

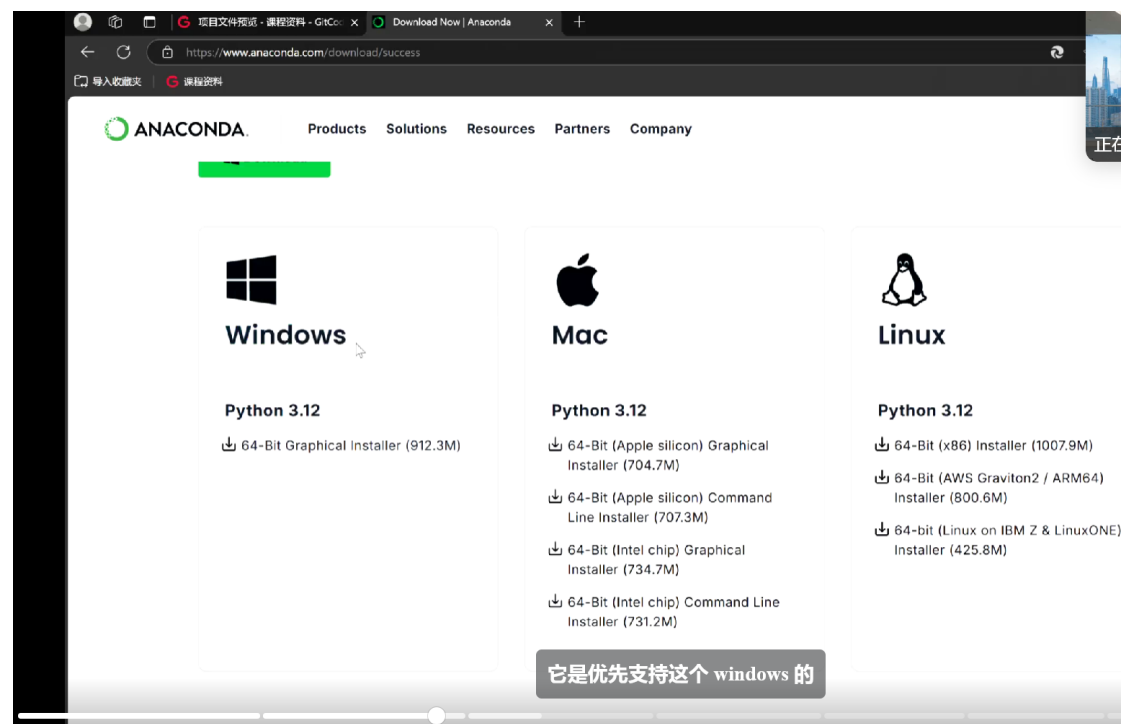
## 一、安装 VS code

### 1、课程网址



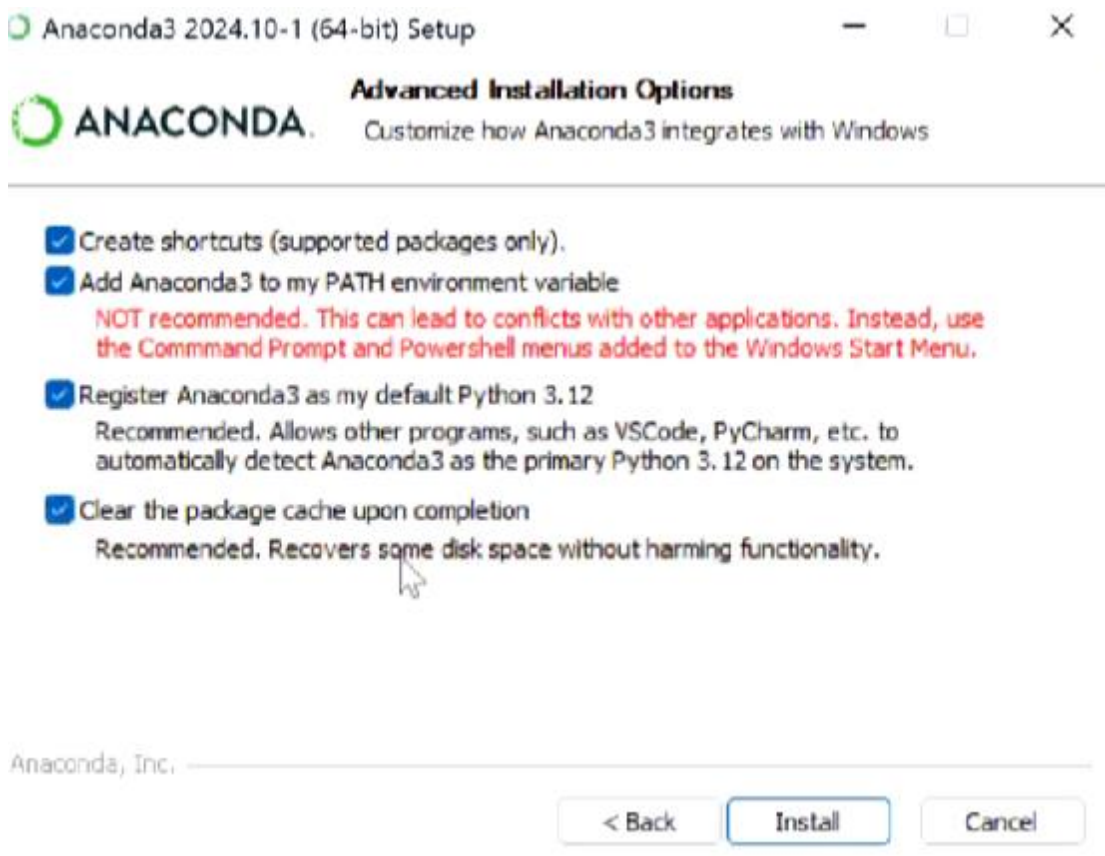
## 二、安装 Git

## 三、ANACONDA 的安装与使用



ANACONDA 是优先服务 Windows 用户的，真正的苹果用户和 Linux 客户不需要用 ANACONDA，主要是让没有计算机基础知识的小白快速上手 python 处理数据。ANACONDA 和 MINICONDA 的区别主要是 ANACONDA 安装了所有软件包, 而 MINICONDA

只安装核心软件包。



勾选 PATH 环境变量之后就可以直接在 git 终端里使用 ANACONDA。  
勾选最后一个选项结束之后清理缓存。

```
qiang@gqm3win CLANGARM64 ~  
$ conda  
bash: conda: command not found  
  
qiang@gqm3win CLANGARM64 ~  
$ which conda  
/c/Users/qiang/anaconda3/Scripts/conda  
  
qiang@gqm3win CLANGARM64 ~  
$ echo $PATH  
/c/Users/qiang/bin:/clangarm64/bin:/usr/local/bin:/usr/bin:/bin:/mingw64/bin:/usr/bin:/c/Users/qiang/bin:/c/Program Files/Parallels/Parallels Tools/Applications:/c/Windows/system32:/c/Windows:/c/Windows/System32/Wbem:/c/Windows/System32/WindowsPowerShell/v1.0:/c/Windows/System32/OpenSSH:/cmd:/mingw64/bin:/usr/bin:/c/Users/qiang/anaconda3:/c/Users/qiang/anaconda3/Library/mingw-w64/bin:/c/Users/qiang/anaconda3/Library/usr/bin:/c/Users/qiang/anaconda3/Library/bin:/c/Users/qiang/anaconda3/Scripts:/c/Users/qiang/AppData/Local/Microsoft/WindowsApps:/c/Users/qiang/AppData/Local/Programs/Microsoft VS Code/bin:/usr/bin/vendor_perl:/usr/bin/core_perl  
  
qiang@gqm3win CLANGARM64 ~  
