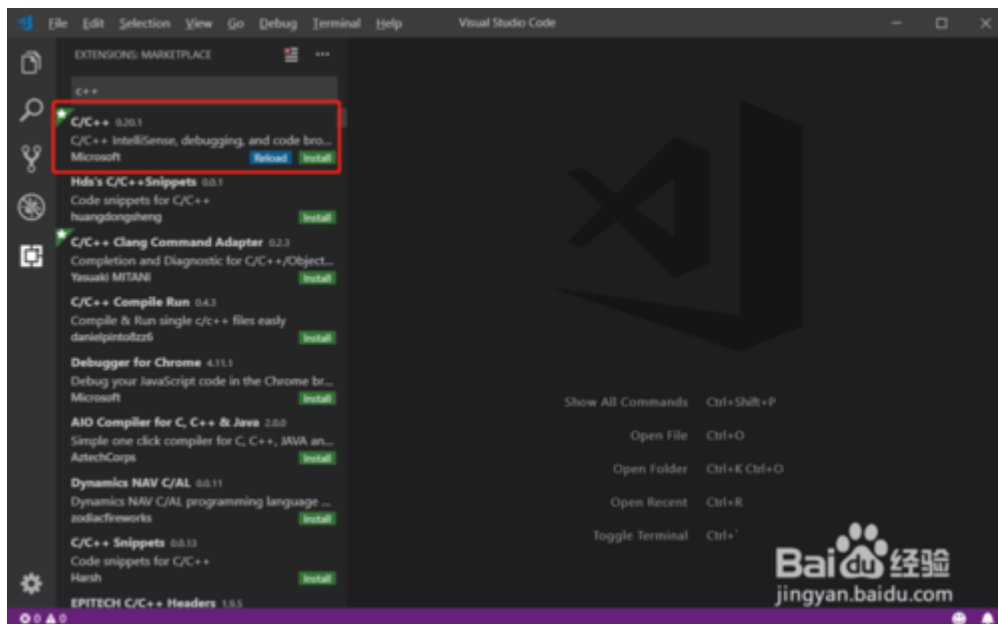


安装C/C++插件

点击左侧Extension按钮来选择安装插件：



搜索C++，点击Install

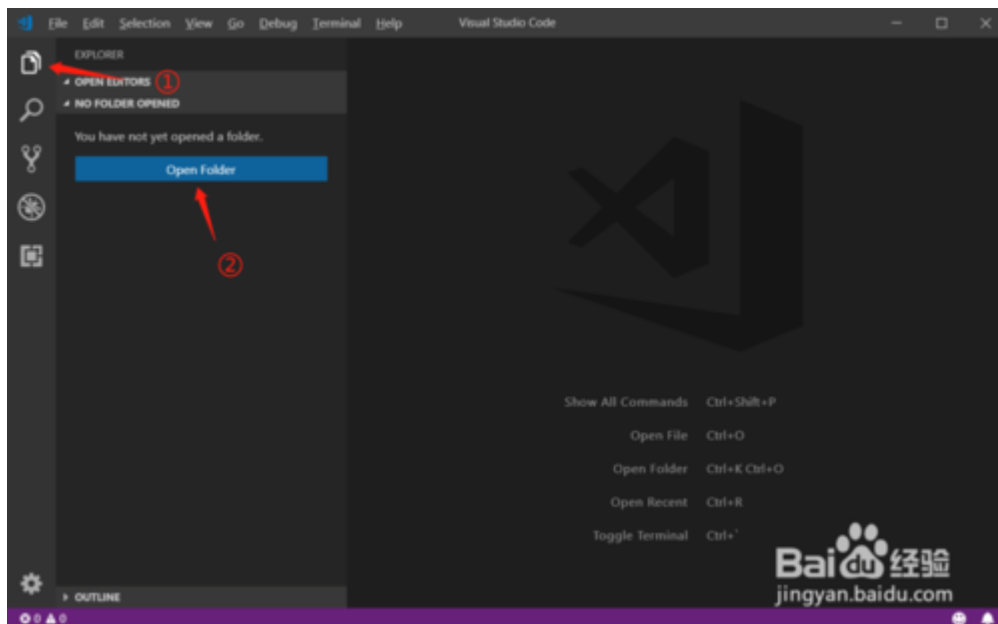


## 新建C/C++工程

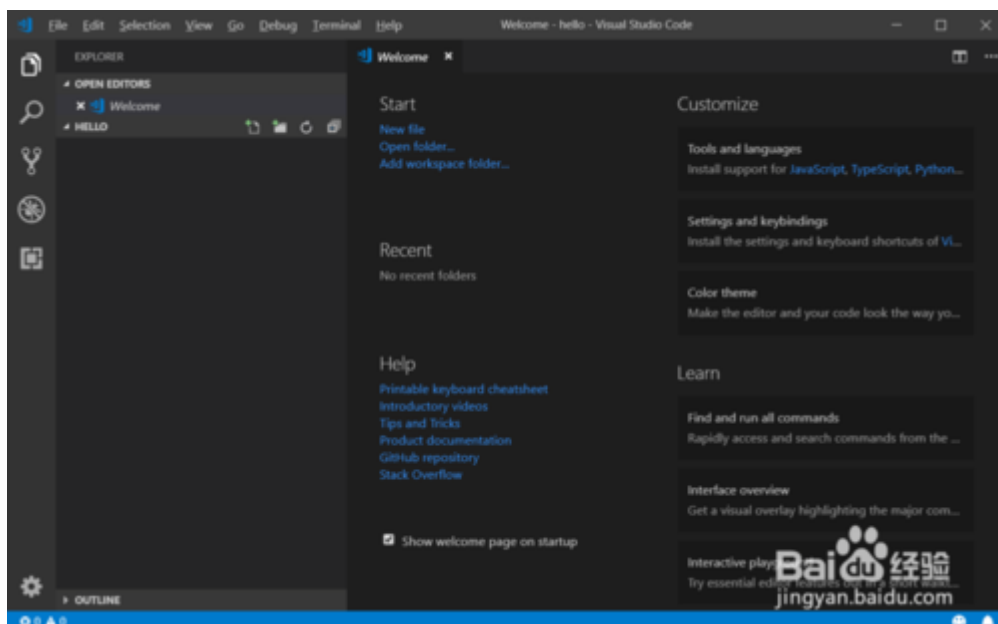
由于Vscode以文件夹的形式管理工程，所以需要新建一个文件夹来管理工程，这里建立一个名为hello的文件夹。



