CSCI2040 Introduction to Python Tutorial on Lab Usage, Python Runtimes, and IDEs

Speaker: Robin Bin Luo

CUHK CSE

香港中文大學計算機科學與工程學系
Department of Computer Science and Engineering
The Chinese University of Hong Kong

Your Tutor

- **Mr. LUO Bin, Robin**
- Email: 1155204978@link.cuhk.edu.hk
- Office: SHB 120
- Consultation hours: Mon 2:15-3:15pm

Outline

- Anaconda Installation (on Your Computer)
- Virtual Environments
- VSCode IDE
- PyCharm IDE
- Package Management
- A Simple Exercise
- Computer Lab Usage
- Appendix:
  - Commands for virtual environment management
  - Commands for package management

Anaconda Installation

On your own laptop computer

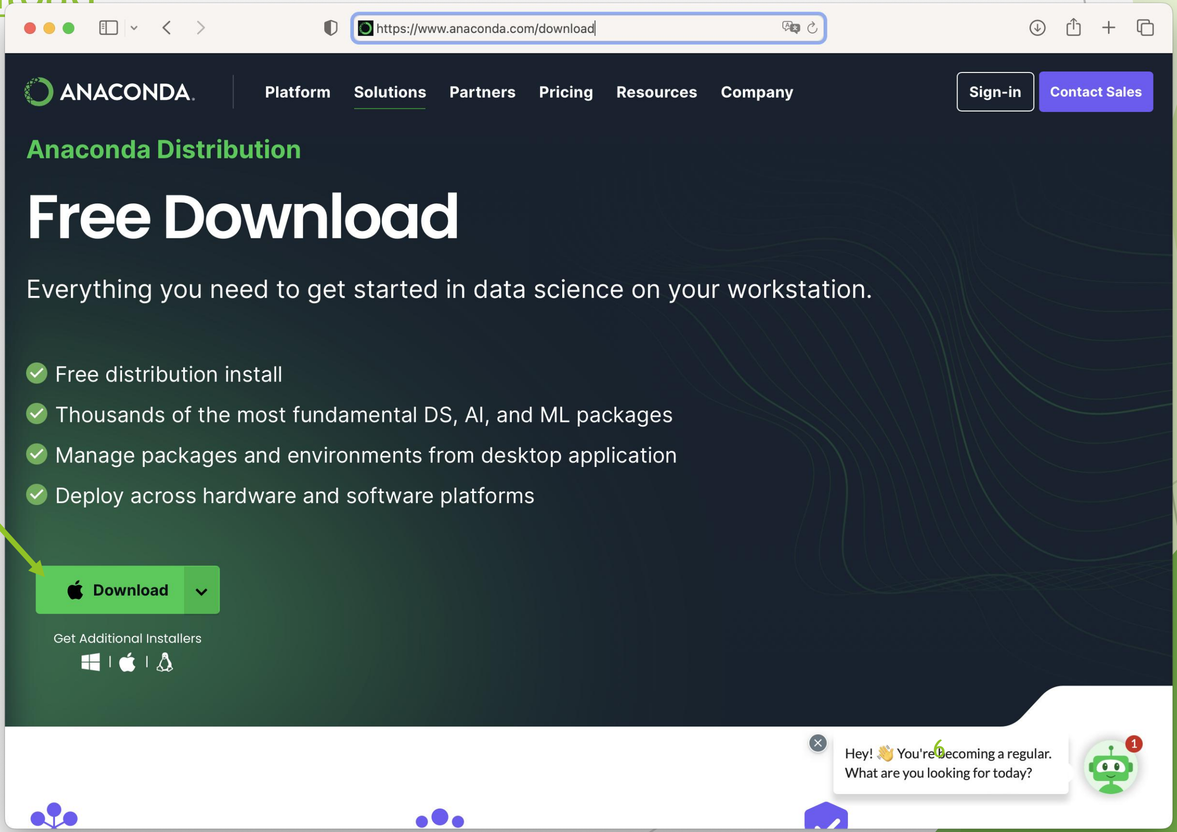
Choices of Python Installation

Besides using our lab PCs, you are highly encouraged to install a Python runtime on your own PC.

- Recap the Introduction lecture:
  1. Official Python runtime
     - Includes a minimal IDE called "IDLE" and a package manager called "pip"
     - But no built-in support of virtual environments
  2. Anaconda (Individual Edition, free)
     - Preinstalls hundreds of packages, a package/environment manager called "conda", and a graphical user interface called Anaconda-Navigator for installing and launching applications.
     - If disk space is not a problem, this is the best option.
  3. Miniconda
     - A minimal free version of Anaconda, which includes a package/environment manager called "conda" and a minimal set of Python packages

Install Anaconda

- Use your browser to open the website: https://www.anaconda.com/download
- Click the “Download” button.



Anaconda Distribution

Free Download

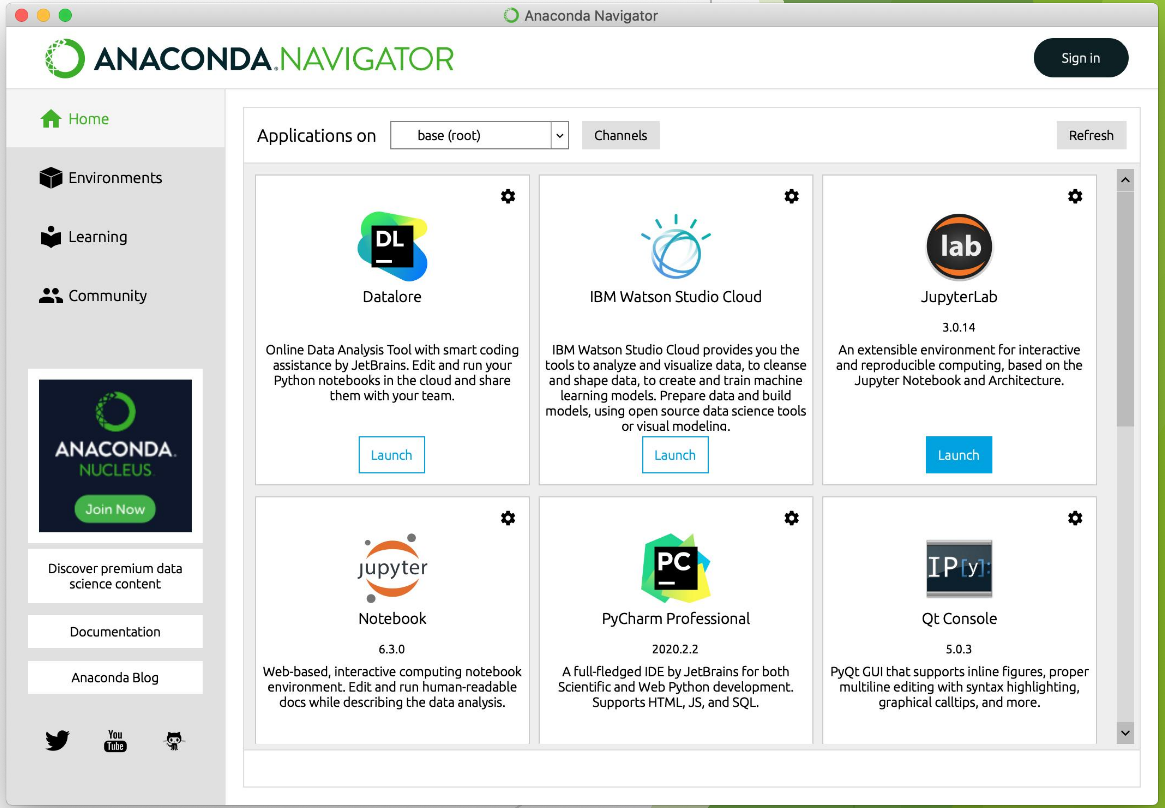
Everything you need to get started in data science on your workstation.

- Free distribution install
- Thousands of the most fundamental DS, AI, and ML packages
- Manage packages and environments from desktop application
- Deploy across hardware and software platforms

Hey! 👋 You're becoming a regular.
What are you looking for today?

Anaconda Navigator

- Anaconda provides a GUI program called Navigator, which brings convenience to users for launching an IDE like PyCharm, VSCode, or other Python-related tools like Jupyter Lab or Notebook, etc.
- Click "Launch" or "Install" (if missing) on each application you need to run.
- It can also be used to manage virtual Python environments and packages.



Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

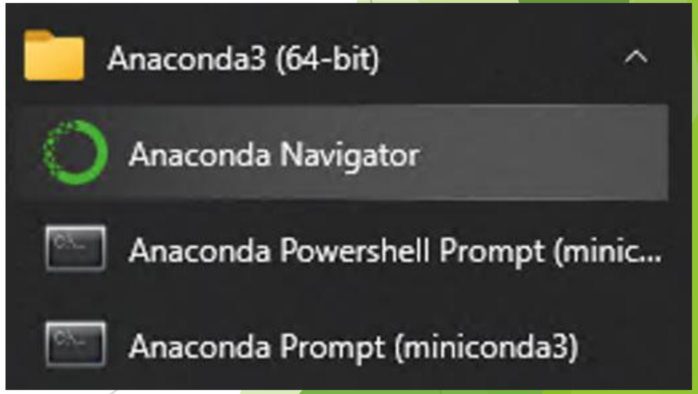
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Anaconda Navigator

- If you've installed Anaconda, you will see "Anaconda Navigator" in the Windows Start Menu or macOS Launchpad.
- If you've installed Miniconda, it is missing but you can install it in an Anaconda Prompt (base environment) by running the command:

```
(base)...> conda install anaconda-navigator
```

- Afterwards, you will see the menu item.


Anaconda3 (64-bit)
Anaconda Navigator
Anaconda Powershell Prompt (minic...
Anaconda Prompt (miniconda3)

Virtual Environments

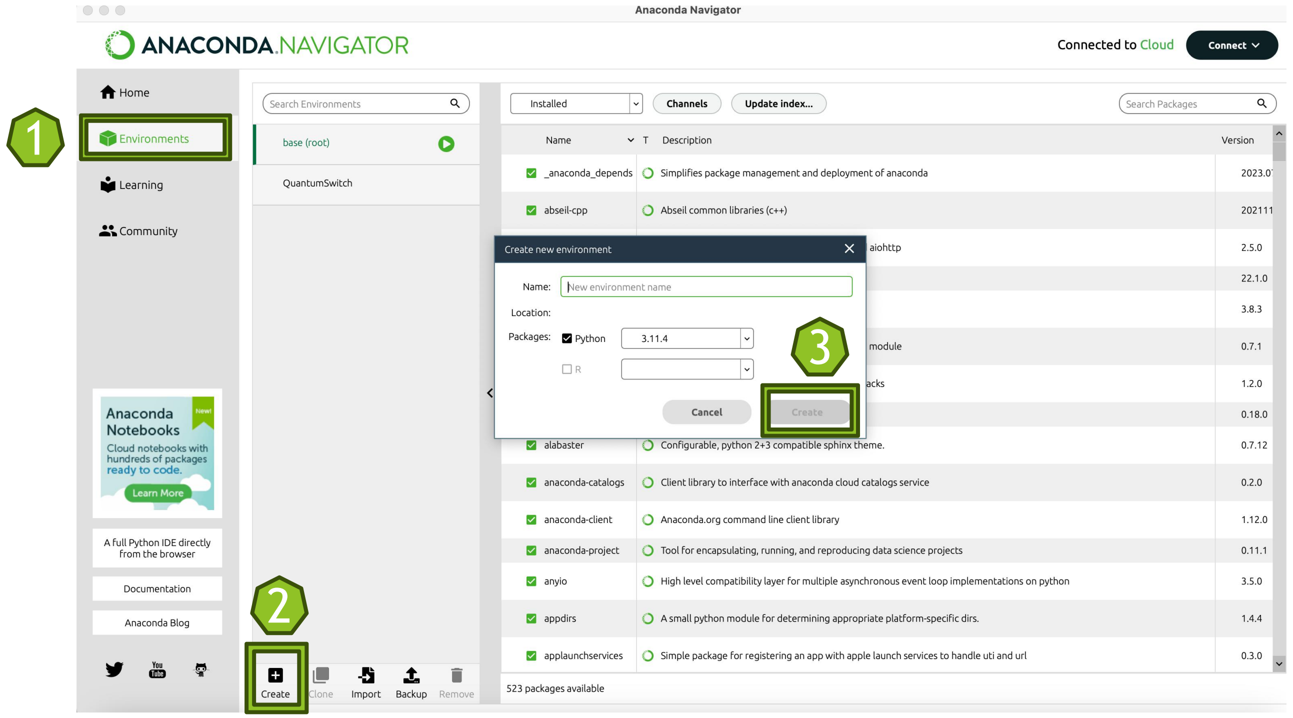
For isolating Python projects

Python Virtual Environments

- A Python virtual environment is a self-contained directory that encapsulates a specific Python interpreter along with its associated libraries, packages, and dependencies.
- It allows you to create **isolated environments** for different Python projects, each with its own set of dependencies, without interference or conflicts between them.
- For example, one project may need an old Python version like 3.6 for running legacy code while another project may require a newer version 3.11 and some 3rd party libraries such as NumPy.

Create a Virtual Environment

- Open the Anaconda Navigator app Anaconda-Navigator after installing Anaconda.
- Click the “Environment”.
- Click “Create” to create a virtual environment named by yourself.



