

第 1 章 机器学习环境安装



第 1 章 环境配置

所谓工欲善其事必先利其器，因此接下来首先需要完成的任务就是将后续所要用到的环境进行安装。总体来说安装过程主要可以分为两大部分：一是 Python 管理环境的安装和配置；另一个就是 IDE 的安装和配置。

1.1 安装 Conda

作为在 Python 开发中一款优秀的包管理工具，Conda 一直以来就有着其独特的优势，尤其是在机器学习和深度学习的开发中。例如最新版本的 Conda 在安装 TensorFlow-gpu 版本时，如果是通过 `conda install` 来进行安装，它能够自动根据 TensorFlow 的版本匹配好对应的 Cuda 驱动以及 cuDNN 的版本号，这一点可谓十分友好。因此下面，笔者就来介绍其基本的安装与使用。

1.1.1 Windows 环境

首先在官网^①下载最新版 Windows 平台下的 Anaconda3 安装包，然后再按照如下安装步骤进行即可。这里顺便提一下，安装 Anaconda 的目的主要是为了使用里面的 Conda 环境管理器，因此这里下载安装的是 Miniconda^②。Anaconda 和 Miniconda 本质上都一样，Anaconda 是拓展自 Miniconda，里面包含了更多的 Python 包，因此也比较大。由于需要创建自己的虚拟环境，所以可以下载更加小巧的 Miniconda（安装过程完全一样）。

