浙江理工大学
ZHEJIANG SCI-TECH UNIVERSITY

1. Introduction

《Python程序设计》 / 教师：罗志一

School of Computer Science and Technology
计算机科学与技术学院

概览

- 开发工具
- 环境配置
- 动手试试

工具1: IDLE

- 下载并安装Python
  - https://www.python.org/downloads/

Want to help test development versions of Python 3.12? Prereleases, Docker images

Active Python Releases

For more information visit the Python Developer's Guide.

Python version	Maintenance status	First released	End of support	Release schedule
3.12	prerelease	2023-10-02 (planned)	2028-10	PEP 693
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	security	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.11.5	Aug. 24, 2023	Download	Release Notes
Python 3.10.13	Aug. 24, 2023	Download	Release Notes
Python 3.9.18	Aug. 24, 2023	Download	Release Notes
Python 3.8.18	Aug. 24, 2023	Download	Release Notes
Python 3.10.12	June 6, 2023	Download	Release Notes
Python 3.11.4	June 6, 2023	Download	Release Notes

View older releases

Want to help test development versions of Python 3.12? Prereleases, Docker images

Active Python Releases

For more information visit the Python Developer's Guide.

Python version	Maintenance status	First released	End of support	Release schedule
3.12	prerelease	2023-10-02 (planned)	2028-10	PEP 693
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	security	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.11.5	Aug. 24, 2023	Download	Release Notes
Python 3.10.13	Aug. 24, 2023	Download	Release Notes
Python 3.9.18	Aug. 24, 2023	Download	Release Notes
Python 3.8.18	Aug. 24, 2023	Download	Release Notes
Python 3.10.12	June 6, 2023	Download	Release Notes
Python 3.11.4	June 6, 2023	Download	Release Notes

View older releases

工具1: IDLE

- 安装Python
  - Windows x86 MSI installer (32-bit)
  - Windows x86-64 MSI installer (64-bit)

Files

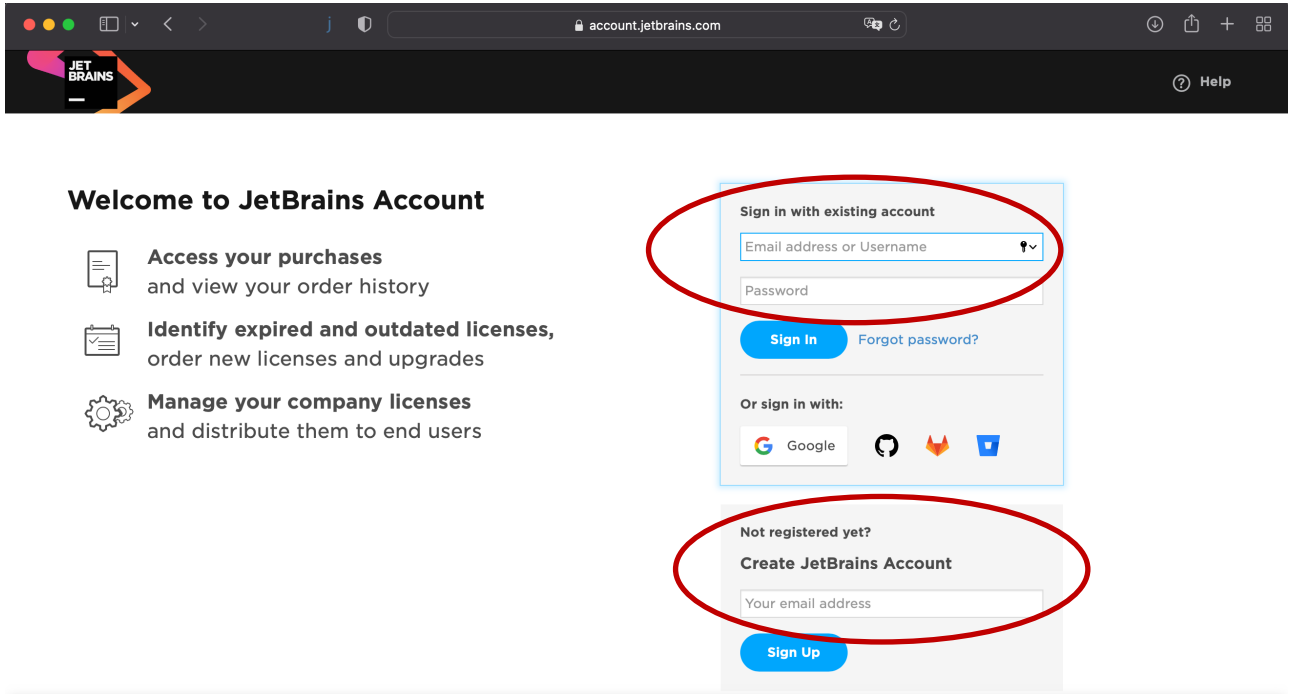
Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		b628f21aae5e2c3006a12380905bb640	26571003	SIG	.sigstore
XZ compressed source tarball	Source release		393856f1b7713aa8bba4b642ab9985d3	20053580	SIG	.sigstore
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	7a24f8b4eeca34899b7d75caaec3bc73	44239554	SIG	.sigstore
Windows embeddable package (32-bit)	Windows		add17856887d34c04a9cfd6c051c4bea	10053367	SIG	.sigstore
Windows embeddable package (64-bit)	Windows		c5e83dc45630df2236720a18170bf941	11170359	SIG	.sigstore
Windows embeddable package (ARM64)	Windows		8fc7d74daf27882f2a32a1b10c3a3a2c	10428395	SIG	.sigstore
Windows installer (32 -bit)	Windows		ac8e48a759a6222ce9332691568fe67a	24662424	SIG	.sigstore
Windows installer (64-bit)	Windows	Recommended	3afd5b0ba1549f5b9a90c1e3aa8f041e	25932664	SIG	.sigstore
Windows installer (ARM64)	Windows	Experimental	cd2bfd6bb39a6c84dbf9d1615b9f53b5	25197192	SIG	.sigstore

