

第 3 章 大数据开发工具装备

★本章导读★

本章将对大数据开发所使用的语言以及工具的安装及基本使用做详细的介绍。工欲善其事必先利其器，了解开发语言和开发工具，才能更快的上手进行大数据处理、分析的开发工作。本章会带读者安装 Python 及 Scala 开发环境，并运行第一个 Scala 程序，开启大数据开发的帷幕。

★知识要点★

通过本章内容的学习，读者将掌握以下知识：

- Python、Scala 语言基本语法
- Anaconda 及 Maven 安装配置
- Jupyter Notebook 和 IDEA 工具的使用
- 写出第一个 Scala 程序

3.1 大数据开发语言

大数据开发语言众多，本节将对大数据处理、分析常用的 Python 和 Scala 语言进行介绍。本书后续的代码开发内容均基于 Python 语言实现，使用 Python 语言是为了帮助读者更快入门、上手大数据，并且 Python 语言也是数据处理分析必备利器。然而 Spark 是基于 Scala 语言进行开发的，所以我们也有必须对 Scala 语言有一定认识。

3.1.1 Python 语言介绍

Python 是一种解释型、面向对象、动态数据类型的高级程序设计语言。Python 由 Guido van Rossum 于 1989 年底发明，第一个公开发行版发行于 1991 年。像 Perl 语言一样，Python 源代码同样遵循 GPL(GNU General Public License) 协议。Python 的设计具有很强的可读性，相比其他语言经常使用英文关键字，其他语言的一些标点符号，它具有比其他语言更有特色的语法结构，图标如图 3-1 所示。



图 3-1 Python 语言图标

Python 语言具有以下特点。

1. 易于学习：Python 有相对较少的关键字，结构简单，语法定义明确，学习起来更加简单。
2. 易于阅读：Python 代码定义更清晰。

3. 易于维护: Python 的成功在于它的源代码是相当容易维护的。
4. 一个广泛的标准库: Python 的最大的优势之一是丰富的库, 可跨平台的, 在 UNIX, Windows 和 Macintosh 兼容很好。
5. 可移植性好: 基于其开放源代码的特性, Python 已经被移植到许多平台。
6. 可扩展性强: 开发人员可以使用 C 或 C++ 完成部分程序, 然后从 Python 程序中进行调用。

3.1.2 Scala 语言介绍

Scala (Scala Language 的简称) 语言是一种能够运行于 JVM 和 .Net 平台之上的通用编程语言, 是一门多范式 (Multi-Paradigm) 的编程语言, 设计初衷是要集成面向对象编程和函数式编程的各种特性。它既可用于大规模应用程序开发, 也可用于脚本编程。它由 Martin Odersky 于 2001 开发, 2004 年开始运行在 JVM 与 .Net 平台之上, 由于其简洁、优雅、类型安全的编程模式而受到关注。Scala 兼容现有的 Java 程序, Scala 源代码被编译成 Java 字节码, 所以它可以运行于 JVM 之上, 并可以调用现有的 Java 类库, 图标如图 3-2 所示。



图 3-2 Scala 语言图标

Scala 语言具有以下特性。

1. 面向对象特性: Scala 是一种纯面向对象的语言, 每个值都是对象。对象的数据类型以及行为由类和特质描述。
2. 函数式编程: Scala 也是一种函数式语言, 其函数也能当成值来使用。Scala 提供了轻量级的语法用以定义匿名函数, 支持高阶函数, 允许嵌套多层函数, 并支持柯里化。
3. 静态类型: Scala 具备类型系统, 通过编译时检查, 保证代码的安全性和一致性。类型系统具体支持泛型类、协变和逆变、类型参数的上下限约束、复合类型等特性。
4. 扩展性: Scala 提供了许多独特的语言机制, 可以以库的形式轻易无缝添加新的语言结构, 可以根据预期类型自动构造闭包。
5. 并发性: Scala 使用 Actor 作为其并发模型, Actor 是类似线程的实体, 通过邮箱收发消息。Actor 可以复用线程, 因此在程序中可以使用数百万个 Actor, 而线程只能创建数千个。在 2.10 之后的版本中, 使用 Akka 作为其默认 Actor 实现。

3.2 Anaconda 安装与配置

3.2.1 Anaconda 介绍

Anaconda 是当前最流行的 python 包管理工具, 因为其功能的便捷性和高效性, 同时支持多种开源的深度学习框架安装, 受到许多从事深度学习的科研工作者以及老师学生的青睐, 图标如图 3-3 所示。

Anaconda 指的是一个开源的 Python 发行版本, 其包含了 conda、Python 等 180 多个科学包及其依赖项。安装 Anaconda 软件包的同时会安装很多其他软件包, 包括 Ipython、Jupyterb Notebook、Numpy、

Scipy、Matplotlib，这些都是数据分析、科学计算常用的软件包。



图 3-3 Anaconda 集成工具图标

Jupyter Notebook 是一种交互式界面工具，我们可以在 Web 界面进行登录使用，每行 Python 代码都可以单独执行，即时看到运行结果，并且运行结果会自动存储，之后再次打开笔记的时候仍然可以使用上次运行的结果，非常方便开发人员调试使用。Jupyter Notebook 可以支持多种语言，如 Python、Java、Shell 等等，图标如图 3-4 所示。



图 3-4 Jupyter Notebook 工具图标

Jupyter Notebook 非常适合数据处理、数据分析人员进行数据分析的开发工作，它强大的交互式界面可以帮助开发人员调试程序，因此，在本书后续章节中，我们会使用 Jupyter Notebook 演示所有的 Spark 代码程序。

3.2.2 Anaconda 下载与安装

本小节将向读者详细介绍 Anaconda 的安装，具体安装步骤如下：

步骤 01：下载 Anaconda

打开 Anaconda 的下载网站 <https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>，选择合适的版本进行下载，读者可以直接在 Linux 中进行下载。如图 3-5 所示。

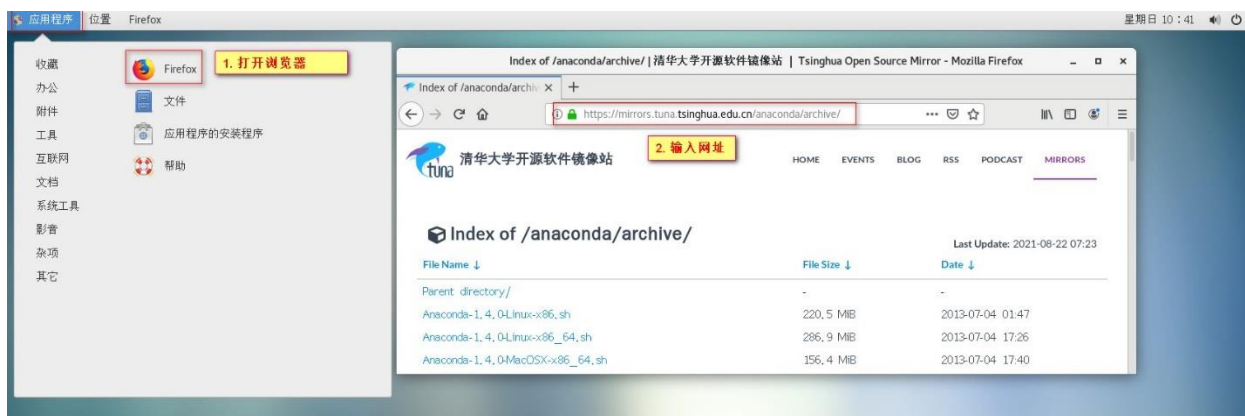


图 3-5 下载 Anaconda

本书使用的 Anaconda 版本是 Anaconda3-5.2.0-Linux-x86_64.sh, 如图 3-6 所示。

Anaconda3-5.2.0-Linux-ppc64le.sh	288.3 MiB	2018-05-31 02:37
Anaconda3-5.2.0-Linux-x86.sh	507.3 MiB	2018-05-31 02:37
Anaconda3-5.2.0-Linux-x86_64.sh	621.6 MiB	2018-05-31 02:38
Anaconda3-5.2.0-MacOSX-x86_64.pkg	613.1 MiB	2018-05-31 02:38
Anaconda3-5.2.0-MacOSX-x86_64.sh	523.3 MiB	2018-05-31 02:39
Anaconda3-5.2.0-Windows-x86.exe	506.3 MiB	2018-05-31 02:41
Anaconda3-5.2.0-Windows-x86_64.exe	631.3 MiB	2018-05-31 02:41
Anaconda3-5.3.0-Linux-ppc64le.sh	305.1 MiB	2018-09-28 06:42

图 3-6 选择 anaconda 版本

由于本书是针对大数据处理分析, 而大数据集群环境均搭建在 Linux 系统中, 所以这里介绍 Linux 环境下 Anaconda 的安装, Linux 环境下 Anaconda 的安装更为复杂, 感兴趣的读者可以自行下载 Windows 版本 Anaconda 软件进行安装, 安装步骤非常简单。

步骤 02: 安装 Anaconda

在下载目录下打开终端, 输入如下命令进行安装。

```
sh Anaconda3-5.2.0-Linux-x86_64.sh
```

执行上述命令之后, 终端屏幕出现如图 3-7 所示内容。

```
[root@master src] # sh Anaconda3-5.2.0-Linux-x86_64.sh
Welcome to Anaconda3 5.2.0

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

图 3-7 Anaconda 安装命令

此时, 按下键盘【Enter】键, 然后连续三次按下【空格】键, 直至出现图 3-8 所示内容。

```
kerberos (krb5, non-Windows platforms)
  A network authentication protocol designed to provide strong authentication
  for client/server applications by using secret-key cryptography.

cryptography
  A Python library which exposes cryptographic recipes and primitives.