然后用Vscode打开该文件夹：

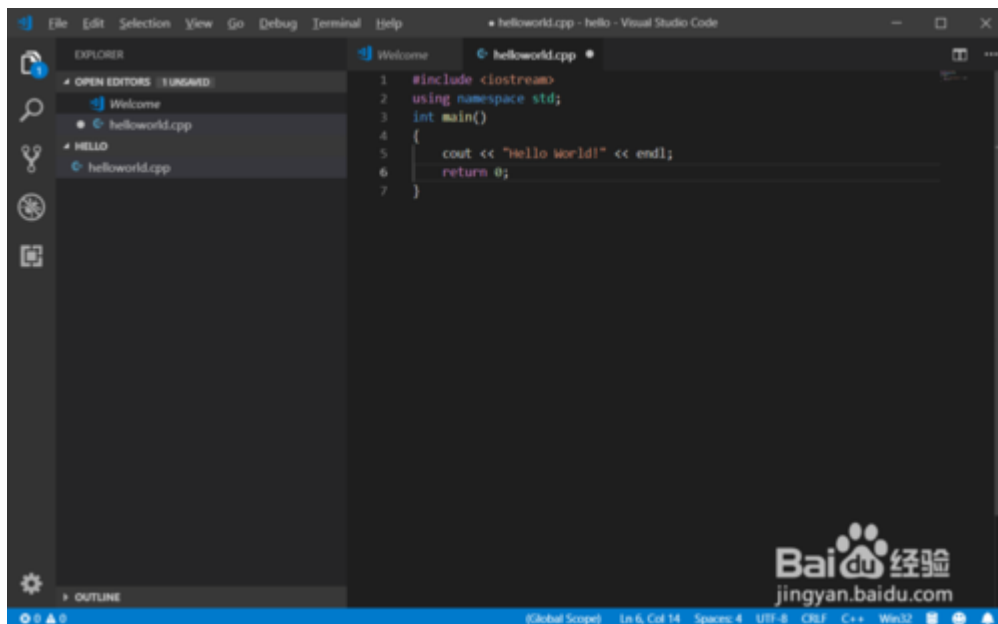


打开后得到一个空的工程：



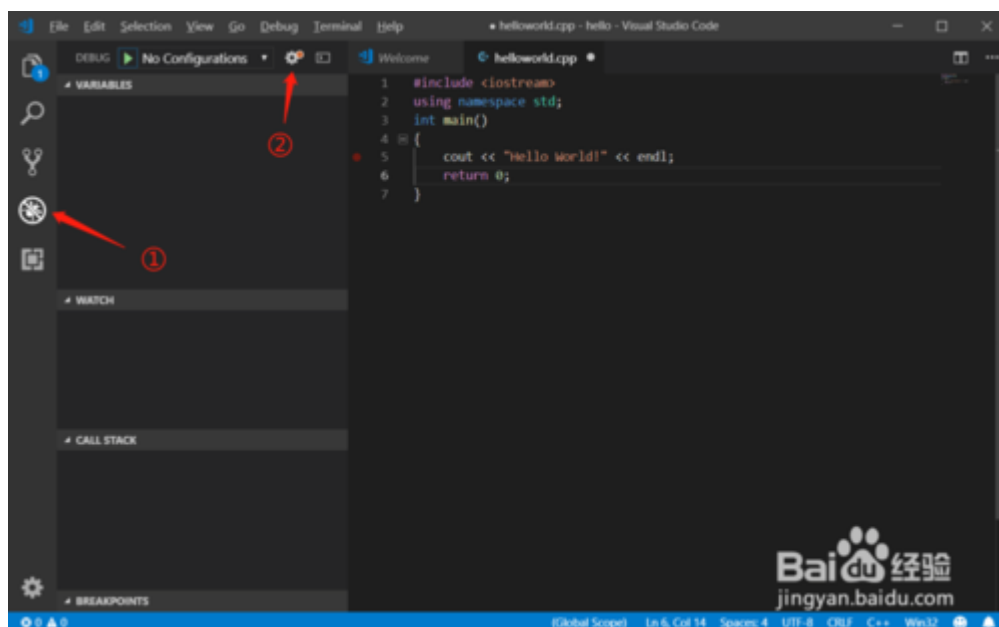
快捷键Ctrl+N新建一个C++文件，取名为helloworld.cpp，并输入一个简单的程序：



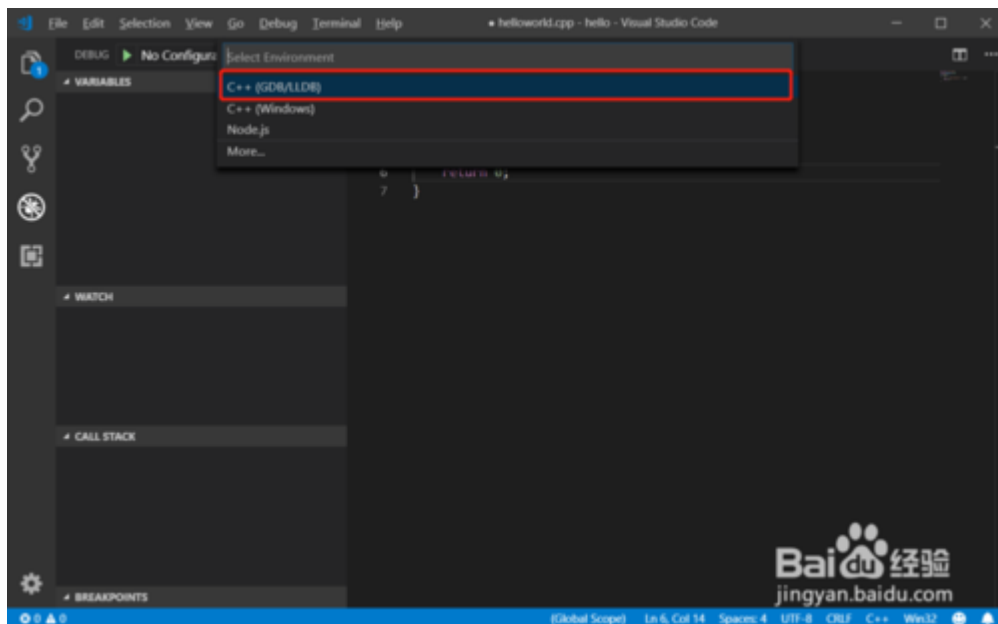


启动配置文件 (launch.json)