1) 安装 Anaconda

双击.exe 后缀的安装包进行安装，如果后续无特殊说明，保持默认直接点击 Next 按钮即可，如图 1-1 所示。

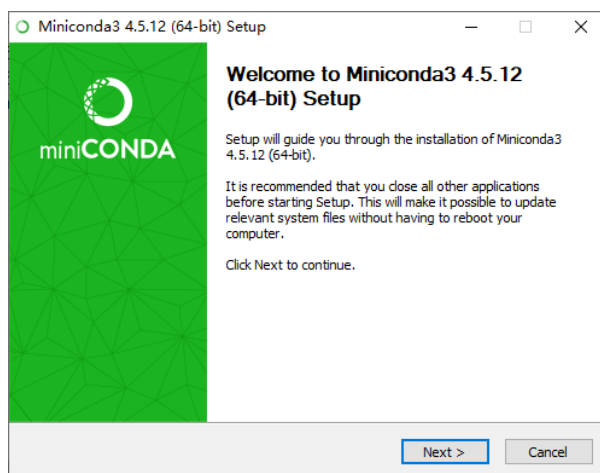


图 1-1 Anaconda 安装图

2) 指定安装目录

在安装过程中还可以自定义指定安装路径，但一般情况下保持默认安装路径即可，如图 1-2 所示。

^① <https://www.anaconda.com/distribution/>

^② <https://docs.conda.io/en/latest/miniconda.html>

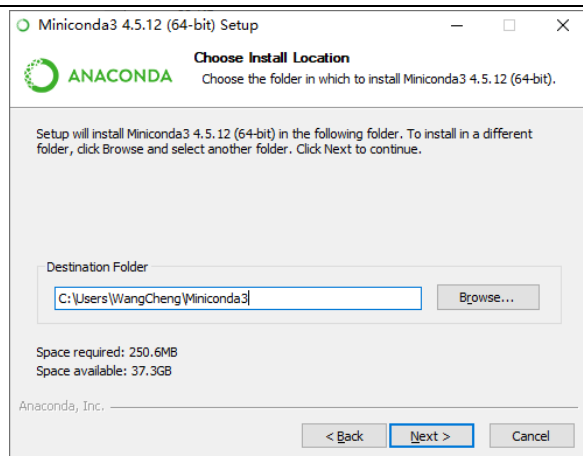


图 1-2 Anaconda 安装路径图

3) 高级设置

当安装过程执行到这一步时, 直接点击 **Install** 即可, 不用勾选任何选项, 如图 1-3 所示。

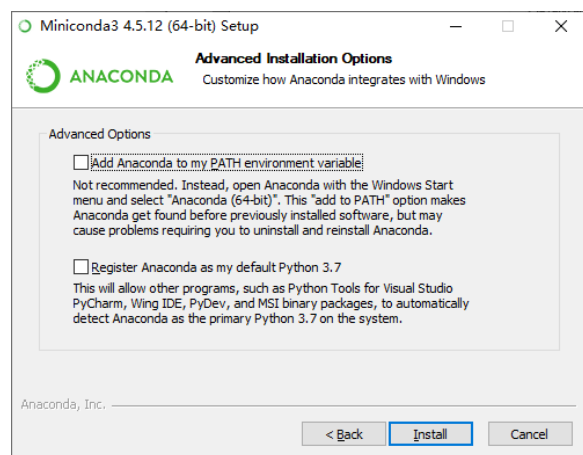


图 1-3 高级设置图

4) 安装完成

安装完成后, 点击 **Finish** 按钮, 如图 1-4 所示。接下来可以通过打开命令行然后输入相关命令来测试是否安装成功。

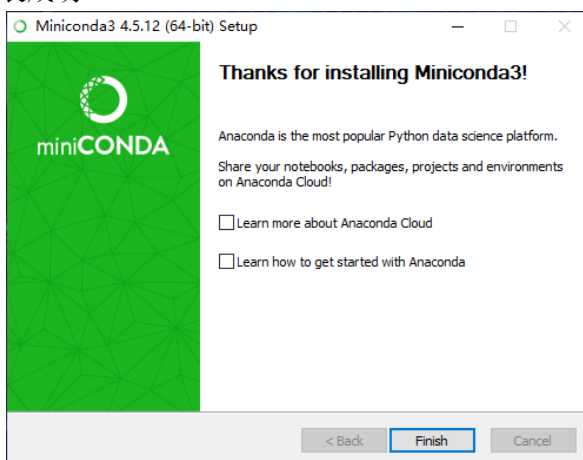


图 1-4 Miniconda 安装完成图



5) 测试

当完成上述安装过后，便可以在开始菜单栏中找到 Anaconda Prompt 命令行，点击打开后并输入 `conda -V`，如果出现相关版本信息则表示安装成功，如图 1-5 所示。

```
■ Anaconda Prompt

(base) C:\Users\WangCheng>conda -V
conda 4.5.12

(base) C:\Users\WangCheng>■
```

图 1-5 测试安装图

1.1.2 Linux 环境

1) 下载 Miniconda

首先，需要在地址^①中找到对应版本的 Miniconda 安装包，并复制对应的链接地址；然后，通过 Linux 中的 `wget` 命令来完成安装包的下载（这里以下载最新版本的 Miniconda 为例），代码如下：

```
# 下载 Miniconda:
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

如果由于网络原因不能完成上述下载过程，也可以从清华大学维护的镜像中^②找到相应 Anaconda 的下载地址，然后同样以 `wget` 命令来进行下载，代码如下：

```
# 下载 Anaconda:
wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-5.3.1-Linux-x86_64.sh
```

2) 安装 Miniconda

在完成安装包下载后，再通过 `bash Miniconda3-latest-Linux-x86_64.sh` 命令进行安装。如果上面下载的是 Anaconda，则对应安装命令为 `bash Anaconda3-5.3.1-Linux-x86_64.sh`，要注意区分。

在上述安装的过程中，一直按回车键即可。在遇到如图 1-6 所示的情况时，输入 `yes`，继续按回车键，直到安装结束。如果没有看到这一步也无妨，继续进行即可。

```
Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes■
```

图 1-6 初始化 Miniconda

^① <https://repo.anaconda.com/miniconda/>

^② <https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>



3) 测试

在安装结束后打开命令行终端，然后输入 `conda -V` 进行测试。如果出现如图 1-7 所示的版本提示信息，则表示安装成功。

```
(base) test@VM-0-15-ubuntu:~$ conda -V
conda 4.8.3
(base) test@VM-0-15-ubuntu:~$
```

图 1-7 测试安装图

但如果出现 `conda: command not found` 的提示，试着执行命令 `source ~/.bashrc`，然后再执行测试命令。在这之后如果依旧提示找不到 `conda` 命令，则再试试通过如下两行命令手动添加环境变量，代码如下：