Do you accept the license terms? [yes|no]
[no] >>> yes
```

图 3-8 Anaconda 授权选择

这是 Anaconda 安装 License 的条款，输入 “yes”，然后按 “Enter” 键继续执行。接下来，会出现提示输入软件安装路径，本书安装的路径为 “/usr/local/src/anaconda3”，如图 3-9 所示。

```
[no] >>> yes

Anaconda3 will now be installed into this location:
/root/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/root/anaconda3] >>> /usr/local/src/anaconda3
```

图 3-9 Anaconda 路径选择

输入完安装路径之后，继续按【Enter】键，软件将开始安装，安装过程最后会提示是否添加环境变量以及是否安装 VSCode，这两处我们均输入 “no”，并按【Enter】执行，最后完成安装，如图 3-10 所示。

```
installation finished.
Do you wish the installer to prepend the Anaconda3 install location
to PATH in your /root/.bashrc ? [yes|no]
[no] >>> no

You may wish to edit your .bashrc to prepend the Anaconda3 install location to PATH:

export PATH=/usr/local/src/anaconda3/bin:$PATH

Thank you for installing Anaconda3!

=====

Anaconda is partnered with Microsoft! Microsoft VSCode is a streamlined
code editor with support for development operations like debugging, task
running and version control.

To install Visual Studio Code, you will need:
- Administrator Privileges
- Internet connectivity

Visual Studio Code License: https://code.visualstudio.com/license

Do you wish to proceed with the installation of Microsoft VSCode? [yes|no]
>>> no
[root@master src] #
```

图 3-10 Anaconda 环境变量设置选择

步骤 03：手动添加环境变量

在终端输入如下命令，使用 gedit 打开 “~/.bashrc” 文件为 Anaconda 添加环境变量。

```
gedit ~/.bashrc
```

在终端输入如下命令，使用 gedit 打开 “~/.bashrc” 文件为 Anaconda 添加环境变量，如图 3-11 所示。

```
export ANACONDA_HOME=/usr/local/src/anaconda3
export PATH=$PATH:$ANACONDA_HOME/bin
```




图 3-11 Anaconda 环境变量添加

步骤 04: 刷新生效环境变量

添加并保存之后关闭文件, 在终端执行如下命令使环境变量生效以及修改 python 默认版本软链接。

```
source ~/.bashrc
cd /usr/bin
rm -f python
ln -s /usr/local/src/anaconda3/bin/python python
```

步骤 05: 查看 Python 和 Anaconda 信息

输入如下命令查看 Python 版本。

```
python --version
conda info -e
```

运行后, 终端屏幕上显示如图 3-12 所示, python 版本为 3.6.5, anaconda 虚拟环境只有 root, 即默认安装版本。

```
[root@master bin]# python -V
Python 3.6.5 :: Anaconda, Inc.
[root@master bin]# conda info -e
# conda environments:
#
base                  * /usr/local/src/anaconda3
```

图 3-12 查看 conda 环境

步骤 06: 子节点安装 anaconda3

同样的方法, 在 slave1、slave2 机器上安装 anaconda。由于运行 Spark 可以采用 local 本地模式运行, 也可以采用 cluster 集群模式运行, 所以如果使用集群模式运行就必须在全部节点安装, 如果读者电脑配置较低, 也可以只在 master 节点安装, 后续章节 Spark 程序采用 local 模式运行即可。

3.2.3 Anaconda 镜像源配置

由于 Anaconda 并不是国产软件, 我们创建的环境以及包都需要自动下载安装成功后才能使用, 而 Anaconda 的服务器在国外, 所以在我们输入命令, 安装所需要的依赖包时, 国内下载速度十分缓慢, 常常出现下载断开无法安装的现象, 于是乎各种镜像就应运而生了, 本书采用了清华镜像源配置, 在终端依次输入如下命令便可以设置清华镜像源及查看设置成功后的信息, 如图 3-13 所示。

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

```
[root@master ~]# conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/
[root@master ~]# conda config --set show_channel_urls yes
[root@master ~]# conda info

active environment : None
user config file : /root/.condarc
populated config files : /root/.condarc
conda version : 4.5.4
conda-build version : 3.10.5
python version : 3.6.5.final.0
base environment : /usr/local/src/anaconda3 (writable)
channel URLs : https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/linux-64
               https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/noarch
```

图 3-13 Anaconda 清华源添加

3.2.4 Jupyter Notebook 配置

以上步骤就完成了 Anaconda 的所有安装步骤。本小节将详细介绍 Jupyter Notebook 的基本使用方法，详情如下所示。

步骤 01：初始化 Jupyter Notebook 配置

首先，我们需要对 Jupyter Notebook 进行一些简单配置，如访问用户权限、访问 IP 地址等等。在终端执行以下命令生成 Jupyter 的默认配置文件

```
jupyter notebook --generate-config
```

```
[root@master ~]# jupyter notebook --generate-config
Writing default config to: /root/.jupyter/jupyter_notebook_config.py
```

步骤 02：修改配置文件

首先，我们要为 Jupyter 生成登录密码（为了方便在 Windows 上访问方便），在终端输入“ipython”，进入 ipython 终端执行创建密码操作，如图 3-14 所示。

```
[root@master ~]# ipython
Python 3.6.5 |Anaconda, Inc.| (default, Apr 29 2018, 16:14:56)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.4.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from notebook.auth import passwd

In [2]: passwd()
Enter password:
Verify password:
Out[2]: 'sha1:f5c02e648ace:361841a95c828d886501e48bd8697a5cc5b6c3de'

In [3]: exit()
[root@master ~]#
```

图 3-14 Jupyter Notebook 密码生成

需要注意的是，在输入密码时，终端界面不会显示所输入的密码，所以为了避免出错，读者可以尽量输入简单的密码，本书采用的密码是“1”，在创建密码完成后，我们需要手动复制一下生成的密码对应的校验码串，即“Out[2]”输出的字符串。

使用 gedit 或者 vim 方式打开“/root/.jupyter/jupyter_notebook_config.py”文件，在打开的配置文件中找到如下代码。

```
## Whether to allow the user to run the notebook as root.
# 运行 root 登录
c.NotebookApp.allow_root = True

## The IP address the notebook server will listen on.
```

```
# 配置 “*” 代表可以所有 ip 地址进行访问
c.NotebookApp.ip = '*'

