

Web 自动化环境安装说明

说明：本文档是基于 Windows 系统的 Web 自动化环境安装说明文档

一、所需要软件

1、Python 开发环境

作用：解析执行 python 语言的代码

下载地址：<https://www.python.org/downloads/release/python-376/>

2、PyCharm 编辑器（Python 集成开发环境）

作用：用来编写 python 代码，

下载地址：<https://www.jetbrains.com/pycharm/download/#section=windows>

3、基于 python 的 selenium 工具包

作用：第三方自动化测试开源框架，提供针对浏览器操作的相关 API

4、WebDriver 浏览器驱动(不同浏览器有不同的 WebDriver)

作用：实现对浏览器的管理，包括创建浏览器实例以及对浏览器的操作

谷歌浏览器驱动下载地址：<http://npm.taobao.org/mirrors/chromedriver/>

火狐浏览器驱动下载地址：<http://npm.taobao.org/mirrors/geckodriver/>

注意版本关系，请下载对应浏览器版本的驱动。

5、浏览器（Chrome、Firefox）

作用：用来访问 WEB 系统的工具

谷歌浏览器下载地址：<https://www.google.cn/chrome/>

火狐浏览器下载地址：<http://www.firefox.com.cn/download/>

6、当前的软件工具包已整理好，各位同学可以通过百度网盘进行下载：

链接：<https://pan.baidu.com/s/1cXwGzgJRpjW9hwknsSm2Yw>

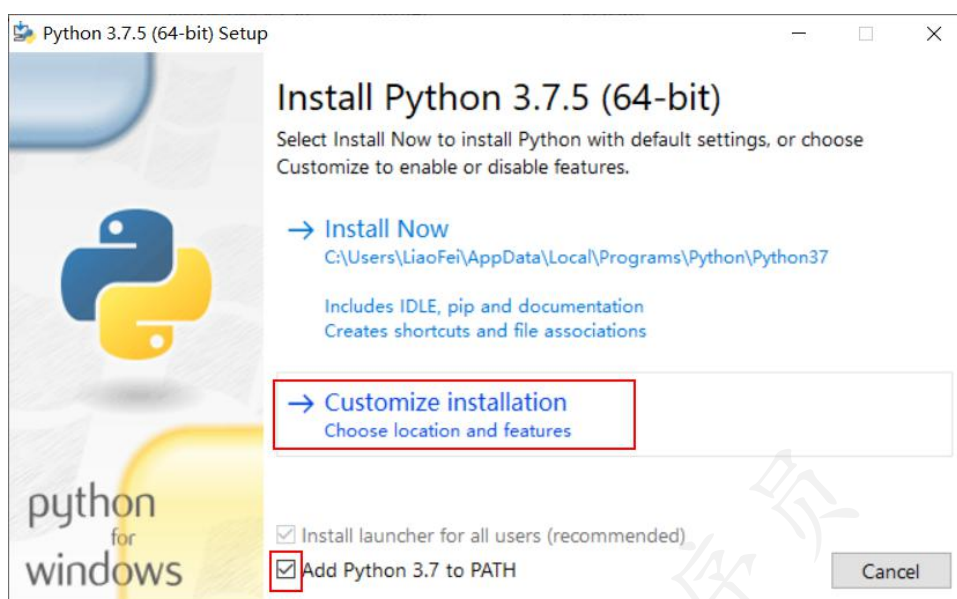
提取码：se0o

二、安装步骤：

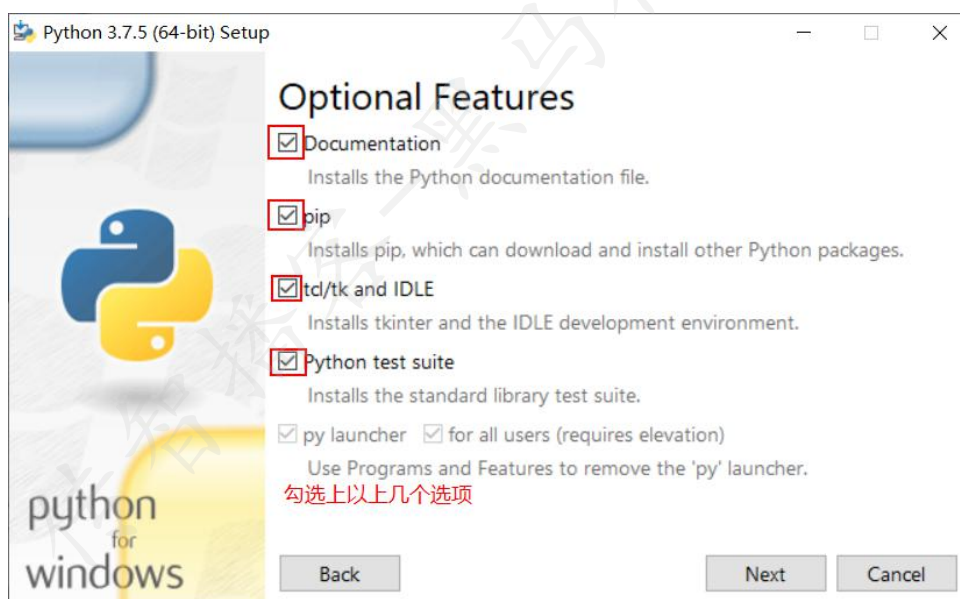
1、python 语言包安装

- 打开下载目录“python 语言安装包”，双击“python-3.7.5-amd64.exe”文件进行安

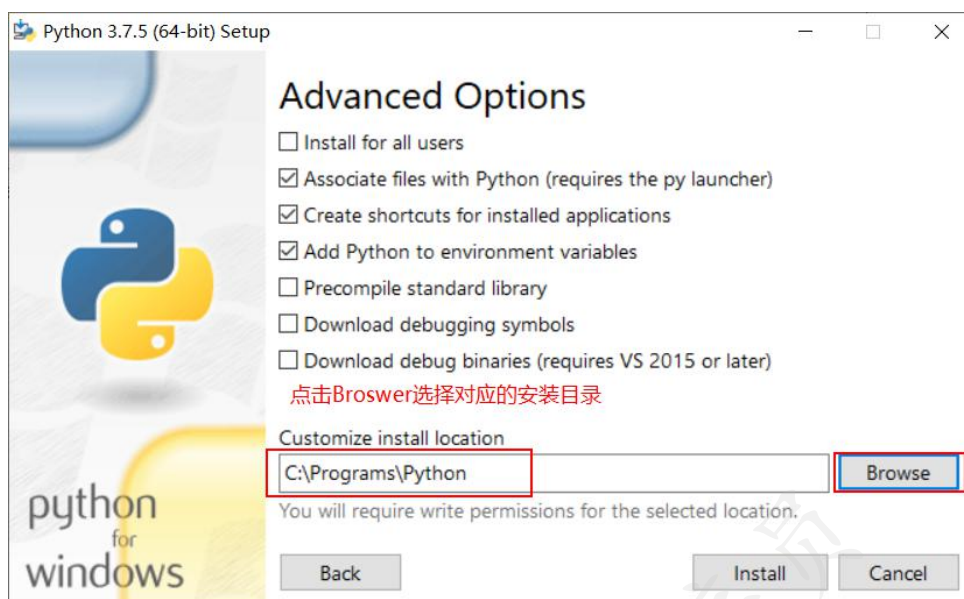
装。安装过程如下图所示：



(图一)

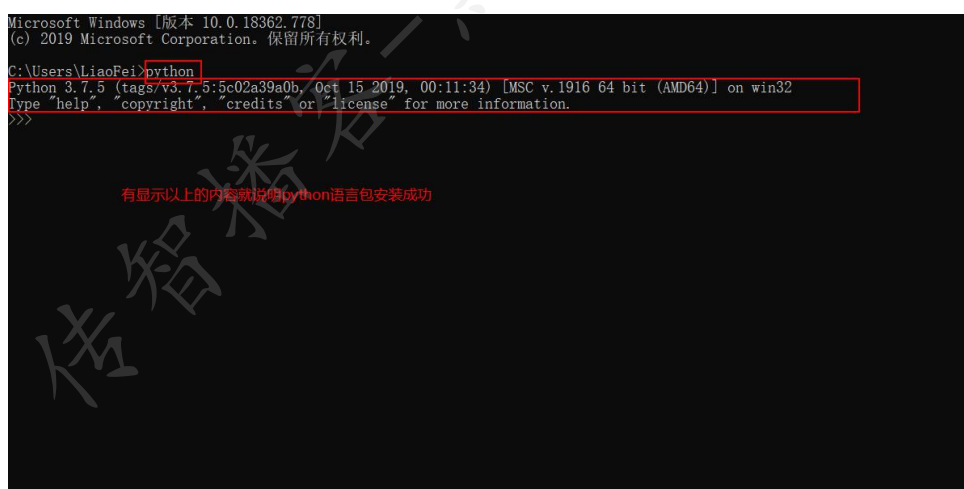


(图二)



(图三)

- 在图三中确定好安装目录之后，点击“Install”按钮进行安装，安装完成之后再点击“close”关闭即可。
- 安装完成之后，通过 cmd 进入到 DOS 命令行，在 DOS 命令行下输入 python 如果出现下图所显示的内容就说明安装成功。



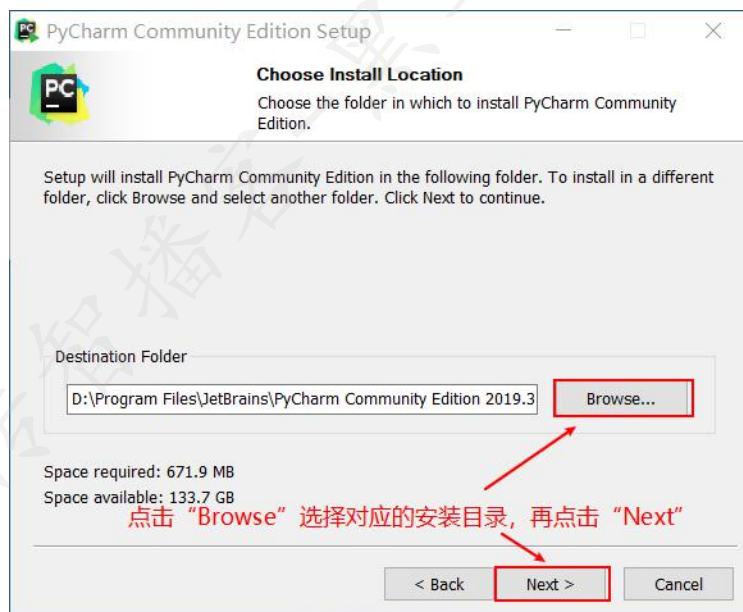
(图四)