点击左边栏的Debug按钮，然后选择配置launch.json文件，操作如下：



选择C++(GDB\LLDB)，自动打开launch.json文件



## 2. Mac os

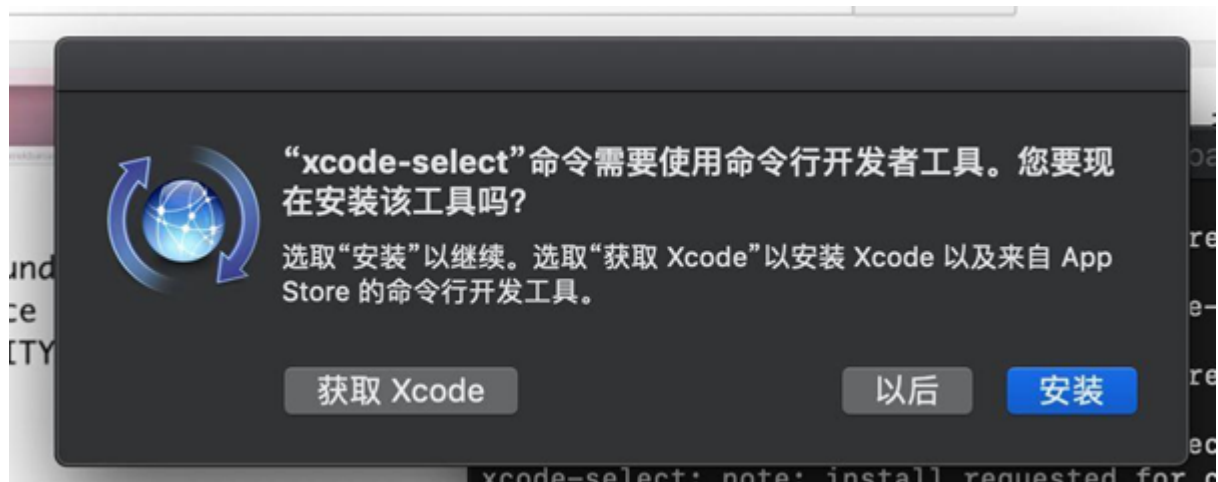
### 1) 安装 xcode

打开 App Store，搜索xcode，进行下载安装。



### 2) 执行命令：

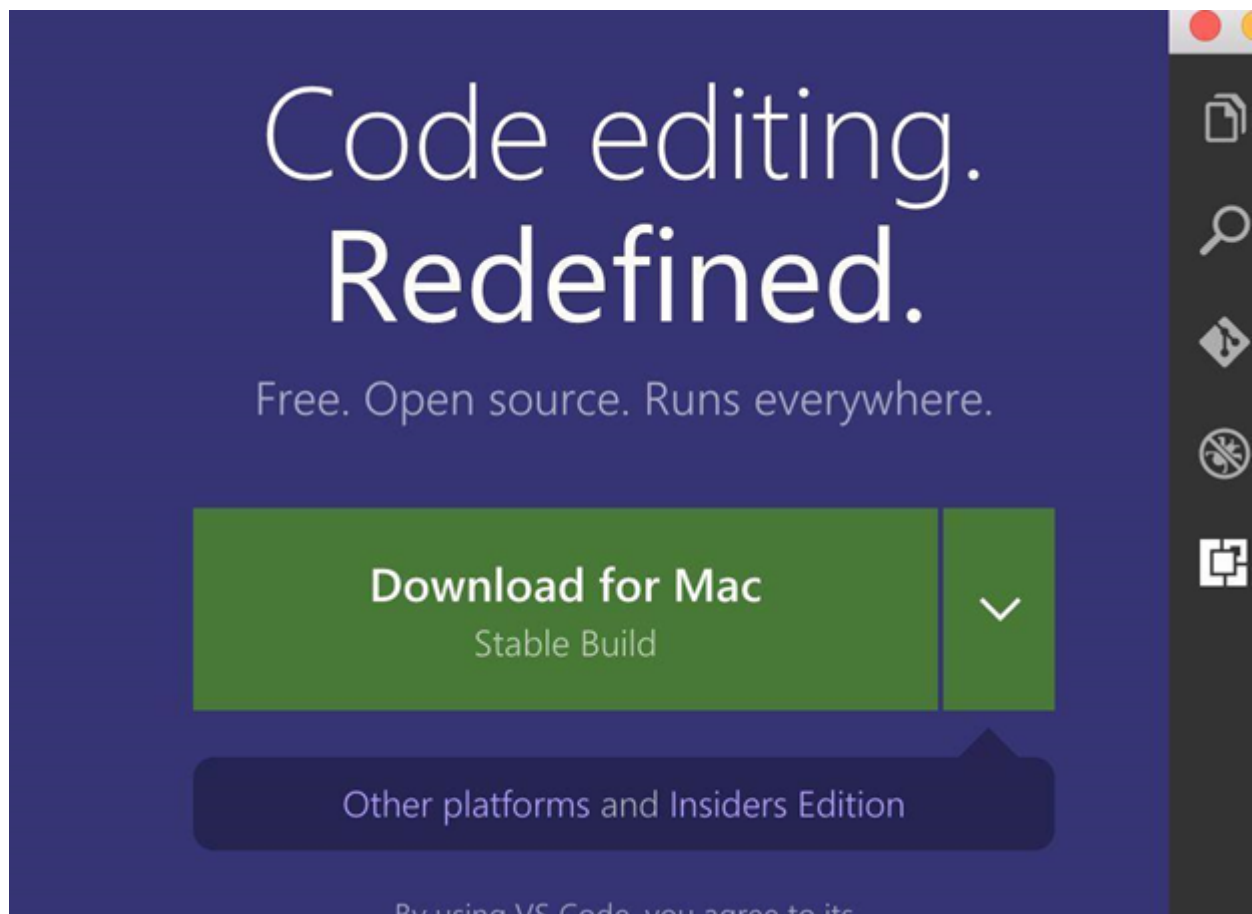
Xcode-select—install



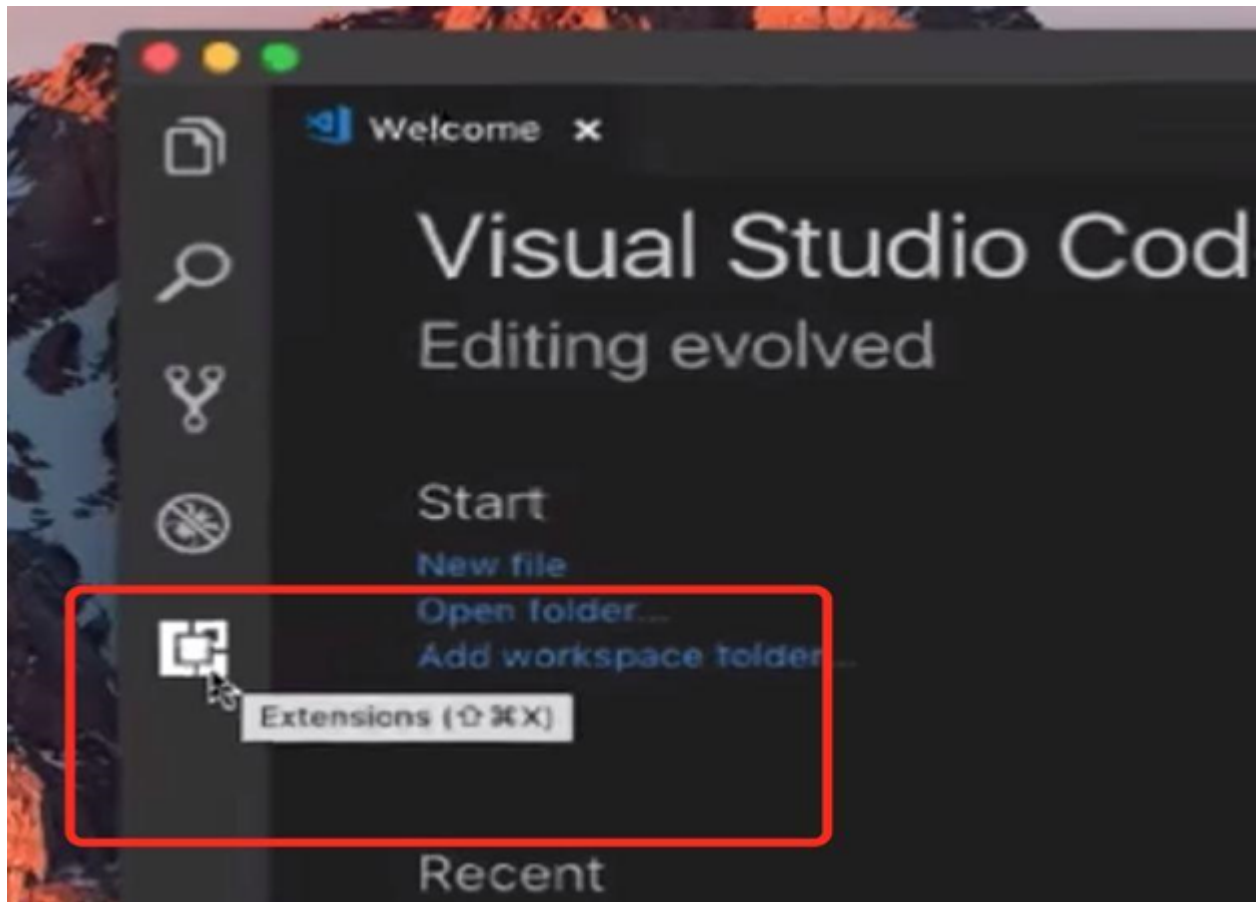
安装命令行工具。

3) 安装VS Code

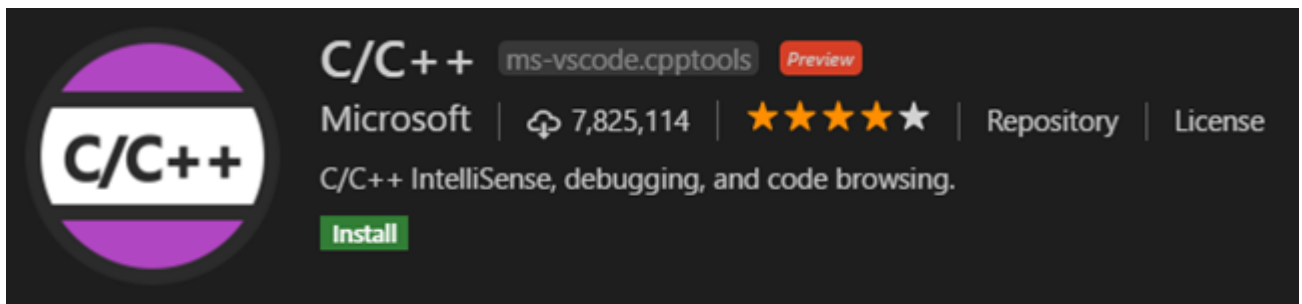
<https://code.visualstudio.com/>



4) 打开VS Code。打开左侧扩展栏



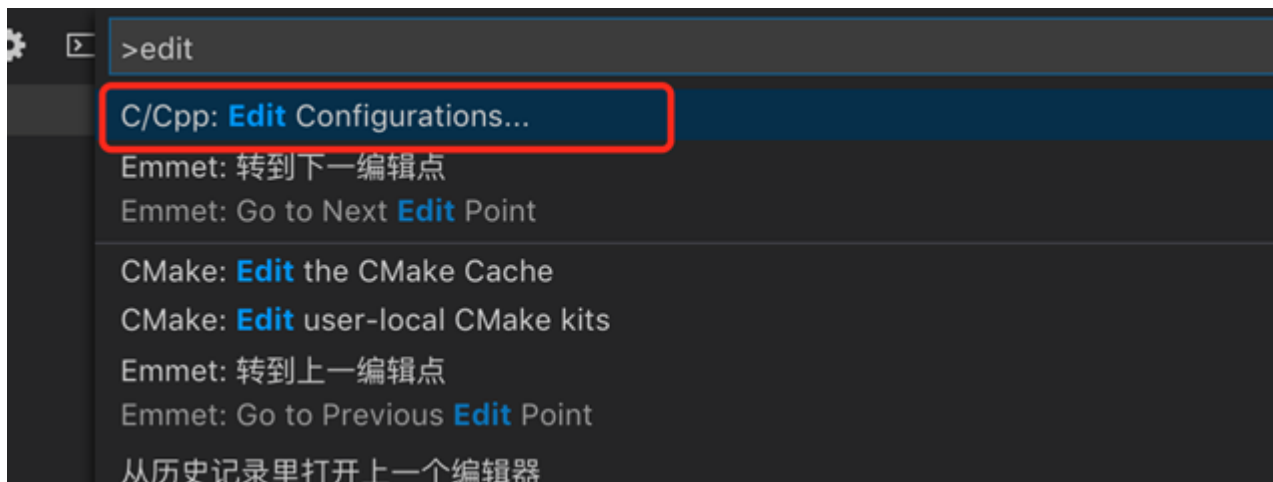
搜索 “c++”



安装该扩展。

5) 打开一个包含.cpp文件的文件夹（没有请自行创建）

“command+shift+p” 打开命令行工具窗口，输入或者选择 “  
Edit Configurations” 命令。

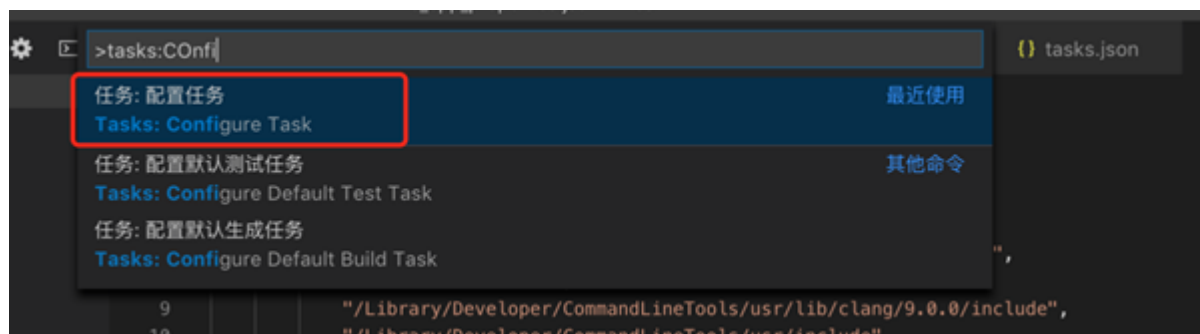


此时会在当前工作空间目录生成.vscode配置目录，同时在配置目录会生成一个c\_cpp\_properties.json文件。

配置include目录：

```
{
  "configurations": [
    {
      "name": "Mac",
      "includePath": [
        "${workspaceFolder}/**",
        "/Library/Developer/CommandLineTools/usr/include/c++/v1",
        "/usr/local/include",
        "/Library/Developer/CommandLineTools/usr/lib/clang/9.0.0/include",
        "/Library/Developer/CommandLineTools/usr/include",
        "/usr/include"
      ],
      "defines": [],
      "macFrameworkPath": [
        "/System/Library/Frameworks",
        "/Library/Frameworks"
      ],
      "compilerPath": "/usr/bin/clang",
      "cStandard": "c11",
      "cppStandard": "c++17",
      "intelliSenseMode": "clang-x64"
    }
  ],
  "version": 4
}
```