Search Environments

base (root)

QuantumSwitch

Installed | Channels | Update index... | Search Packages

Name	T	Description	Version
✓ _anaconda_depends		Simplifies package management and deployment of anaconda	2023.0
✓ abseil-cpp		Abseil common libraries (c++)	202111
		aiohttp	2.5.0
			22.1.0
			3.8.3
		module	0.7.1
		acks	1.2.0
			0.18.0
✓ alabaster		Configurable, python 2+3 compatible sphinx theme.	0.7.12
✓ anaconda-catalogs		Client library to interface with anaconda cloud catalogs service	0.2.0
✓ anaconda-client		Anaconda.org command line client library	1.12.0
✓ anaconda-project		Tool for encapsulating, running, and reproducing data science projects	0.11.1
✓ anyio		High level compatibility layer for multiple asynchronous event loop implementations on python	3.5.0
✓ appdirs		A small python module for determining appropriate platform-specific dirs.	1.4.4
✓ applaunchservices		Simple package for registering an app with apple launch services to handle uti and url	0.3.0

523 packages available

Create new environment

Name: New environment name

Location:

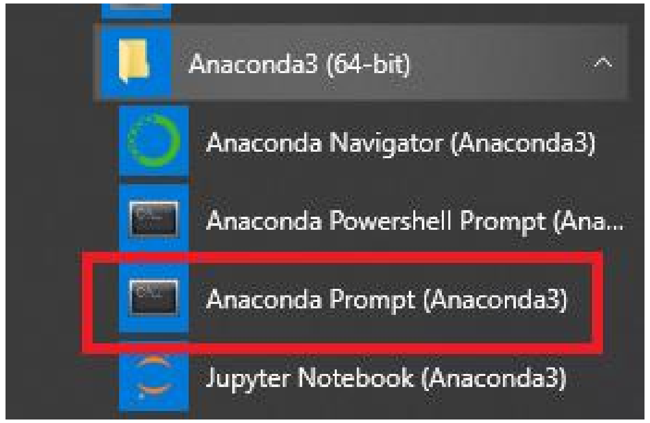
Packages: [✓] Python 3.11.4

☐ R

Cancel | Create

Create a Virtual Environment via the Command Line

- Refer to the Appendix for how to use a terminal to type commands for managing a virtual environment.

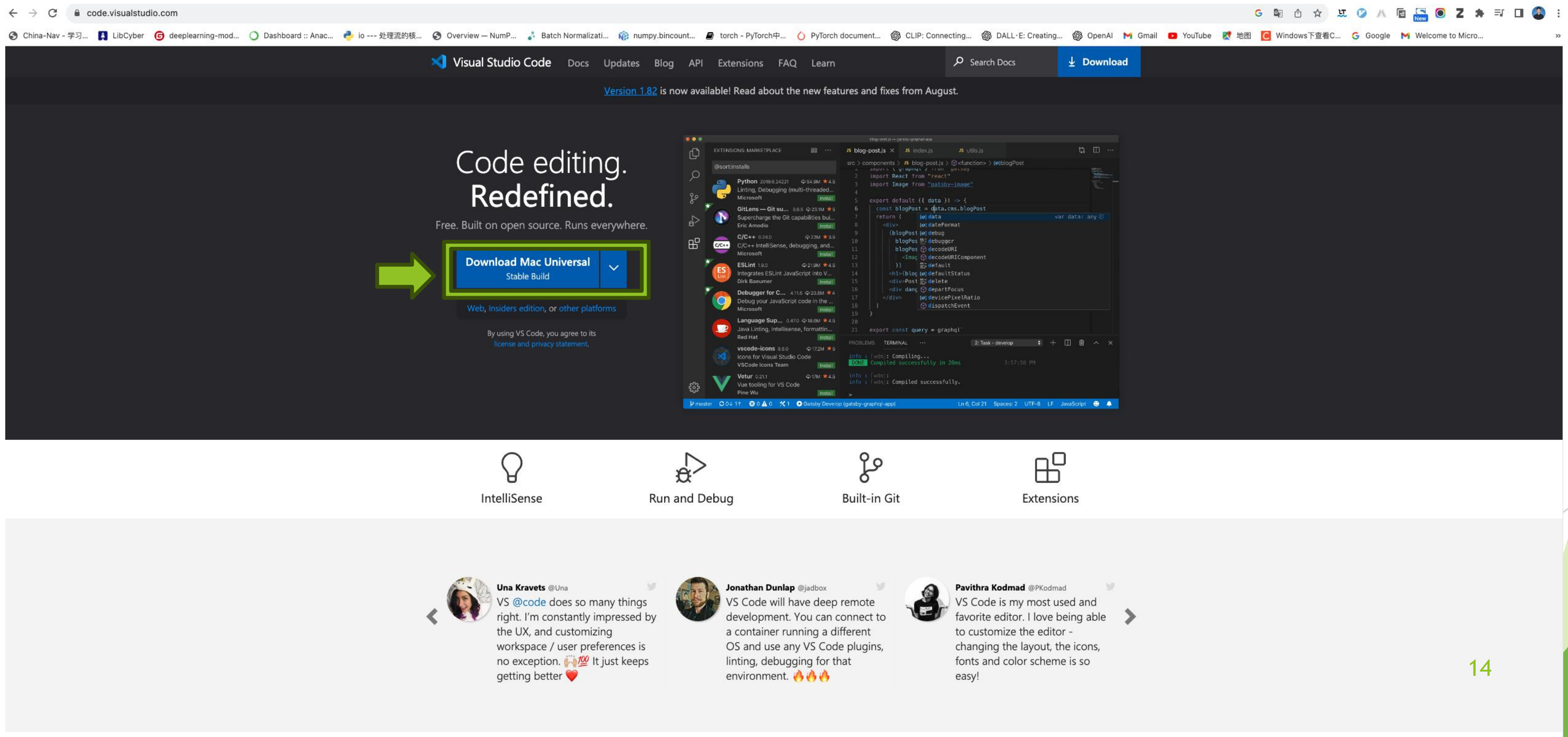

Anaconda3 (64-bit)
Anaconda Navigator (Anaconda3)
Anaconda Powershell Prompt (Ana...
Anaconda Prompt (Anaconda3)
Jupyter Notebook (Anaconda3)

Visual Studio Code IDE

From Microsoft

Use VSCode to Write Code

- Assume that you have installed VSCode in your computer. If not, download it from this website: https://code.visualstudio.com and run the installer.



Version 1.82 is now available! Read about the new features and fixes from August.

Code editing.
Redefined.

Free. Built on open source. Runs everywhere.

Web, Insiders edition, or other platforms

By using VS Code, you agree to its license and privacy statement.

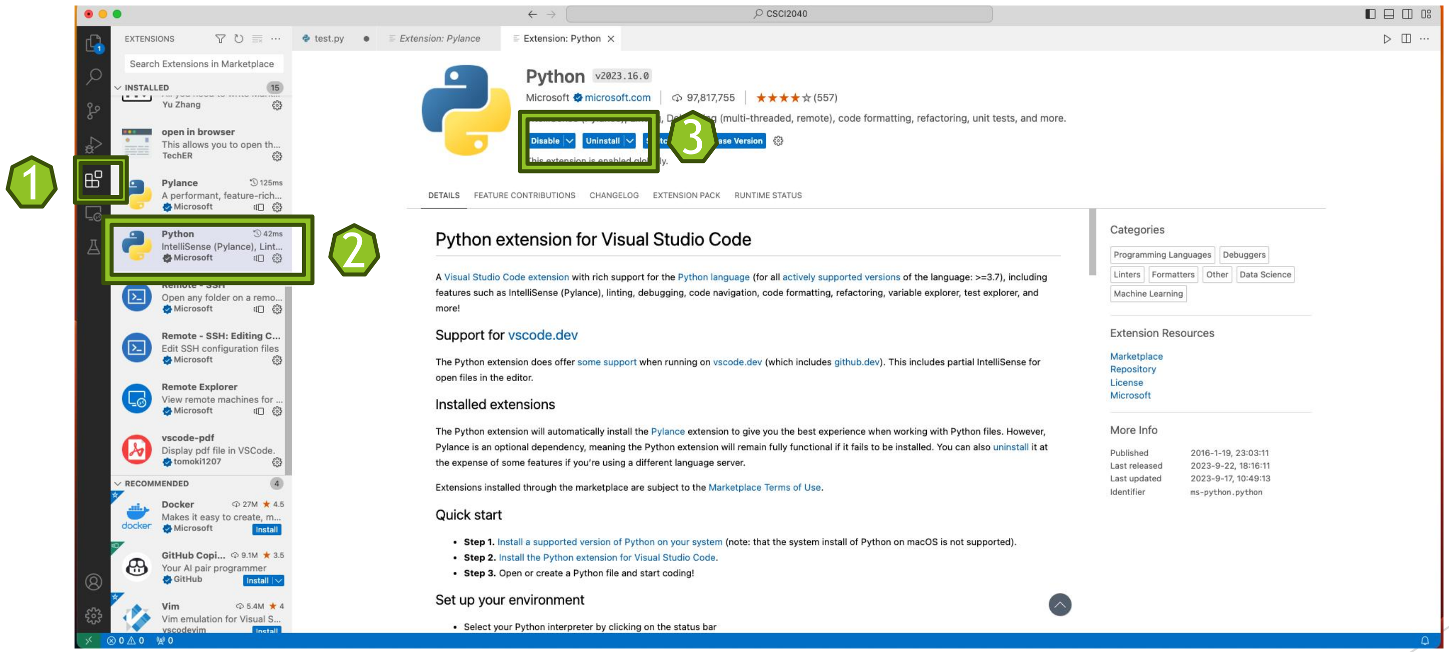
Una Kravets @Una
VS @code does so many things right. I'm constantly impressed by the UX, and customizing workspace / user preferences is no exception. 🙌💯 It just keeps getting better ❤️

Jonathan Dunlap @jadbox
VS Code will have deep remote development. You can connect to a container running a different OS and use any VS Code plugins, linting, debugging for that environment. 🔥🔥🔥

Pavithra Kodmad @PKodmad
VS Code is my most used and favorite editor. I love being able to customize the editor - changing the layout, the icons, fonts and color scheme is so easy!

Use VSCode to Write Code

- Open the VSCode and Install the Python Extension
  - To install Python Extension, open VSCode, go to the Extensions view by clicking on the Extensions icon in the Activity Bar on the left side of the window, and search for ‘Python’. Click ‘Install’ to add it.



Yu Zhang

open in browser
This allows you to open th...
TechER

Pylance 125ms
A performant, feature-rich...
Microsoft

Python 42ms
IntelliSense (Pylance), Lint...
Microsoft

Open any folder on a remo...
Microsoft

Remote - SSH: Editing C...
Edit SSH configuration files
Microsoft

Remote Explorer
View remote machines for ...
Microsoft

vscode-pdf
Display pdf file in VSCode.
tomoki1207

Docker 27M ★ 4.5
Makes it easy to create, m...
Microsoft

GitHub Copi... 9.1M ★ 3.5
Your AI pair programmer
GitHub

Vim 5.4M ★ 4
Vim emulation for Visual S...
vscodevim

Python v2023.16.0
Microsoft microsoft.com | 97,817,755 | ★★★★☆ (557)

This extension is enabled globally.

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported versions of the language: >=3.7), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Support for vscode.dev

The Python extension does offer some support when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.

Installed extensions

The Python extension will automatically install the Pylance extension to give you the best experience when working with Python files. However, Pylance is an optional dependency, meaning the Python extension will remain fully functional if it fails to be installed. You can also uninstall it at the expense of some features if you're using a different language server.

Extensions installed through the marketplace are subject to the Marketplace Terms of Use.

Quick start

- **Step 1.** Install a supported version of Python on your system (note: that the system install of Python on macOS is not supported).
- **Step 2.** Install the Python extension for Visual Studio Code.
- **Step 3.** Open or create a Python file and start coding!

Set up your environment

- Select your Python interpreter by clicking on the status bar

Categories

Programming Languages | Debuggers | Linters | Formatters | Other | Data Science | Machine Learning

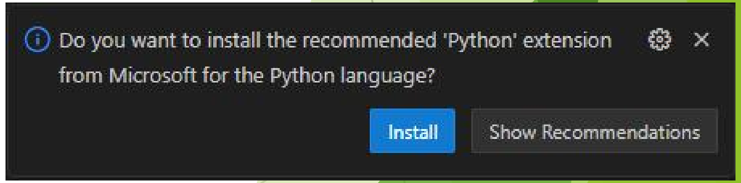
Extension Resources

Marketplace
Repository
License
Microsoft

More Info

Published	2016-1-19, 23:03:11
Last released	2023-9-22, 18:16:11
Last updated	2023-9-17, 10:49:13
Identifier	ms-python.python

An alternative way to install the extension is to create a .py file via the **New File…** menu. Then VSCode will show the following dialog. Click Install.

