

实验课 01

实验1-1 GitHub的初始化

步骤1 注册并登录GitHub

用浏览器打开 [GitHub](https://github.com)。

如果还没有GitHub账户，请点击右上角的Sign up按钮，注册一个新账号。

然后登录GitHub账号。

步骤2 新建一个仓库

单击左侧的New按钮，新建仓库。

Repositories

New

Find a repository...

在新建仓库页面的"Repository name"中输入仓库名称，其他保持默认选项，单击下方的"Create repository"按钮，完成新建仓库。

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *

soobun

Repository name *

Great repository names are short and memorable. Need inspiration? How about [ideal-octo-enigma?](#)

Description (optional)

☒



Public

Anyone on the internet can see this repository. You choose who can commit.

☐



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None

Add a license: None



Create repository

实验1-2 配置编程环境

步骤1 确认系统架构

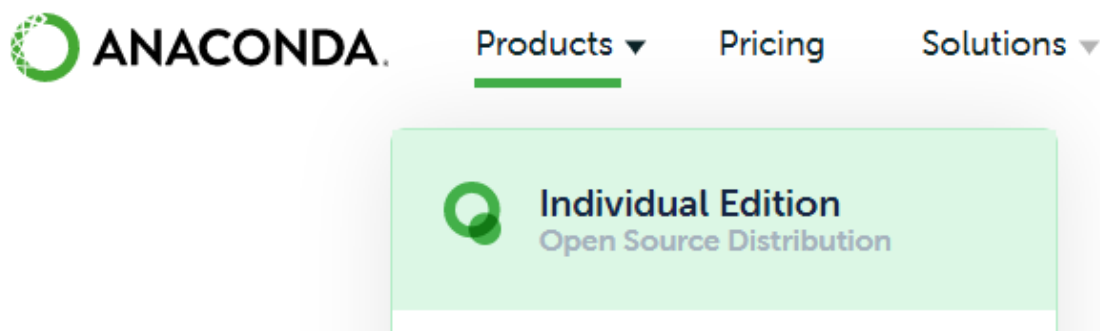
按下Win+R键，弹出“运行”窗口，在“打开”输入框中输入 `cmd` 后回车，启动命令提示符。

在命令行界面输入 `wmic OS get OSArchitecture` 后回车，记录操作系统的架构是32位还是64位

步骤2 下载并安装Anaconda

Anaconda是一个免费开源的Python和R语言的发行版本，用于计算科学（数据科学、机器学习、大数据处理和预测分析），Anaconda致力于简化包管理和部署。Anaconda的包使用的是软件包管理系统Conda进行管理。

浏览器打开[Anaconda的主页](#)，并在上方Product中单击Individual Edition。

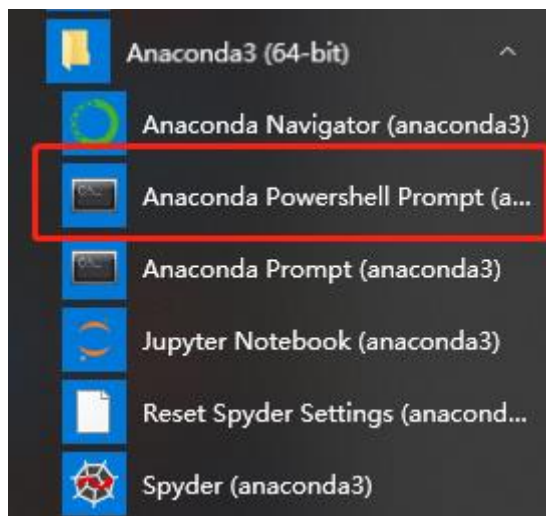


将页面滚动至最下方，根据步骤1记录的操作系统架构选择合适的版本下载安装包，下载完成后运行安装程序，一路点击下一步即可。

步骤3 Anaconda-切换默认下载源

当通过Anaconda安装第三方库的时候，默认是从Anaconda国外的服务器下载安装包并进行安装。我们可以通过修改默认源，让Anaconda从清华大学的镜像站中下载安装包，这样就可以节省很多时间。

打开开始菜单，然后打开 Anaconda3(64-bit) 的文件夹，然后单击 `Anaconda Powershell Prompt (anaconda3)`，启动命令行模式的Anaconda。



