

实验1：语言规范与工具实验



案例1. 安装python学习相关软件

任务

通过百度下载并安装软件下列软件。

1. python
2. git
3. jupyter
4. opencv, numpy
5. VSCode

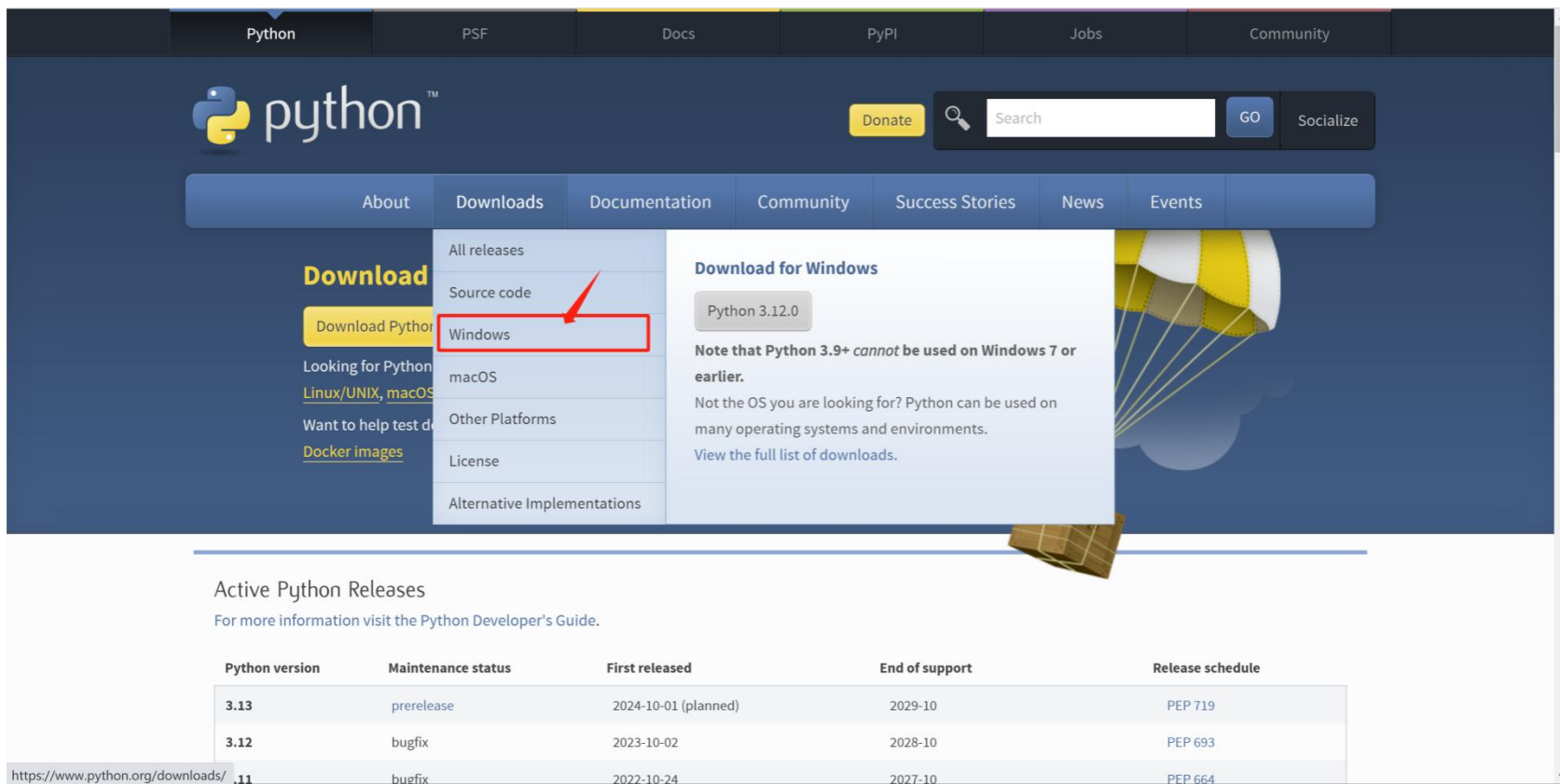


实验目标与案例

实验名称	实验目标	实验案例
实验1：语言规范与工具实验	<ol style="list-style-type: none">1. 能使用如下方式编写，运行Python代码（非程序）<ol style="list-style-type: none">1.1. vscode编辑器；1.2. python交互式编程；1.3. jupyter编写；2. 能使用git上传代码；	<p>案例1. 安装python学习相关软件（本机，使用平台的不用安装）</p> <ol style="list-style-type: none">1. 安装python2. 安装git3. 安装jupyter4. 安装opencv, numpy5. 安装VSCode <p>案例2. 编译编写一个程序，打印一个表格。</p> <ol style="list-style-type: none">1. VSCode源代码版2. Jupyter版本 <p>案例3. 使用git上传代码到服务器，并使用markdown（说明作业完成情况）。</p>