2、Pycharm(python 编辑器)安装

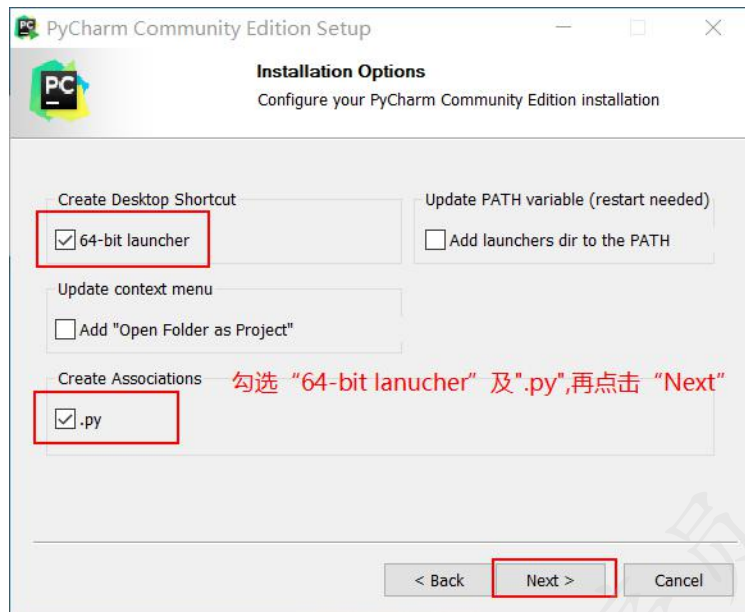
- 打开下载目录“python 编辑器工具-Pycharm”，双击“pycharm-community-2019.3.5.exe”文件进行安装。安装过程如下图所示：



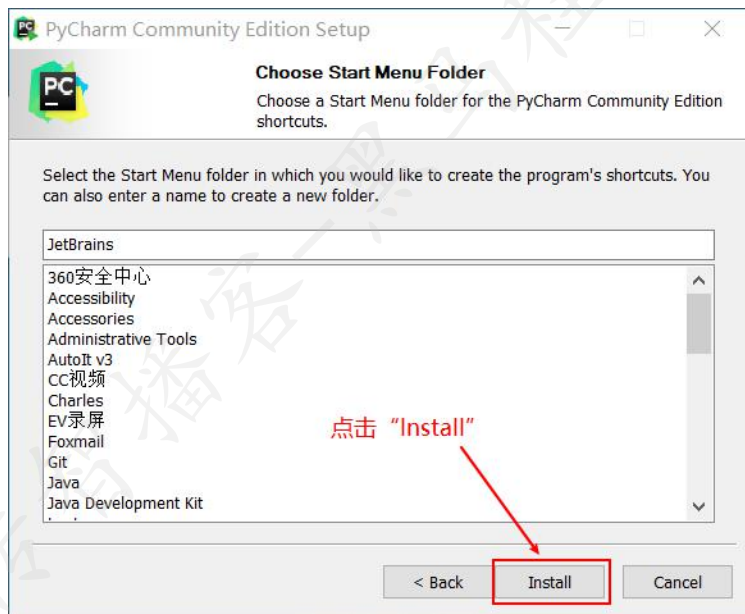
(图四)



(图五)



(图六)

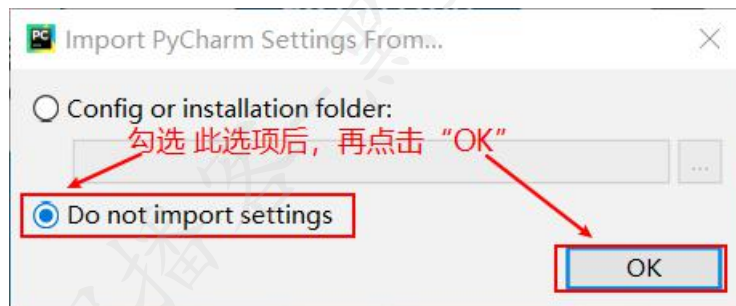


(图七)

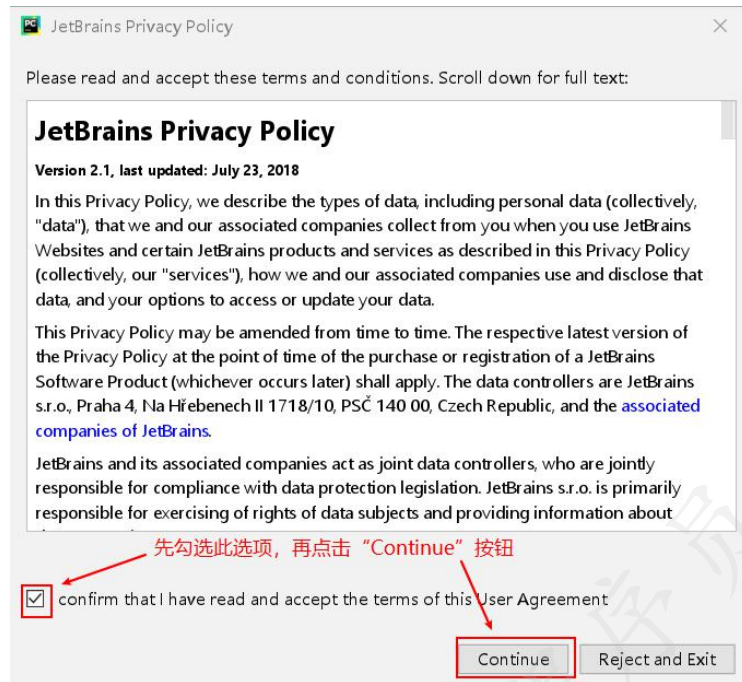


(图八)