在命令行中分别输入这三行语句，输入完一行语句后按一下回车以执行。

```
conda config --add channels
```

```
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free
```

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main

conda config --set show_channel_urls yes
```

步骤4 Anaconda-创建新环境

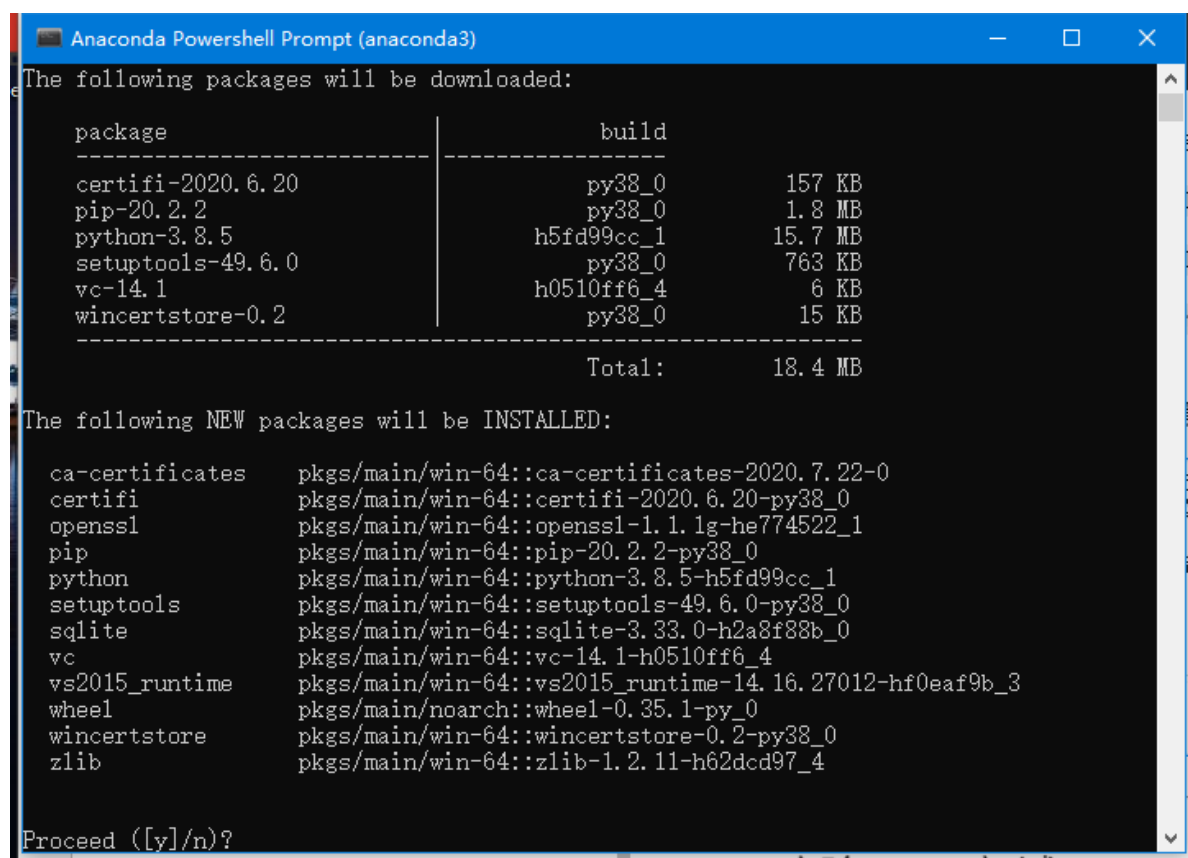
Anaconda支持创建多个虚拟环境，这样方便我们管理多个版本的Python环境，例如我们玩图像处理时可能需要安装大量的深度学习框架，但是进行简单的数据分析时就没必要加载这些库，我们可以轻松地切换环境以适应实际问题解决的需要（类似于手机的双卡双待，生活一个号工作一个号）。

启动Anaconda Powershell Prompt，在命令行我们可以看到最前方有 (base) 字样，这代表我们目前处于base环境下。

现在我们创建一个新环境，名称为test，安装python的版本是3.8。

在命令行中输入 `conda create -n test python=3.8`，按回车键执行。

系统稍后会列出将要安装的库名称及下载源，直接按回车键开始下载并自动安装。