案例指导：1. python下- 载官网

1. 官网：<https://www.python.org/>



选择合适的下载版本：

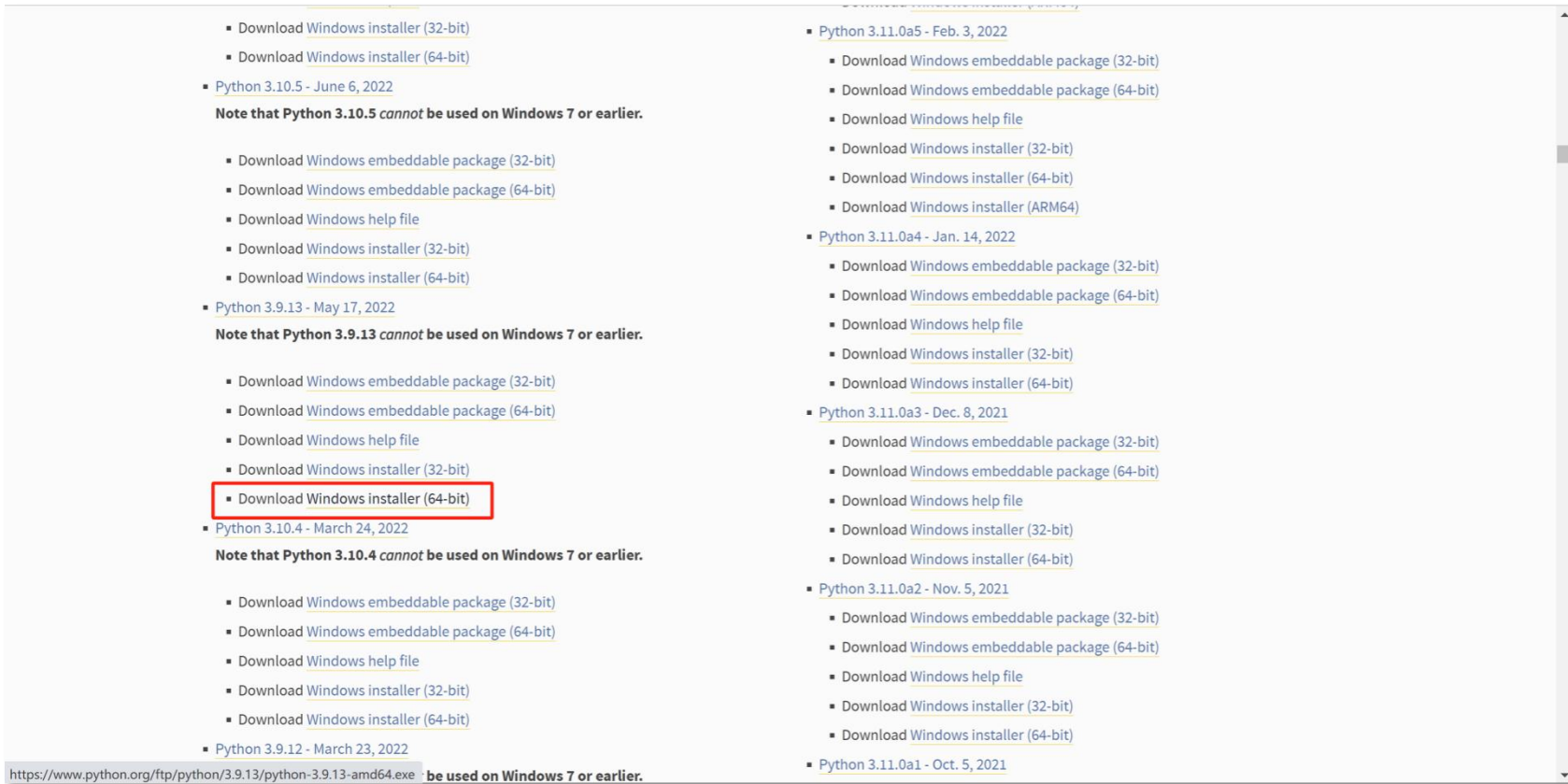
1. 操作系统Window
2. 版本

不建议下载最新版本，建议下载3.9版本比较稳定，大部分软件都升级支持Python3.9



案例指导：1. python-下载页面

1. 选择合适版本，并且有下载文件的版本。



选择合适的位数：

1. 64位

注：

embeddable版本是解压安装版本，需要手工配置环境变量。

下载慢可以使用迅雷工具。

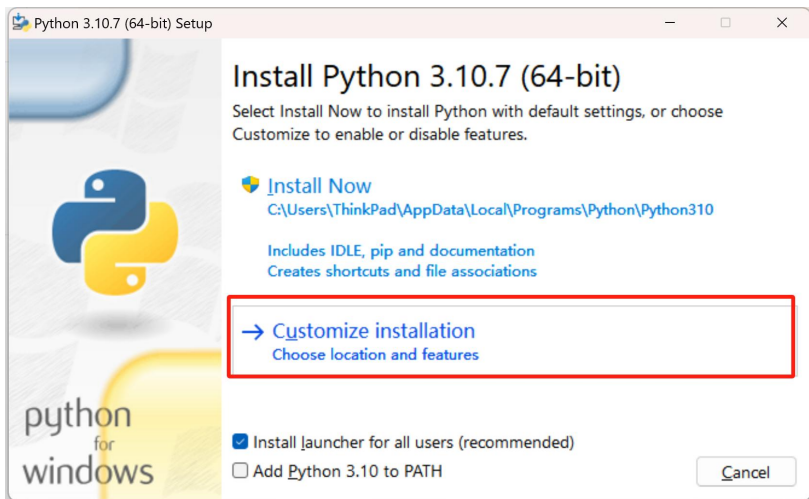
我们选择3.9.13版本。



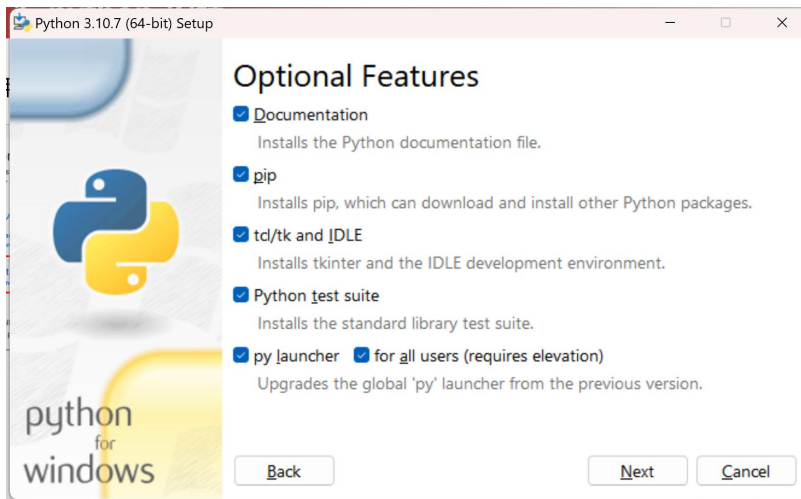
没有下载文件的不要选择。

案例指导：1. python-安装

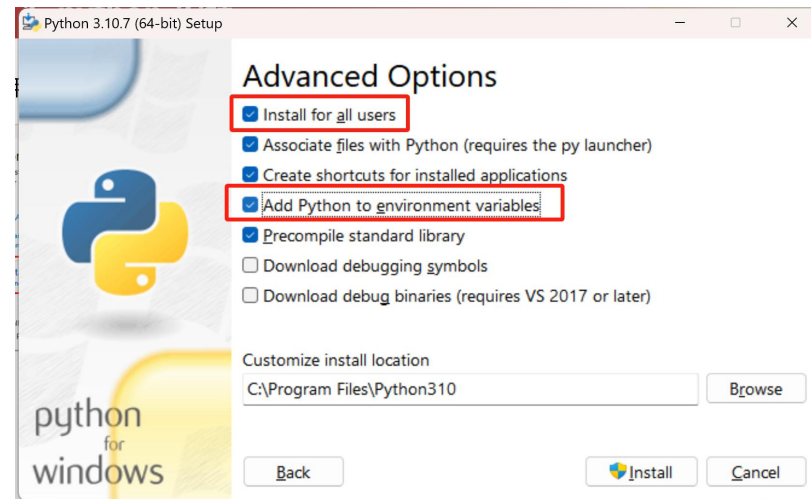
python安装是使用安装包安装，一般比较简单，但是在安装的时候有几个选项需要注意：



选择定制安装



直接点击next



勾选：Install for all users
勾选：Add Python to environment variables
最后点击install按钮安装

案例指导：1. python-验证

验证是否能找到安装的程序：python与pip

```
命令提示符
C:\Users\ThinkPad>where python
C:\Program Files\Python39\python.exe
C:\Users\ThinkPad\AppData\Local\Microsoft\WindowsApps\python.exe

C:\Users\ThinkPad>where pip
C:\Program Files\Python39\Scripts\pip.exe
C:\Users\ThinkPad\AppData\Roaming\Python\Python39\Scripts\pip.exe

C:\Users\ThinkPad>
```