工具2: Pycharm

- 下载并安装Pycharm
  - https://www.jetbrains.com/zh-cn/pycharm/download/

工具2: Pycharm

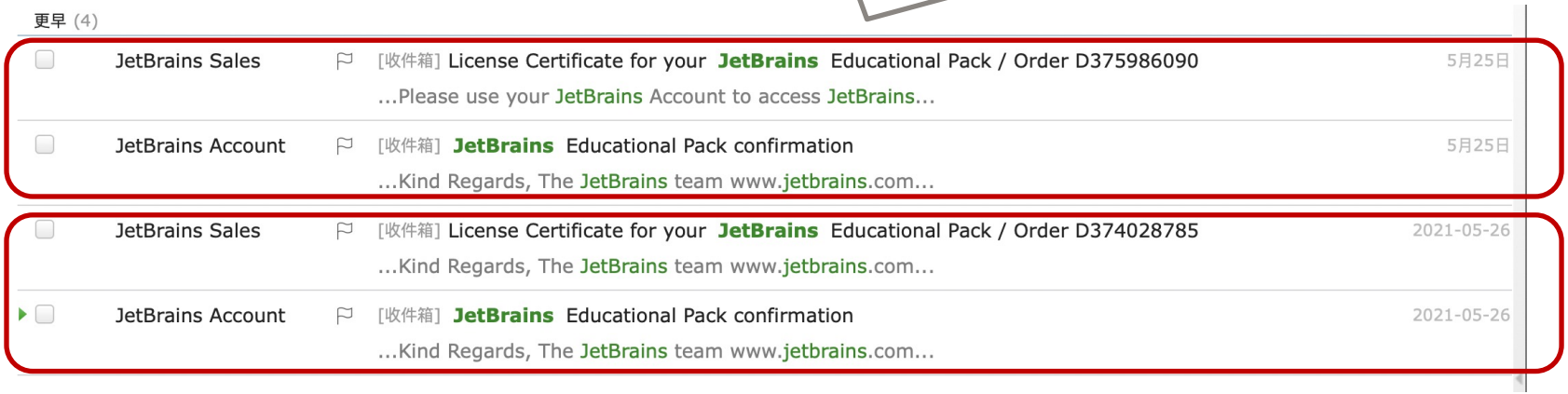
- 用学校邮箱注册JetBrains账户并登录



工具2: Pycharm

- 申请免费的学生或教师license