## The directory to use for notebooks and kernels.
# 此处路径可以自定义
c.NotebookApp.notebook_dir = u'/usr/code/'

# The string should be of the form type:salt:hashed-password.
# 此处密码内容为上一步复制的所生成的校验码字符串
c.NotebookApp.password = u'sha1:f5c02e648ace:361841a95c828d886501e48bd8697a5cc5b6c3de'
```

温馨提示：

步骤 03：启动 Jupyter Notebook

在终端中输入“jupyter notebook”命令之后，然后按“Enter”键，即可启动 jupyter，在终端可以看到启动日志，启动之后系统会自动弹出浏览器 Web 登录页面。

jupyter notebook

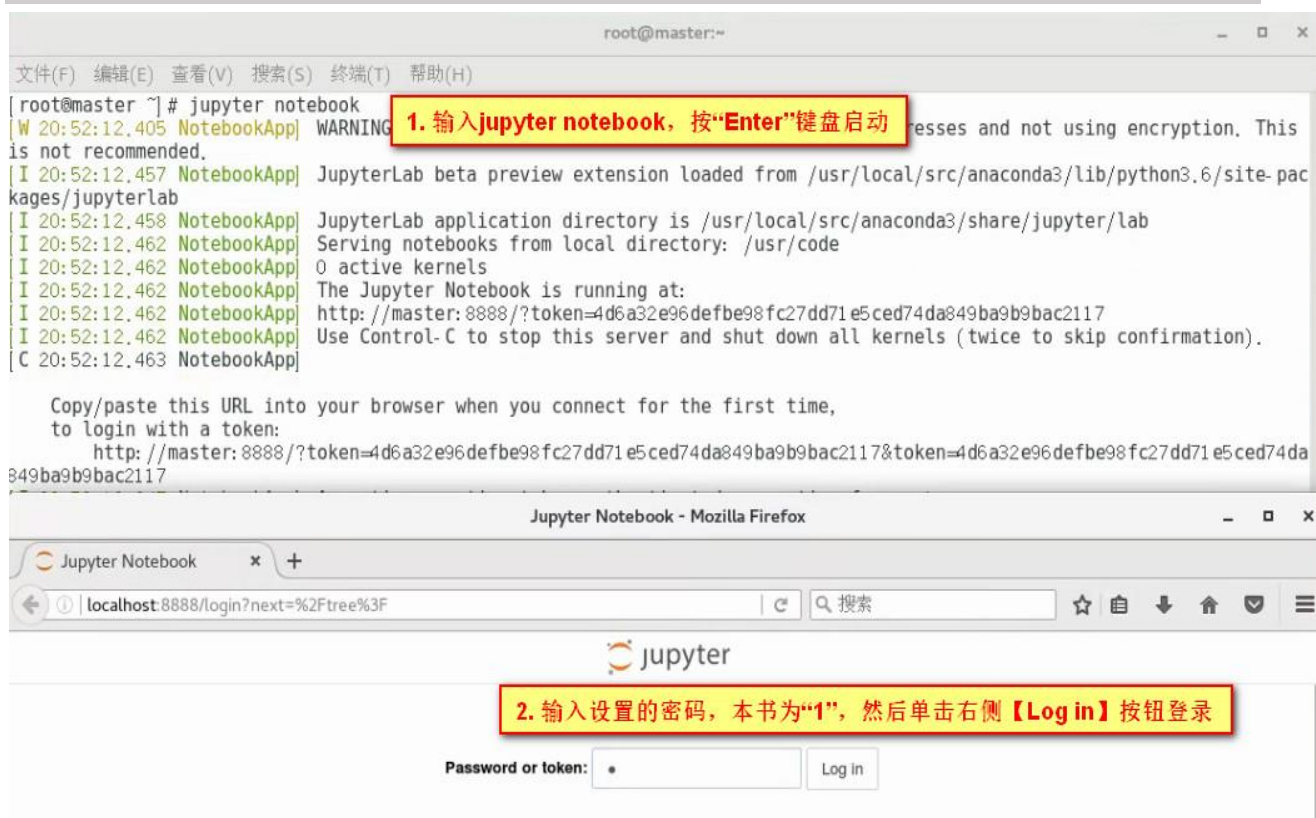


图 3-15 Jupyter Notebook 登录

当然也可以在 Windows 上的浏览器中进行访问，Jupyter Notebook 默认端口为 8888，在 Windows 上浏览器中输入访问地址如下所示。

192.168.75.100:8888

或

master:8888

3.2.5 Jupyter Notebook 初体验

在启动 Jupyter Notebook 之后，接下来，通过新建 Notebook、重命名 Notebook、添加/运行代码、插入代码单元格等基本操作使没有接触过 Jupyter Notebook 的读者能够快速上手掌握它的使用方法。

(1) 新建 Notebook

通过右上角【New】下拉菜单选择【Python3】，并确定，完成 Notebook 的新建工作，如图 3-16 所示。

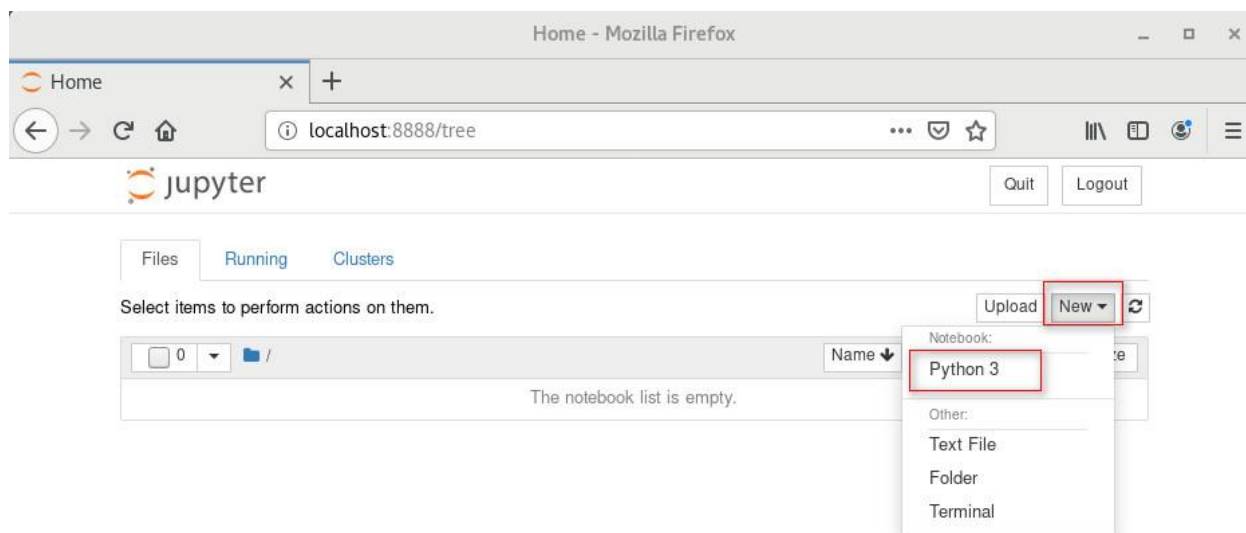


图 3-16 创建 Notebook 文件

(2) Notebook 重命名

首先单击【Untitled】，在弹出的【Rename Notebook】对话框中输入“First”，最后单击【Rename】按钮，为 Notebook 重命名，如图 3-17 所示。

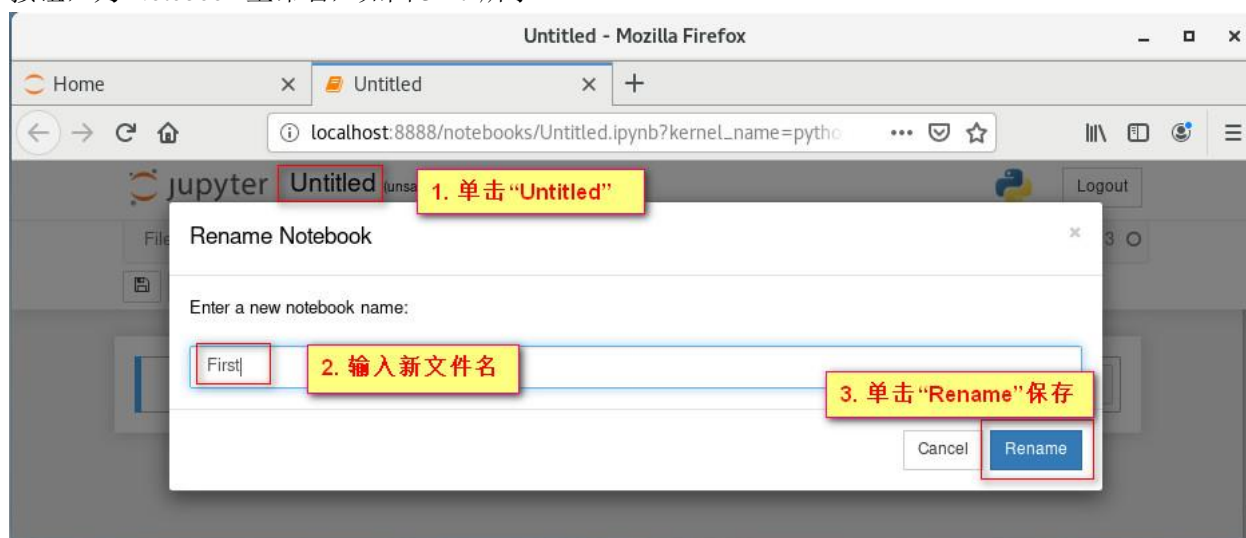


图 3-17 命名 Notebook 文件

(3) Notebook 运行代码

在上一步中完成了代码的编写，接下来进行代码的运行方法演示，点击要运行的代码行前方空白处，例如单击“In[1]”前方空白处，使得该行前方出现蓝色竖线，代表选中了该行，然后单击菜单栏中【Run】按钮，即可运行本行代码。例如“In[2]”中代码运行结果，运行代码方法如图 3-18 所示。

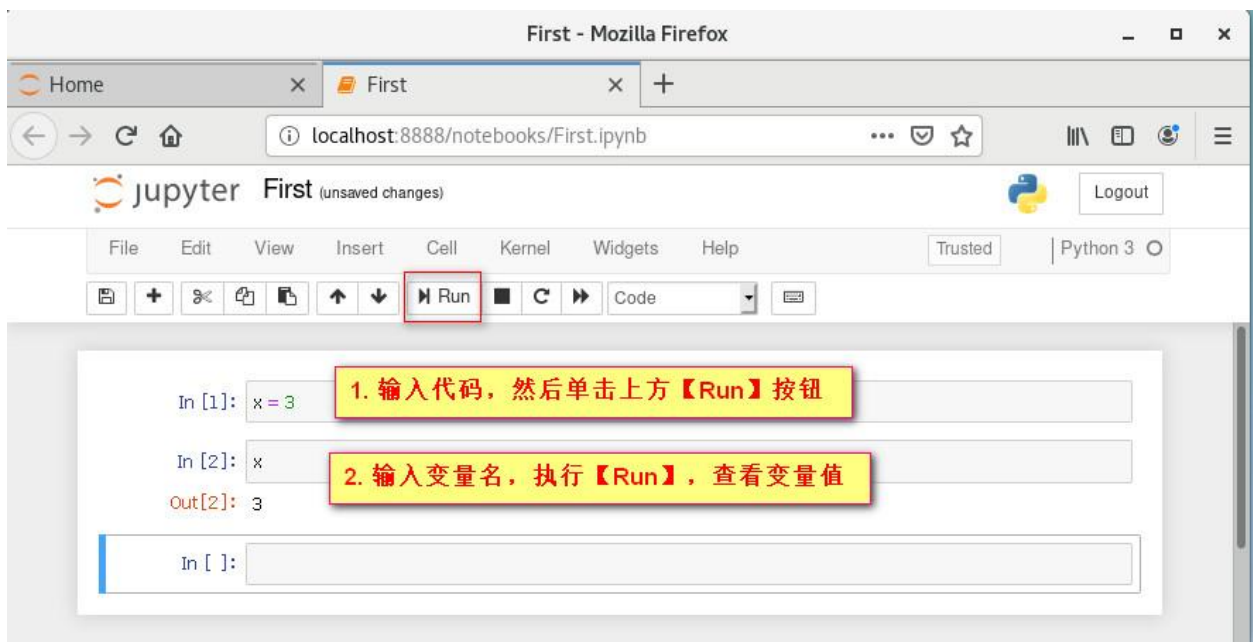


图 3-18 运行 Notebook 文件

(4) 查看变量值



从上图可以看出，Notebook 中有两部分构成，分别是“In[2]”和“Out[2]”，其中 In 代表输入代码或命令，方括号中的数字代表这是第几次的代码，数字会根据插入的代码输入行和运行代码的次数而自增，同样，Out 部分代表对应 In 部分代码的运行输出结果。Notebook 可以存储代码中间变量的结果，例如图中变量 x，直接在代码输入行中输入变量 x，然后进行运行，即可查看当前变量的值。

图 3-19 查看 Notebook 文件中变量值

(5) 插入单元格

上面步骤介绍了代码运行以及输入和输出的方法，当我们需要新增代码时，我们可以使用插入功能。点击菜单栏中的“Insert”菜单，在弹出的菜单栏中有两个选项，分别是“Insert Cell Above”和“Insert Cell Below”，分别代表在当前行的上方和下方插入新的代码行，如图 3-20 所示。

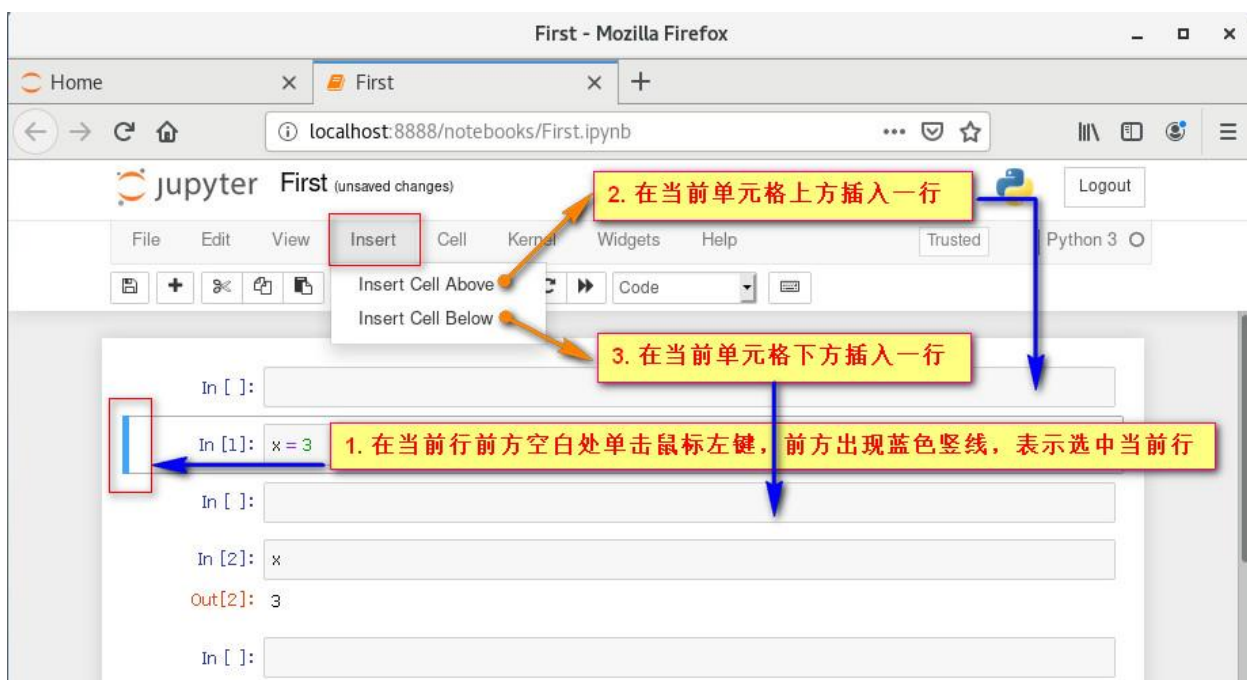


图 3-20 Notebook 文件插入行

(6) 添加注释

在 Notebook 中添加注释的方法和 Python 语法完全一致，下面展示单行注释和代码块注释的在 Notebook 中的添加，注释代码内容如图 3-21 所示。