如果找不到安装的python与pip程序，请设置PATH环境变量。

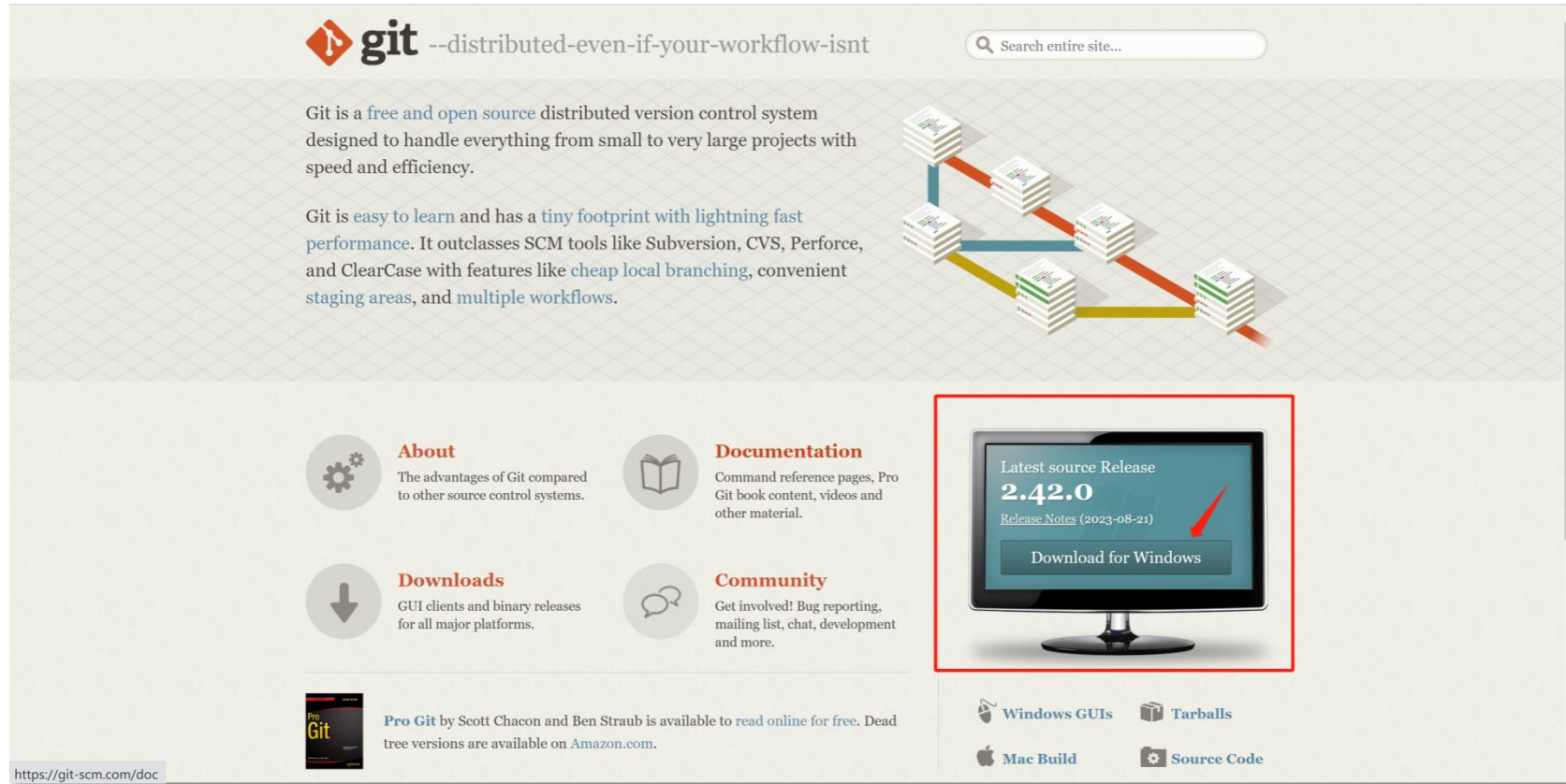
通过执行python与pip来验证python是否安装成功。

```
命令提示符 - python
C:\Users\ThinkPad>python
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```



案例指导：2. git官网

官网：<https://git-scm.com/>



The screenshot shows the Git website homepage. At the top left is the Git logo with the tagline "--distributed-even-if-your-workflow-isnt". A search bar is located at the top right. The main content area features a diagram of a distributed version control system with multiple nodes and branches. Below this, there are four main sections: "About", "Documentation", "Downloads", and "Community". The "Downloads" section is highlighted with a red box and contains a "Latest source Release 2.42.0" section with a "Download for Windows" button. At the bottom, there are links for "Windows GUIs", "Mac Build", "Tarballs", and "Source Code".

git --distributed-even-if-your-workflow-isnt

Search entire site...

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.42.0
Release Notes (2023-08-21)
Download for Windows

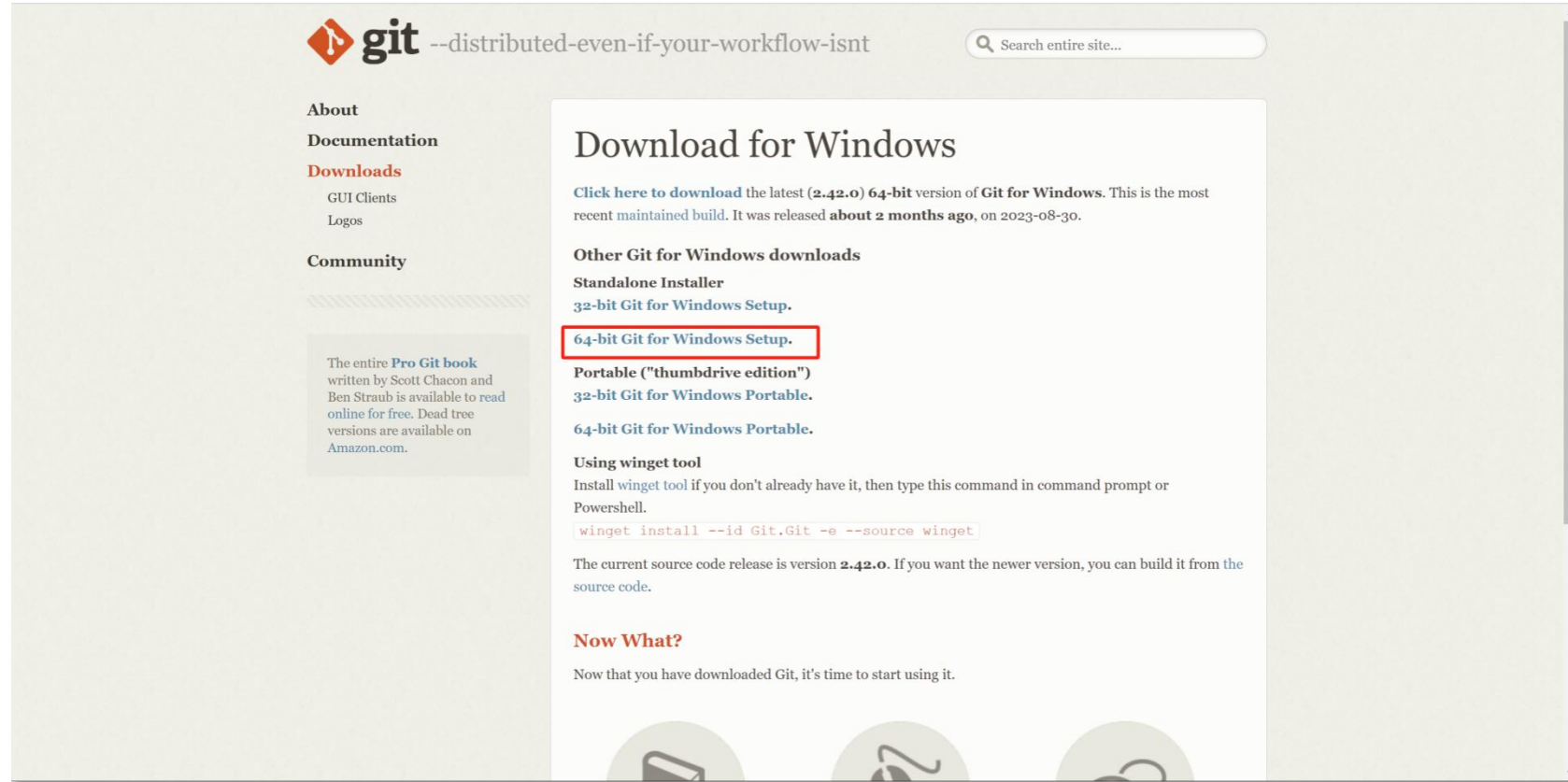
Windows GUIs | Mac Build | Tarballs | Source Code

<https://git-scm.com/doc>

下载最新本版本。

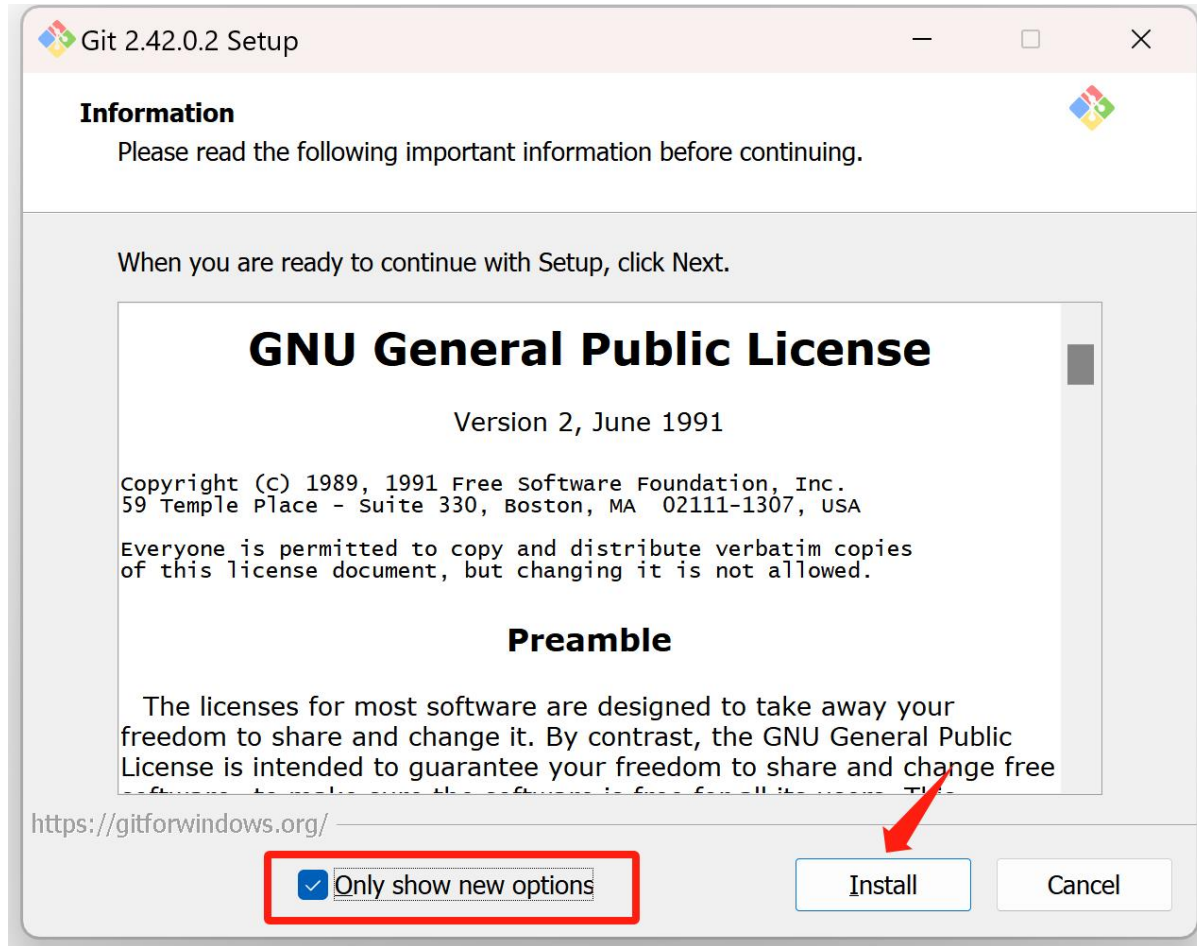


案例指导：2. git下载



下载标准安装版本。
速度慢，请使用迅雷。

案例指导：2. git安装



勾选：Only show new options。
然后，点击install直接安装。

案例指导：2. git验证

```
命令提示符
C:\Users\ThinkPad>where git
C:\Program Files\Git\cmd\git.exe

C:\Users\ThinkPad>git --help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          [--config-env=<name>=<envvar>] <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