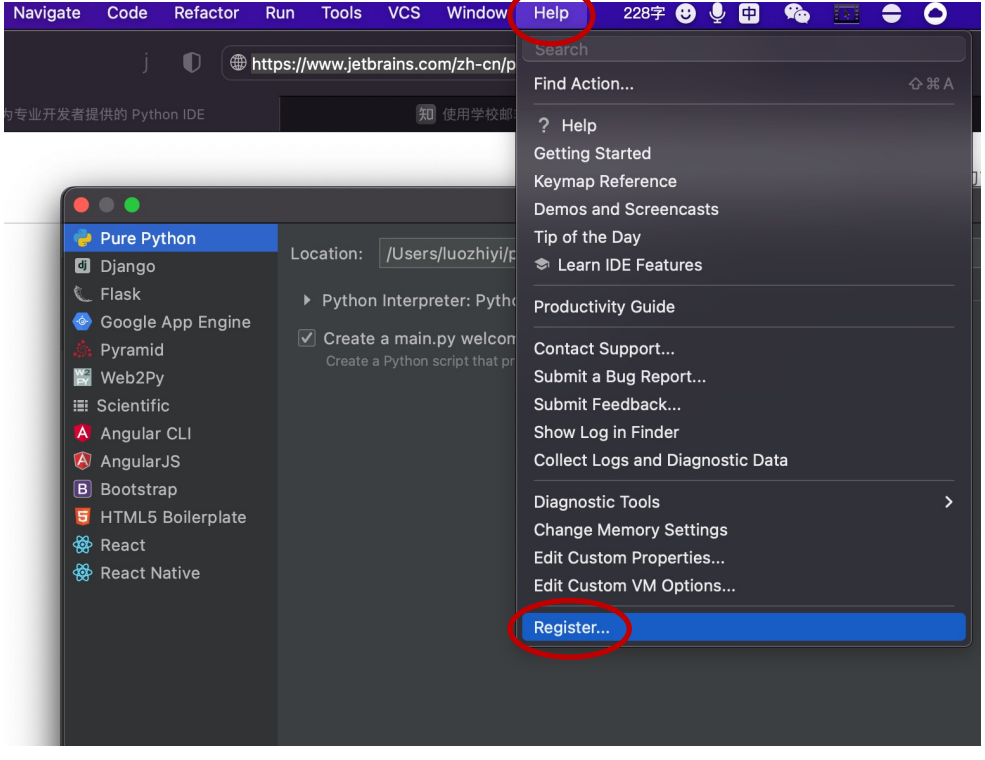
2. 第二年license到期，申请新的license。



1. 首次申请license，有效期一年。

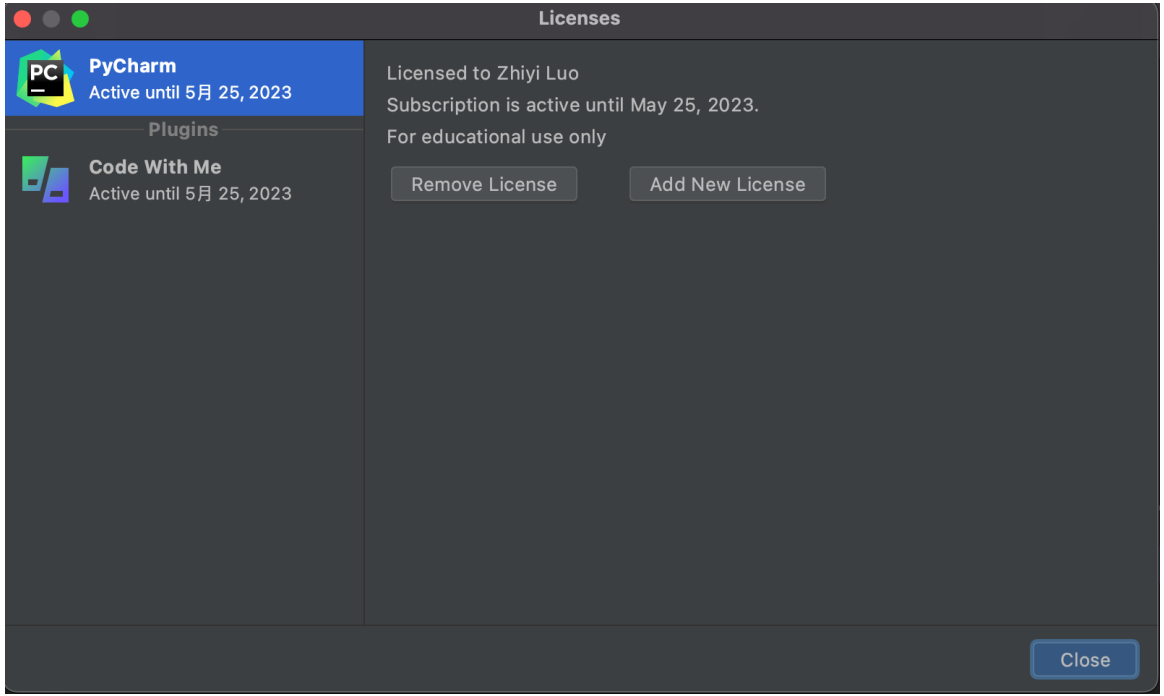
工具2: Pycharm

- 用申请到的license激活Pycharm



工具2: Pycharm

- 用申请到的license激活Pycharm



环境配置: Anaconda