$
```

PATH 就是找文件路径的优先级，用:分隔

```
CLANGARM64:/c/Users/qiang x + v
qiang@gqm3win CLANGARM64 ~
$ python
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

在 git 终端中使用 python, 要用 3.12 的版本, 因为我们使用的 python 软件包未必支持 3.13python 解释器

```
CLANGARM64:/c/Users/qiang x + v
qiang@gqm3win CLANGARM64 ~
$ python
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()

qiang@gqm3win CLANGARM64 ~
$ ipython
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.27.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print('hello')
hello

In [2]:
Do you really want to exit ([y]/n)? y

qiang@gqm3win CLANGARM64 ~
$ python
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello')
hello
>>>
```

ipython 和 python 在功能上的一些区别, ipython 在功能上有很多增强

```
CLANGARM64/c/Users/qiang x + v
ntuser.dat.LOG1
ntuser.dat.LOG2
NTUSER.DAT{52f8dc78-cdd7-11ec-8680-00224848b4d3}.TM.blf
NTUSER.DAT{52f8dc78-cdd7-11ec-8680-00224848b4d3}.TMContainer000000000000000001.regtra
ns-ms
NTUSER.DAT{52f8dc78-cdd7-11ec-8680-00224848b4d3}.TMContainer000000000000000002.regtra
ns-ms
ntuser.ini
Pictures/
PrintHood@
Recent@
repo/
'Saved Games'/
Searches/
SendTo@
'Start Menu'@
Templates@
Videos/
'WPS Cloud Files'/
WPSDrive/

qiang@gqm3win CLANGARM64 ~
$ cd repo

qiang@gqm3win CLANGARM64 ~/repo
$ code .