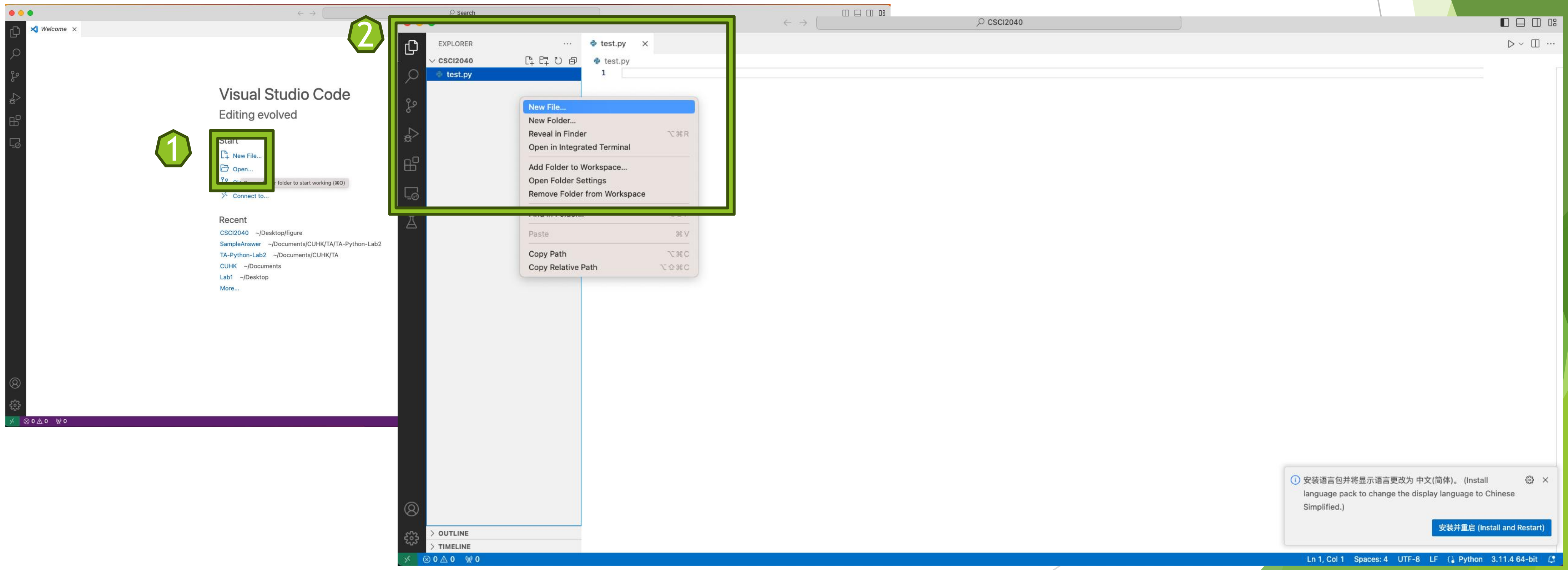


Do you want to install the recommended 'Python' extension from Microsoft for the Python language?

Use VSCode to Write Code

First, create a new folder, say “CSCI2040” on your Desktop using the File Explorer app on Windows or the Finder app on macOS.

- Click File > Open Folder to open the folder, and create a new “.py file”, say test.py, in the folder by clicking the right button of your mouse.

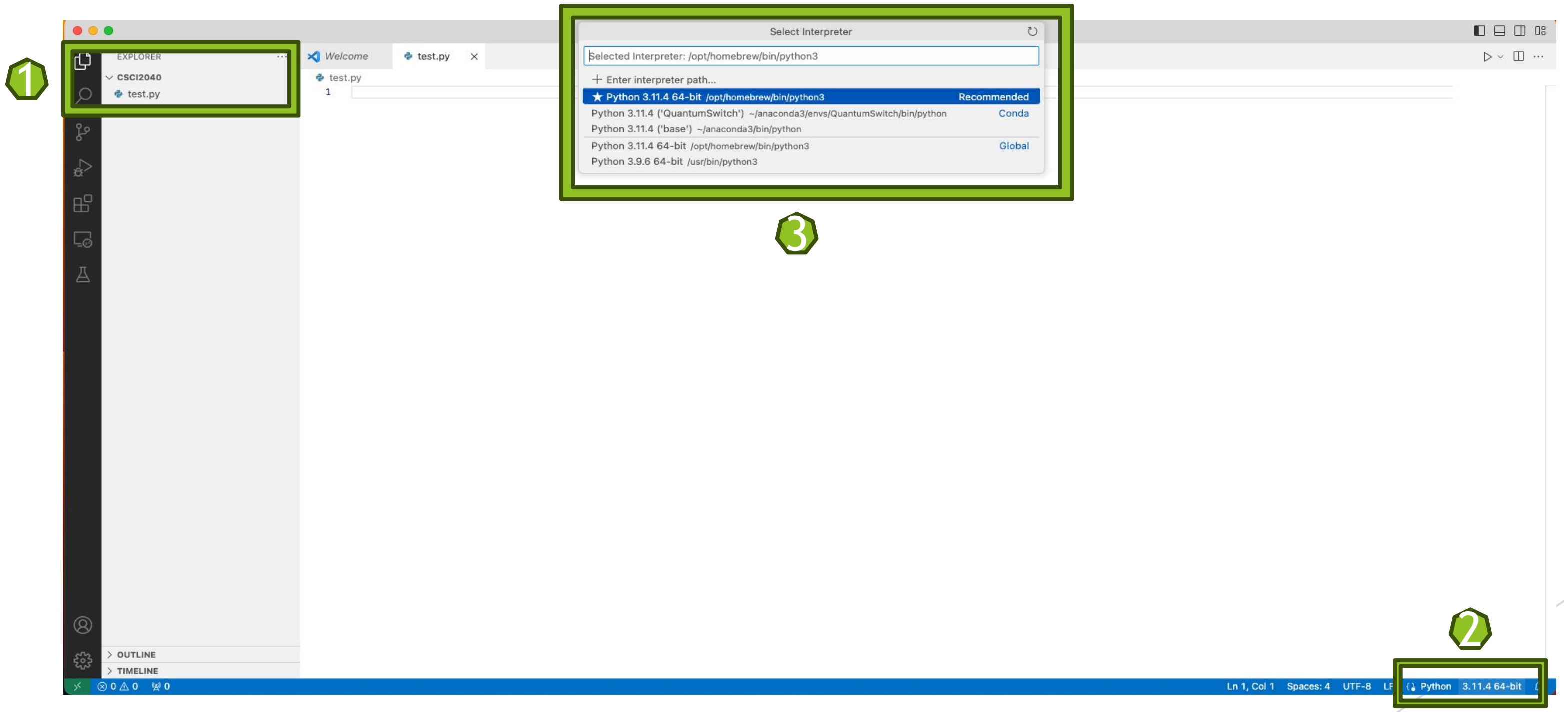


安装语言包并将显示语言更改为 中文(简体)。(Install language pack to change the display language to Chinese Simplified.)

安装并重启 (Install and Restart)

Use VSCode to Write Code in a Virtual Environment

- Click the right-bottom corner of the bottom status bar to choose which environments to use or Open the Command Palette (Win: Ctrl+Shift+P/Mac: Command+Shift+P), type ‘Python: Select Interpreter’, and hit Enter. A list of available interpreters will appear. Select the one that corresponds to your Anaconda environment.



Select Interpreter

Selected Interpreter: /opt/homebrew/bin/python3

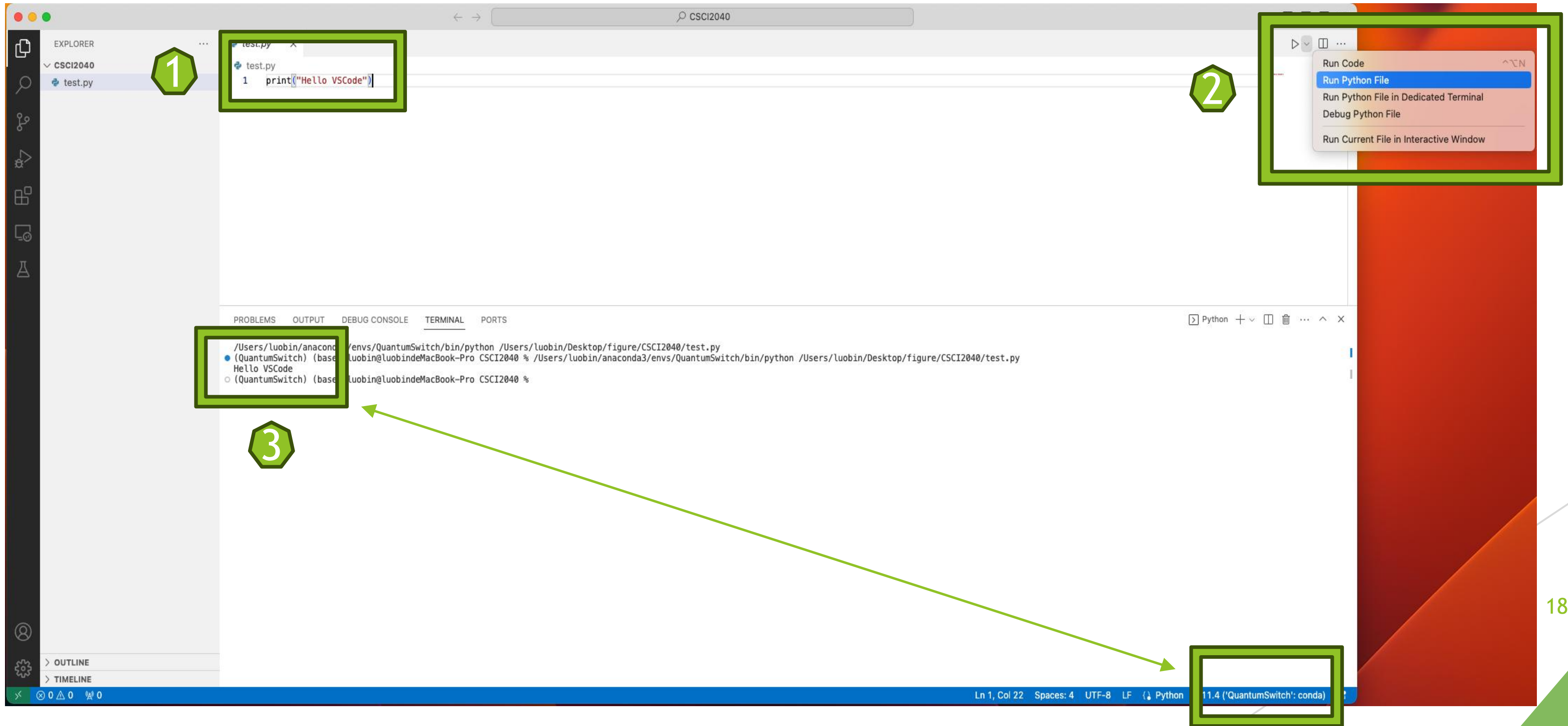
+ Enter interpreter path...

★ Python 3.11.4 64-bit /opt/homebrew/bin/python3	Recommended
Python 3.11.4 ('QuantumSwitch') ~/anaconda3/envs/QuantumSwitch/bin/python	Conda
Python 3.11.4 ('base') ~/anaconda3/bin/python	
Python 3.11.4 64-bit /opt/homebrew/bin/python3	Global
Python 3.9.6 64-bit /usr/bin/python3	

Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.11.4 64-bit

Use VSCode to Write Code in a Virtual Environment

- Verify the environment activation: you can write print("Hello VSCode") in the test.py and run this .py file. The terminal should show the name of your Anaconda environment before the prompt, indicating that the environment is active.



```python
print("Hello VSCode")
```

```
/Users/luobin/anacond /envs/QuantumSwitch/bin/python /Users/luobin/Desktop/figure/CSCI2040/test.py
(QuantumSwitch) (base luobin@luobindeMacBook-Pro CSCI2040 % /Users/luobin/anaconda3/envs/QuantumSwitch/bin/python /Users/luobin/Desktop/figure/CSCI2040/test.py
Hello VSCode
(QuantumSwitch) (base luobin@luobindeMacBook-Pro CSCI2040 %
```

Use VSCode to Write Code in a Virtual Environment

- Press Ctrl-S (or Cmd-S on macOS) to save your .py source file.

- See if you can locate the file in macOS Finder or Windows Explorer.
  - Be sure you know how to do so; you need to locate and attach the completed Python script files for a lab assignment submission to Blackboard.

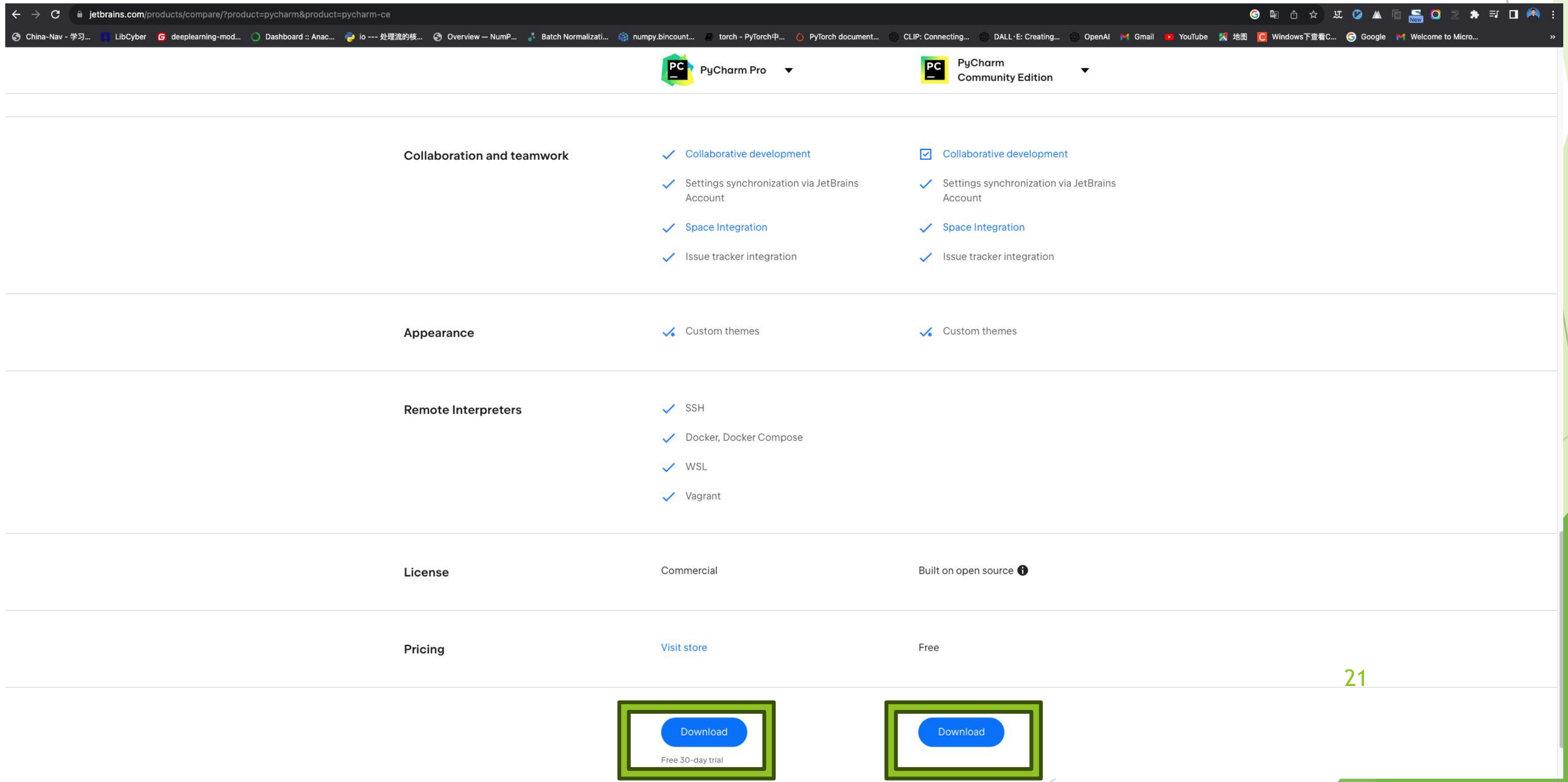
- For more information on using Anaconda Environment in VSCode, you may refer to this guide: https://saturncloud.io/blog/activating-anaconda-environment-in-vscode-a-guide-for-data-scientists/