6) “command+shift+p” 打开命令行工具窗口，输入或者选择 “Tasks: Configure Task”



```

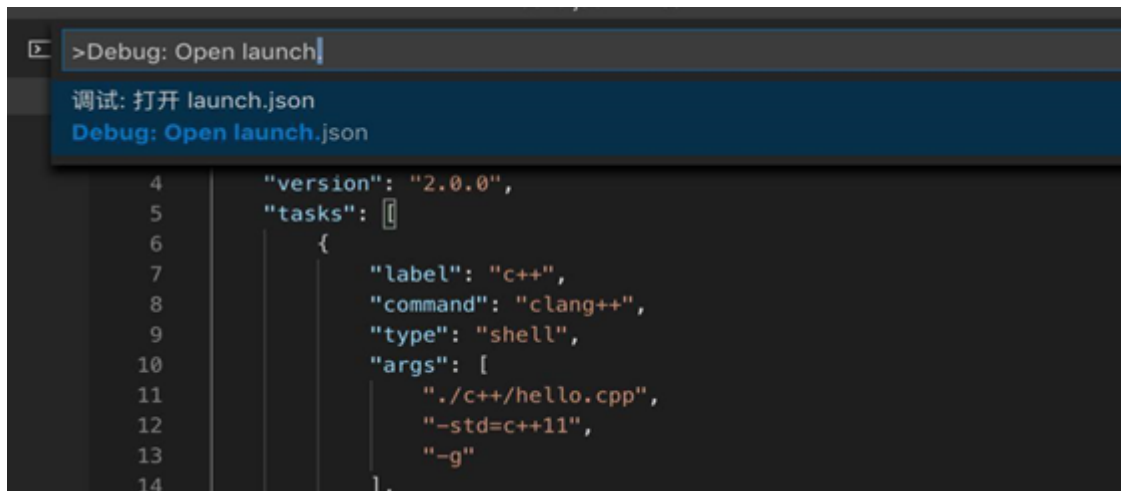
{
  // See https://go.microsoft.com/fwlink/?LinkId=733558
  // for the documentation about the tasks.json format
  "version": "2.0.0",
  "tasks": [
    {
      "label": "c++",
      "command": "clang++",
      "type": "shell",
      "args": [
        "./c++/hello.cpp",
        "-std=c++11",
        "-g"
      ],
      "presentation": {
        "echo": true,
        "reveal": "always",
        "focus": false,
        "panel": "shared"
      }
    }
  ]
}

```

## 7) 配置launch.json

“command+shift+p” 打开命令行工具窗口，输入或者

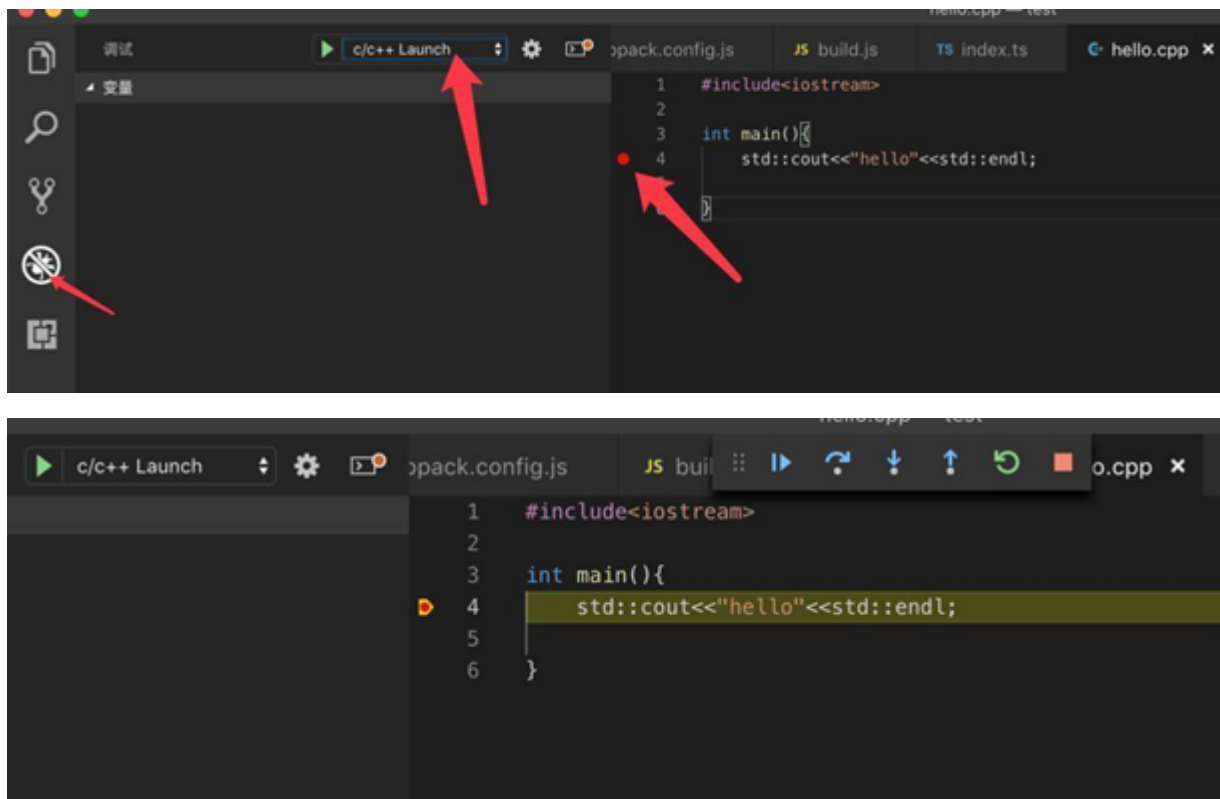
选择Debug: Open launch.json命令。



修改内容如下：

```
{
  // 使用 IntelliSense 了解相关属性。
  // 悬停以查看现有属性的描述。
  // 欲了解更多信息，请访问: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
  "configurations": [
    {
      "name": "c/c++ Launch",
      "type": "cppdbg",
      "request": "launch",
      "program": "${workspaceFolder}/a.out",
      "args": [],
      "stopAtEntry": false,
      "cwd": "${workspaceFolder}",
      "environment": [],
      "externalConsole": true,
      "MIMode": "lldb",
      "preLaunchTask": "c++"
    }
  ]
}
```