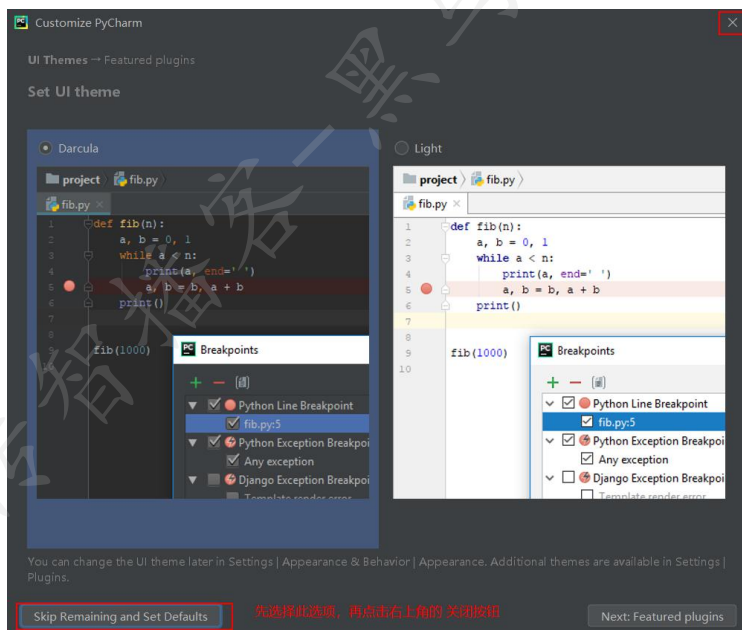
- 安装完成之后会自动运行 pycharm 编辑器工具, 首次运行时需要进行相关的配置, 配置步骤如下图所示:



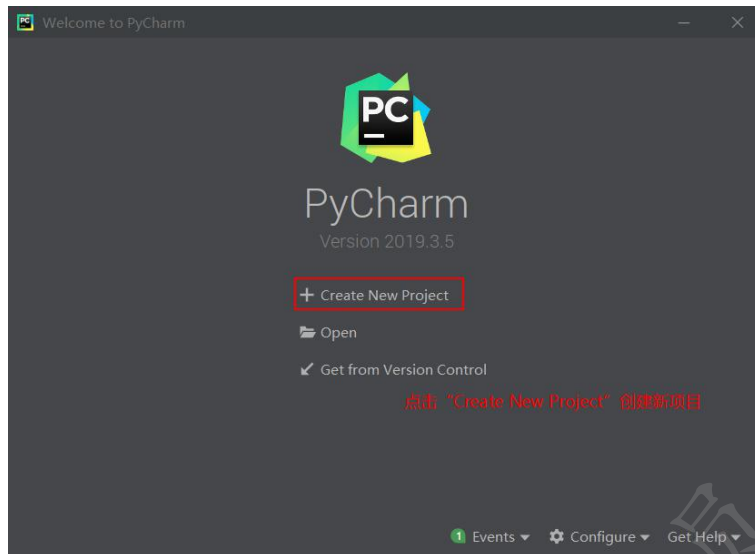
(图九)



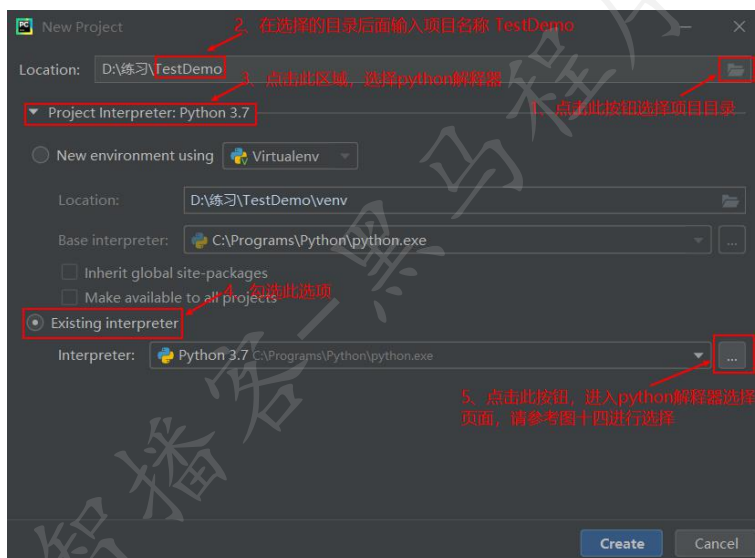
(图十)



(图十一)