```
In [3]: # 添加单行注释
        .....
        代码块注释
        .....
```

图 3-21 Notebook 文件添加注释

(7) 切换 Markdown 模式

Notebook 默认单元格为代码模式，即 Code。但是 Notebook 支持 Code、Markdown 等多种文本格式，下面我们介绍最常使用到的 Markdown 文本的编写，步骤如图 3-22 所示。

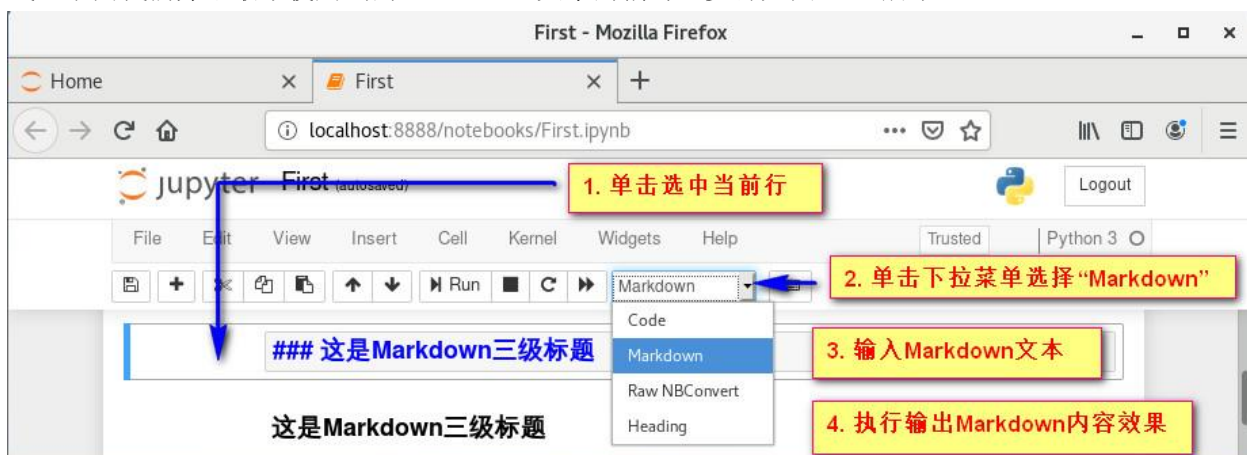


图 3-22 Notebook 文件切换 Markdown

(8) 关闭与打开 Notebook

Notebook 文件的关闭操作非常简单，和关闭普通网页没有区别，直接点击浏览器标签页的“×”按钮即可，如图 3-23 所示。

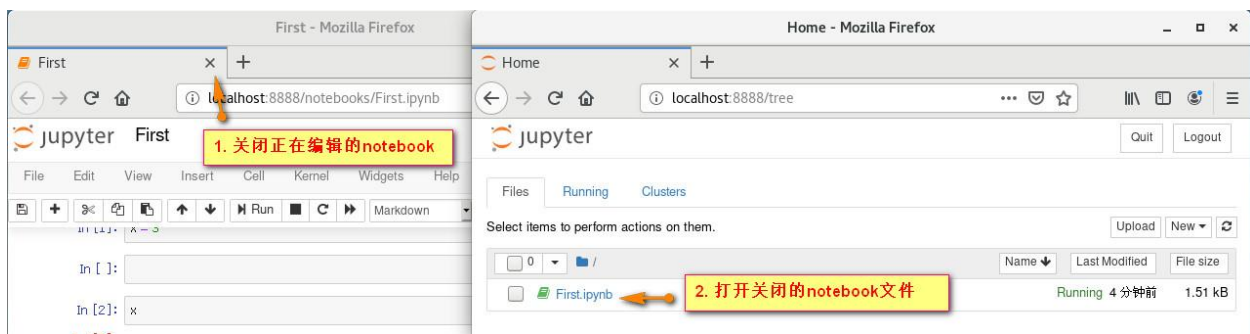


图 3-23 Notebook 文件关闭

3.3 Scala 安装

3.3.1 Windows 安装 Scala

Scala 下载官网为: <https://www.scala-lang.org/download/2.12.10.html> , 本书采用的是 2.12.10 版本, 需要分别下载 Windows 版本和 Linux 版本, 如图 3-24 所示。

You can find the installer download links for other operating systems, as well as documentation and source code archives for Scala 2.12.10 below.

Archive		System	Size
scala-2.12.10.tgz	Linux version	Mac OS X, Unix, Cygwin	19.71M
scala-2.12.10.msi	Windows version	Windows (msi installer)	124M
scala-2.12.10.zip		Windows	19.75M
scala-2.12.10.deb		Debian	144.88M
scala-2.12.10.rpm		RPM package	124.52M
scala-docs-2.12.10.tgz		API docs	53.21M
scala-docs-2.12.10.zip		API docs	107.63M
scala-sources-2.12.10.tar.gz		Sources	

图 3-24 Scala 安装包下载

Windows 具体安装步骤如下所示。