- 下载Anaconda Installer
  - 官方下载链接: https://www.anaconda.com/products/distribution

Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

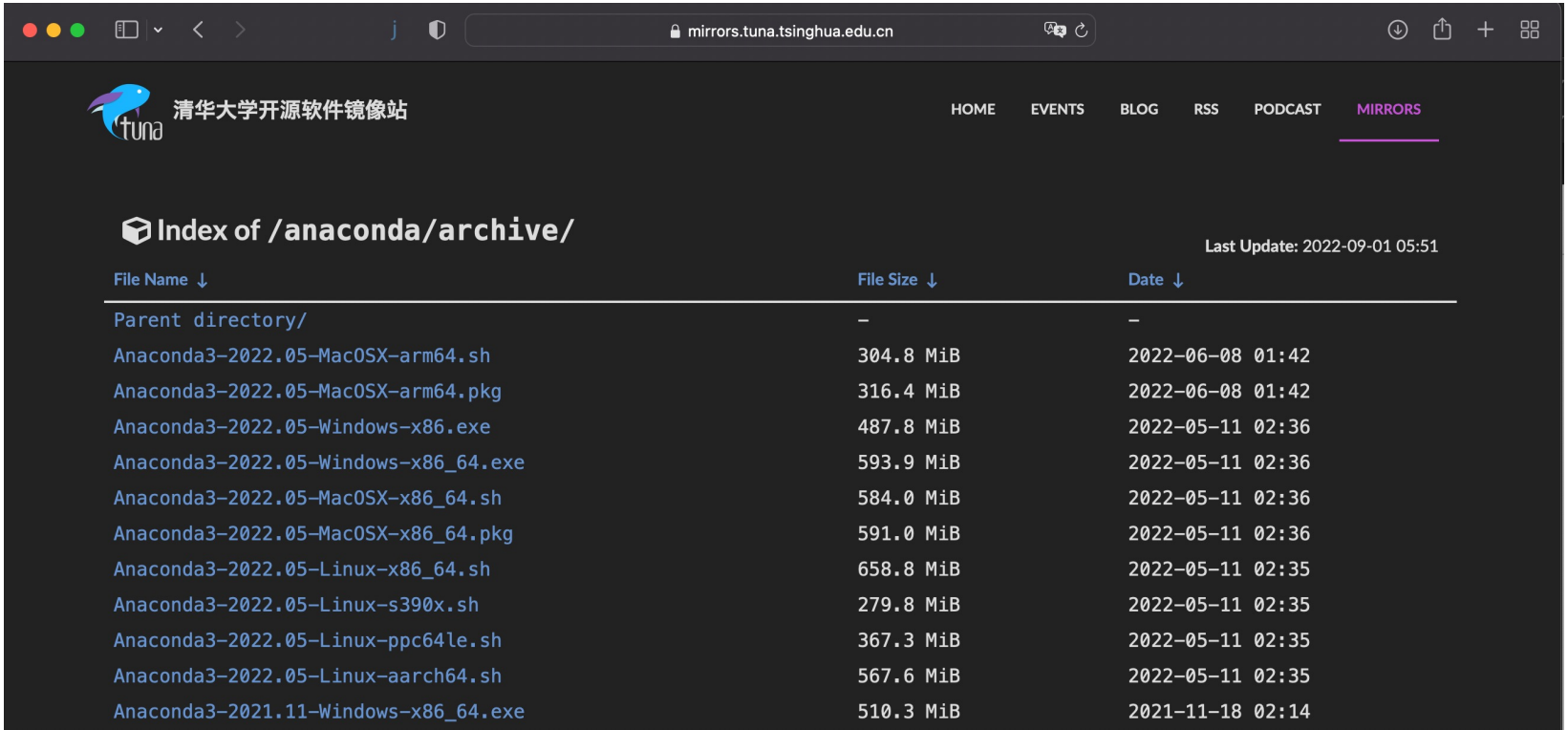
浏览器会根据当前使用的操作系统为你推荐安装文件，点击“Download”下载即可。