PyCharm IDE

From JetBrains

Use PyCharm to Write Code

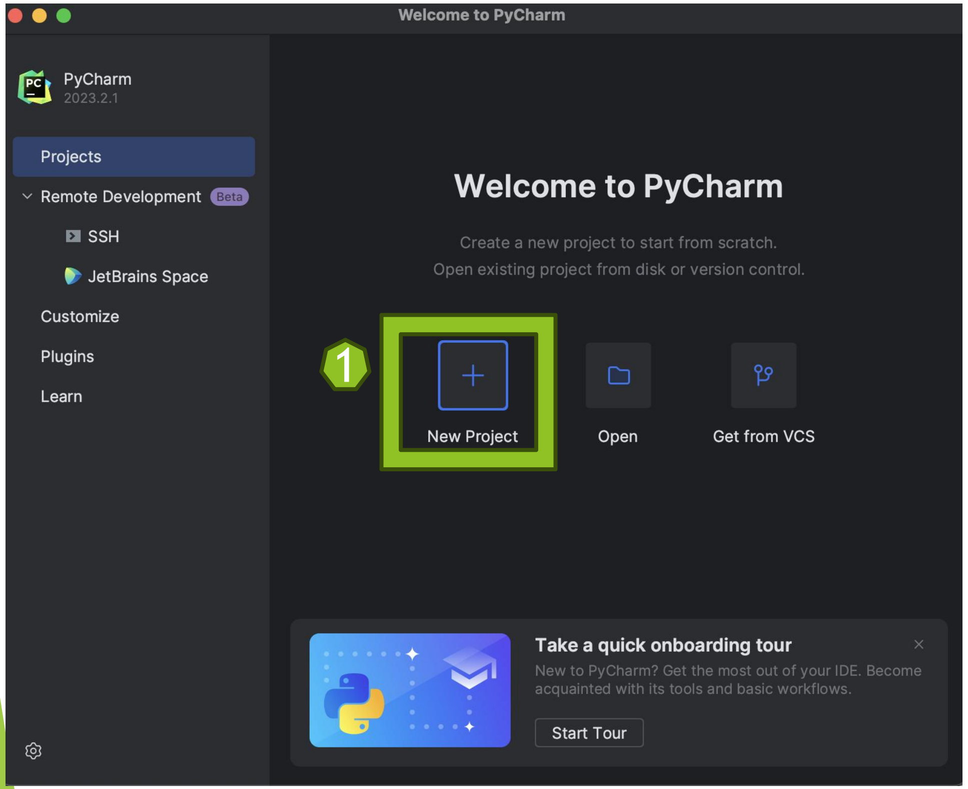
- Download PyCharm or PyCharm Pro from the following website: https://www.jetbrains.com/products/compare/?product=pycharm&product=pycharm-ce

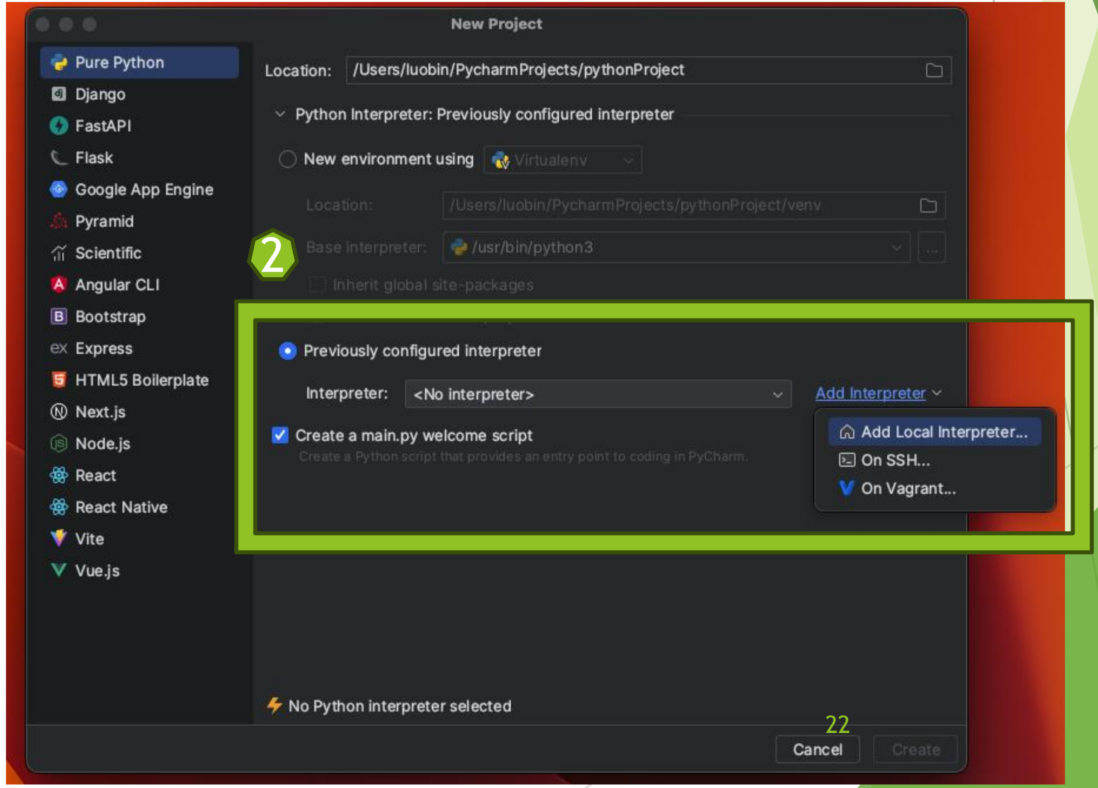


	PyCharm Pro	PyCharm Community Edition
Collaboration and teamwork	✓ Collaborative development ✓ Settings synchronization via JetBrains Account ✓ Space Integration ✓ Issue tracker integration	✓ Collaborative development ✓ Settings synchronization via JetBrains Account ✓ Space Integration ✓ Issue tracker integration
Appearance	✓ Custom themes	✓ Custom themes
Remote Interpreters	✓ SSH ✓ Docker, Docker Compose ✓ WSL ✓ Vagrant	
License	Commercial	Built on open source
Pricing	Visit store	Free
	Download Free 30-day trial	Download

Use PyCharm to Write Code

- Open PyCharm and start to create a new project with new environment.

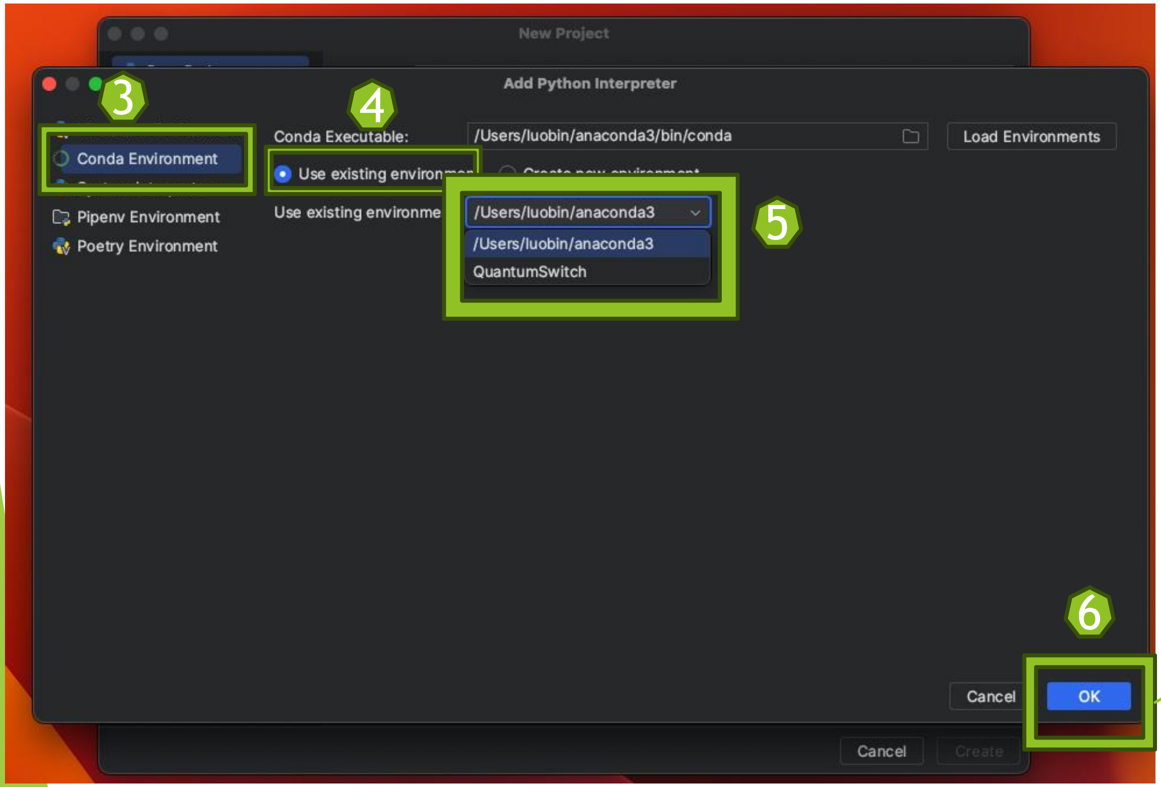

Welcome to PyCharm
PyCharm
2023.2.1
Projects
Remote Development Beta
SSH
JetBrains Space
Customize
Plugins
Learn
Welcome to PyCharm
Create a new project to start from scratch.
Open existing project from disk or version control.
1
New Project
Open
Get from VCS
Take a quick onboarding tour
New to PyCharm? Get the most out of your IDE. Become acquainted with its tools and basic workflows.
Start Tour



Location: /Users/luobin/PycharmProjects/pythonProject
Python Interpreter: Previously configured interpreter
New environment using Virtualenv
Location: /Users/luobin/PycharmProjects/pythonProject/venv
Base interpreter: /usr/bin/python3
Inherit global site-packages
Previously configured interpreter
Interpreter: <No interpreter>
Create a main.py welcome script
Create a Python script that provides an entry point to coding in PyCharm.
No Python interpreter selected

Use PyCharm to Write Code

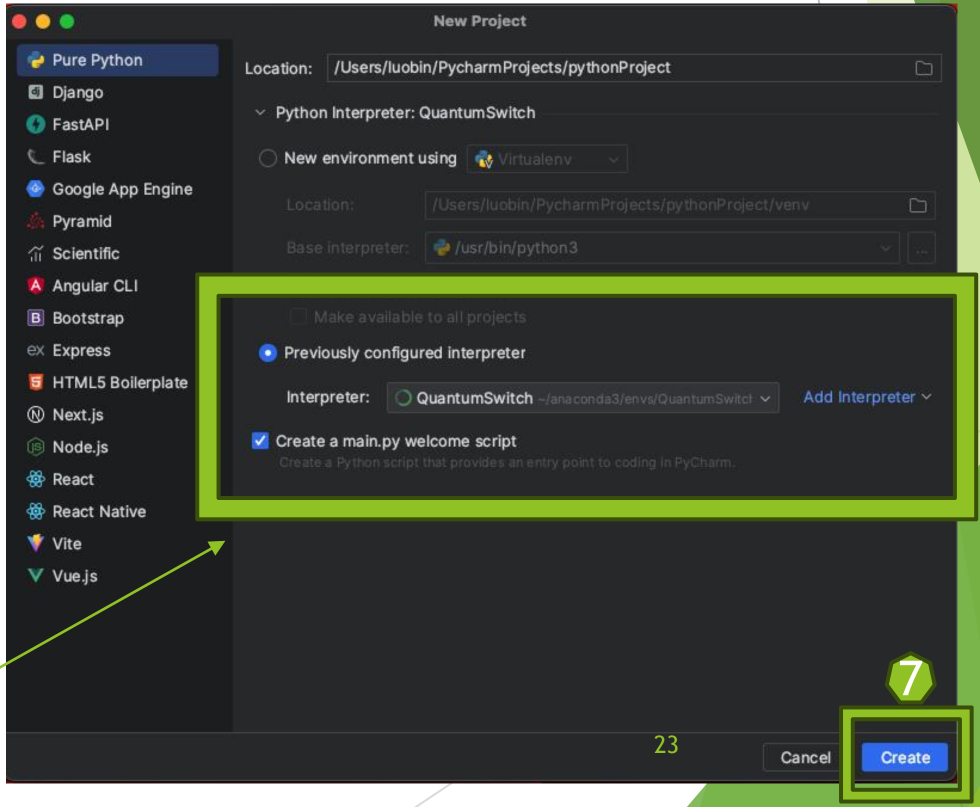
- Open PyCharm and start to create a new project with new environment.