步骤 01: 双击 Scala 安装包 `scala-2.12.10.msi`, 如图 3-25 所示, 并单击【Next】按钮。

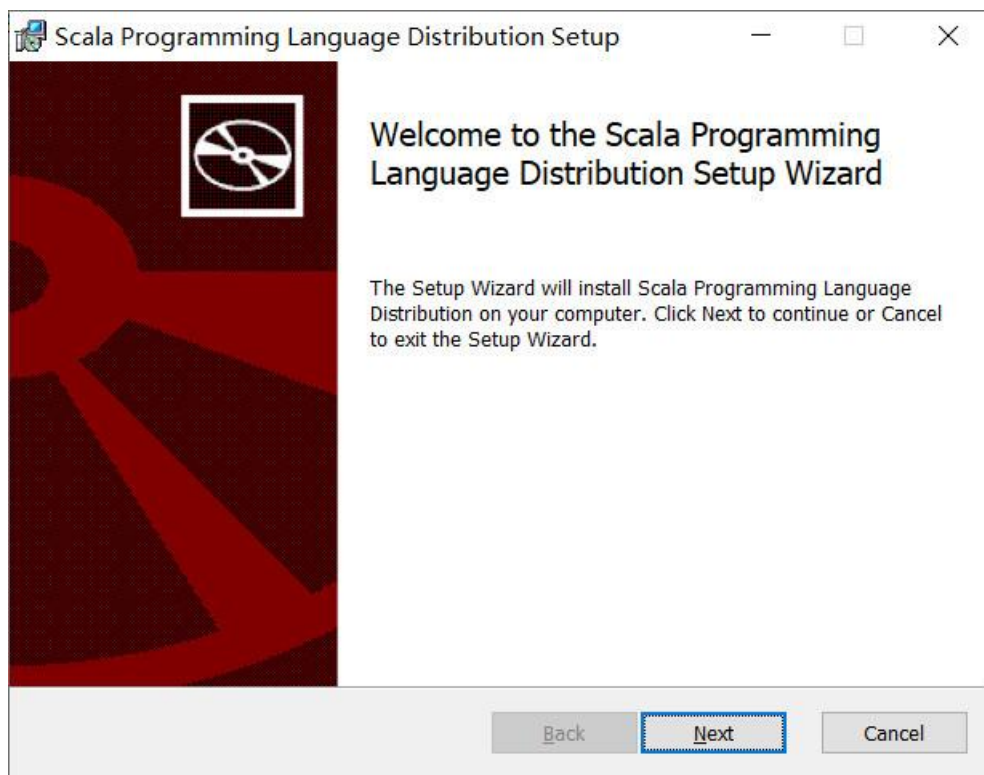


图 3-25 Scala Setup

步骤 02: 勾选接受许可，然后单击【Next】按钮，如图 3-26 所示。

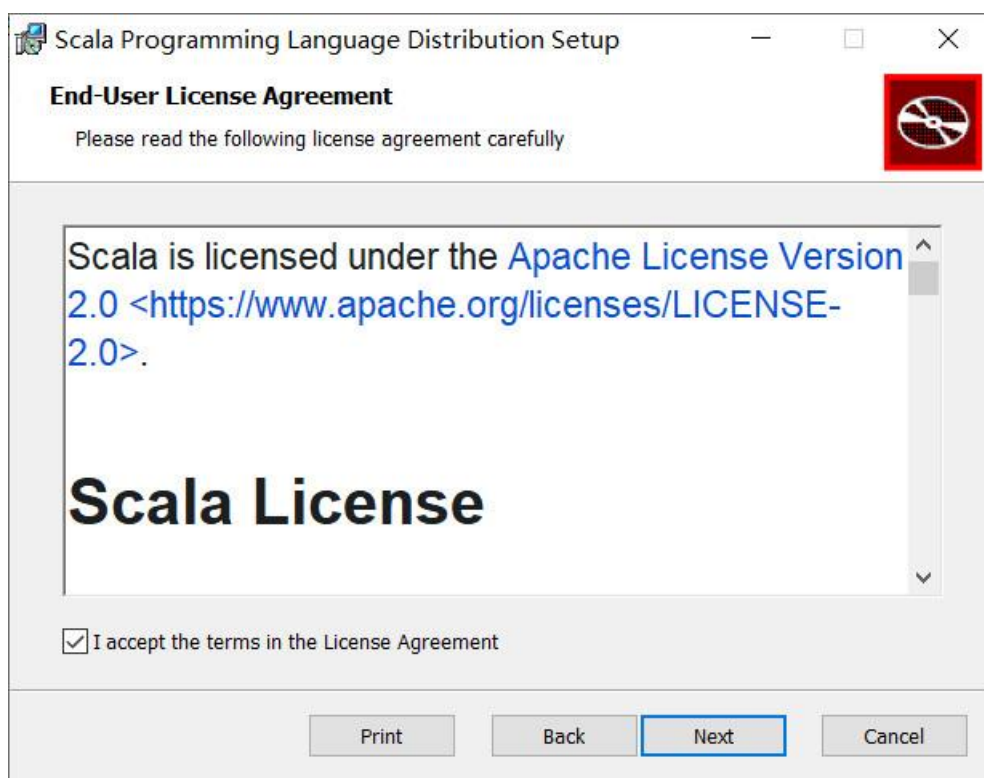


图 3-26 勾选接受许可

步骤 03: 接下来，单击【Browse...】按钮，选择 Scala 安装的路径，默认软件是安装在 C 盘位置。选择好安装位置之后，继续单击【Next】按钮，如图 3-27 所示。接下来单击【Install】按钮，进入安装界面，如图 3-28 所示，等待完成安装，如图 3-29 所示。

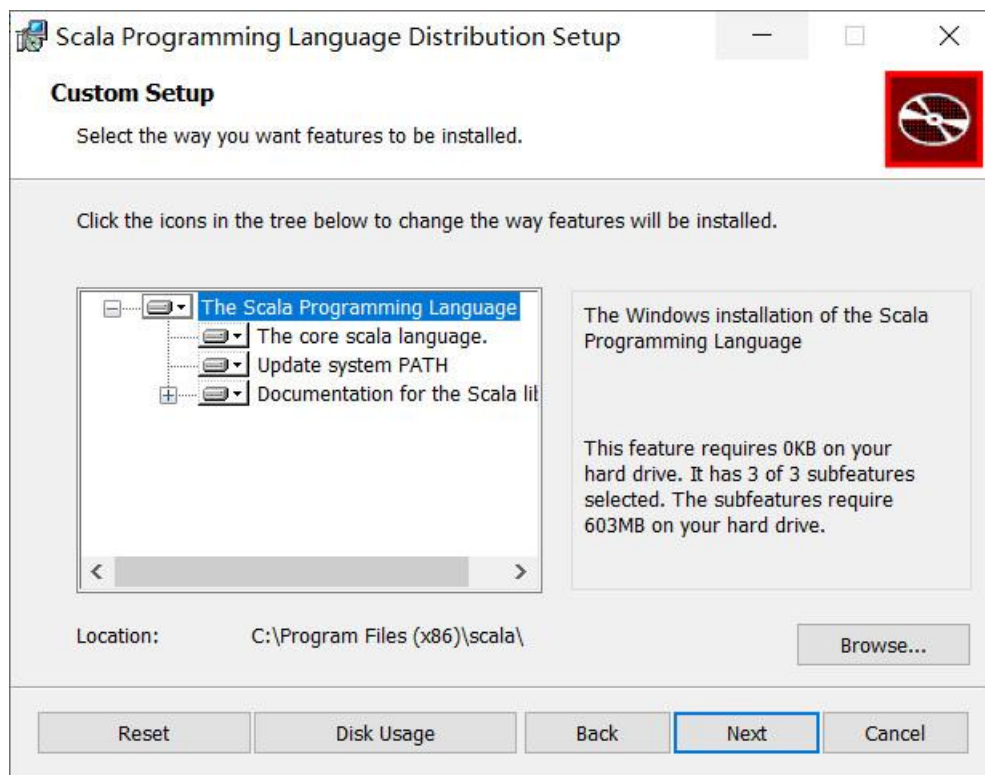


图 3-27 默认安装位置

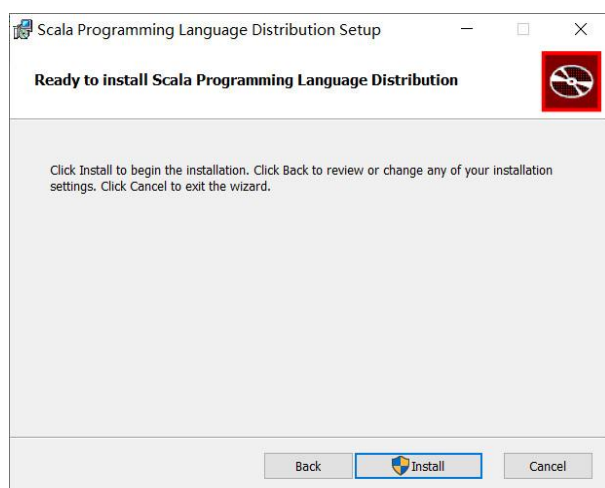


图 3-28 安装

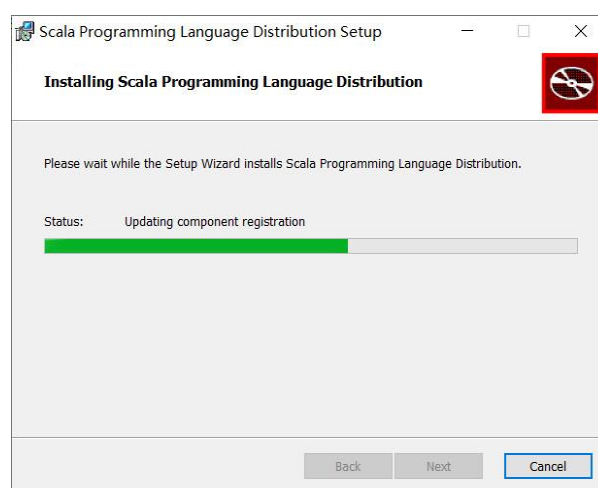


图 3-29 安装过程

步骤 04: 最终完成安装后，单击【Finish】按钮，关闭界面即可，如图 3-30 所示。

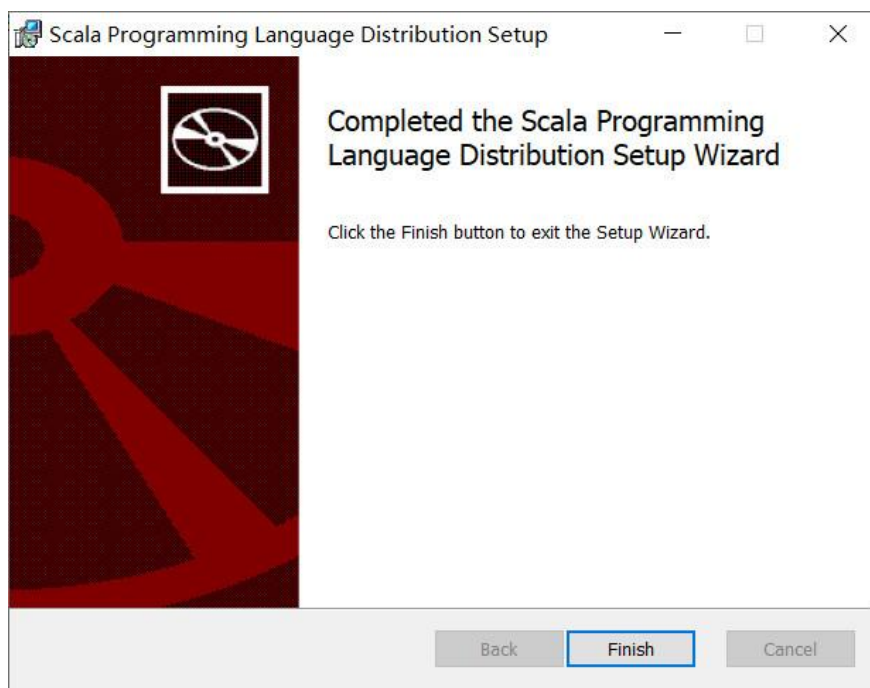


图 3-30 安装完成

步骤 04: 同安装 JDK 软件, 安装 Scala 同样需要配置环境变量, 打开系统环境变量配置的窗口, 在【系统变量】部分, 单击【新建】, 创建“SCALA_HOME”变量, 变量值为安装 SCALA 时的路径, 如图 3-31 所示, 完成之后单击【确定】按钮。



图 3-31 Scala 环境变量

步骤 05: 在【系统变量】中找到“Path”变量, 双击打开, 在变量值中添加“%SCALA_HOME%\bin”, 如图 3-32 所示。以上完成之后, 点击【确定】按钮关闭环境变量配置的窗口即可。

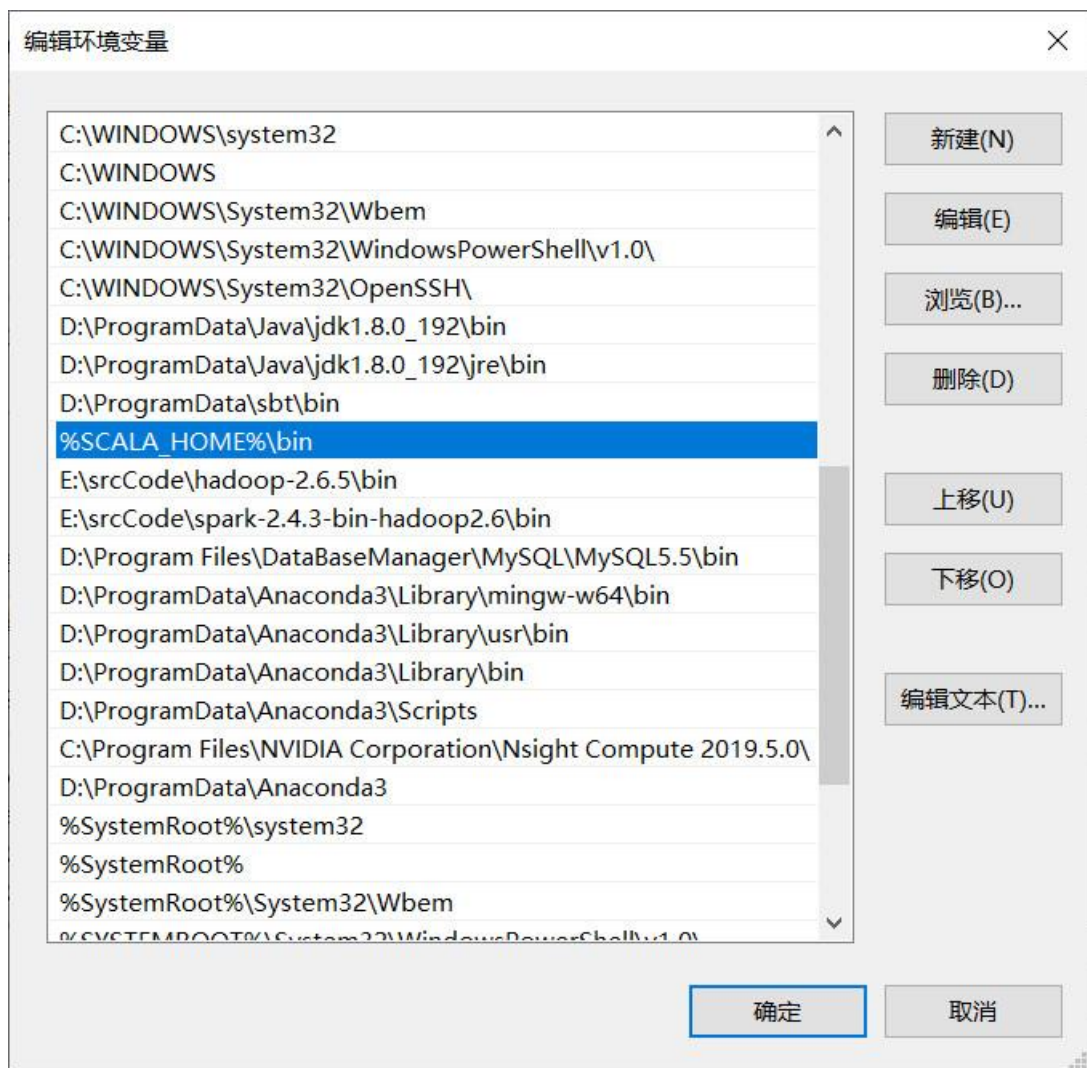


图 3-32 系统环境变量

步骤 06: 最后测试环境变量配置是否正确。按下键盘【Win+R】，在弹出的运行窗口中输入“cmd”然后回车，如图 3-33 所示，进入 cmd 命令窗口，在其中输入“scala -version”，若弹出如图 3-34 所示的信息，则代表环境变量配置成功。

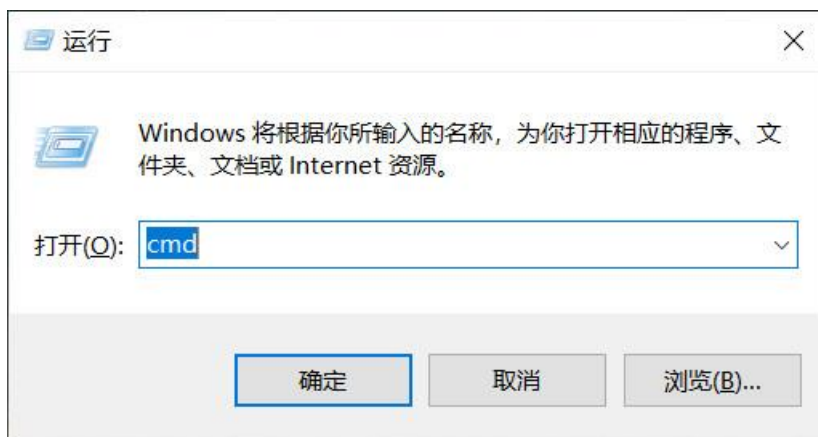


图 3-33 cmd 命令打开