```
echo 'export PATH="/home/username/anaconda3/bin:$PATH"' >> ~/.bashrc
source ~/.bashrc
```

然后再执行测试命令便能看到正确的版本提示信息。

1.2 替换源

在安装完成 Miniconda 后（无论是在哪个平台下），为了加快后续 Python 包安装过程中的下载速度，这里需要将默认的 conda 源和 pip 源替换成清华对应的镜像源。替换方式如下：

❑ 替换 conda 源

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda
/pkgs/free/
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda
/pkgs/main/
conda config --set show_channel_urls yes
```

❑ 替换 pip 源

```
pip install pip -U
pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```

如果只是临时使用一下某个 pip 源的话，可以用如下方式进行 Python 包的安装：

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

1.3 Conda 环境管理

由于在实际项目开发过程中，可能会根据情况使用到不同版本的 Python 解释器或者是一些相互不兼容的 Python 包，例如一个项目依赖的 Python 版本是 3.6 而另外一个却是 2.7，显然这两者是不能够同时存在于一个环境中。此时，便可以通过 Miniconda 中的 Conda 环境管理器来进行方便的创建与管理。接下来，将会依次介绍虚拟环境的安装与使用。

1.3.1 虚拟环境安装

在完成 Miniconda 安装后，可以通过使用 `conda create -n env_name` 来创建一个名为 `env_name` 的虚拟环境。同时，如果你需要一个特定的 Python 版本，那么可以通过命令 `conda create -n env_name python=3.6` 来创建一个名为 `env_name`，Python 版本为 3.6 的虚拟环境。

下面，以安装一个名为 `py36`，且同时指定 Python 版本为 3.6 的过程为例进行演示。



1) 创建新环境

输入 `conda create -n py36 python=3.6` 创建新环境，如图 1-8 所示。

```
(base) test@VM-0-15-ubuntu:~$ conda create -n py36 python=3.6
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.8.3
  latest version: 4.9.2
```

图 1-8 虚拟环境安装图

2) 继续安装

在执行上一步的命令后，便会看到如图 1-9 所示的提示内容，直接按回车键即可。从图中可以看到，上一步的命令将会安装一个 Python 版本为 3.6.12 的虚拟环境。

```
pip                pkgs/main/linux-64::pip-20.3.3-py36h06a4308_0
python             pkgs/main/linux-64::python-3.6.12-hcff3b4d_2
readline           pkgs/main/linux-64::readline-8.0-h7b6447c_0
setuptools         pkgs/main/linux-64::setuptools-51.1.2-py36h06a4308_4
sqlite             pkgs/main/linux-64::sqlite-3.33.0-h62c20be_0
tk                 pkgs/main/linux-64::tk-8.6.10-hbc83047_0
wheel              pkgs/main/noarch::wheel-0.36.2-pyhd3eb1b0_0
xz                 pkgs/main/linux-64::xz-5.2.5-h7b6447c_0
zlib               pkgs/main/linux-64::zlib-1.2.11-h7b6447c_3

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

图 1-9 Python 环境安装过程图

3) 完成安装

出现如图 1-10 所示的提示表示安装成功。

```
Downloading and Extracting Packages
pip-20.3.3                | 1.8 MB |
setuptools-51.1.2         | 730 KB |
certifi-2020.12.5         | 140 KB |
libffi-3.3                | 50 KB  |
sqlite-3.33.0             | 1.1 MB |
ca-certificates-2020      | 121 KB |
libedit-3.1.20191231     | 116 KB |
tk-8.6.10                 | 3.0 MB |
openssl-1.1.1i            | 2.5 MB |
python-3.6.12             | 29.7 MB |
wheel-0.36.2              | 33 KB  |
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

图 1-10 安装完成示意图

如果是在后续使用过程中想再次更换某个环境中的 Python 版本，则可以先进入对应的虚拟环境，然后用以下命令来完成 Python 版本的更换，代码如下：

```
conda install python==3.6.7
```



1.3.2 虚拟环境使用

在完成环境的创建后，可以通过命令 `conda activate env_name` 进入到对应的虚拟环境。同时，还可以使用命令 `conda env list` 来列出当前存在的所有虚拟环境；通过 `conda remove -n env_name --all` 来删除名为 `env_name` 的虚拟环境。同时如果需要在对应的虚拟环境中安装相应的 Python 包，那么可以使用 `pip install package_name` 来完成，如图 1-11 所示。最后，使用 `conda deactivate` 命令退出相应的虚拟环境。