环境配置: Anaconda

- 从清华大学镜像下载Anaconda
  - 如果安装包下载速度过慢，可以使用Anaconda国内源（例如：清华大学镜像）进行下载。
  - Anaconda清华源的下载列表链接为：https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/

环境配置: Anaconda

- 从清华大学镜像下载Anaconda



环境配置: Anaconda

- 从清华大学镜像下载Anaconda
  - 下载完成之后，双击安装包进行安装。

环境配置: Anaconda

- 从清华大学镜像下载Anaconda
  - 使用帮助文档在：
    https://mirrors.tuna.tsinghua.edu.cn/help/anaconda/
    - 在用户目录下配置.condarc文件
    - macOS和Linux用户可直接在用户目录/home/username下创建 .condarc文件，并写入配置内容。
    - Windows用户无法直接创建名为.condarc的文件，可先执行 conda config –set show_channel_urls yes命令生成该文件，再写入配置内容。

环境配置: Anaconda

- 创建Python虚拟环境
  - conda create -n <环境名称>
  - 例如，可以使用如下命令将新建环境命名为pycourse，并指定该环境的Python版本为3.9: **conda create -n pycourse python=3.9**
- 查看Anaconda中的所有虚拟环境
  - conda info --envs
- 激活指定环境
  - conda activate <环境名称>
- 退出当前激活的环境
  - conda deactivate

如果.condarc中配置了清华源
此处无需翻墙即可成功。

动手试试!

运行 Python

- 使用交互解释器

```
06:44:29 with luozhiyi in ~ via ©base
➜ python
Python 3.8.5 (default, Sep  4 2020, 02:22:02)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

运行 Python

- 使用Python文件

创建并编写一个Python程序文件，并运行：

1. 打开文件编辑器
2. 键入文本 `print(61)`
3. 保存并将文件命名为 61.py.
4. 打开终端窗口
5. 输入以下命令运行python程序：

```
$ python 61.py
```

你将会看到一行输出：61

将Python用作计算器

第一个例子: Hello World!

```
print(‘Hello World’)
```

第一个例子: Hello World!

```
# Define a main() function that prints a little greeting.
def main():
    print('Hello World')

main()
```

第一个例子: Hello World!

```
# Define a main() function that prints a little greeting.
def main():
    print('Hello World')

main()
```

缩进：四个空格

第一个例子: Hello World!

```
# Define a main() function that prints a little greeting.
def main():
    print(‘Hello World’)

# This is the standard boilerplate that calls the main() function.
if __name__ == '__main__':
    main()
```

第一个例子: Hello World!

```
import sys

# Define a main() function that prints a little greeting.
def main():
    # Get the name from the command line, using 'World' as a fallback.
    if len(sys.argv) >= 2:
        name = sys.argv[1]
    else:
        name = 'World'
    print('Hello', name)

# This is the standard boilerplate that calls the main() function.
if __name__ == '__main__':
    main()
```

第一个例子: Hello World!

```
import sys

# Define a main() function that prints a little greeting.
def main():
    # Get the name from the command line, using 'World' as a fallback.
    if len(sys.argv) >= 2:
        name = sys.argv[1]
    else:
        name = 'World'
    print('Hello', name)

# This is the standard boilerplate that calls the main() function.
if __name__ == '__main__':
    main()
```

此处应有多少空格?