#### 8) 开启调试



中途可能会提醒控制终端，需要赋予权限，允许即可。

```
问题  输出  调试控制台  终端

> Executing task: clang++ ./c++/hello.cpp -std=c++11 -g <

终端将被任务重用，按任意键关闭。
█
```

最后如果终端有类似的提示，输入回车结束终端调用。

可能遇到的问题：

可能遇到 “xcrun: error: invalid active developer path (/Library/Developer/CommandLineTools), missing xcrun at: /Library/Developer/CommandLineTools/usr/bin/xcrun”

这样的报错，终端输入命令：

```
sudo xcode-select --switch /Applications/Xcode.app
```

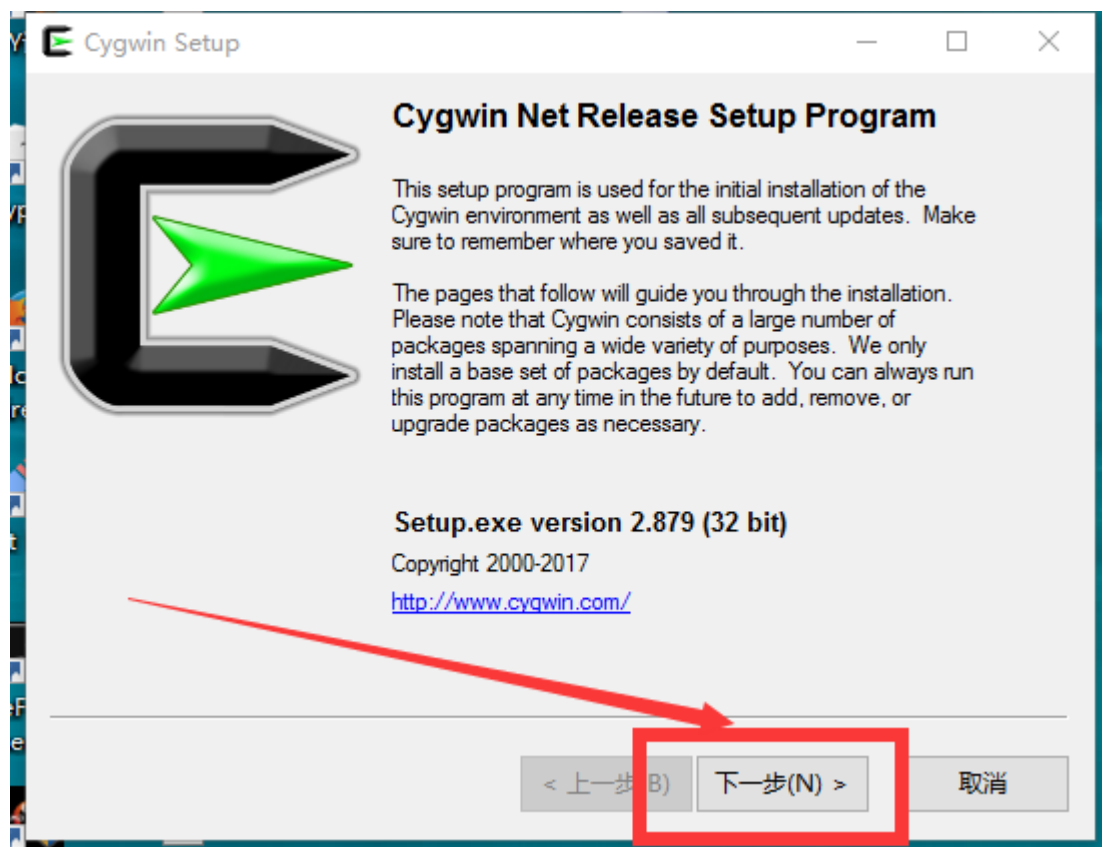
### 三、eclipse配置c语言开发环境

#### 1.下载Cygwin 并安装

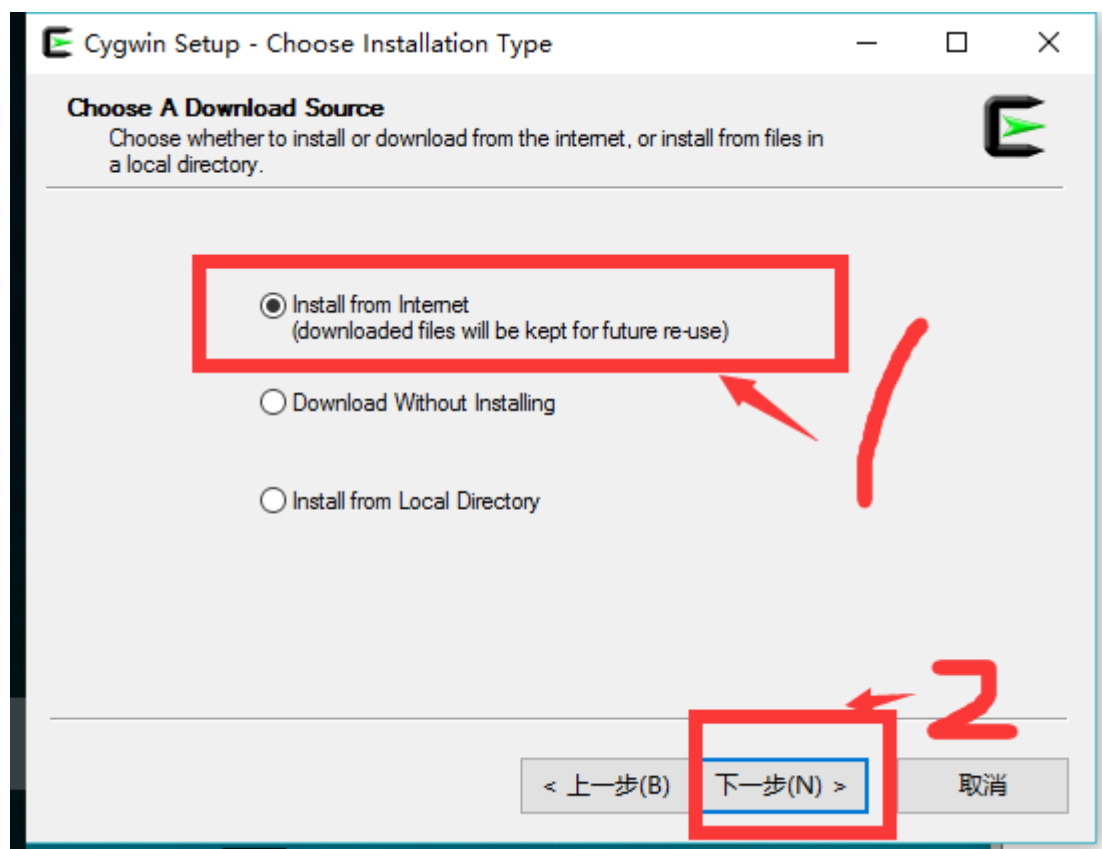
下载地址:<http://www.cygwin.cn/site/install/>