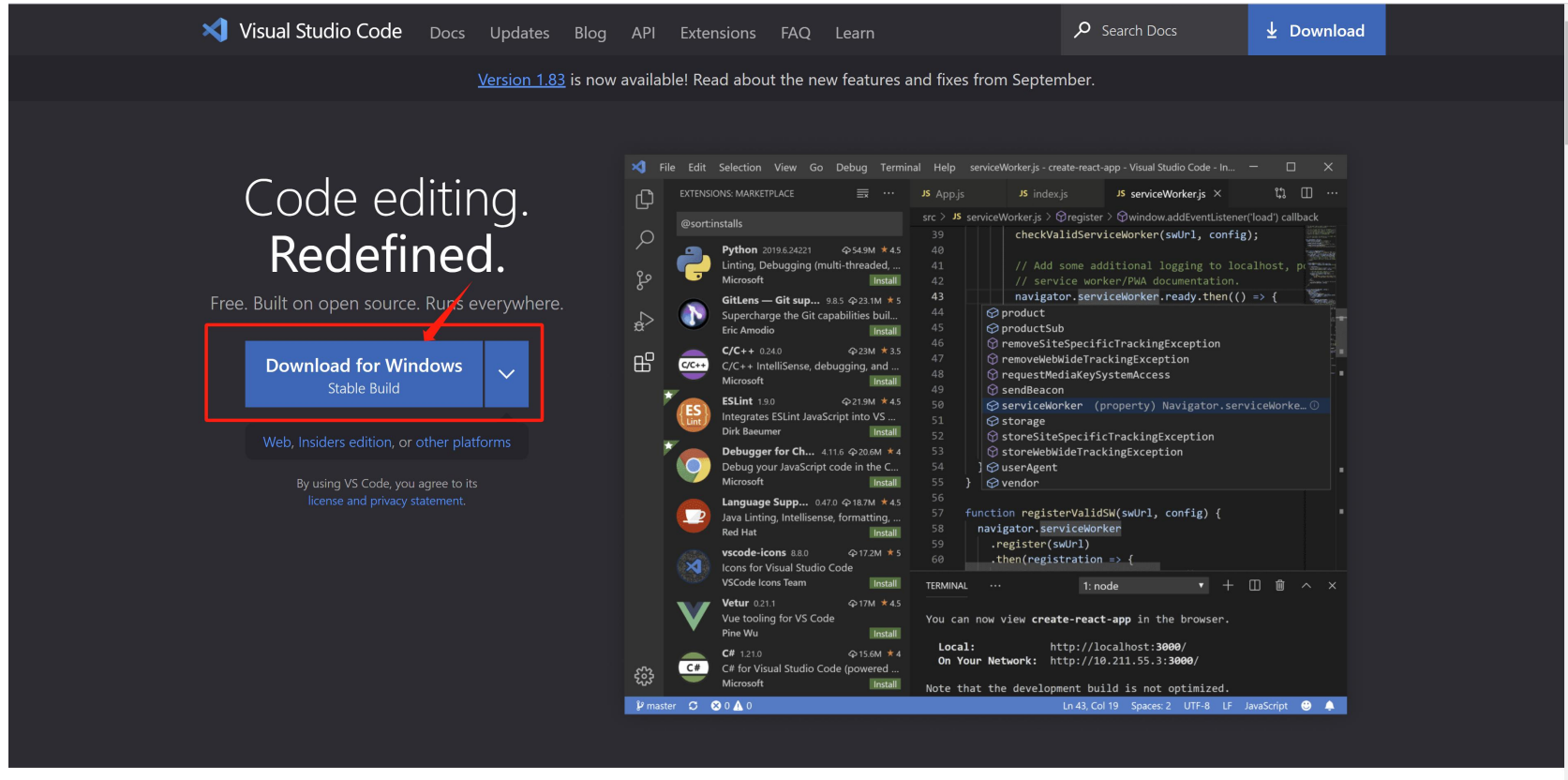
examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  diff      Show changes between commits, commit and working tree, etc
  grep      Print lines matching a pattern
  log       Show commit logs
  show      Show various types of objects
  status    Show the working tree status
```

1. 使用where验证是否能找到安装程序，找不到请设置PATH环境变量。

2. 直接执行验证。

案例指导：3. VSCode官网

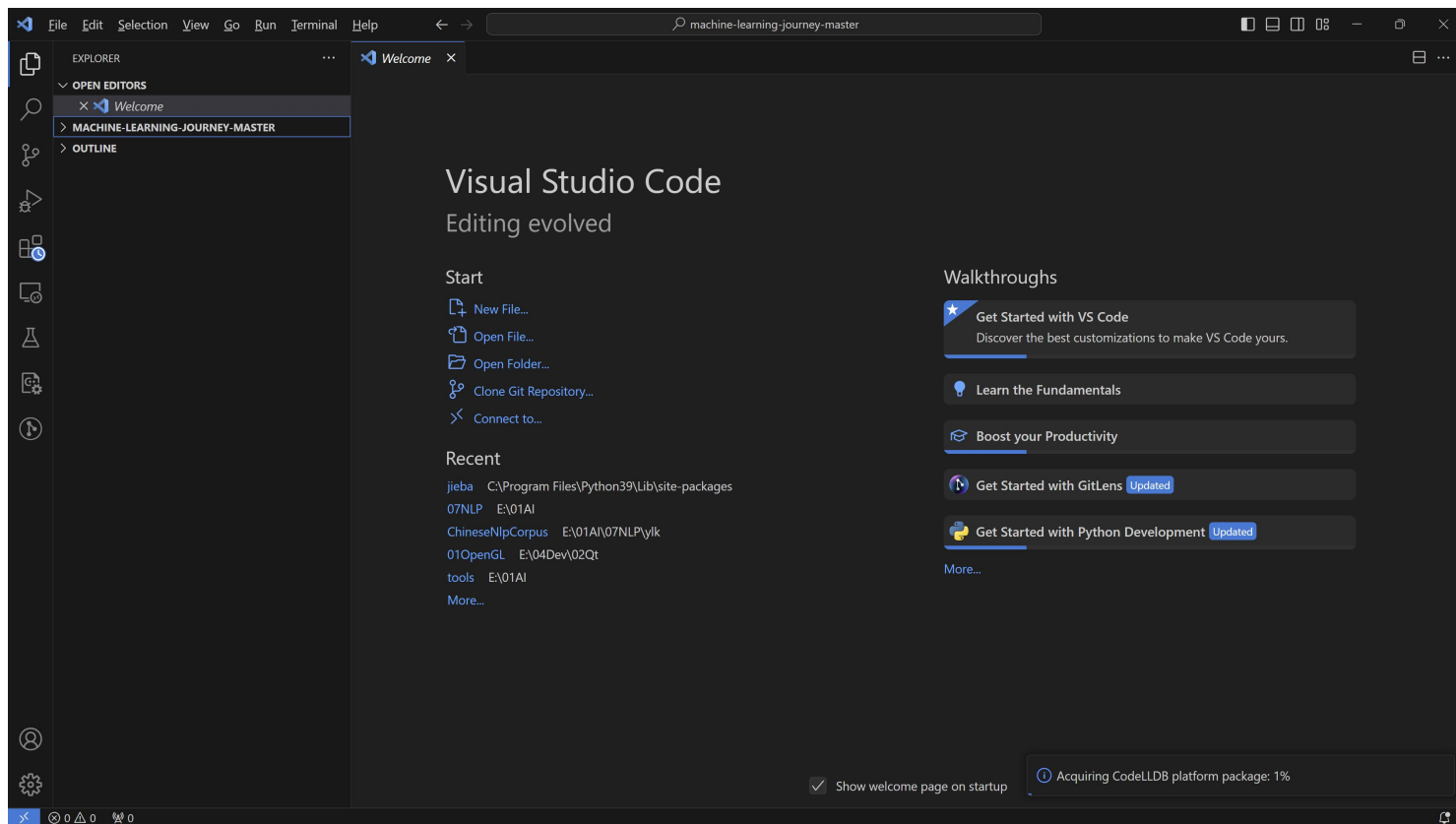
<https://code.visualstudio.com/>



点击可以直接下载

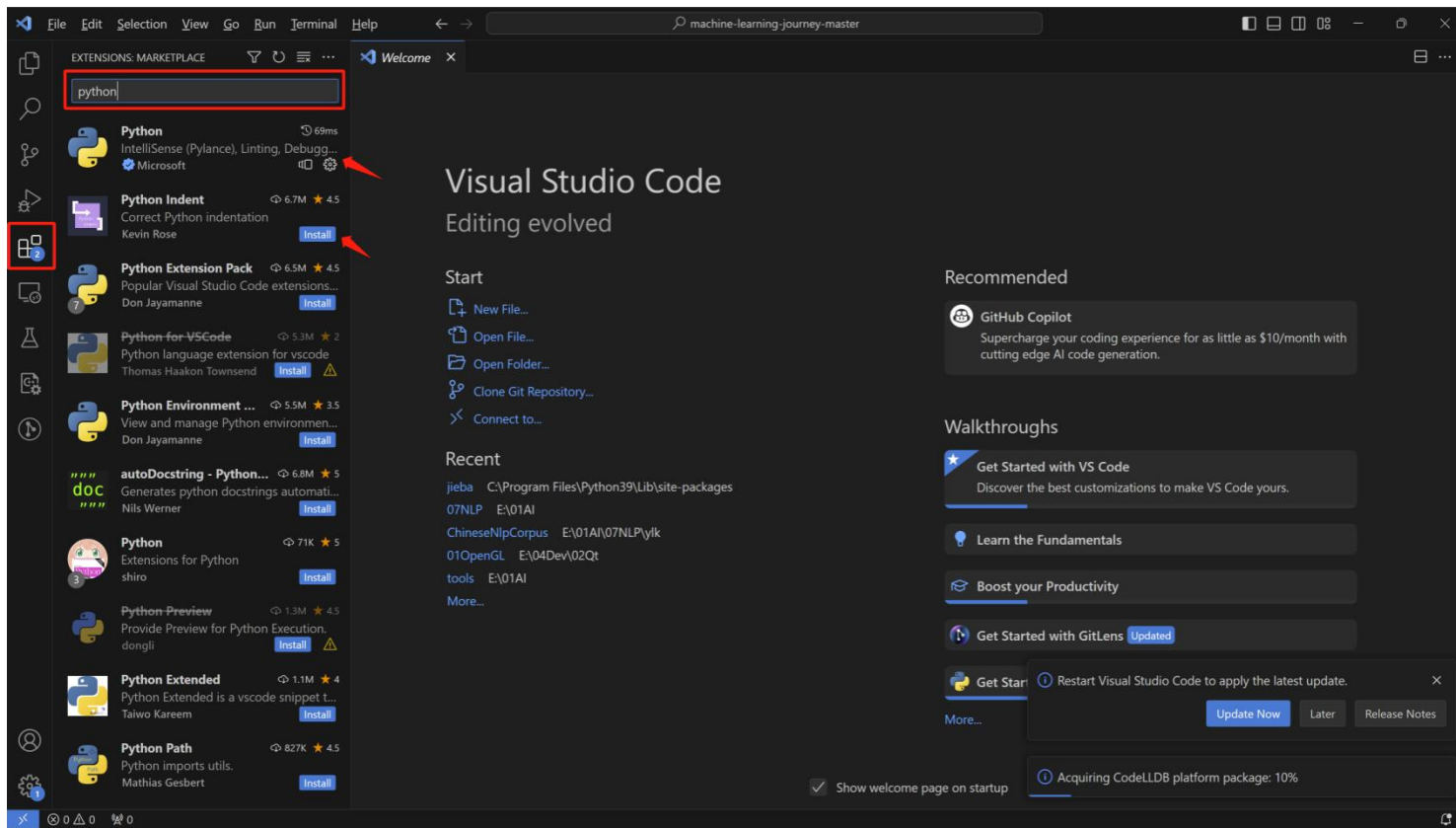
案例指导：3. VSCode安装与验证

因为使用安装包。按照向导式指示进行安装即可，安装完毕，在开始菜单中找到安装的VSCode直接运行。启动的VSCode如下。



案例指导：3. VSCode配置Python语法

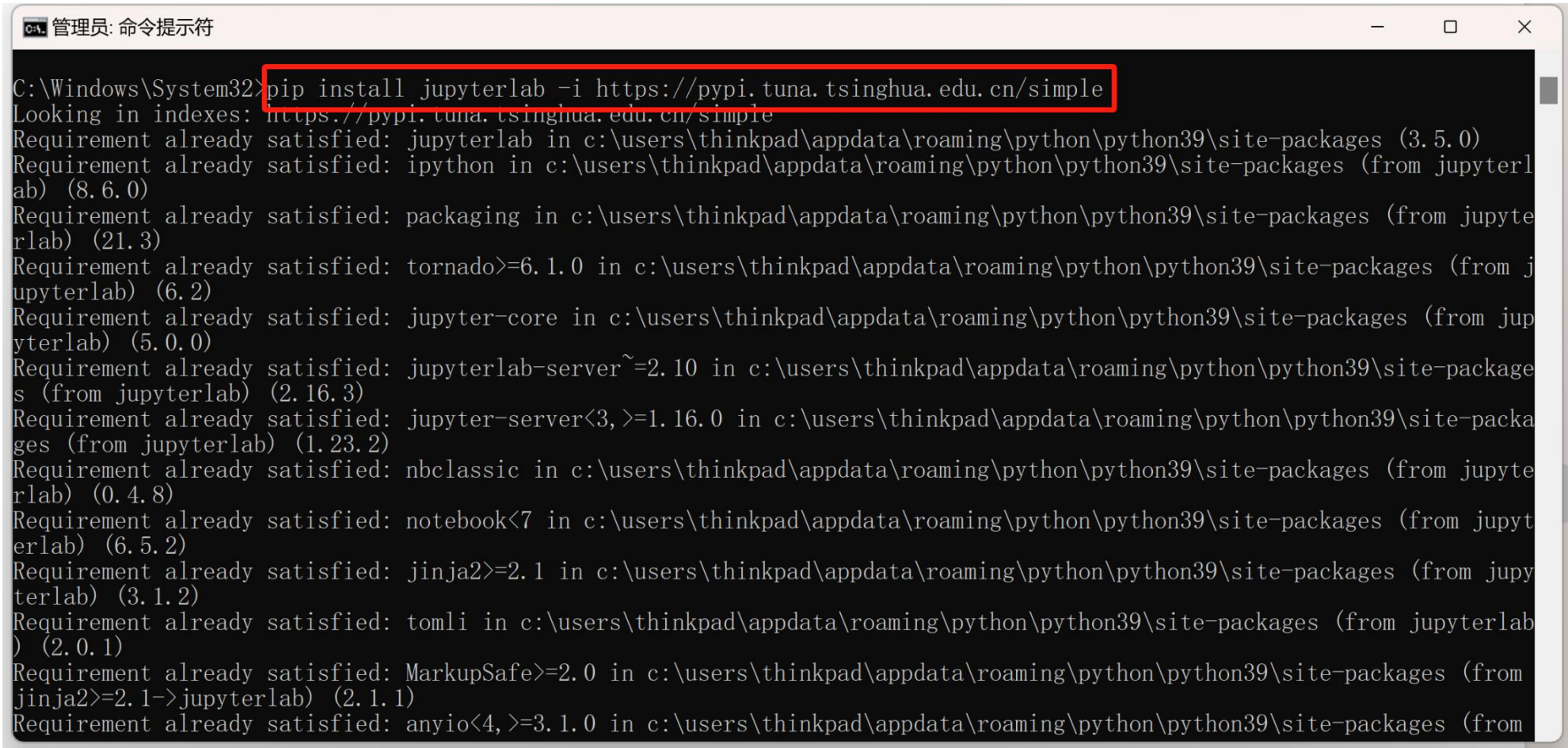
点击模块库图标，搜索python可以安装自己感兴趣的VSCode的python扩展，当然也可以安装C/C++扩展。



案例指导：4. Jupyter安装

安装指令： `pip install jupyterlab -i https://pypi.tuna.tsinghua.edu.cn/simple`

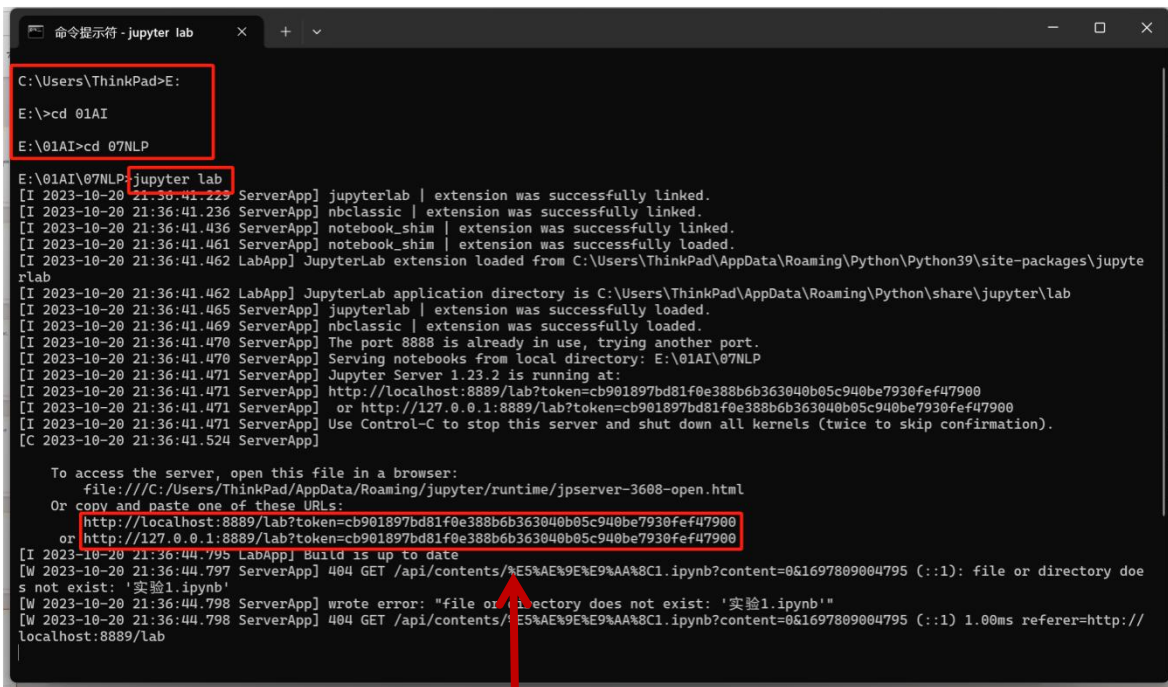
注： 使用**管理员身份**打开命令提示符终端。