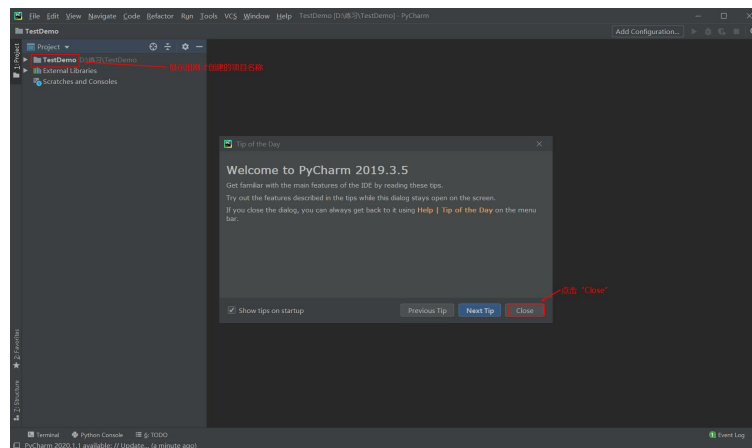
(图十二)



(图十三)

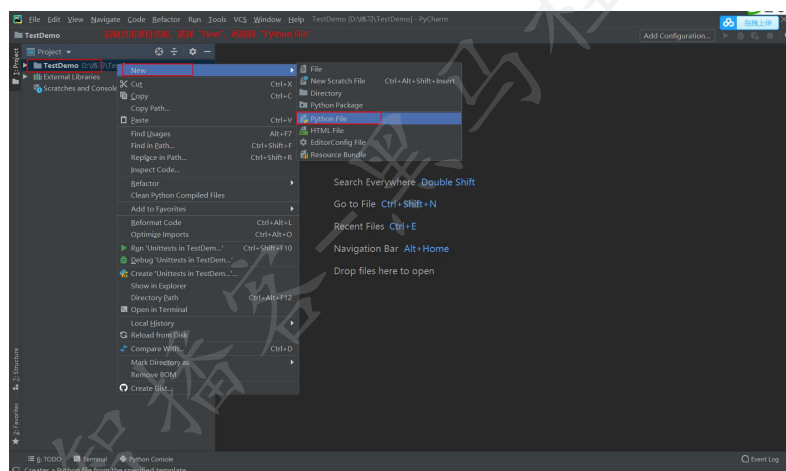


(图十四)

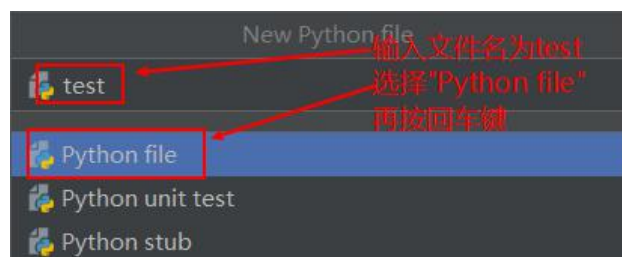


(图十五)

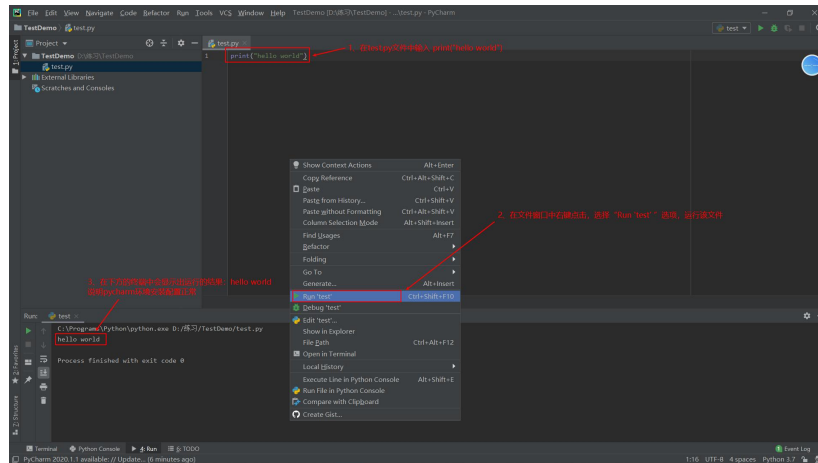
- 如果能看到图十五所示的图片，那么 pycharm 工具的首次配置就完成了。接下来在项目中创建 python 文件，写一个示例代码看能不能正常运行。具体步骤如下图所示：



(图十六)



(图十七)



(图十八)