```
Anaconda Powershell Prompt (anaconda3)
The following packages will be downloaded:

package                                     build                                     157 KB
-----                                     -
certifi-2020.6.20                           py38_0
pip-20.2.2                                   py38_0                                   1.8 MB
python-3.8.5                                h5fd99cc_1                               15.7 MB
setuptools-49.6.0                           py38_0                                   763 KB
vc-14.1                                       h0510ff6_4                               6 KB
wincertstore-0.2                             py38_0                                   15 KB
-----                                     -
Total:                                     18.4 MB

The following NEW packages will be INSTALLED:

ca-certificates      pkgs/main/win-64::ca-certificates-2020.7.22-0
certifi              pkgs/main/win-64::certifi-2020.6.20-py38_0
openssl              pkgs/main/win-64::openssl-1.1.1g-he774522_1
pip                  pkgs/main/win-64::pip-20.2.2-py38_0
python               pkgs/main/win-64::python-3.8.5-h5fd99cc_1
setuptools           pkgs/main/win-64::setuptools-49.6.0-py38_0
sqlite               pkgs/main/win-64::sqlite-3.33.0-h2a8f88b_0
vc                   pkgs/main/win-64::vc-14.1-h0510ff6_4
vs2015_runtime       pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_3
wheel                pkgs/main/noarch::wheel-0.35.1-py_0
wincertstore         pkgs/main/win-64::wincertstore-0.2-py38_0
zlib                 pkgs/main/win-64::zlib-1.2.11-h62dcd97_4