```
(py36) test@VM-0-15-ubuntu:~$ pip install numpy
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting numpy
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/14/34569f38f3a08/numpy-1.19.5-cp36-cp36m-manylinux2010_x86_64.whl
    |████████████████████| 14.8 MB 475 kB/s
Installing collected packages: numpy
Successfully installed numpy-1.19.5
```

图 1-11 Python 包安装示意图

如果需要安装值得版本号的 Python 包可以通过如下命令进行

```
pip install jieba==0.42.1
```

1.4 PyCharm 安装与配置

在 Python 开发中，最常用的 IDE 就是 PyCharm，从名字也可以看出它是专门为 Python 开发而设计。首先需要去 PyCharm 官网^①下载离线安装包，如图 1-12 所示。

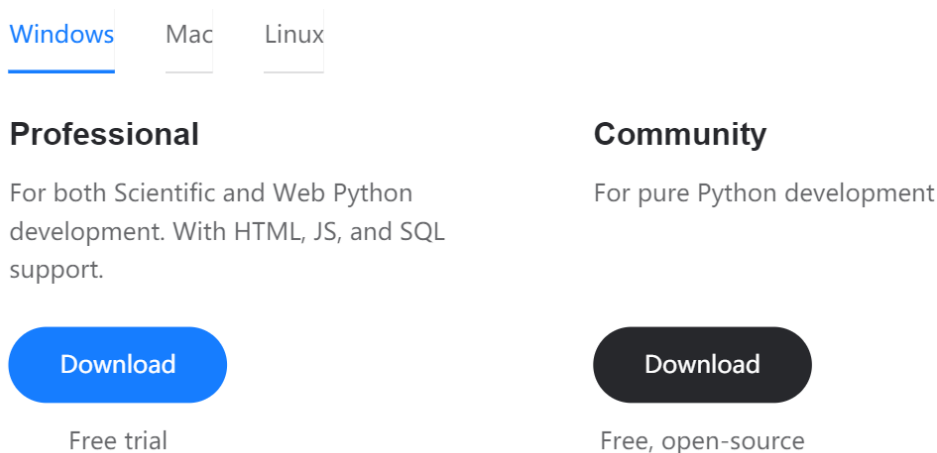


图 1-12 下载 PyCharm

页面提供了两种版本：专业版和社区版，前者收费后者免费。对于初学者来说社区版的就已经足够了。点击 **Download** 按钮，然后等待下载完成。

1) 安装 PyCharm

双击下载好的安装包，然后持续点击 **Next** 按钮。当执行到，图 1-13 所示的位置时，可以勾选上此处的两个选项，然后接着继续点击 **Next** 按钮即可安装完成。