- 如果运行示例文件能够看到图十八所示的结果, 就说明 pycharm 的安装和配置是正常的。

代码: `print("hello world")`

3、Selenium 工具包安装

注意: Selenium 工具包提供在线安装和离线安装两种方式, 选择其中一种安装方式即可。

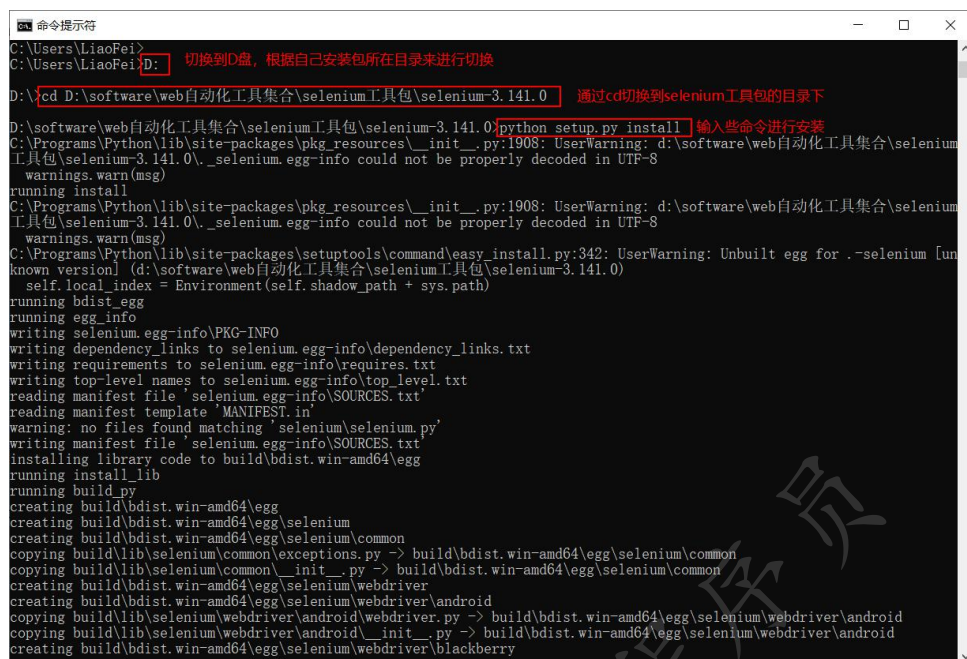
- 在线安装方式: 进入到 DOS 命令行中, 直接输入以下命令 (如下图十九): `pip install selenium`



(图十九)

- 离线安装方式: 在 DOS 命令行中, 进入到 selenium 工具包的目录下面, 然后

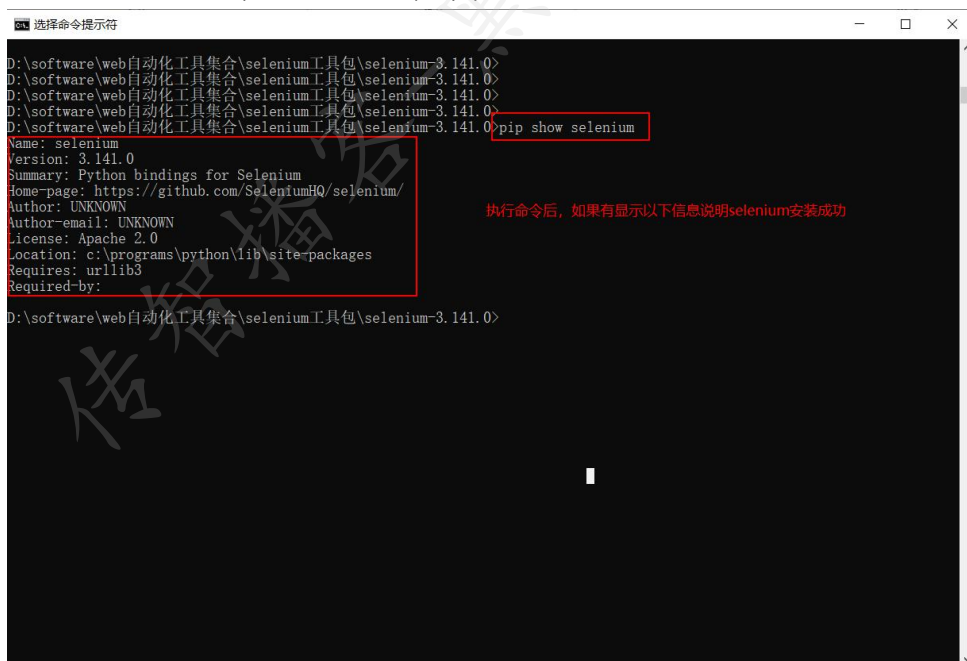
执行以下命令进行安装(如下图二十): `python setup.py install`



```
C:\Users\LiaoFei>
C:\Users\LiaoFei> D: 切换到D盘, 根据自己安装包所在目录来进行切换
D:\> cd D:\software\web自动化工具集合\selenium工具包\selenium-3.141.0 通过cd切换到selenium工具包的目录下
D:\software\web自动化工具集合\selenium工具包\selenium-3.141.0> python setup.py install 输入些命令进行安装
C:\Programs\Python\lib\site-packages\pkg_resources\__init__.py:1908: UserWarning: d:\software\web自动化工具集合\selenium
工具包\selenium-3.141.0\._selenium.egg-info could not be properly decoded in UTF-8
  warnings.warn(msg)
running install
C:\Programs\Python\lib\site-packages\setuptools\command\easy_install.py:342: UserWarning: Unbuilt egg for .selenium [un
known version] (d:\software\web自动化工具集合\selenium工具包\selenium-3.141.0)
  self.local_index = Environment(self.shadow_path + sys.path)
running bdist_egg
running egg_info
writing selenium.egg-info\PKG-INFO
writing dependency links to selenium.egg-info\dependency_links.txt
writing requirements to selenium.egg-info\requires.txt
writing top-level names to selenium.egg-info\top_level.txt
reading manifest file 'selenium.egg-info\SOURCES.txt'
reading manifest template 'MANIFEST.in'
warning: no files found matching 'selenium\selenium.py'
writing manifest file 'selenium.egg-info\SOURCES.txt'
installing library code to build\bdist.win-amd64\egg
running install_lib
running build_py
creating build\bdist.win-amd64\egg
creating build\bdist.win-amd64\egg\selenium
creating build\bdist.win-amd64\egg\selenium\common
copying build\lib\selenium\common\exceptions.py -> build\bdist.win-amd64\egg\selenium\common
copying build\lib\selenium\common\__init__.py -> build\bdist.win-amd64\egg\selenium\common
creating build\bdist.win-amd64\egg\selenium\webdriver
creating build\bdist.win-amd64\egg\selenium\webdriver\android
copying build\lib\selenium\webdriver\android\webdriver.py -> build\bdist.win-amd64\egg\selenium\webdriver\android
copying build\lib\selenium\webdriver\android\__init__.py -> build\bdist.win-amd64\egg\selenium\webdriver\android
creating build\bdist.win-amd64\egg\selenium\webdriver\blackberry
```