Conda Executable: /Users/luobin/anaconda3/bin/conda

Use existing environment / Create new environment

Use existing environme: /Users/luobin/anaconda3
- /Users/luobin/anaconda3
- QuantumSwitch



Location: /Users/luobin/PycharmProjects/pythonProject

Python Interpreter: QuantumSwitch

New environment using Virtualenv

Location: /Users/luobin/PycharmProjects/pythonProject/venv

Base interpreter: /usr/bin/python3

Make available to all projects

Previously configured interpreter

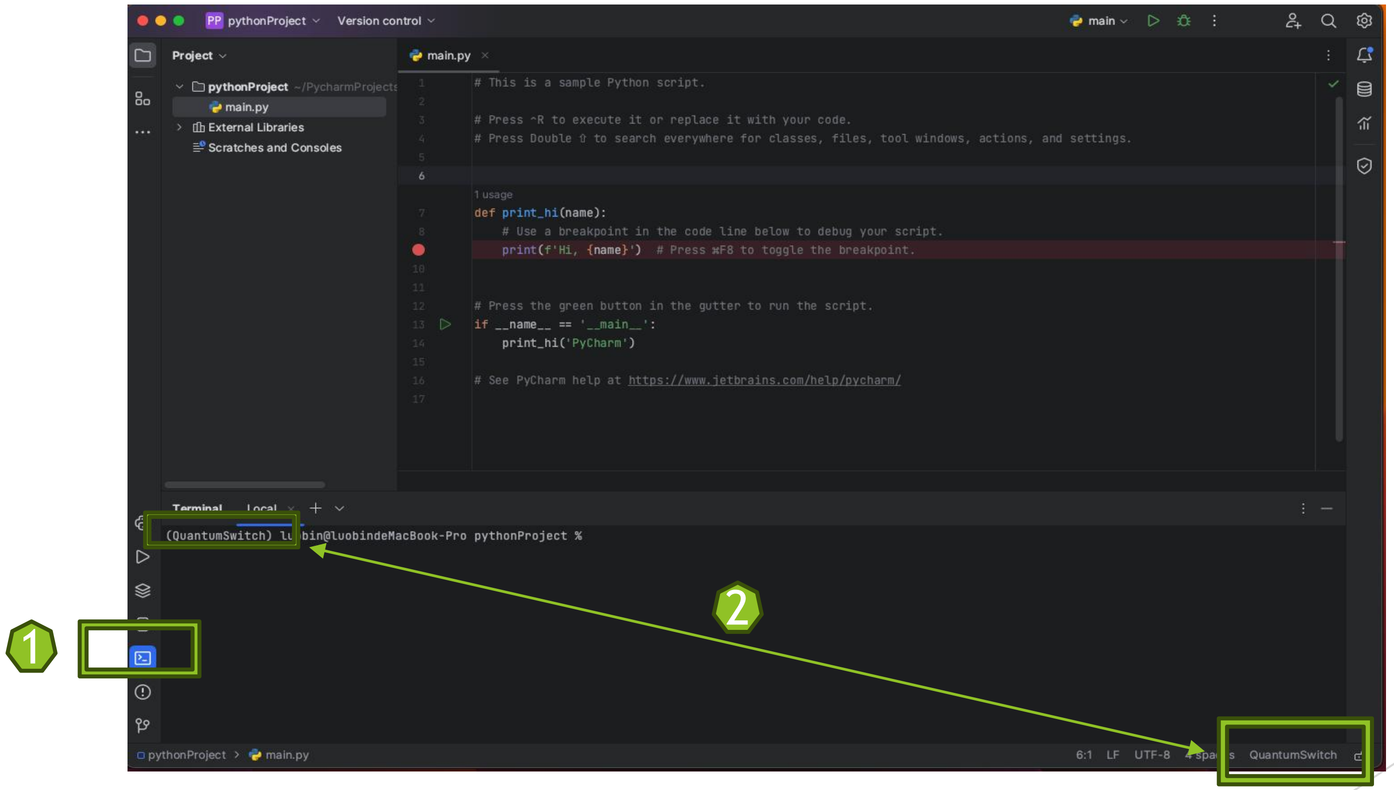
Interpreter: QuantumSwitch ~/anaconda3/envs/QuantumSwitch

Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

Use PyCharm to Write Code

- Verify the environment activation: The terminal should show the name of your Anaconda environment before the prompt, indicating that the environment is active.

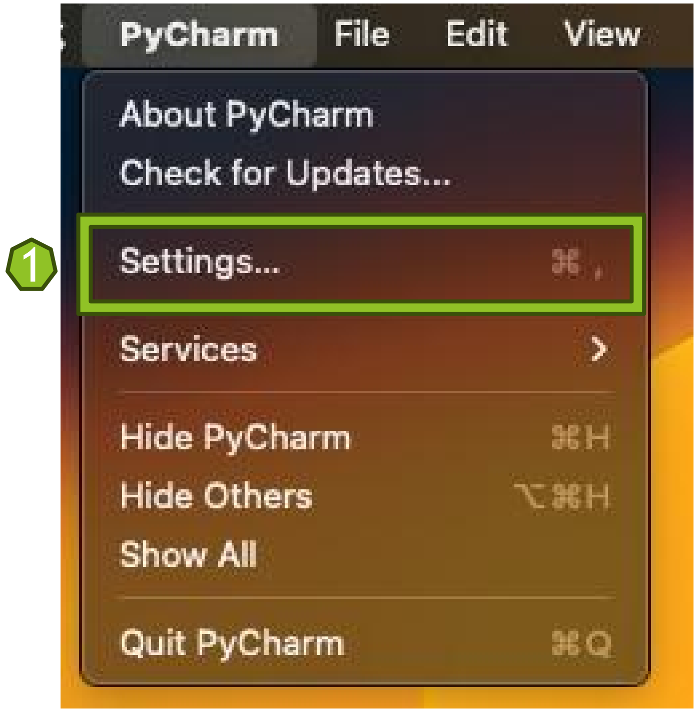


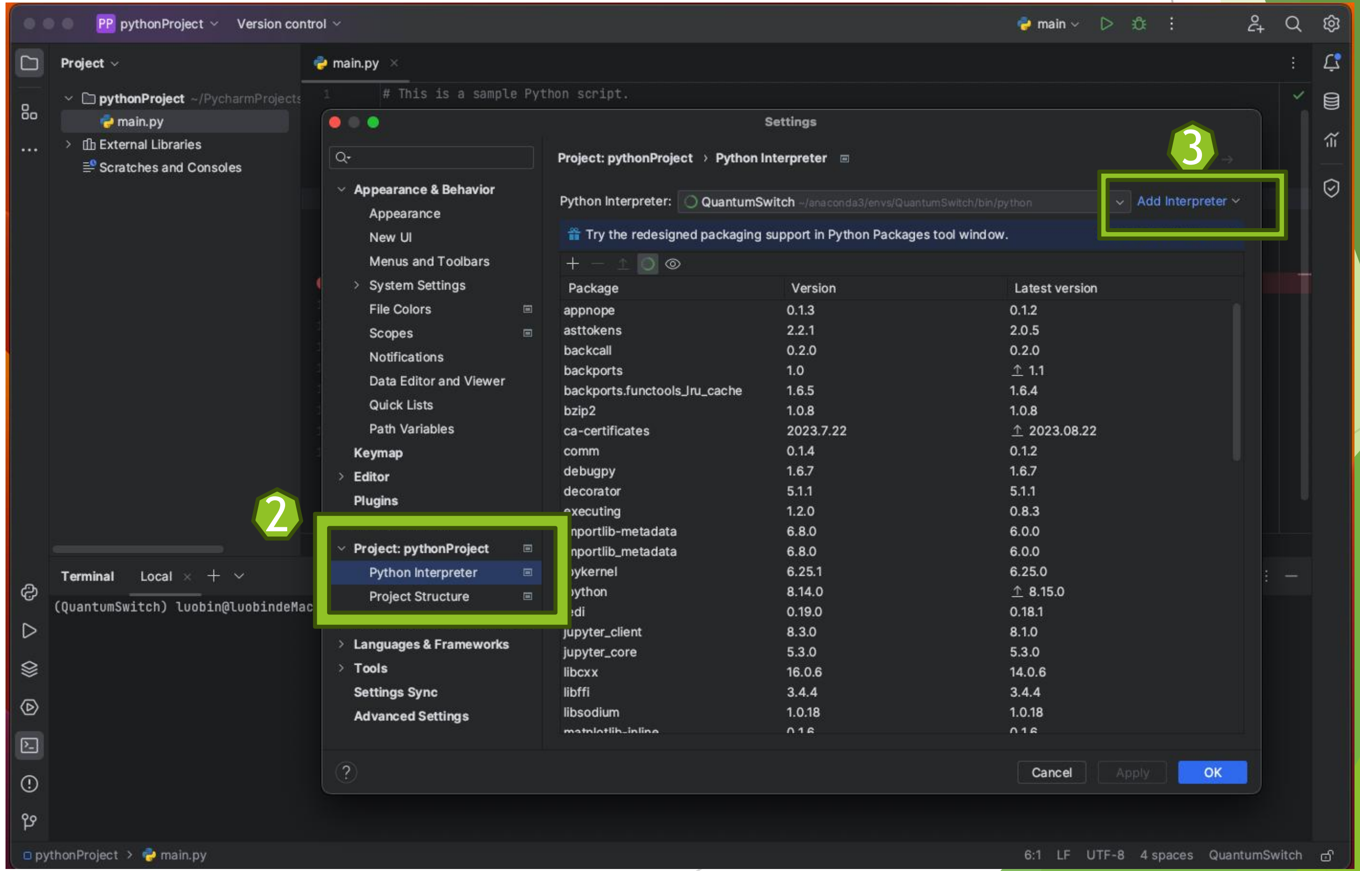
```python
# This is a sample Python script.

# Press ^R to execute it or replace it with your code.
# Press Double ⇧ to search everywhere for classes, files, tool windows, actions, and settings.


def print_hi(name):
    # Use a breakpoint in the code line below to debug your script.
    print(f'Hi, {name}')  # Press ⌘F8 to toggle the breakpoint.


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')

# See PyCharm help at https://www.jetbrains.com/help/pycharm/
```

```
(QuantumSwitch) luobin@luobindeMacBook-Pro pythonProject %
```

Use PyCharm to Write Code in a Virtual Environment

- Change environment in your existing project and file.





```
# This is a sample Python script.
```

```
(QuantumSwitch) luobin@luobindeMac
```

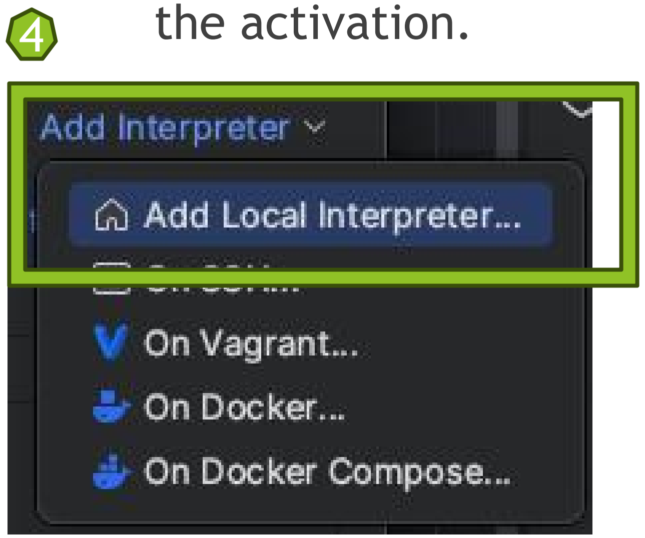
Python Interpreter: QuantumSwitch ~/anaconda3/envs/QuantumSwitch/bin/python