^① <https://www.jetbrains.com/pycharm/download/#section=windows>

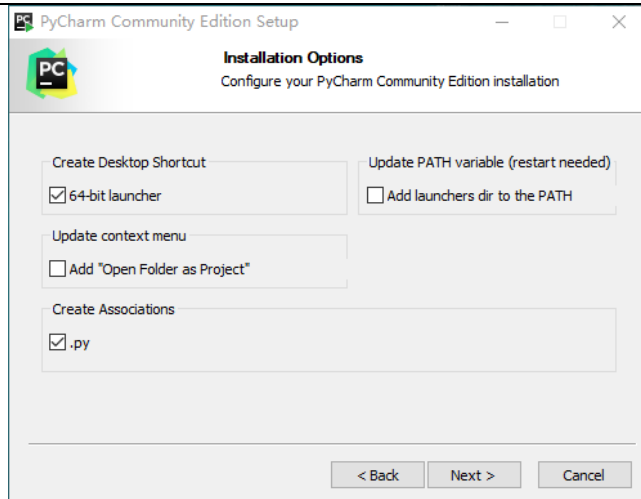


图 1-13 PyCharm 安装选择图

2) 配置 PyCharm

在安装完成后双击 PyCharm 图标，在第一次打开时可能会有如图 1-14 所示的提示。

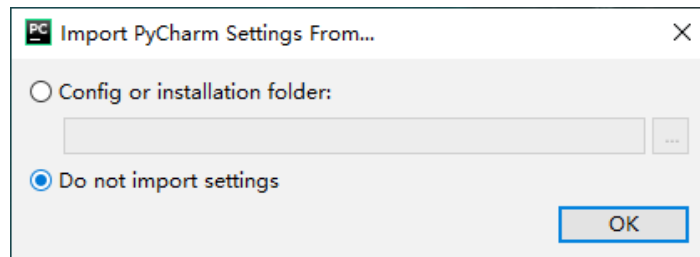


图 1-14 PyCharm 启动图

此时选择 Do not import settings，点击 OK 按钮即可。最后，再点击图 1-15 所示的 New Project 按钮创建一个新的工程。

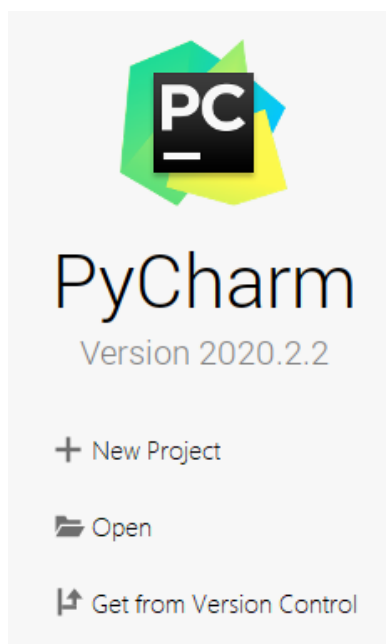


图 1-15 创建新工程图

按照如图 1-16 所示的内容输入相应的工程名称和选择对应的 Python 解释器。

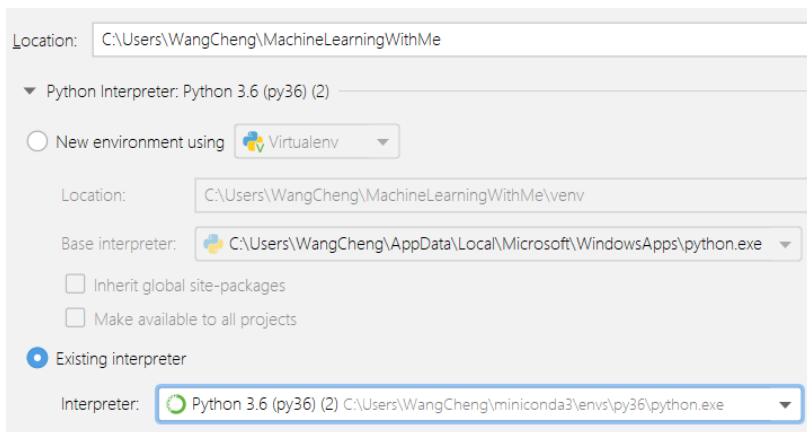


图 1-16 Python 工程配置图

通常图 1-16 中 Interpreter 的路径为 C:\Users\Username\Miniconda3\envs\py36\python.exe，且这里安装的是 Miniconda，所以路径里是 Miniconda3，其具体的选择方法如图 1-17 所示。

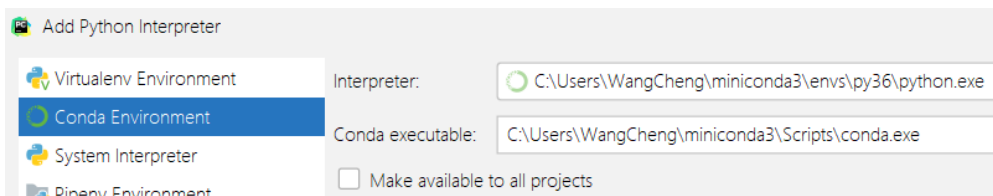


图 1-17 Python 解释器选择图

在解释器选择完成后点击 OK 按钮，便能够回到图 1-16 所示的页面，最后点击 Create 按钮即可完成工程的创建。

3) 更换解释器