```
命令提示符
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Henry>scala -version
Scala code runner version 2.11.8 -- Copyright 2002-2016, LAMP/EPFL

C:\Users\Henry>
```

图 3-34 Scala 版本检查

3.3.2 Linux 安装 Scala

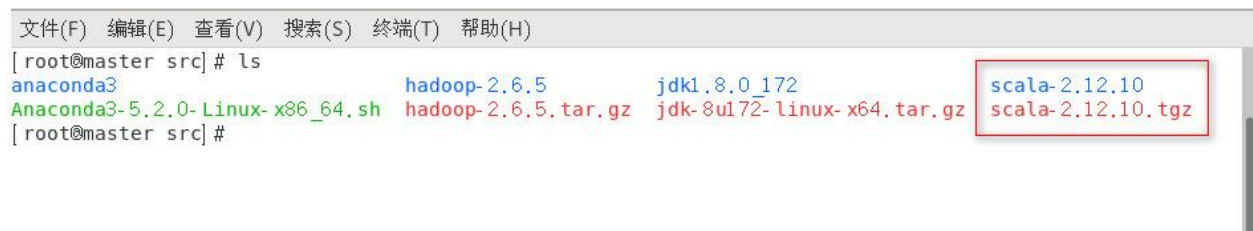
Linux 系统安装 Scala 相对比较简单，类似于 Linux 安装 JAVA 的方法，具体安装步骤如下所示。

步骤 01：首先需要将下载的安装包 `scala-2.12.10.tgz` 放置到 “`/usr/local/src`” 目录下，并执行如下命令 “`tar -zxvf scala-2.12.10.tgz`” 进行解压。