```
管理员: 命令提示符
C:\Windows\System32>pip install jupyterlab -i https://pypi.tuna.tsinghua.edu.cn/simple
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Requirement already satisfied: jupyterlab in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (3.5.0)
Requirement already satisfied: ipython in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (8.6.0)
Requirement already satisfied: packaging in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (21.3)
Requirement already satisfied: tornado>=6.1.0 in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (6.2)
Requirement already satisfied: jupyter-core in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (5.0.0)
Requirement already satisfied: jupyterlab-server~=2.10 in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (2.16.3)
Requirement already satisfied: jupyter-server<3,>=1.16.0 in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (1.23.2)
Requirement already satisfied: nbclassic in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (0.4.8)
Requirement already satisfied: notebook<7 in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (6.5.2)
Requirement already satisfied: jinja2>=2.1 in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (3.1.2)
Requirement already satisfied: tomli in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jupyterlab) (2.0.1)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from jinja2>=2.1->jupyterlab) (2.1.1)
Requirement already satisfied: anyio<4,>=3.1.0 in c:\users\thinkpad\appdata\roaming\python\python39\site-packages (from
```

案例指导：4. Jupyter启动

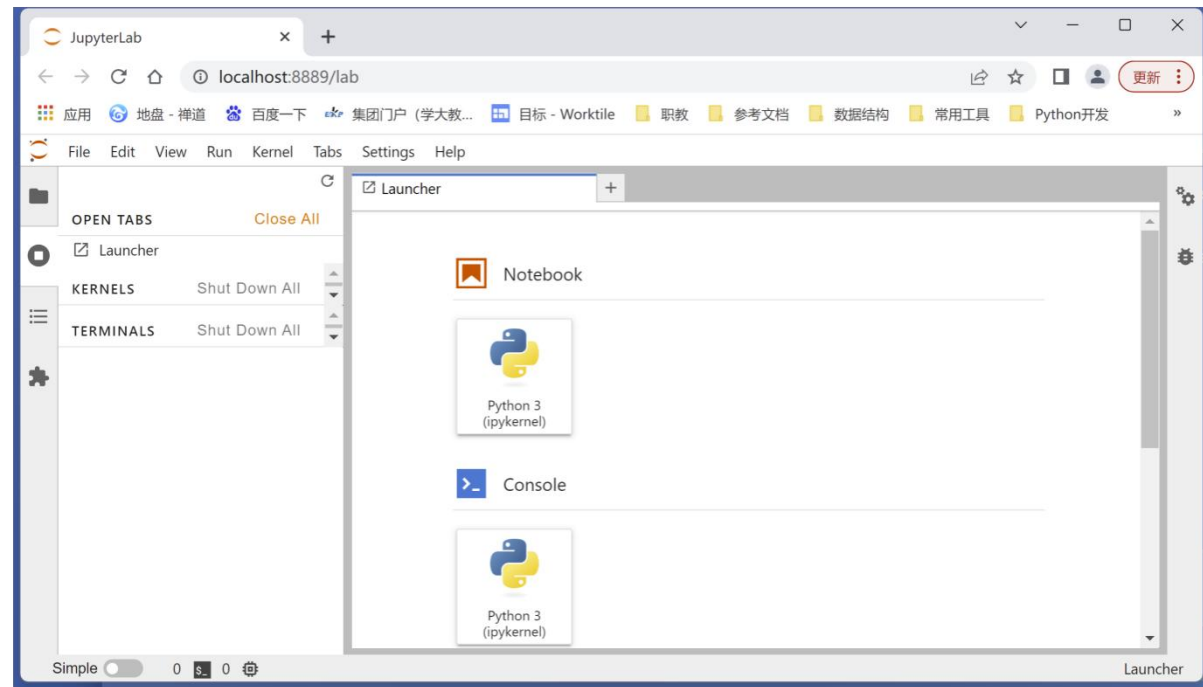
1. 切换到自己的工作路径。
2. 使用命令：`jupyter lab`启动。
3. 使用浏览器访问（默认会自动启动浏览器）