如果后续过程中需要更换虚拟环境（解释器），则可先点击 File→Settings，再点击其中的 Project Interpreter，然后点击右上角的设置按钮，如图 1-18 所示。这样便可以回到图 1-16 所示相同的配置页面，最后选择相应的环境即可。

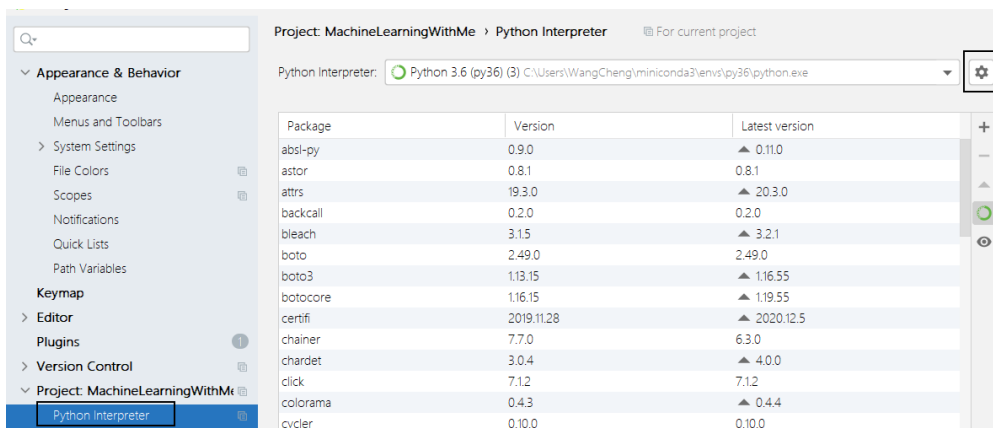


图 1-18 Python 解释器更换图

4) 运行示例

鼠标指向工程名，单击 New，然后选择 Python File，输入文件名即可创建新的 Python 文件，如图 1-19 所示。

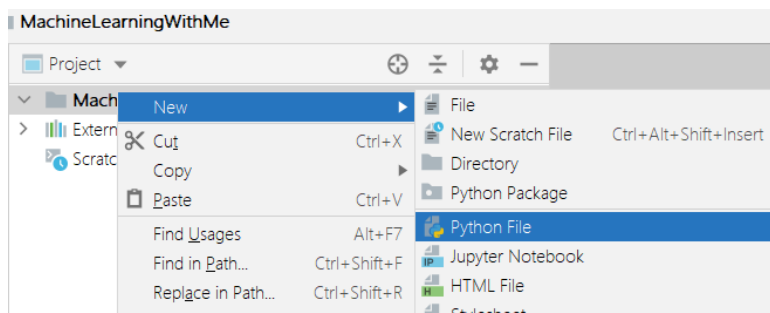


图 1-19 Pycharm 新建文件图

在空白处输入代码后，右击选择 Run test 即可运行该程序，如图 1-20 所示，也可使用快捷键 Ctrl+Shift+F10。

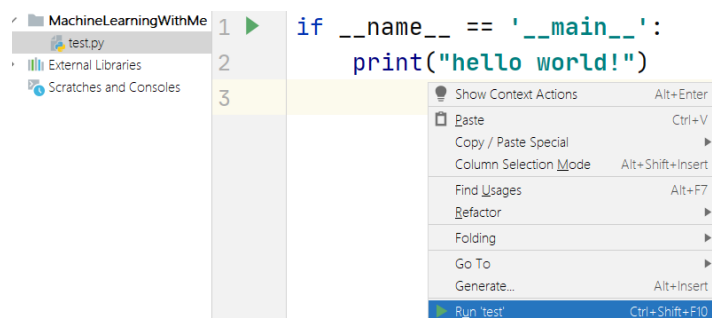
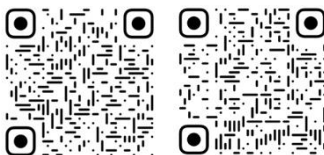


图 1-20 代码运行图

1.5 小结

在本章中，笔者首先介绍了如何在 Windows 和 Linux 两种环境中来安装和配置 Conda 管理器；接着介绍了如何一步步来创建一个新的虚拟环境和安装 Python 包；最后介绍了如何下载安装和配置 PyCharm 开发环境，同时还以一行简单的代码进行了示例。到这里，详细大家对于 Python 环境的开发与配置已经有了一个较为清晰的认知。

本次内容就到此结束，感谢您的阅读！如果你觉得上述内容对你有所帮助，欢迎分享至一位你的朋友！若有任何疑问与建议，请添加笔者微信'nulls8'或加群进行交流。青山不改，绿水长流，我们月来客栈见！



扫码关注月来客栈可获得更多优质内容！