```
cd /usr/local/src/
tar -zxvf scala-2.12.10.tgz

# 查看是否解压成功
ls
```

查看是否解压成功，如图 3-35 所示。



```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@master src] # ls
anaconda3          hadoop-2.6.5      jdk1.8.0_172      scala-2.12.10
Anaconda3-5.2.0-Linux-x86_64.sh  hadoop-2.6.5.tar.gz  jdk-8u172-linux-x64.tar.gz  scala-2.12.10.tgz
[root@master src] #
```

图 3-35 解压 Scala 安装包文件

步骤 02：安装配置 Scala 环境变量，执行代码如下，添加完之后保存并退出。

```
vim ~/.bashrc

# 在最下面一行加入如下内容
export SCALA_HOME=/usr/local/src/scala-2.12.10
export PATH=$PATH:$SCALA_HOME/bin

# 退出后刷新环境变量
source ~/.bashrc

# 查看 scala 是否安装成功
scala -version
```

编辑后的结果如图 3-36 所示。



```
root@master:/usr/local/src/scala-2.12.10
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

export JAVA_HOME=/usr/local/src/jdk1.8.0_172
export JRE_HOME=/usr/local/src/jdk1.8.0_172/jre
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin

export HADOOP_HOME=/usr/local/src/hadoop-2.6.5
export PATH=$PATH:$HADOOP_HOME/bin

export ANACONDA_HOME=/usr/local/src/anaconda3
export PATH=$PATH:$ANACONDA_HOME/bin

export SCALA_HOME=/usr/local/src/scala-2.12.10
export PATH=$PATH:$SCALA_HOME/bin
```

28,0-1 底端

图 3-36 添加 Scala 环境变量

执行命令查看 Scala 是否安装配置成功，如图 3-37 所示。



```
root@master:/usr/local/src/scala-2.12.10
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@master scala-2.12.10]# scala -version
Scala code runner version 2.12.10 -- Copyright 2002-2019, LAMP/EPFL and
Lightbend, Inc.
[root@master scala-2.12.10]# █
```

图 3-37 Scala 安装成功验证

步骤 03：同样的方法，分别在 slave1 和 slave2 节点安装并配置 Scala，最后验证每个节点都能够正确使用 Scala。

至此，Linux 中安装 Scala 的内容已经完成。

3.3.3 Scala 交互式体验

Spark 是基于 Scala 语言进行开发的，所以我们有必要掌握 Scala 语言简单的使用方法，便于在今后更深层次的阅读 Spark 源码或使用 Scala 语言进行 Spark 开发工作。Scala 交互式体验步骤如下所示。

步骤 01：在 Linux 终端输入“scala”，然后按下【Enter】键，进入到 scala 交互界面，同样，也可以在 Windows 的 cmd 中输入“scala”，然后按下【Enter】键，进入到 scala 交互界面，这里在 Linux 下进行演示，如图 3-38 所示。


```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@master src] # scala
Welcome to Scala 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_172).
Type in expressions for evaluation. Or try :help.

scala>
```

图 3-38 Scala 交互式

步骤 02: 进入 scala 交互界面之后, 这里简单的介绍一下几种常用变量类型, 执行代码如下所示。

```
// scala 语言使用 val 或 var 关键字定义变量, 其中 val 为不可变类型变量, var 为可变类型变量
val x = 3           // 整数
val y = "This is a string" // 字符串
val z = true        // 布尔类型
val f = 1.265        // 浮点数
```

运行结果如图 3-39 所示。

```
[root@master ~] # scala
Welcome to Scala 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_172).
Type in expressions for evaluation. Or try :help.

scala> val x = 3
x: Int = 3

scala> val y = "This is a string."
y: String = This is a string.