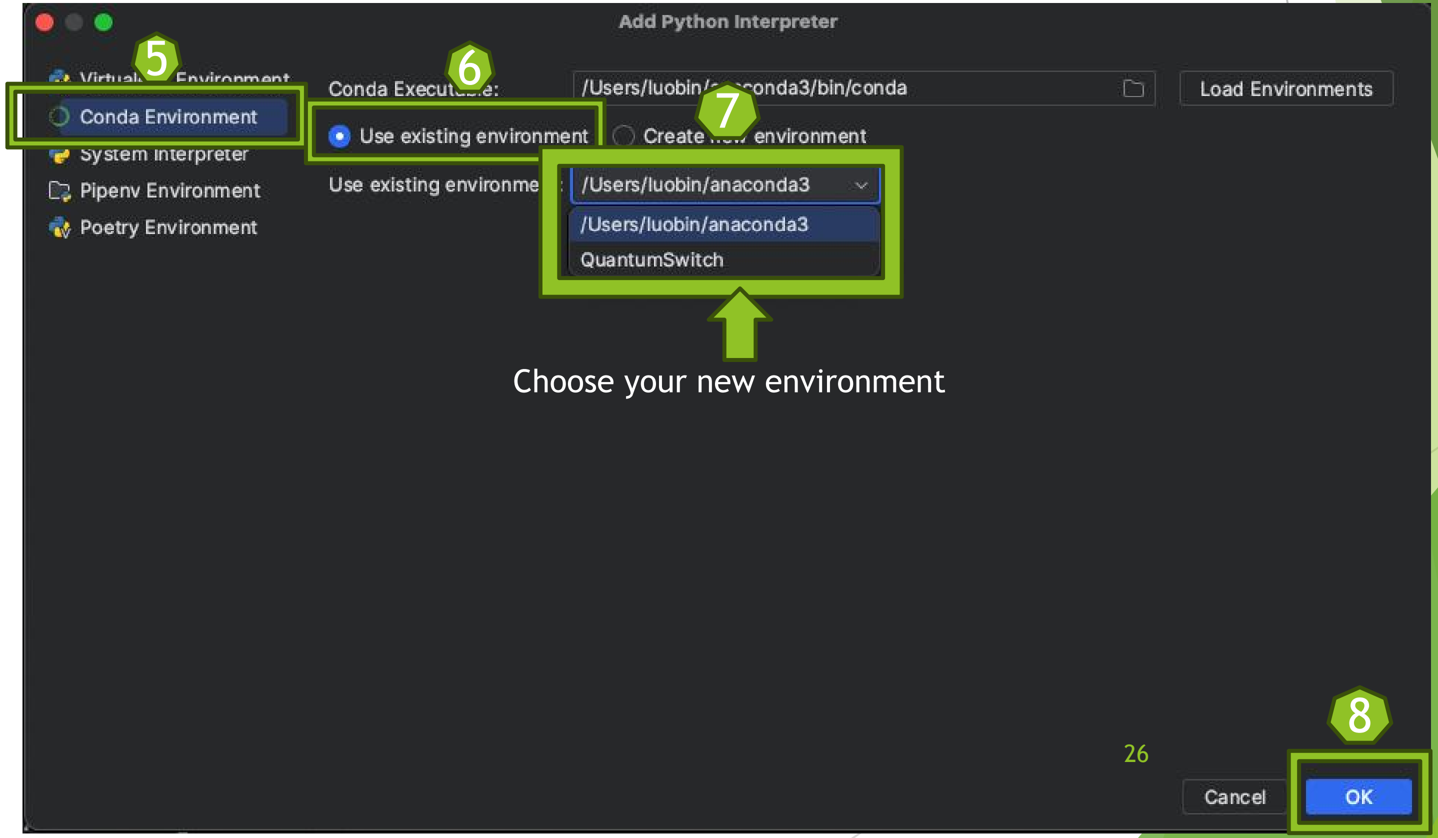
Try the redesigned packaging support in Python Packages tool window.

Package	Version	Latest version
appnope	0.1.3	0.1.2
asttokens	2.2.1	2.0.5
backcall	0.2.0	0.2.0
backports	1.0	↑ 1.1
backports.functools_lru_cache	1.6.5	1.6.4
bzip2	1.0.8	1.0.8
ca-certificates	2023.7.22	↑ 2023.08.22
comm	0.1.4	0.1.2
debugpy	1.6.7	1.6.7
decorator	5.1.1	5.1.1
executing	1.2.0	0.8.3
nportlib-metadata	6.8.0	6.0.0
nportlib_metadata	6.8.0	6.0.0
ykernel	6.25.1	6.25.0
ython	8.14.0	↑ 8.15.0
edi	0.19.0	0.18.1
jupyter_client	8.3.0	8.1.0
jupyter_core	5.3.0	5.3.0
libcxx	16.0.6	14.0.6
libffi	3.4.4	3.4.4
libsodium	1.0.18	1.0.18
matplotlib-inline	0.1.6	0.1.6

Use PyCharm to Write Code in a Virtual Environment

- Change environment in your existing project and file and remember to verify the activation.





Conda Executable: /Users/luobin/anaconda3/bin/conda

Use existing environment

Use existing environment: /Users/luobin/anaconda3

/Users/luobin/anaconda3
QuantumSwitch

Choose your new environment

Package Management

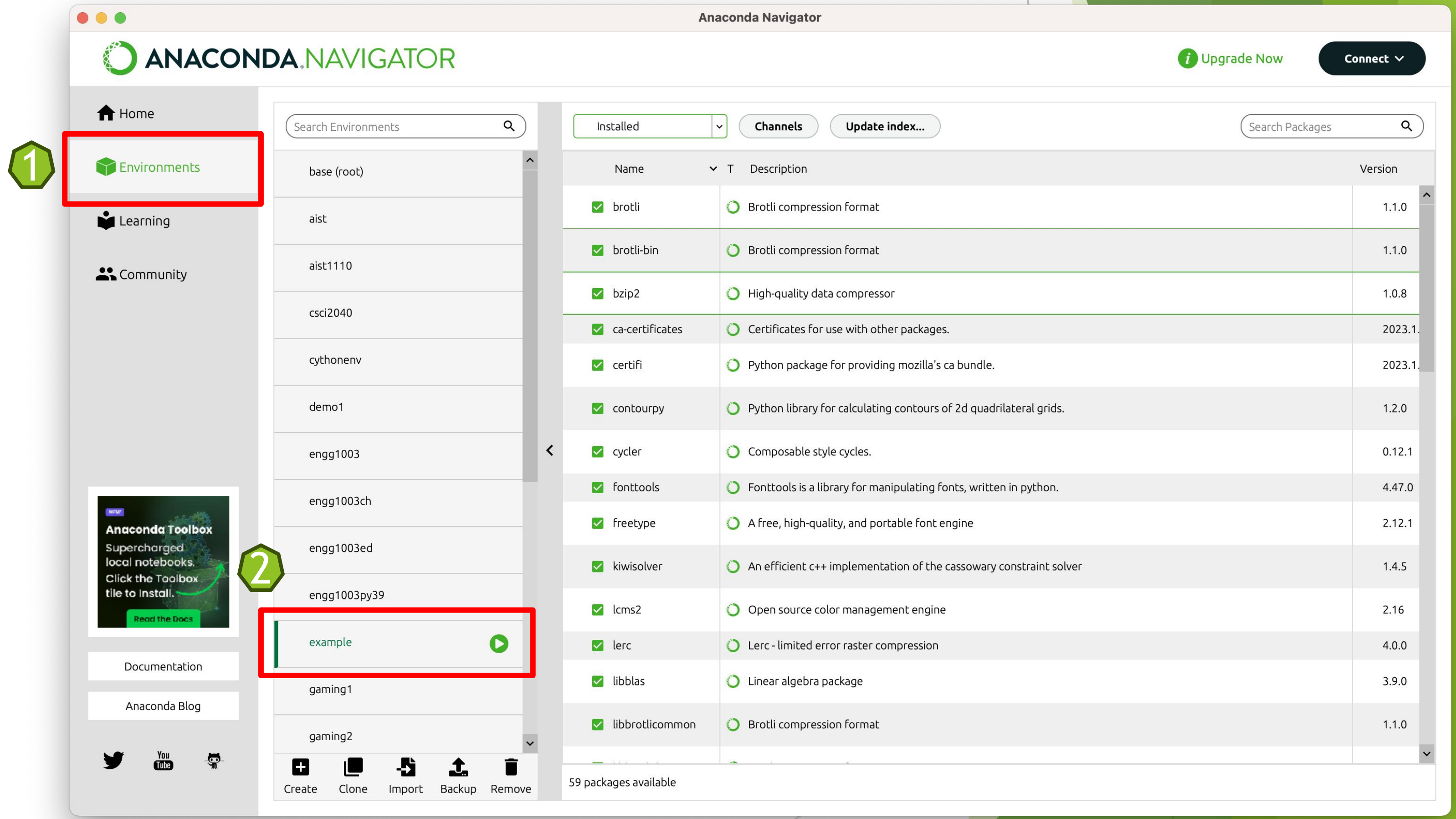
Package Management

- A (Python) package means some library of code that can be installed on your computer. It contains useful code that can be reused and imported into your program to save your program development time.
- Package management: *install*, *update*, *uninstall*, *list installed packages*
- Package managers:
  - **conda**
  - **pip**
    - A recursive acronym that stands for "Pip Installs Packages" or "preferred installer program".

both provided by Anaconda

Manage Packages in Anaconda Navigator

- Click the Environments tab.
- Choose an environment.
- The GUI will show the list of installed packages under the chosen environment.



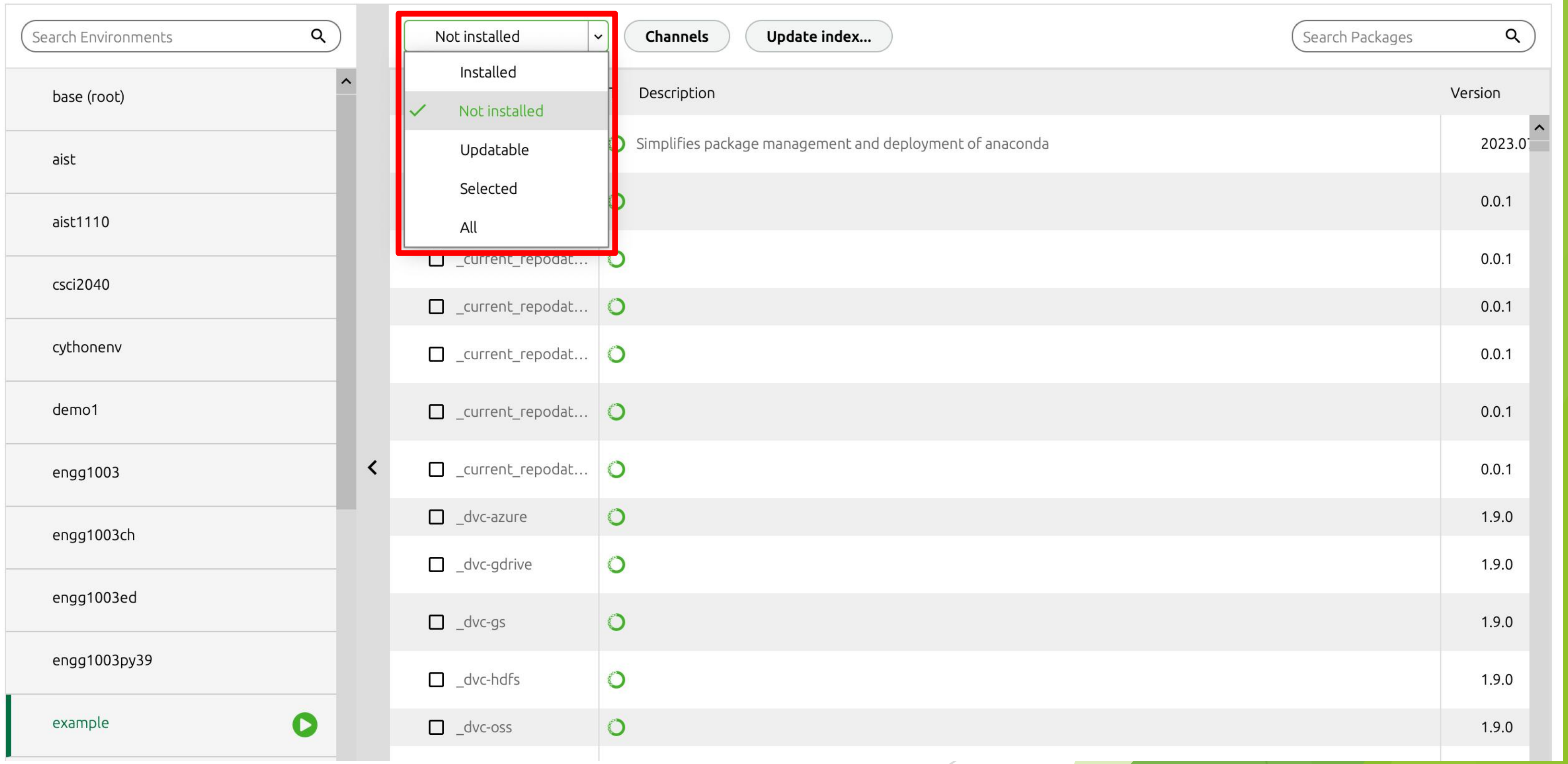
Environments: base (root), aist, aist1110, csci2040, cythonenv, demo1, engg1003, engg1003ch, engg1003ed, engg1003py39, example, gaming1, gaming2

Name	T	Description	Version
brotli		Brotli compression format	1.1.0
brotli-bin		Brotli compression format	1.1.0
bzip2		High-quality data compressor	1.0.8
ca-certificates		Certificates for use with other packages.	2023.1.
certifi		Python package for providing mozilla's ca bundle.	2023.1.
contourpy		Python library for calculating contours of 2d quadrilateral grids.	1.2.0
cycler		Composable style cycles.	0.12.1
fonttools		Fonttools is a library for manipulating fonts, written in python.	4.47.0
freetype		A free, high-quality, and portable font engine	2.12.1
kiwisolver		An efficient c++ implementation of the cassowary constraint solver	1.4.5
lcms2		Open source color management engine	2.16
lerc		Lerc - limited error raster compression	4.0.0
libblas		Linear algebra package	3.9.0
libbrotlicommon		Brotli compression format	1.1.0