```
C:\Users\ThinkPad>E:
E:>cd 01AI
E:\01AI>cd 07NLP
E:\01AI\07NLP>jupyter lab
[I 2023-10-20 21:36:41.229 ServerApp] jupyterlab | extension was successfully linked.
[I 2023-10-20 21:36:41.236 ServerApp] nbclassic | extension was successfully linked.
[I 2023-10-20 21:36:41.436 ServerApp] notebook_shim | extension was successfully linked.
[I 2023-10-20 21:36:41.461 ServerApp] notebook_shim | extension was successfully loaded.
[I 2023-10-20 21:36:41.462 LabApp] JupyterLab extension loaded from C:\Users\ThinkPad\AppData\Roaming\Python\Python39\site-packages\jupyterlab
[I 2023-10-20 21:36:41.462 LabApp] JupyterLab application directory is C:\Users\ThinkPad\AppData\Roaming\Python\share\jupyterlab
[I 2023-10-20 21:36:41.465 ServerApp] jupyterlab | extension was successfully loaded.
[I 2023-10-20 21:36:41.469 ServerApp] nbclassic | extension was successfully loaded.
[I 2023-10-20 21:36:41.470 ServerApp] The port 8888 is already in use, trying another port.
[I 2023-10-20 21:36:41.470 ServerApp] Serving notebooks from local directory: E:\01AI\07NLP
[I 2023-10-20 21:36:41.471 ServerApp] Jupyter Server 1.23.2 is running at:
[I 2023-10-20 21:36:41.471 ServerApp] http://localhost:8889/lab?token=cb901897bd81f0e388b6b363040b05c940be7930fef47900
[I 2023-10-20 21:36:41.471 ServerApp] or http://127.0.0.1:8889/lab?token=cb901897bd81f0e388b6b363040b05c940be7930fef47900
[I 2023-10-20 21:36:41.471 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2023-10-20 21:36:41.524 ServerApp]

To access the server, open this file in a browser:
file:///C:/Users/ThinkPad/AppData/Roaming/jupyter/runtime/jpservice-3608-open.html
Or copy and paste one of these URLs:
http://localhost:8889/lab?token=cb901897bd81f0e388b6b363040b05c940be7930fef47900
or http://127.0.0.1:8889/lab?token=cb901897bd81f0e388b6b363040b05c940be7930fef47900
[I 2023-10-20 21:36:44.795 LabApp] Build is up to date
[W 2023-10-20 21:36:44.797 ServerApp] 404 GET /api/contents/%E5%AE%9E%9%AA%8C1.ipynb?content=061697809004795 (::1): file or directory does not exist: '实验1.ipynb'
[W 2023-10-20 21:36:44.798 ServerApp] wrote error: "file or directory does not exist: '实验1.ipynb'"
[W 2023-10-20 21:36:44.798 ServerApp] 404 GET /api/contents/%E5%AE%9E%9%AA%8C1.ipynb?content=061697809004795 (::1) 1.00ms referer=http://localhost:8889/lab
```

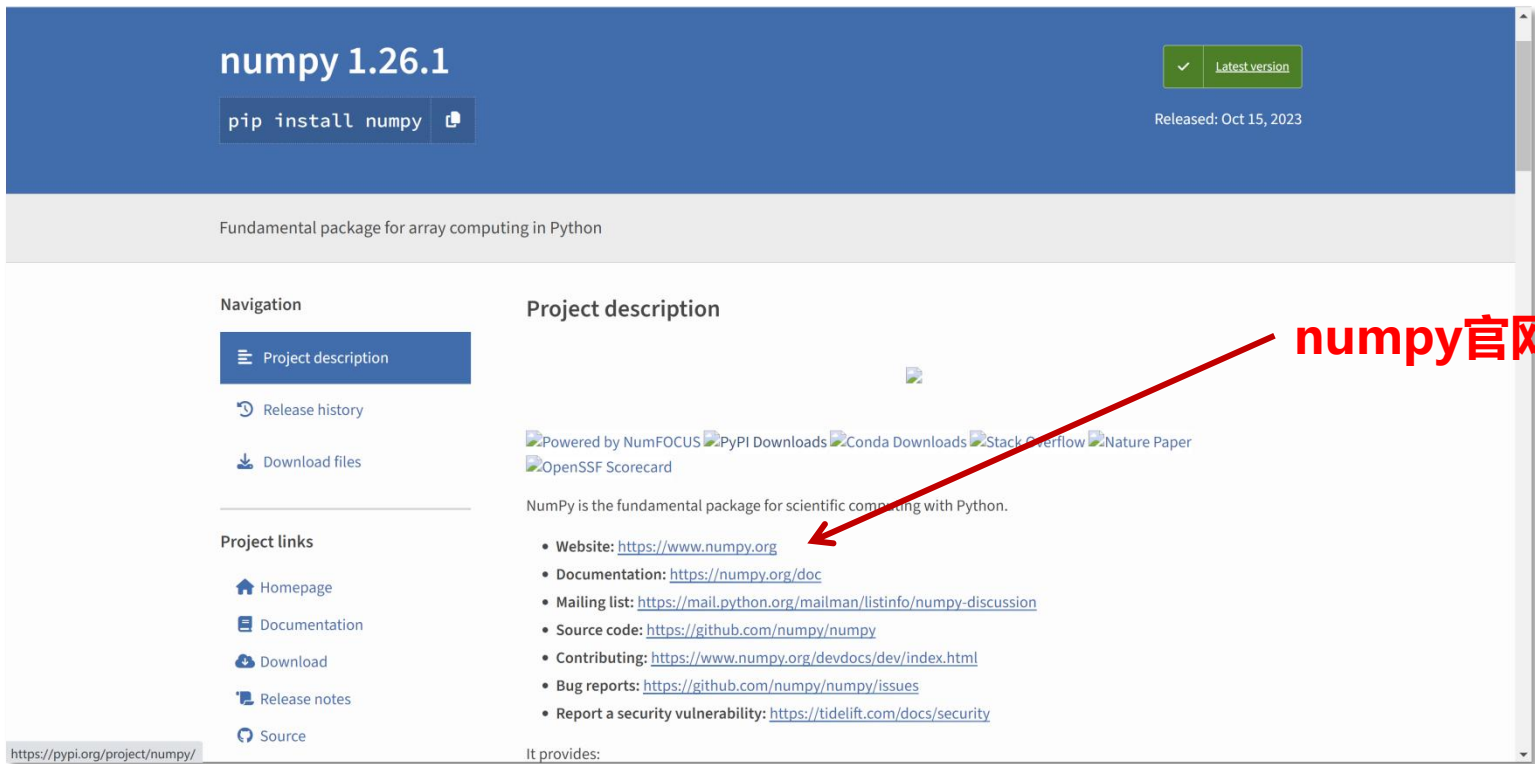
访问路径



使用浏览器访问

案例指导：4. numpy安装

pip安装官网：<https://pypi.org/project/numpy/>



numpy官网：<https://numpy.org/>

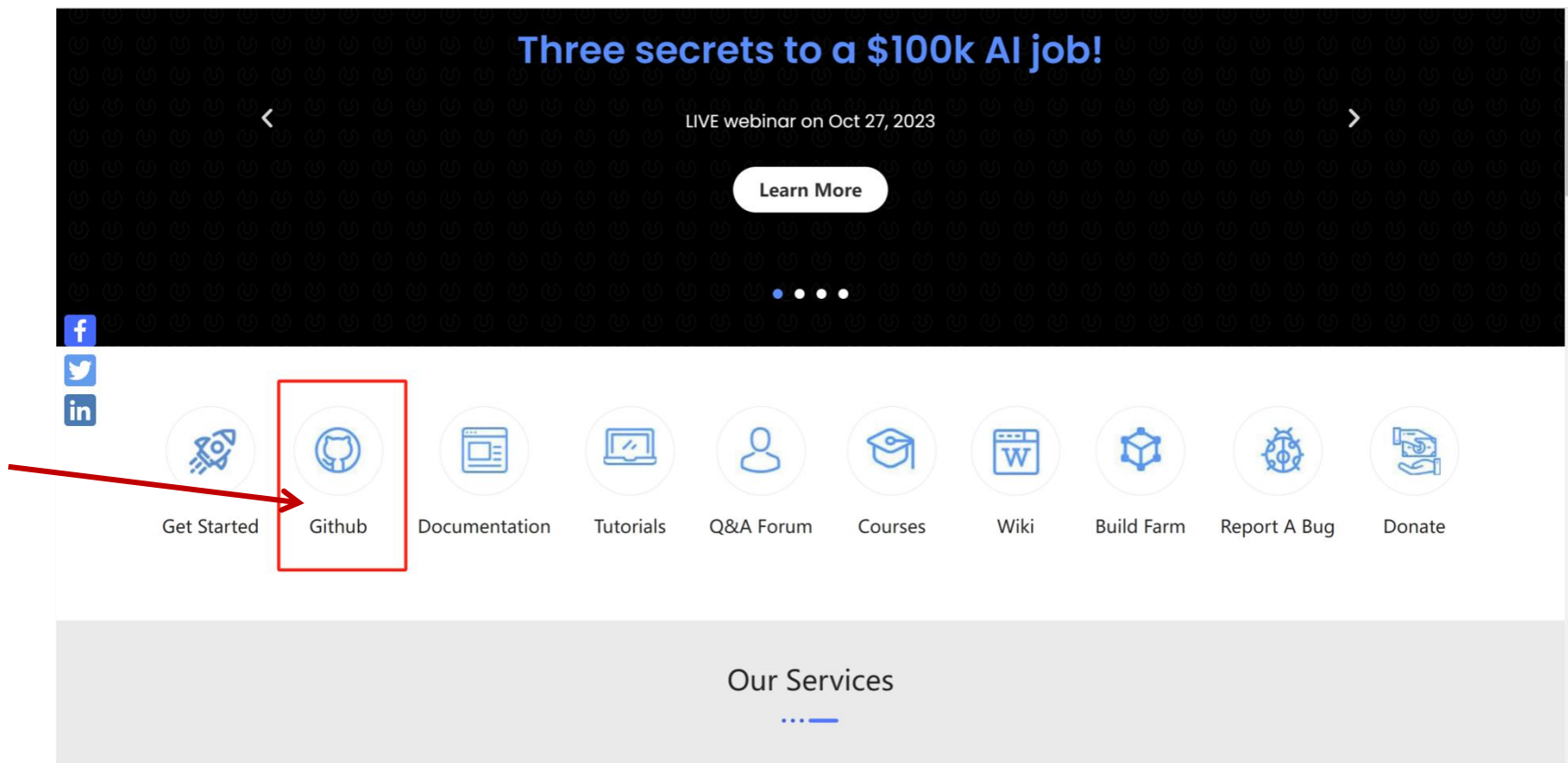
使用**管理员身份**安装，安装指令：

`pip install numpy -i https://pypi.tuna.tsinghua.edu.cn/simple`

案例指导：5. opencv官网

OpenCV官网: <https://opencv.org/>

在github访问源代码与安装方式。



案例指导：5. opencv源代码与安装 (github)

opencv / opencv Public

<> Code Issues 2.4k Pull requests 113 Actions Wiki Security Insights

4.x 6 branches 123 tags

kallaballa Merge pull request #23894 from kallaballa:blobFromImagesWithParams c2f909f 2 hours ago

- .github CI: enable RISC-V for 4.x branch
- 3rdparty Merge pull request #23109 from seanm:misc-warnings
- apps Add missing <stream> includes
- cmake Merge pull request #23929 from CNClareChen:4.x
- data Merge pull request #22727 from su77ungr:patch-1
- doc Added ArUco marker size check for Aruco and Charuco boards.
- include exclude opencv_contrib modules
- modules Merge pull request #23894 from kallaballa:blobFromImagesWithParams 2 hours ago
- platforms Merge pull request #24305 from hanliutong:toolchain last month
- samples Clean up the Universal Intrinsic API. last week
- .editorconfig add .editorconfig 5 years ago
- .gitattributes cmake: generate and install ffmpeg-download.ps1 5 years ago
- .gitignore Merge pull request #17165 from komakai:objc-binding 3 years ago
- CMakeLists.txt cmake: revise OPENCV_DNN_BACKEND_DEFAULT integration last month
- CONTRIBUTING.md migration: github.com/opencv/opencv 7 years ago

asmorkalov released this 3 weeks ago · 336 commits to 4.x since this release · 4.8.1 · 5199850

OpenCV 4.8.1 has been released.
Change log is [here](#).

SHA256 sum for packages:

```
a8bfe53cf9d7fa94ed5d85d61cf028348a6a1268b8027c9bc95ca96edd90ba13 opencv-4.8.1-android-sdk.zip
0689312a9de439757618a412b266dc5ee75d2e32aef9a9eac32c3f808ade06331 opencv-4.8.1-ios-framework.zip
037dd1c5c172237b0a00ee822684e5ac453028e1933111a2c547b62061f8b7e4 opencv-4.8.1-windows.exe
```

Assets 5

- opencv-4.8.1-android-sdk.zip 189 MB 3 weeks ago
- opencv-4.8.1-ios-framework.zip 197 MB 3 weeks ago
- opencv-4.8.1-windows.exe 168 MB 3 weeks ago
- Source code (zip) 3 weeks ago
- Source code (tar.gz) 3 weeks ago

39 4 2 14 2 53 people reacted

Report repository

Releases 61

- OpenCV 4.8.1 Latest 3 weeks ago

+ 60 releases

Sponsor this project

点击Release可以访问发布版本。
github访问速度确实糟糕。
可以访问国内加速站点。
也可以访问pip官方网站。



案例指导：5. opencv的pip站点

The screenshot shows the OpenCV pip installation page. On the left, there is a list of maintainers with their names and GitHub profiles. The main content area contains instructions for installing OpenCV via pip. It includes a 'Requires: Python >=3.6' section, a 'Maintainers' list, and a list of installation options. The options are:

- Option 1 - Main modules package: `pip install opencv-python`
- Option 2 - Full package (contains both main modules and contrib/extra modules): `pip install opencv-contrib-python`
- Option 3 - Headless main modules package: `pip install opencv-python-headless`
- Option 4 - Headless full package (contains both main modules and contrib/extra modules): `pip install opencv-contrib-python-headless`

Below the options, there is a section for importing the package, showing the code `import cv2`. At the bottom, it mentions that all packages contain Haar cascade files and provides a shortcut `cv2.data.harcascades`.

pip install opencv-python -i https://pypi.tuna.tsinghua.edu.cn/simple

pip install opencv-contrib-python -i https://pypi.tuna.tsinghua.edu.cn/simple

案例指导：5. opencv的国内源代码站点

github的克隆代码：<https://gitee.com/explore>

The screenshot shows the Gitee website homepage. At the top, there is a navigation bar with the Gitee logo, links for '开源软件' (Open Source Software), '企业版' (Enterprise Edition), '高校版' (University Edition), '私有云' (Private Cloud), and '博客' (Blog). A search bar is located on the right side of the navigation bar. Below the navigation bar, there is a large search bar with a '搜索' (Search) button. To the right of the search bar, there are several hot search terms: '物联网' (IoT), '鸿蒙' (HarmonyOS), 'OceanBase', 'Serverless', and '微服务' (Microservices). Below the search bar, there are several logos for popular open-source projects: AntV, apollo, 滴滴 | 开源 (Didi | Open Source), DolphinScheduler, dotNET, and a logo for China. The main content area is divided into two columns. The left column contains a sidebar with categories: '全部推荐项目' (All Recommended Projects), '前沿技术' (Cutting-edge Technology), 'OpenHarmony', '程序开发' (Program Development), '人工智能' (Artificial Intelligence), '区块链' (Blockchain), '微信开发' (WeChat Development), '企业应用' (Enterprise Applications), '建站系统' (Website Building Systems), '应用工具' (Application Tools), 'WEB应用开发' (WEB Application Development), and 'DevOps/运维/网管' (DevOps/Operations/Network Management). The right column displays a list of recommended projects. The first project is 'tencentcloud/tencentcloud-sdk-java' (Tencent Cloud API 3.0 SDK for Java) with 5 stars. The second project is '大古/light-chaser' (Light Chaser) with 76 stars. The third project is 'Quard/quardCRT' (quardCRT) with 11 stars. The fourth project is '刘文韬/六轴机器人仿真软件' (Six-axis robot simulation software) with 85 stars. On the right side of the main content area, there are two columns: '今日热门' (Today's Popular) and '本周热门' (This Week's Popular). The '今日热门' column lists '芋道源码/yudao-ui-admin-vb...' (1K stars), 'TOPIAM/topiam-eiam' (1.4K stars), and 'hefengbao/jingmo' (31 stars). The '本周热门' column lists 'lengleng/pig' (41.1K stars) and 'RL/RL云验证' (74 stars). A small tooltip with a question mark and the text '点此查找更多帮助' (Click here to find more help) is visible over the 'hefengbao/jingmo' project.



案例2. 编译编写一个程序，打印一个表格

任务

编写一个python程序，使用变量与制表符，打印一个成绩表格效果如右图。

数据：

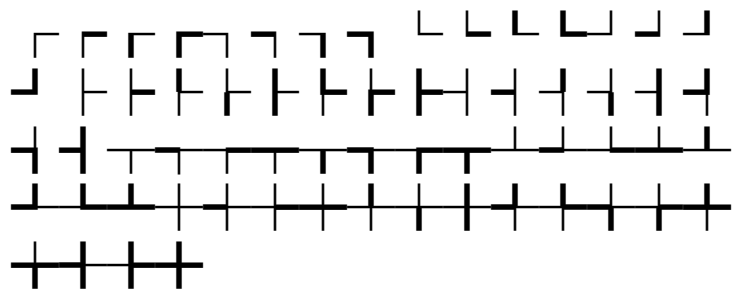
赵德柱 90.56

黄金华 96.68

姓名	成绩
赵德柱	90.56
黄金华	96.68

案例指导：制表符号

使用输入法提供的特殊字符，或者百度搜索“特殊字符”可以得到制表符号。



搜狗输入法提供的特殊字符输入（直观）。





案例指导：参考步骤

1. 定义表格中需要输出的数据。
2. 使用输出语句，一行行打印表格（内容是制表符与数据）。
 - 2.1. 数据的输出使用格式字符串。F" {}"



案例指导：参考实现与运行效果 (VSCode与Jupyter)

VSCode代码：