(图二十)

- 安装完成之后, 需要检查 selenium 工具包是否安装成功。在 DOS 命令行中输入以下命令(如下图二十一): `pip show selenium`



```
D:\software\web自动化工具集合\selenium工具包\selenium-3.141.0>
D:\software\web自动化工具集合\selenium工具包\selenium-3.141.0>
D:\software\web自动化工具集合\selenium工具包\selenium-3.141.0>
D:\software\web自动化工具集合\selenium工具包\selenium-3.141.0>
D:\software\web自动化工具集合\selenium工具包\selenium-3.141.0> pip show selenium
Name: selenium
Version: 3.141.0
Summary: Python bindings for Selenium
Home-page: https://github.com/SeleniumHQ/selenium/
Author: UNKNOWN
Author-email: UNKNOWN
License: Apache 2.0
Location: c:\programs\python\lib\site-packages
Requires: urllib3
Required-by:
D:\software\web自动化工具集合\selenium工具包\selenium-3.141.0>
```

(图二十一)

4、Chrome 浏览器及火狐浏览器安装

浏览器安装包在工具集中已提供给大家, 大家直接安装就行, 课程当中自动化示例都是以

Chrome 浏览器为主。如果大家已经安装了 Chrome 浏览器或者火狐浏览器建议卸载，重新安装。

- 建议大家安装 Chrome 81 的版本

software > web自动化工具集合 > 浏览器安装包 > 谷歌浏览器	
名称	修改日期
78版本	2020/5/18 18:09
81版本	2020/5/18 18:09

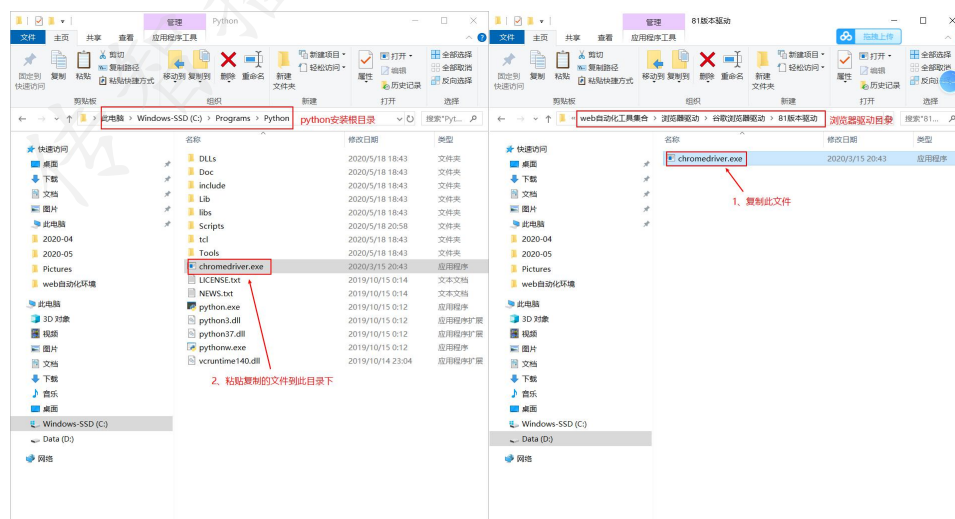
(图二十二)

- 建议大家安装火狐浏览器 76 的版本

5、浏览器驱动安装

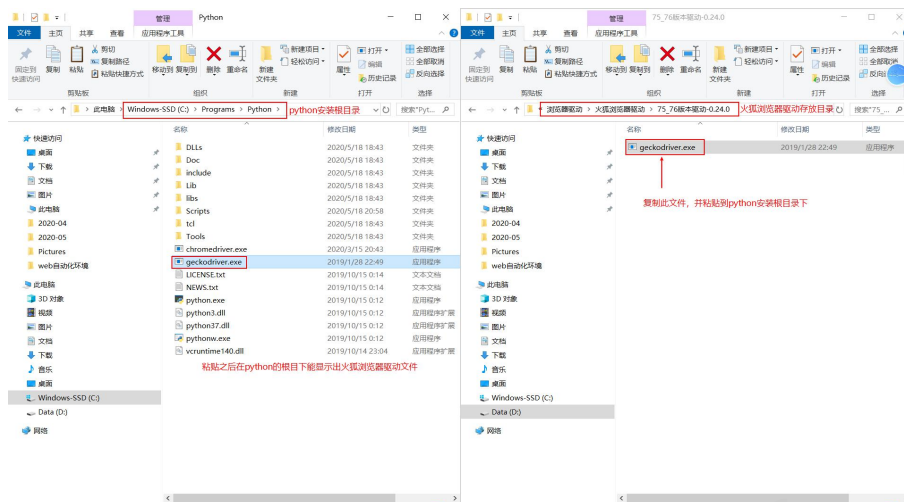
- 安装 Chrome 浏览器驱动：将下载的工具集当中的谷歌浏览器驱动文件拷贝贝 python 的安装根目录即可。