59 packages available

Manage Packages in Anaconda Navigator

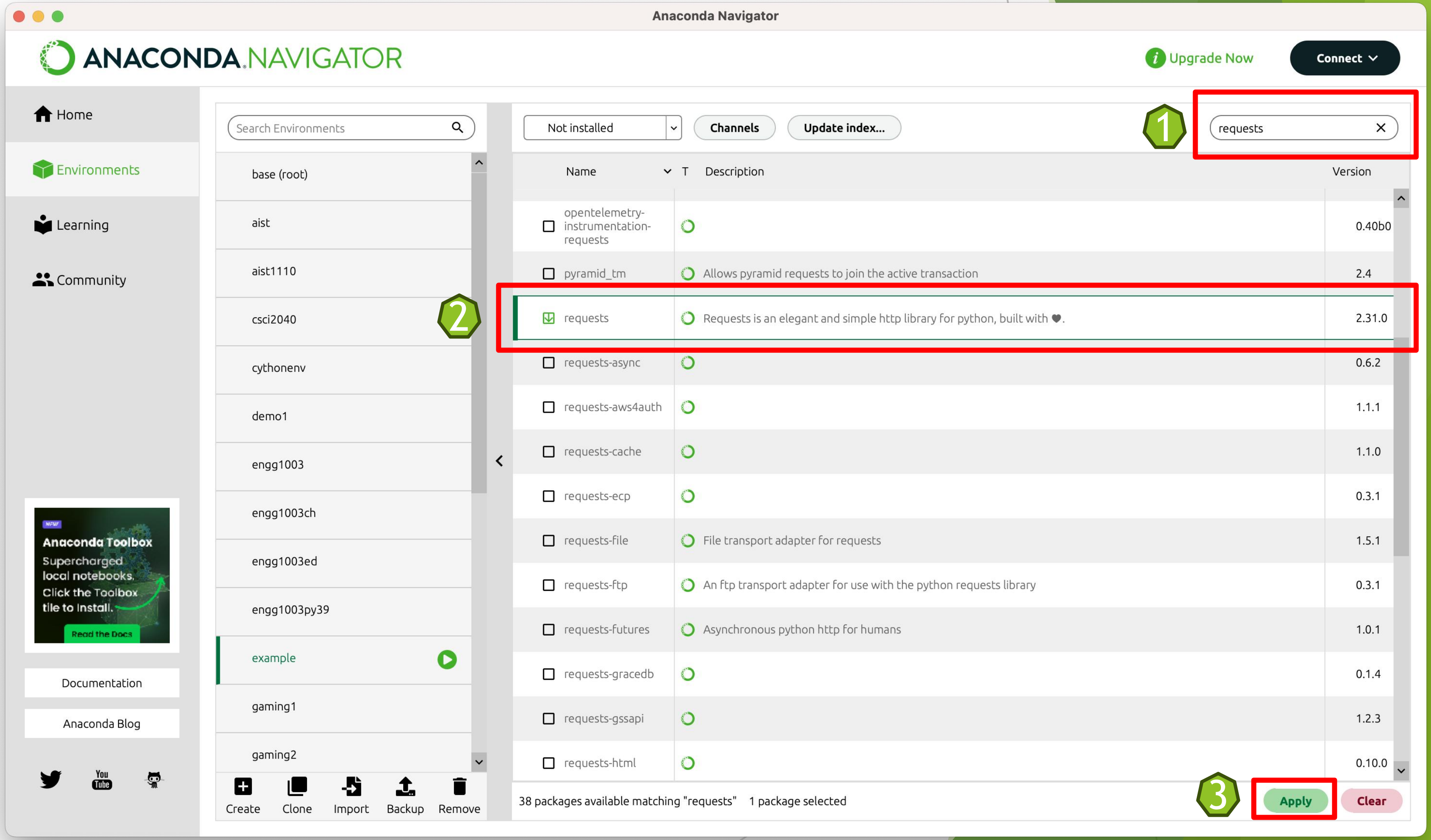
- To install a new package in the chosen environment, click the drop-down menu and choose “Not installed”.
- The list shows all packages that are not yet installed in the environment.



Name		Description	Version
		Simplifies package management and deployment of anaconda	2023.0
			0.0.1
☐ _current_repodat...			0.0.1
☐ _current_repodat...			0.0.1
☐ _current_repodat...			0.0.1
☐ _current_repodat...			0.0.1
☐ _current_repodat...			0.0.1
☐ _dvc-azure			1.9.0
☐ _dvc-gdrive			1.9.0
☐ _dvc-gs			1.9.0
☐ _dvc-hdfs			1.9.0
☐ _dvc-oss			1.9.0

Manage Packages in Anaconda Navigator

- Use the search bar to search for the desired package to install.
- For example, type "requests" to search for the requests package (a library for making HTTP requests).
- Check the box of the package to install and click the Apply button.



base (root)
aist
aist1110
csci2040
cythonenv
demo1
engg1003
engg1003ch
engg1003ed
engg1003py39
example
gaming1
gaming2

requests

Name	T	Description	Version
☐ opentelemetry-instrumentation-requests			0.40b0
☐ pyramid_tm		Allows pyramid requests to join the active transaction	2.4
☑ requests		Requests is an elegant and simple http library for python, built with ♥.	2.31.0
☐ requests-async			0.6.2
☐ requests-aws4auth			1.1.1
☐ requests-cache			1.1.0
☐ requests-ecp			0.3.1
☐ requests-file		File transport adapter for requests	1.5.1
☐ requests-ftp		An ftp transport adapter for use with the python requests library	0.3.1
☐ requests-futures		Asynchronous python http for humans	1.0.1
☐ requests-gracedb			0.1.4
☐ requests-gssapi			1.2.3
☐ requests-html			0.10.0

38 packages available matching "requests"   1 package selected

Exercise

- Create a virtual environment named “lab0” with Python 3.11.
- In the lab0 environment, install a package called requests.
- Create a Python script file named get_web.py with the following lines of code using your favorite editor or IDE.

```
import requests
r = requests.get('https://api.spotify.com/')
print(r.status_code)
```

- Run it in the IDE or by executing python get_web.py in the terminal.
- Seeing “200” printed indicates the website access was successful.
- No submission is needed.

Computer Lab Usage

SHB 924

Computer Laboratory SHB 924

- Door Access: Tap your student ID card over the door access reader.
  - Access is granted to students who have officially enrolled in this course.
- Opening hours: Mon – Fri, 9am - 9pm

- Please use the area of SHB 924A.
- Suggest taking the same seat every time.
  - Why? You can access files and settings you left behind last time.
- Log in a Windows PC using your CUHK account and OnePass password.
  - For students who enrolled during the add/drop period, please raise your hand to seek the supporting TA to use a temporary user account to log in the PC for you.

Computer Laboratory Regulations

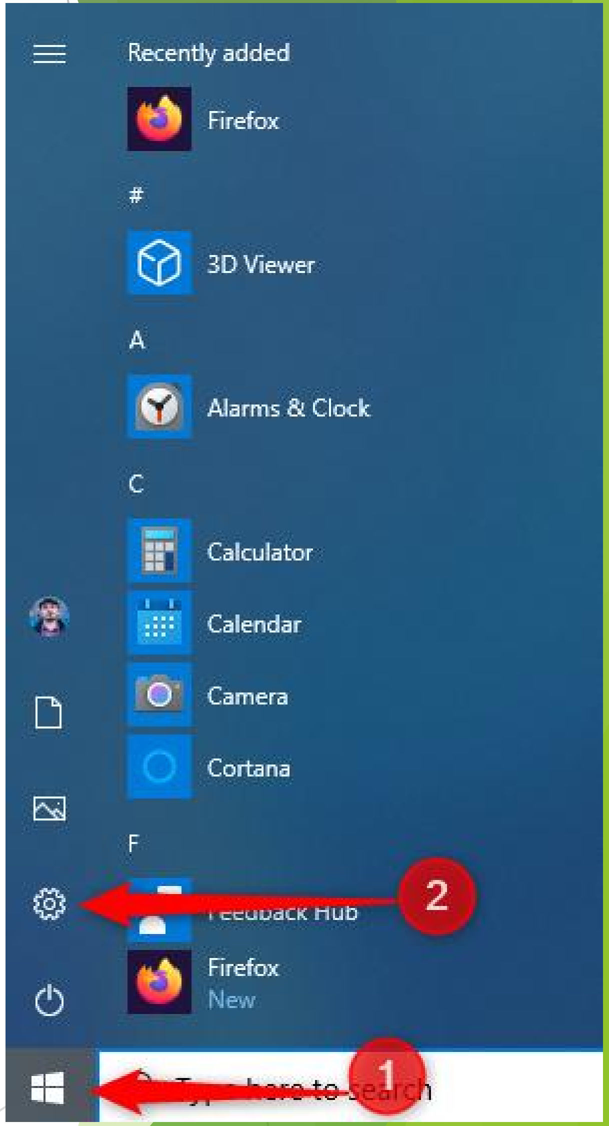
- Please observe computer laboratory regulations.
- For example,
  - Don’t take away or destroy any equipment.
  - No eating and drinking.
  - No computer games.
  - Users should work quietly. Don’t play music or hold group discussions that may disturb other users in the lab.

Back up Your Files

- Files saved under your user folder on the lab PC you logged in won't be deleted. You can access them by signing in the same PC next time.
- But we strongly recommend that you should still **back up your source files** stored on the lab Windows PC before you leave.
- Suggestion: Bring a USB thumb drive for doing the backup.
- Alternative ways of backup: by email, CUHK OneDrive, Google Drive
  - But these ways require some setup effort.
- Protect the **privacy** of your lab exercise solutions.
  - Don't save your files under the Users\Public folder which is visible to other users.
  - Files stored under your own user folder (Desktop, Documents, Download, …) are not accessible by others.

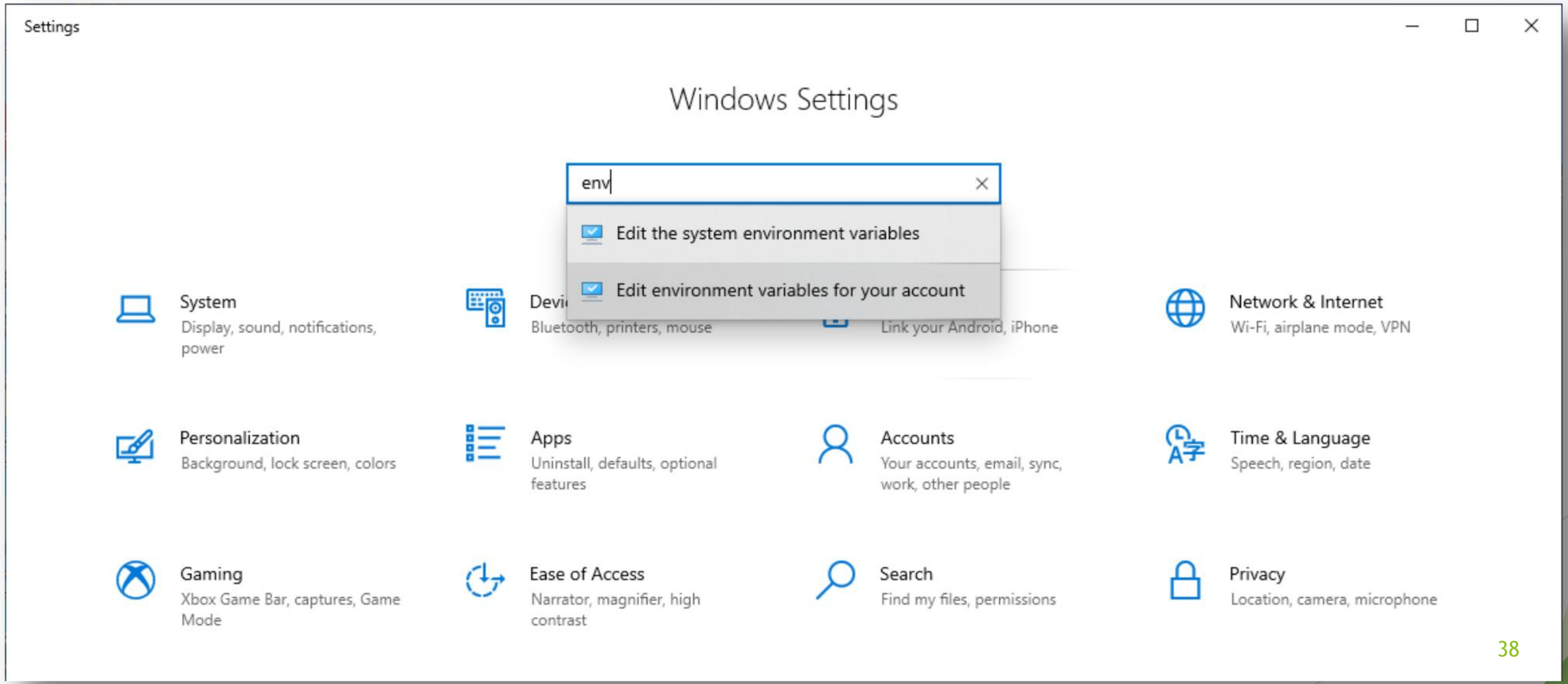
Proxy Server Setting on Lab PCs

- Our lab computers are behind firewall.
- To allow Anaconda's package manager like conda or pip to download Python packages from the Internet, you need to configure the proxy server setting on the lab PC you've logged in.
- The setting is user-based and local to the PC. If you move to another PC, you will need to set this up again.
- Click the Windows icon, and then the Gear to enter Windows Settings.



Proxy Server Setting on Lab PCs

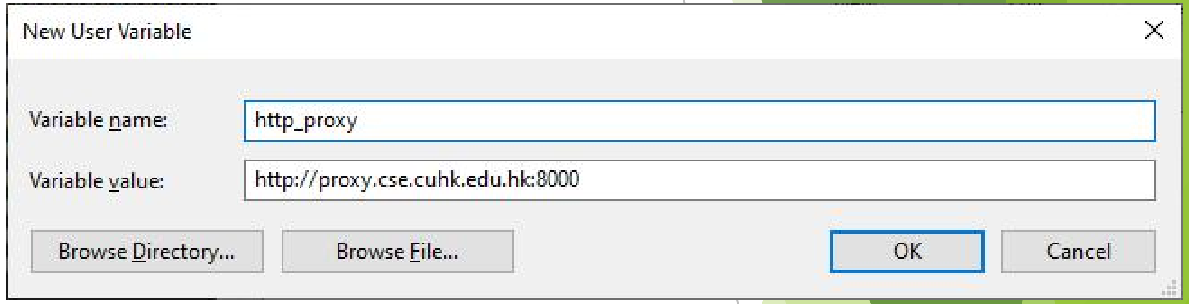
- Search “env” and choose “**Edit environment variables for your account**”.