Proceed ([y]/n)?
```

步骤5 Anaconda-查看环境列表及切换环境

在命令行输入 `conda info -e` 后回车，系统会列出当前电脑已安装的环境列表。

现在我们切换至刚才新建的test环境，准备安装Jupyter。

输入 `conda activate test` 后回车，我们看到命令行最前端字样已变成 (test)，这说明我们已经成功切换至test环境。

如果想删除一个环境，请输入 `conda remove -n test --all`，这样就会删除test环境及环境中所有的库，操作需谨慎！

步骤6 Anaconda-安装Jupyter

我们以后的实验代码都是在Jupyter Notebook上编写并调试，它具有良好的交互式界面。

首先使用 `conda list` 命令查看当前环境中所有已经安装的库。

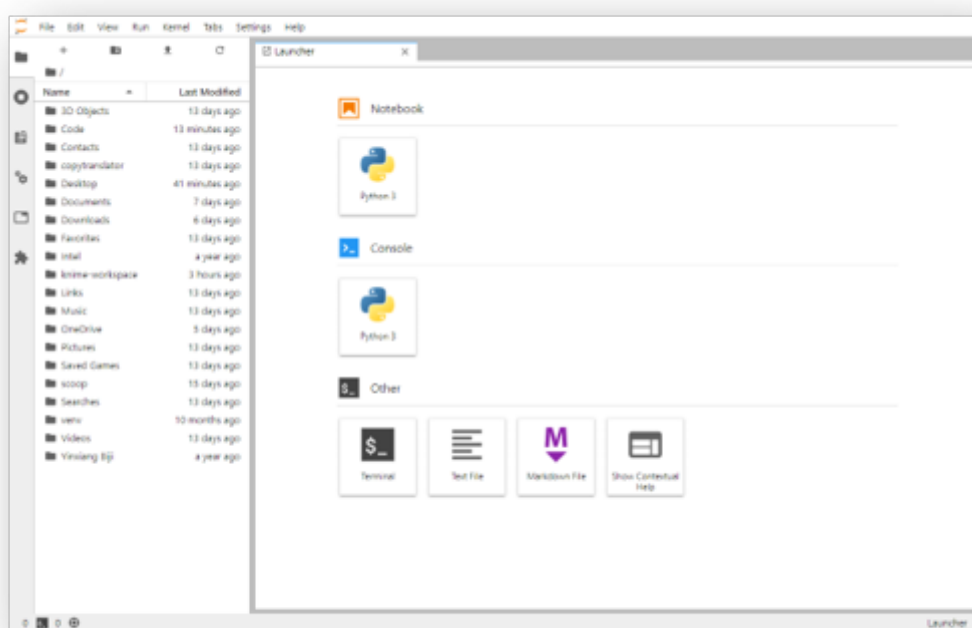
命令行切换至test环境后，输入 `conda install jupyterlab` 后回车安装。

安装完成后，测试一下是否能够运行Jupyter。

通过Windows资源管理器在D盘或者其他盘新建一个文件夹，以后可以把自己写的代码都放在里面。假设我们在D盘新建了一个mycode的文件夹。

在命令行中首先输入 `D:` 切换至D盘，然后输入 `cd mycode` 后回车，将命令行当前目录切换为mycode。

在命令行中输入 `jupyter lab` 后回车，稍等一下会自动启动浏览器，显示如下界面，说明你的编程环境安装成功了，你可以新建一个Notebook写一个Hello World程序。



步骤7 安装Git（可选）

在[下载页面](#)中选择合适的版本下载安装包并运行安装程序。

实验1-3 Git初试（选做）

步骤1 设置个人信息

在多人协作开发中需要明确代码编辑者的信息，所以在使用Git前我们首先设置一下自己的个人信息

启动Git Bash，输入 `git config --global user.name "Your Name"` 后回车（其中引号内的Your Name替换成你的英文名）

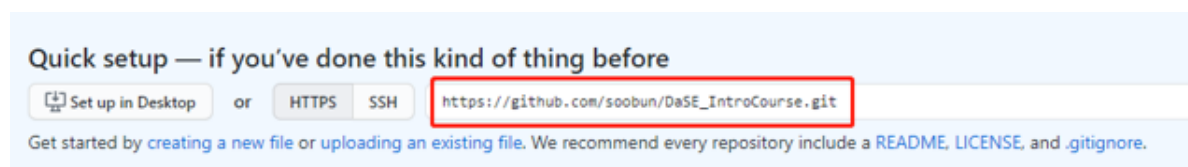
然后再输入 `git config --global user.email your_email@xyz.com` 后回车（替换为你的电子邮箱地址）

步骤2 克隆远程仓库至本地

假设已在GitHub上新建了一个名为"DaSE_IntroCourse"的仓库

登录至GitHub并进入个人主页，单击"DaSE_IntroCourse"仓库

在仓库"Quick Setup"提示界面中复制代码仓库地址



启动Git Bash，通过 `cd` 命令切换至你想存放代码的文件夹

输入 `git clone`（注意clone后再输入一个空格）后**不要按**回车，在命令行界面按一下鼠标右键即可粘贴刚才复制的地址，然后再按下回车键，开始克隆远程仓库至本地

步骤3 提交第一份代码

通过 `cd` 命令进入代码仓库文件夹，然后输入 `code` . 启动Visual Studio Code，新建一个简单的Python程序文件并保存（假设文件名为hello.py）

返回命令行窗口，输入 `git add hello.py` 后回车，将Python程序文件加入“暂存区域”

输入 `git commit -m "Add the hello.py file."`（`-m` 参数后引号内的内容为提交注释，可根据实际情况自定义注释）后回车，将程序文件正式提交至版本仓库中

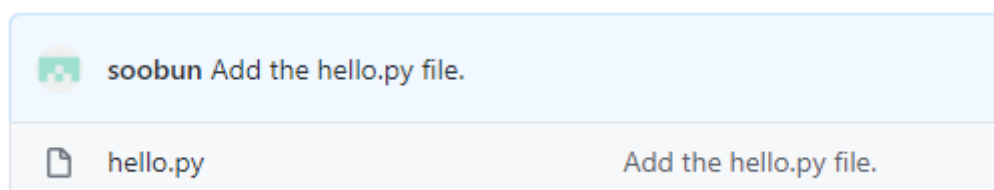
步骤4 推送至远程仓库

不过目前最新的版本仓库仍在我们本地主机，为了以后供多人协作和代码备份，我们需要将本地仓库推送（Push）至远程仓库GitHub中

在命令行中输入 `git push` 后回车，将本地仓库推送至GitHub

可能会弹出登录GitHub的窗口，输入GitHub账号和密码后登录即可

通过浏览器登录至GitHub并进入仓库查看，发现GitHub中保存了刚才新建的hello.py文件及版本注释内容



实验练习01

1. 在自己的电脑中完成环境配置
2. 创建一个GitHub账号并新建一个仓库
3. 完成Python语言的[入门课程](#)