注意：工具集里面有两个版本的谷歌浏览器驱动，如果大家安装的 81 版本的谷歌浏览器，那么大家就把 81 版本的浏览器驱动拷贝贝到 python 的安装根目录。如果大家安装的 78 版本的谷歌浏览器，那么大家就把 78 版本的浏览器驱动拷贝贝到 python 的安装根目录。只选择其中一个即可。如图二十三所示：



(图二十三)

- 安装火狐浏览器驱动：将下载的工具集当中的火狐浏览器驱动文件拷贝贝到 python 安装根目录下。如图二十四所示：



(图二十四)

6、Web 自动化环境验证

- 在 Pycharm 编辑器工具中新建一个 python 文件，并输入以下示例代码(如图二十五所示):

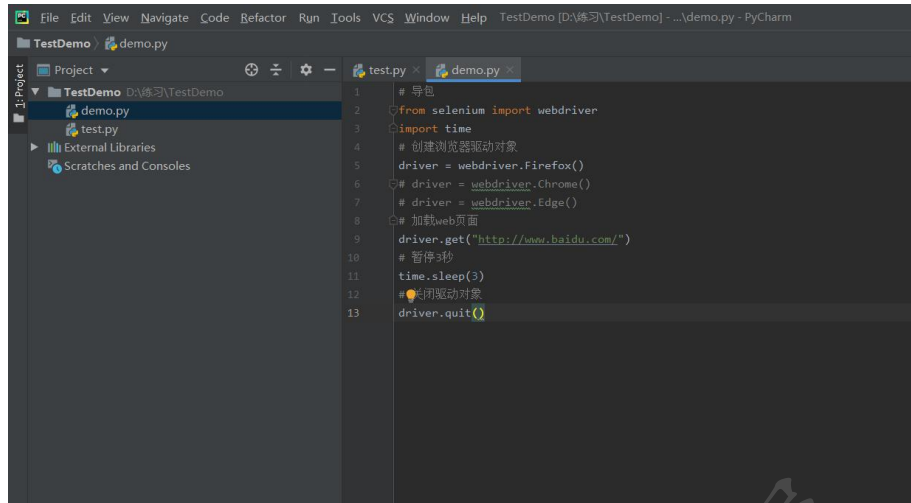
```
# 导包
from selenium import webdriver
import time

# 创建浏览器驱动对象
driver = webdriver.Chrome()
# driver = webdriver.Chrome()
# driver = webdriver.Edge()

# 加载web页面
driver.get("http://www.baidu.com/")

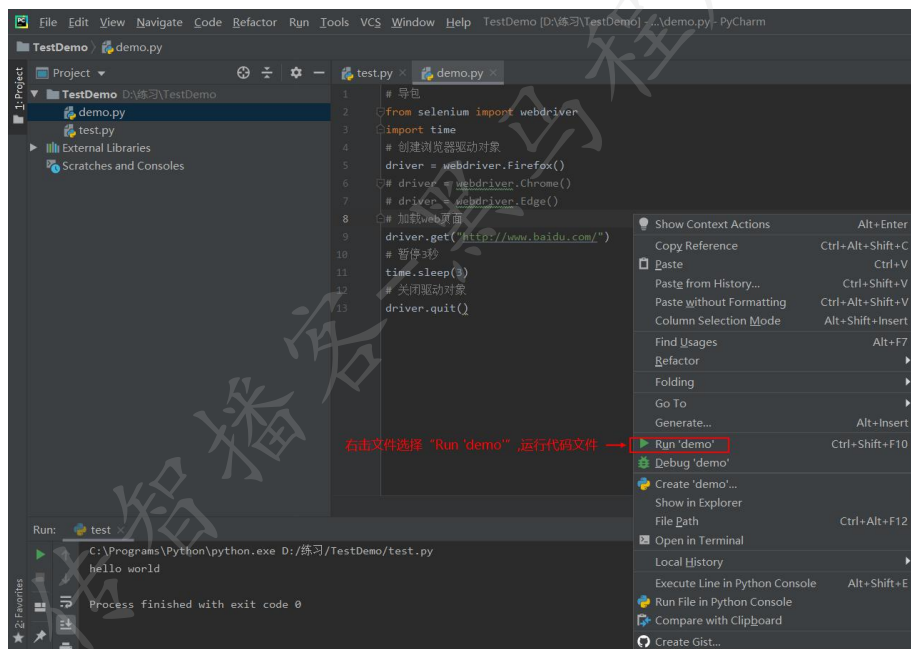
# 暂停3秒
time.sleep(3)

# 关闭驱动对象
driver.quit()
```



(图二十五)

- 在示例代码文件中 右击 选择“Run 'demo' ”运行代码。



(图二十六)

- 运行代码过程中，如果能够打开 Chrome 浏览器并进入到百度网站，并且在等待 3 秒钟之后，自动关闭浏览器就说明 Web 的自动化测试环境搭建成功。