Settings
Windows Settings
env
Edit the system environment variables
Edit environment variables for your account
System
Display, sound, notifications, power
Devi
Bluetooth, printers, mouse
Link your Android, iPhone
Network & Internet
Wi-Fi, airplane mode, VPN
Personalization
Background, lock screen, colors
Apps
Uninstall, defaults, optional features
Accounts
Your accounts, email, sync, work, other people
Time & Language
Speech, region, date
Gaming
Xbox Game Bar, captures, Game Mode
Ease of Access
Narrator, magnifier, high contrast
Search
Find my files, permissions
Privacy
Location, camera, microphone

Proxy Server Setting on Lab PCs

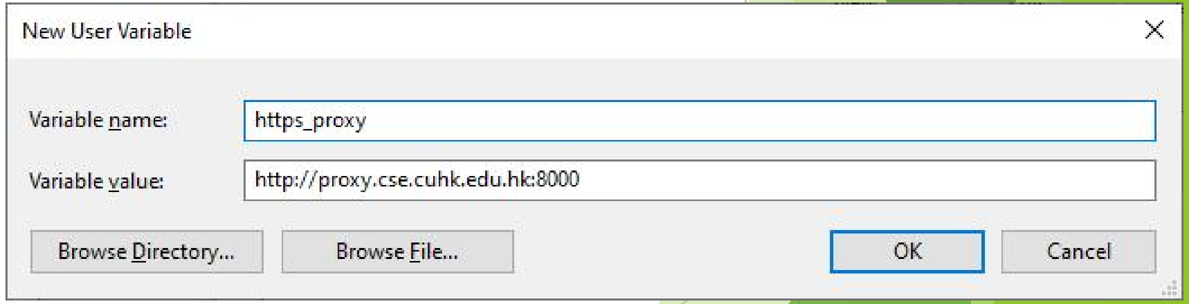
- Click the **New...** button to create two environment variables:
  - Variable name: **http_proxy**
  - Variable value: **http://proxy.cse.cuhk.edu.hk:8000**

  - Variable name: **https_proxy**
  - Variable value: **http://proxy.cse.cuhk.edu.hk:8000**
- Be careful not to make any typos here. Note that we use "http://..." for both variables.


New User Variable

Variable name: http_proxy

Variable value: http://proxy.cse.cuhk.edu.hk:8000

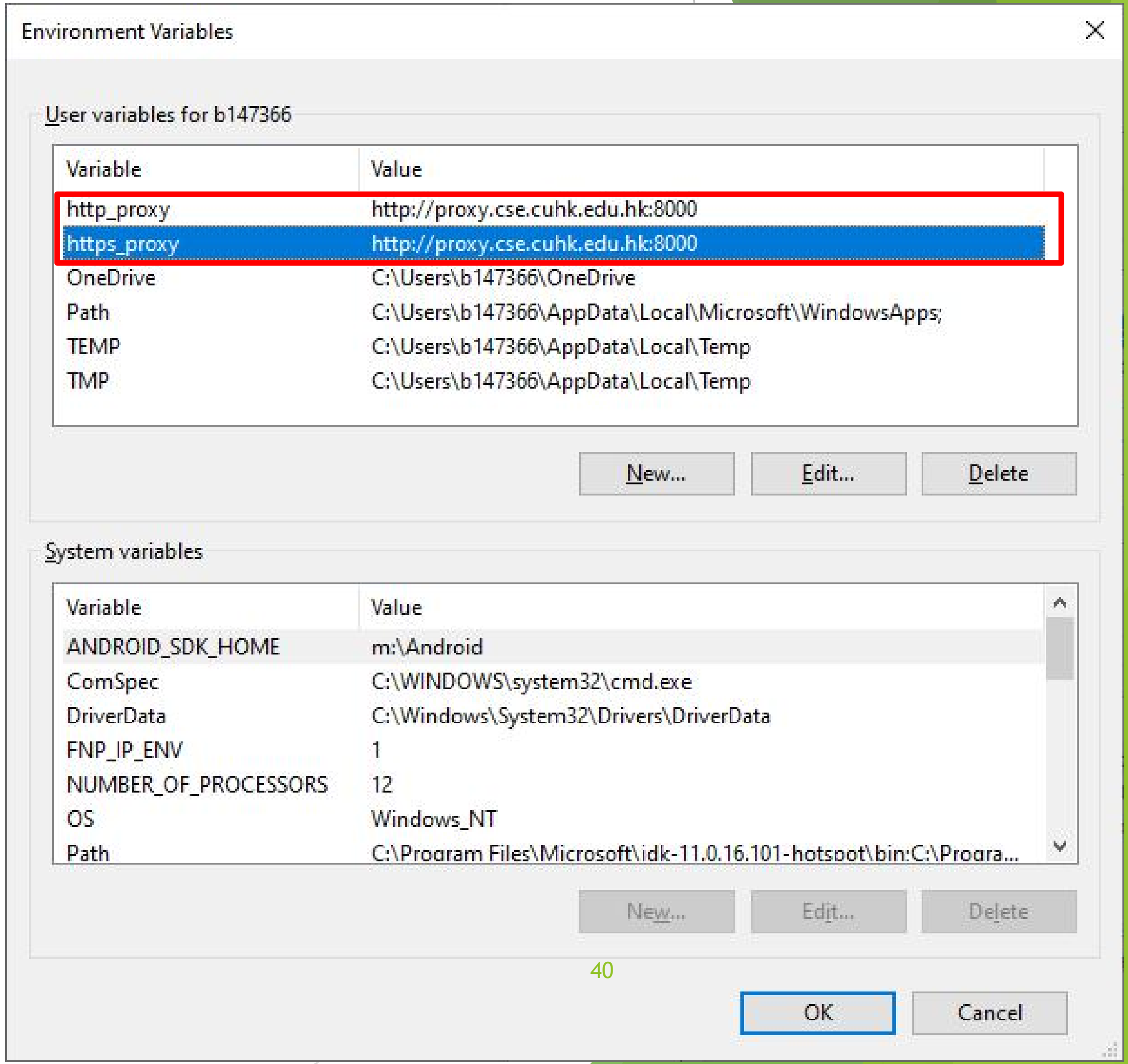

New User Variable

Variable name: https_proxy

Variable value: http://proxy.cse.cuhk.edu.hk:8000

Proxy Server Setting on Lab PCs

- The result is shown on the right screenshot.
- From now on, Anaconda can access the Internet for downloading packages.



Environment Variables

User variables for b147366

Variable	Value
http_proxy	http://proxy.cse.cuhk.edu.hk:8000
https_proxy	http://proxy.cse.cuhk.edu.hk:8000
OneDrive	C:\Users\b147366\OneDrive
Path	C:\Users\b147366\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\b147366\AppData\Local\Temp
TMP	C:\Users\b147366\AppData\Local\Temp

System variables

Variable	Value
ANDROID_SDK_HOME	m:\Android
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
FNP_IP_ENV	1
NUMBER_OF_PROCESSORS	12
OS	Windows_NT
Path	C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin;C:\Progra...

IDEs for Python Coding

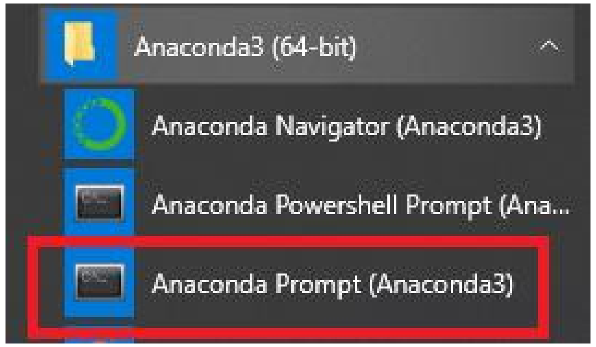
- Our lab PCs have installed Anaconda, VSCode and PyCharm for your Python coding.
- Microsoft's **Visual Studio Code** (VS Code) is very good choice of IDE (Integrated Development Environment) for Python coding that we recommend.
  - Download from: https://code.visualstudio.com
- JetBrains' **PyCharm** is another good IDE. Our lab provides the Community Edition.
  - The Professional Edition provides more features, e.g., running a Jupyter notebook.
    - Professional vs. Community
  - Academic users (you may use your CUHK email) can register a free educational license to use the Professional Edition.
  - Download the Professional Edition from https://www.jetbrains.com/pycharm/download/

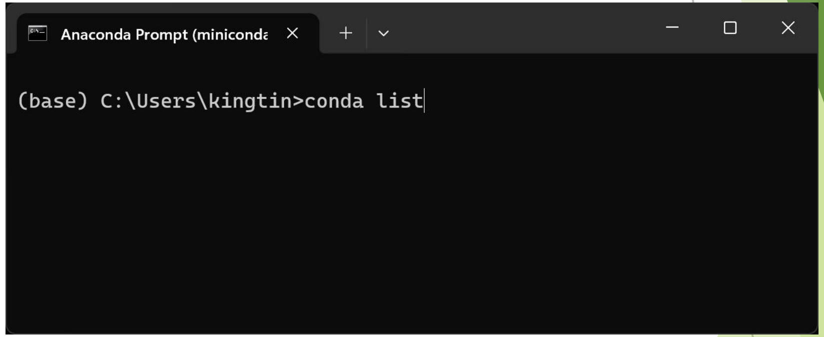
Appendix

Commands for Managing Virtual Environments and Packages

Creating a Virtual Environment in Command Line

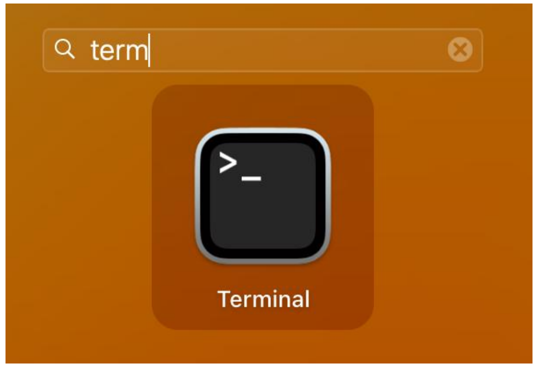
- On Windows, click Start menu, search for “Anaconda Prompt”, and click to open.

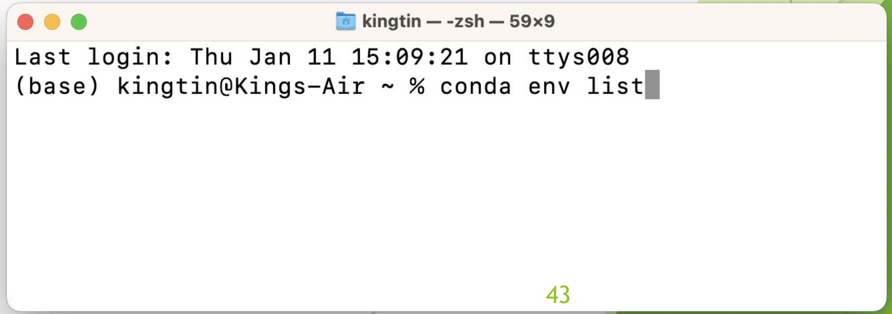

Anaconda3 (64-bit)
Anaconda Navigator (Anaconda3)
Anaconda Powershell Prompt (Ana...
Anaconda Prompt (Anaconda3)



```
(base) C:\Users\kingtin>conda list
```

- On macOS, press cmd + space, type “terminal”, hit Enter to the base environment.





```
Last login: Thu Jan 11 15:09:21 on ttys008
(base) kingtin@Kings-Air ~ % conda env list
```

Creating a Virtual Environment in Command Line

- Create a virtual environment:

```
conda create ––name <name> python=<version>
                 or –n <name>      (optional)
```

- For example, create an enviornment named py311env with python 3.11:

```
conda create ––name py311env python=3.11
```

- List all environments(s):

```
conda env list
```

Best practice: Give your virtual environment a meaningful name, e.g. dev (for development), prod (for production)

Activating a Virtual Environment

- Before you can start installing or using packages in your virtual environment, you'll need to *activate* it, which means putting the virtual environment-specific **python** and **pip** executables into your shell's PATH.

```
conda activate p311env
```

Your virtual environment's name

- The command prompt will change from (base) to (p311env).

Leaving a Virtual Environment

- To leave your virtual environment, simply run:

```
(py311env)...> conda deactivate
```

- To re-enter the virtual environment, just follow the same instructions above about activating a virtual environment.

- To remove a virtual environment,

Your virtual environment's name

```
conda env remove –n py311env
```

Using conda to Manage Python Packages

- Install packages:

```
conda install SomePackage              # latest version
conda install SomePackage=1.0.4        # specific version
conda install SomePackage'>=1.0.4'     # minimum version
```

e.g.

```
conda install matplotlib      # install a chart plotting tool
```

```
conda install numpy scipy     # install multiple packages
```

- Uninstall packages:

```
conda uninstall SomePackage
```

- Update packages:

```
conda update SomePackage
```

- List installed packages:

```
conda list
```

Using pip to Manage Python Packages

- Install packages:

```
pip install SomePackage             # latest version
pip install SomePackage==1.0.4      # specific version
pip install SomePackage'>=1.0.4'  # minimum version
```

e.g.

```
pip install matplotlib    # install a chart plotting tool
```

```
pip install numpy scipy   # install multiple packages
```

- Uninstall packages:

```
pip uninstall SomePackage
```

- Update packages:

```
pip install ––upgrade SomePackage
```

- List installed packages:

```
pip list
```

Recap Overall

- Create your virtual environment:

```
conda create –n env
```

- Activate it:

```
conda activate env
```

- Install a package in env:

```
(env) > pip install matplotlib
```

(for example)

- Bring up python for execution:

```
(env) > python
```

or

```
(env) > python script.py
```

- Leave the environment:

```
(env) > conda deactivate
```

The End