```
name1 = "赵德柱"  
name2 = "黄金花"  
score1 = 90.67  
score2 = 99.56  
print(" | | ")  
print(F" | {name1} | {score1} | ")  
print(" | | ")  
print(F" | {name2} | {score2} | ")  
print(" | | ")
```

赵德柱	90.67
黄金花	99.56

Jupyter代码：

```
name1 = "赵德柱"  
name2 = "黄金花"  
score1 = 90.67  
score2 = 99.56  
print(" | | ")  
print(F" | \033[31m{name1}\033[0m | {score1} | ")  
print(" | | ")  
print(F" | \033[31m{name2} \033[0m | {score2} | ")  
print(" | | ")
```

赵德柱	90.67
黄金花	99.56

注：这里使用了转义序列知识。



案例指导：参考实现与运行效果 (VsCode)

代码：

```
name1 = "赵德柱"  
name2 = "黄金花"  
score1 = 90.67  
score2 = 99.56  
print(" | | ")  
print(F" | {name1} | {score1} | ")  
print(" | | ")  
print(F" | {name2} | {score2} | ")  
print(" | | ")
```

赵德柱	90.67
黄金花	99.56



案例3. 使用git上传代码并使用markdown

任务

使用git按照要求上传代码到服务器。

1. 在git服务器注册账号。
2. 在账号下建立仓库。
3. 使用git在本地建本地仓库。
4. 配置本地仓库。
5. 编写仓库README
6. 编写本地仓库更新脚本。



案例指导：git服务器

http://39.106.24.154:8906/



注册 登录

进入登录页面



案例指导：git服务器注册

The screenshot shows a web registration form for Git. The form is titled "注册" (Registration) and is centered on the page. It contains the following fields and elements:

- Navigation links: 首页 (Home), 发现 (Discover), 帮助 (Help) on the left; 注册 (Register), 登录 (Login) on the right.
- Form title: 注册
- Fields: 用户名* (Username), 邮箱* (Email), 密码* (Password), 确认密码* (Confirm Password), 验证码* (Captcha).
- Captcha: A handwritten-style image of the number "97813".
- Submit button: 创建帐户 (Create Account)
- Footer: 已经注册? 立即登录! (Already registered? Log in immediately!)
- Page footer: © 2023 Gogs, 页面: 0ms 模板: 0ms, 简体中文 | 官方网站

注册属于常规技能，直接输入注册信息，创建帐户。

案例指导：git服务器登录并创建仓库



必须：输入仓库名。
可选：编辑仓库描述。
可选：指定授权许可类型。

案例指导：git服务器的友情帮助

The screenshot shows a web interface for a Git repository named 'AITools' by user 'yangqiang'. The page includes a navigation bar with '文件', '工单管理 0', 'Wiki', and '仓库设置'. Below the navigation bar, there is a '快速帮助' (Quick Help) section. The first part of the help section is titled '克隆当前仓库' (Clone current repository) and provides a URL: 'http://39.106.24.154:8906/yangqiang/AITools.git'. The second part is titled '从命令行创建一个新的仓库' (Create a new repository from the command line) and contains a list of terminal commands: 'touch README.md', 'git init', 'git add README.md', 'git commit -m "first commit"', 'git remote add origin http://39.106.24.154:8906/yangqiang/AITools.git', and 'git push -u origin master'. The third part is titled '从命令行推送已经创建的仓库' (Push already created repository from the command line) and contains the commands: 'git remote add origin http://39.106.24.154:8906/yangqiang/AITools.git' and 'git push -u origin master'. A red box highlights the first set of commands, and a red arrow points from the text '本地仓库创建与配置指令。' to this box. The Windows taskbar is visible at the bottom of the screenshot.

本地仓库创建与配置指令。

案例指导：使用git创建本地仓库并配置

1. 选择一个本地目录，使用git指令把目录创建一个仓库，
 - git init
2. 新建README.md文件（空的也可以）
 - echo "" >README.md
3. 添加服务仓库到本地git配置
 - git remote add origin http://39.106.24.154:8906/yangqiang/AITools.git

```
命令提示符
E:\AITools>cd
E:\AITools
E:\AITools>git init
Initialized empty Git repository in E:/AITools/.git/
E:\AITools>echo "" >README.md
E:\AITools>git remote add origin http://39.106.24.154:8906/yangqiang/AITools.git
E:\AITools>
```

注意：在使用的时候，可能会提示配置用户名与邮箱，可以根据提示直接设置。

案例指导：使用markdown编写README.md内容

1. 标题：#
2. 引用：>
3. 列表：-
4. 分割线：----
5. 代码块：``语言类型``
6. 图像：![说明](图像文件名)
7. 强调：``
8. 加粗：**文本**
9. 数学公式：\$Latex\$或者\$\$Latex\$\$
10. 表格：|--|

```
1  ### 第一次课
2  ##### I. 内容
3  >1. 语言要素规范
4  >2. python解释器与开发工具
5  >3. 数据定义
6  >4. 数据运算与运算符
7
8  ##### II. 要求
9  >1. 能编写代码并运行代码
10
11 ##### III. 实验1：语言规范与工具实验
12 ##### 实验要求
13 >1. 能使用如下方式编写，运行Python代码（非程序）
14 > 1.1. vscode编辑器；
15 > 1.2. python交互式编程；
16 > 1.3. jupyter编写；
17 >2. 能使用git上传代码；
18
19 ##### 实验案例
20 - 案例1. 安装python学习相关软件（本机，使用平台的不用安装）
21   1. 安装python
22   2. 安装git
23   3. 安装jupyter
24   4. 安装opencv, numpy
25   5. 安装VSCode
26 - 案例2. 编译编写一个程序，打印一个表格。
27   1. VSCode源代码版
28   2. Jupyter版本
29 - 案例3. 使用git上传代码到服务器，并使用markdown（说明作业完成情况）。
30
```

Line 16, Column 22

案例指导：使用git更新服务器仓库

@git add *

@git commit -m "教学资料"

@git push -u origin master

注意：建议把上传命令编程脚本。

```
命令提示符
E:\AITools>git add *
warning: in the working copy of 'Class01/codes/class01.ipynb', LF will be replaced by CRLF the next time Git touches it

E:\AITools>git commit -m "教学资料"
[master (root-commit) ale9762] 教学资料
9 files changed, 239 insertions(+)
create mode 100644 Class01/codes/app/app.py
create mode 100644 Class01/codes/class01.ipynb
create mode 100644 Class01/codes/first.py
create mode 100644 Class01/codes/graphic/draw.py
create mode 100644 Class01/note01.md
create mode 100644 "Class01/\345\256\236\351\252\2141_\350\257\255\350\250\200\350\247\204\350\214\203\344\270\216\345\267\245\345\205\267\345\256\236\351\252\214.pptx"
create mode 100644 "Class01/\347\254\25401\346\254\241\350\257\276 \347\274\226\347\250\213\350\257\255\350\250\200\357\274\232\350\257\255\350\250\200\347\273\223\346\236\204\344\270\216\346\225\260\346\215\256.pptx"
create mode 100644 README.md
create mode 100644 commit.bat

E:\AITools>git push -u origin master
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (15/15), 11.86 MiB | 5.01 MiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0
To http://39.106.24.154:8906/yangqiang/AITools.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

E:\AITools>
```



案例指导：查看服务器提交效果

Class01	a1e97620a1	教学资料	8 分钟之前
README.md	cb346fee22	鐵欄口壁勒枳	4 分钟之前
commit.bat	27b4c49e41	教学资料	1 分钟之前

README.md

第一次课

I. 内容

1.语言要素规范 2.python解释器与开发工具 3.数据定义 4.数据运算与运算符

II. 要求

1. 能编写代码并运行代码

III. 实验1：语言规范与工具实验

实验要求

1. 能使用如下方式编写，运行Python代码（非程序） 1.1. vscode编辑器； 1.2. python交互式编程； 1.3. jupyter编写；
2. 能使用git上传代码；

实验案例

- 案例1. 安装python学习相关软件（本机，使用平台的不用安装）
 - i. 安装python
 - ii. 安装git
 - iii. 安装jupyter
 - iv. 安装opencv, numpy
 - v. 安装VSCode

正在连接...

答疑联系

13338629985(微信)