双击安装cygwin 单击下一步

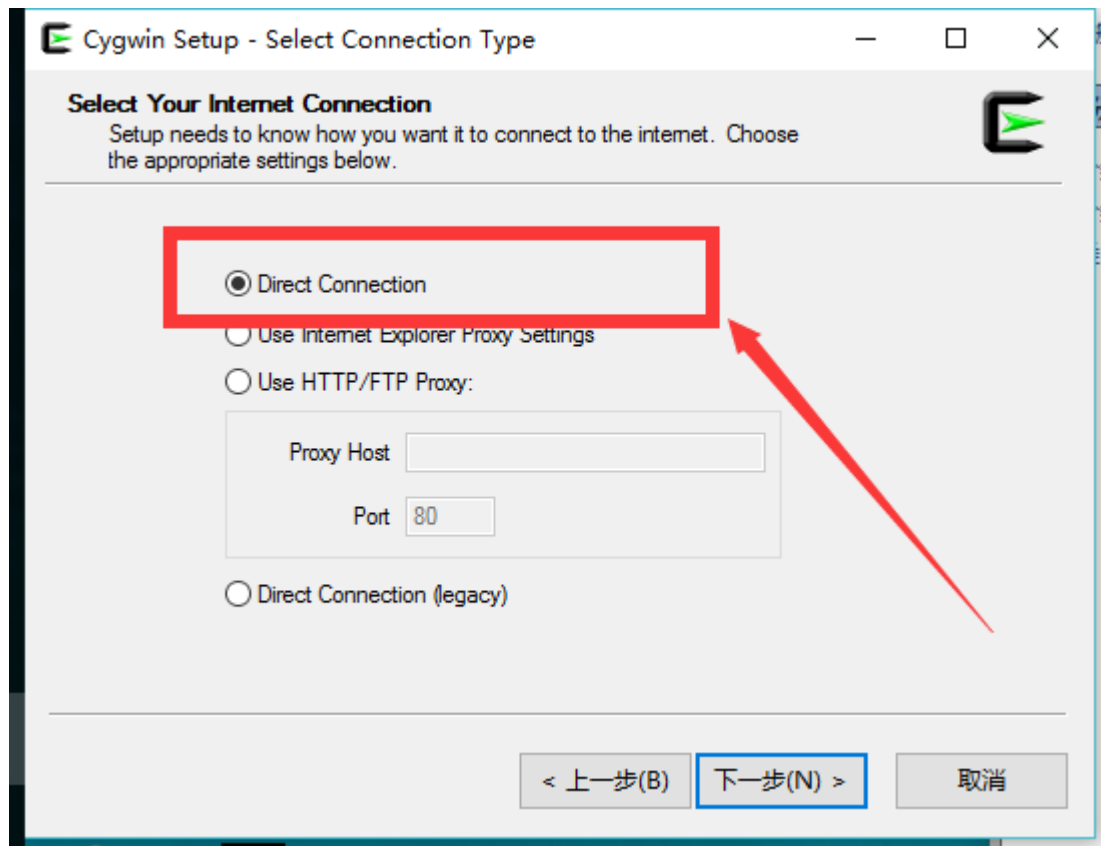




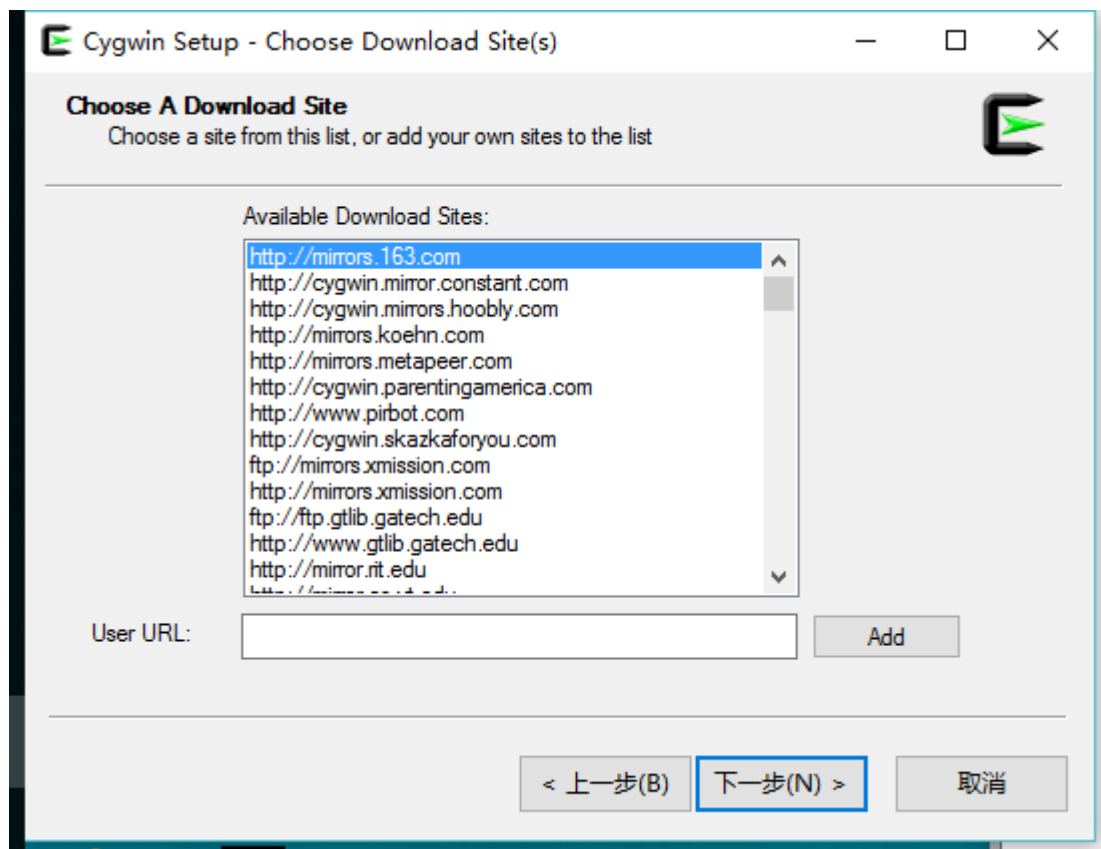
选择在线安装



选择安装路径然后点击下一步

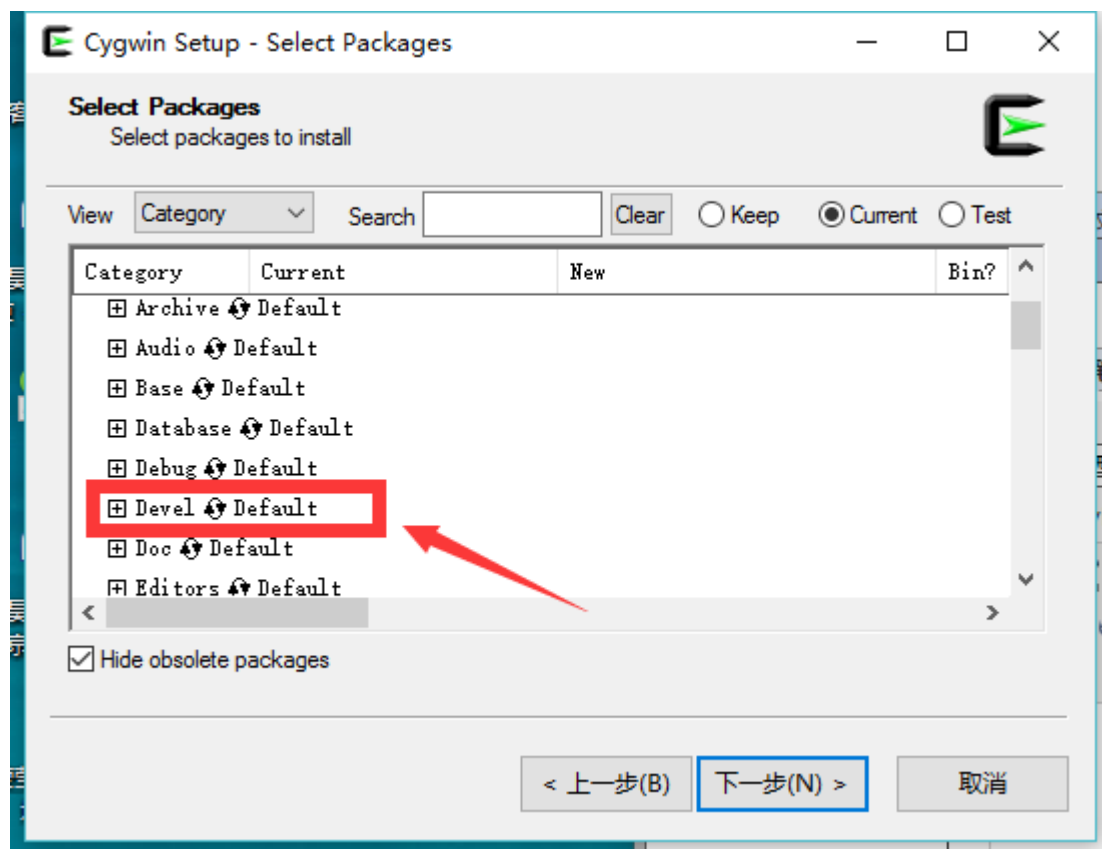


选择直接连接



选择站点这里我选择使用163地址:<http://mirrors.163.com/cygwin/>

特别注意:这里一定要选择安装g++和gcc 默认是不安装的