qiang@gqm3win CLANGARM64 ~/repo
$
```

在 git 终端输入 code 就是调用 vs code，在这里可以直接用 python

四、访问 Gitcode，注册账户，保持登录

五、生成自己的 SSH 密钥，将公钥添加到自己的 GitCode 安全设置里

#### 1. 打开终端

- 在 Linux/macOS 上，打开系统自带的终端
- 在 Windows 上，可以使用 Cmd、Power Shell 或 Git Bash

#### 2. 输入生成密钥的命令

运行以下命令，将 `your_email@example.com` 替换为您的邮箱：

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

- `-t ed25519`：指定密钥类型为 ED25519。
- `-C "your_email@example.com"`：添加注释，通常使用您的邮箱地址，便于识别密钥。

#### 3. 选择密钥保存位置

命令执行后，您将看到以下提示：

```
Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/user/.ssh/id_ed25519):
```

- 按 **Enter** 键接受默认位置 (`~/.ssh/id_ed25519`)。
- 如果需要自定义保存路径，可以输入新的路径，例如：`~/.ssh/my_custom_key`。



```
edy@edydeMacBook-Pro ~ % ssh-keygen -t ed25519 -C "your_email@example.com" 替换为您的邮箱地址
Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/edy/.ssh/id_ed25519):
/Users/edy/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase): 重复输入两次安全密钥 (可按回车跳过)
Enter same passphrase again:
Your identification has been saved in /Users/edy/.ssh/id_ed25519
Your public key has been saved in /Users/edy/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:gYmpnMkP7dSuoc18rN/dg0HQ9GZp9lgLhzFnG7ZEYx4 your_email@example.com
The key's randomart image is:
+--[ED25519 256]--+
|      o. o.E      |
|      o.O.. % *   |
|      o O.. X *   |
| o = . . * * .   |
| B o ..S . o     |
|      = . .       |
|      .+ . o      |
| o oo+ ....      |
| *+o . . . .     |
+-----[SHA256]-----+
edy@edydeMacBook-Pro ~ % cat .ssh/id_ed25519.pub 复制此 ssh 公钥
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPAtuN6D3rxXuSCs79BzbF6qfVpgiPwTa+bubJ1QygFz your_email@example.com
edy@edydeMacBook-Pro ~ %
```

## 添加 SSH 公钥到 GITCODE

### 1. 复制 SSH 公钥内容

从保存 SSH 密钥的文件中复制你的 SSH 密钥的公钥，以下命令可以将 ED25519 的信息保存到指定操作系统的剪贴板中：

macOS

```
pbcopy < ~/.ssh/id_ed25519.pub
```

Linux (需要 xclip 软件包)

```
xclip -sel clip < ~/.ssh/id_ed25519.pub
```

Windows 上的 Git Bash

```
cat ~/.ssh/id_ed25519.pub | clip
```

如果你使用的是 RSA 密钥，相应地替换即可

### 2. 登录 GitCode，进入「个人设置」->「安全设置」->「SSH 公钥」

### 3. 点击「+ SSH 公钥」

4. 在“公钥名称”一栏中，为公钥添加一个描述性名称

5. 将复制的公钥内容粘贴到“公钥”文本框中

6. 点击「新建」完成操作

### 新建 SSH 公钥



The screenshot shows a web form titled "新建 SSH 公钥" (New SSH Key). It has two main input fields and two buttons at the bottom. Red arrows and text annotations are overlaid on the form to guide the user:

- 1. 输入自定义公钥名称** (Enter custom key name): An arrow points to the "公钥名称" (Key Name) field, which contains the text "SSH公钥3".
- 2. 输入 SSH 公钥** (Enter SSH key): An arrow points to the "公钥" (Key) field, which contains a long alphanumeric string: "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPAtuN6D3rxuSCs79BzbF6qfVpgiPwTa+bubJ1QygFz your\_email@example.com". Above this field is a link that says "了解如何生成 SSH Key." (Learn how to generate an SSH key.).
- 3. 点击保存** (Click save): An arrow points to the "新建" (New) button at the bottom left.

The "取消" (Cancel) button is located to the right of the "新建" button.

在终端里，右击是复制粘贴，CTRL+c 是终止运行