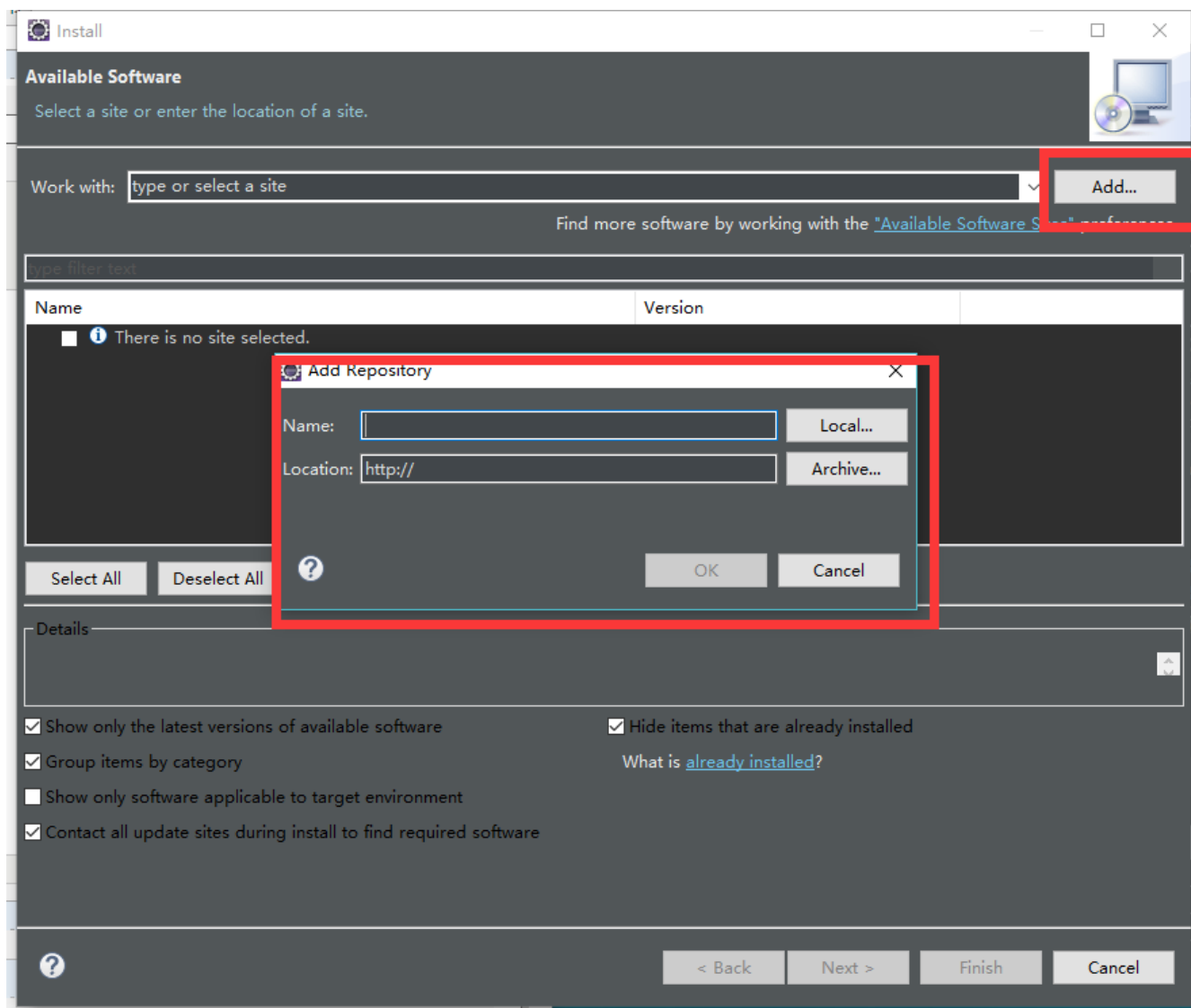
然后一直下一步就好了...

然后我们打开eclipse>Help>Install New Software

名字随便输入 我输入的是CDT

location输入 <http://download.eclipse.org/tools/cdt/releases/9.2>

到此eclipse就会自动帮你将CDT下载好了.....



重启eclipse以后点击 open Perspective 看到C/C++以后证明安装成功,这时就可以编写C程序了