scala> val z = true
z: Boolean = true

scala> val f = 1.265
f: Double = 1.265

scala> █
```

图 3-39 Scala 交互式运行结果

3.4 Maven 安装与配置

本小节我们将介绍在 Windows 下如何安装 Maven 工具。介绍 Maven 安装的目的在于接下来方便我们创建简单的 Scala 工程。

3.4.1 Maven 介绍

Maven 是目前最流行的 Java 项目构建系统，Maven 项目对象模型(POM)，可以通过一小段描述信息来管理项目的构建，报告和文档的软件项目管理工具，当然也可以用于我们创建 Scala 项目，Maven 主页如下图 3-40 所示。

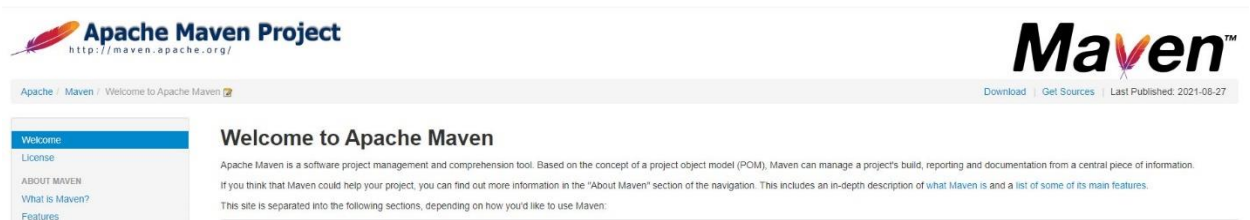


图 3-40 Maven 主页

3.4.2 Maven 下载与安装

Maven 具体安装步骤如下所示。

步骤 01: Maven 可以通过官网选择合适的版本进行下载，本书采用的是 Maven 3.6.0 版本，Maven 官网下载地址为：<https://maven.apache.org/docs/history.html>，找到对应版本之后的【release-notes】单击，进入下载界面，如图 3-41 所示。

Maven Releases History

Date format is: YYYY-MM-DD

Maven 3.1+

Release Date	Version	Required Java Version	Links
2021-08-04	3.8.2	Java 7	announce , release notes , reference documentation
2021-04-04	3.8.1		announce , release notes , reference documentation
2019-11-25	3.6.3		announce , release notes , reference documentation
2019-08-27	3.6.2		announce , release notes , reference documentation
2019-04-04	3.6.1		announce , release notes , reference documentation
2018-10-24	3.6.0		announce , release notes , reference documentation
2018-06-21	3.5.4		announce , release notes , reference documentation
2018-03-08	3.5.3		announce , release notes , reference documentation
2017-10-24	3.5.2		announce , release notes , reference documentation
2017-04-07	3.5.0		announce , release notes , reference documentation
2017-03-24	3.5.0-beta-1		announce , release notes , reference documentation
2017-02-28	3.5.0-alpha-1		announce , release notes , reference documentation
2015-11-14	3.3.9		announce , release notes , reference documentation
2015-04-28	3.3.3		announce , release notes , reference documentation
2015-03-18	3.3.1		announce , release notes , reference documentation

图 3-41 Maven 下载页面

步骤 02: 下载完成之后，解压安装包，如图 3-42 所示。



图 3-42 Maven 解压展示

步骤 03: 完成安装包解压之后, 复制所在路径, 同样的, 进行 Maven 环境变量的配置, 创建“MAVEN_HOME”环境变量, 并在系统环境变量 Path 中添加相关内容, 如图 3-43 所示。配置完成之后, 单击【确定】按钮, 关闭环境变量配置窗口即可。

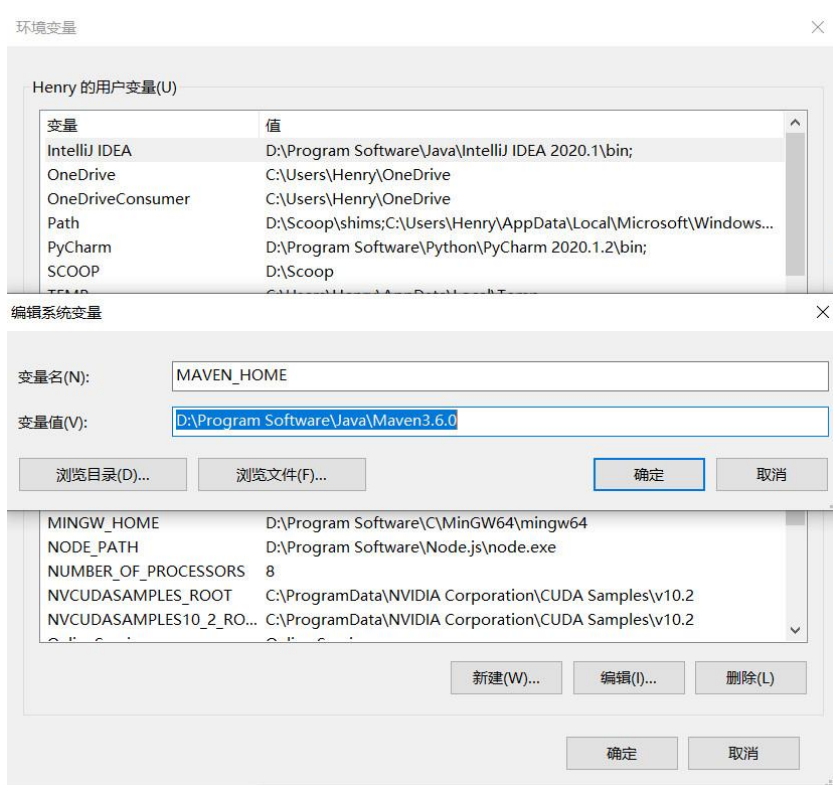


图 3-43 MAEVN 环境变量配置

接下来, 讲 Maven 添加到系统环境变量 Path 中, 使得系统可以访问到 Maven, 添加系统环境变量 Path 的方法如图 3-44 所示。

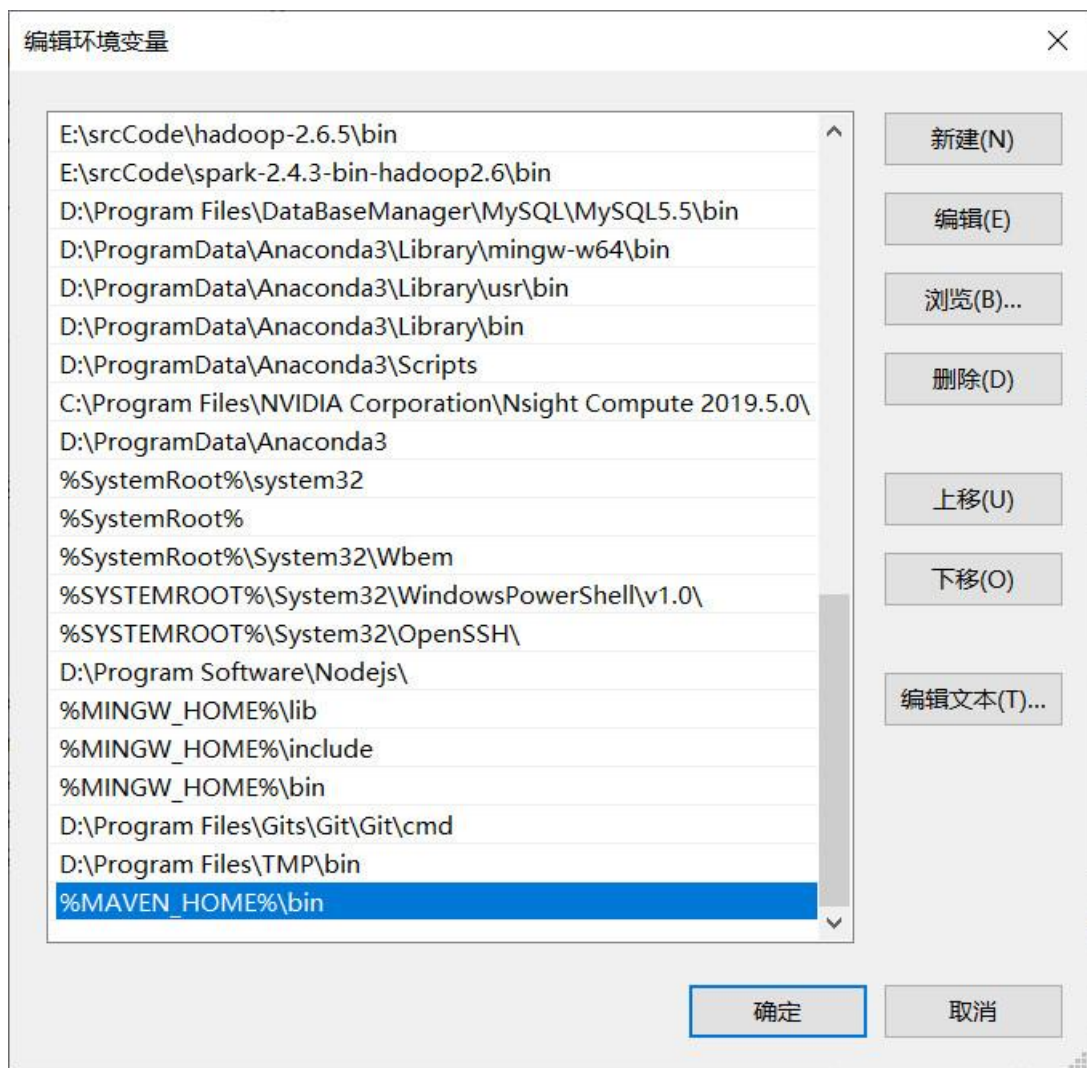


图 3-44 系统环境变量展示

步骤 04: 最后, 进行 Maven 安装配置的验证工作, 打开“cmd”, 输入“mvn -v”, 查看环境变量是否配置成功并生效。如图 3-44 所示, 输出结果中展示了 Maven 的版本是 3.6.0, Java 的版本是 1.8.0-192, 当前系统版本为 windows 10.0。至此, 说明 Maven 已安装配置成功。

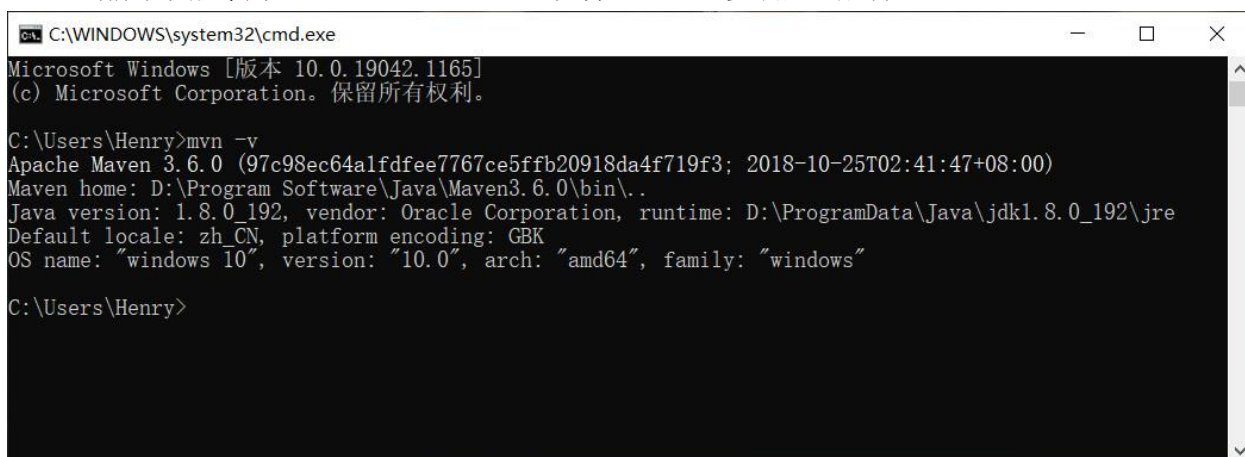


图 3-44 Maven 安装成功验证

3.4.3 Maven 镜像源配置

修改 Maven 镜像源的目的同 Anaconda，修改方法比较简单。找到 Maven 的安装目录下，进入 conf 文件夹中，打开 setting.xml 文件，在 “mirrors” 位置添加如下代码，然后保存并关闭。

```
<mirror>
  <id>nexus-aliyun</id>
  <name>nexus-aliyun</name>
  <url>http://maven.aliyun.com/nexus/content/groups/public</url>
  <mirrorOf>central</mirrorOf>
</mirror>
```

添加后如图 3-45 所示。

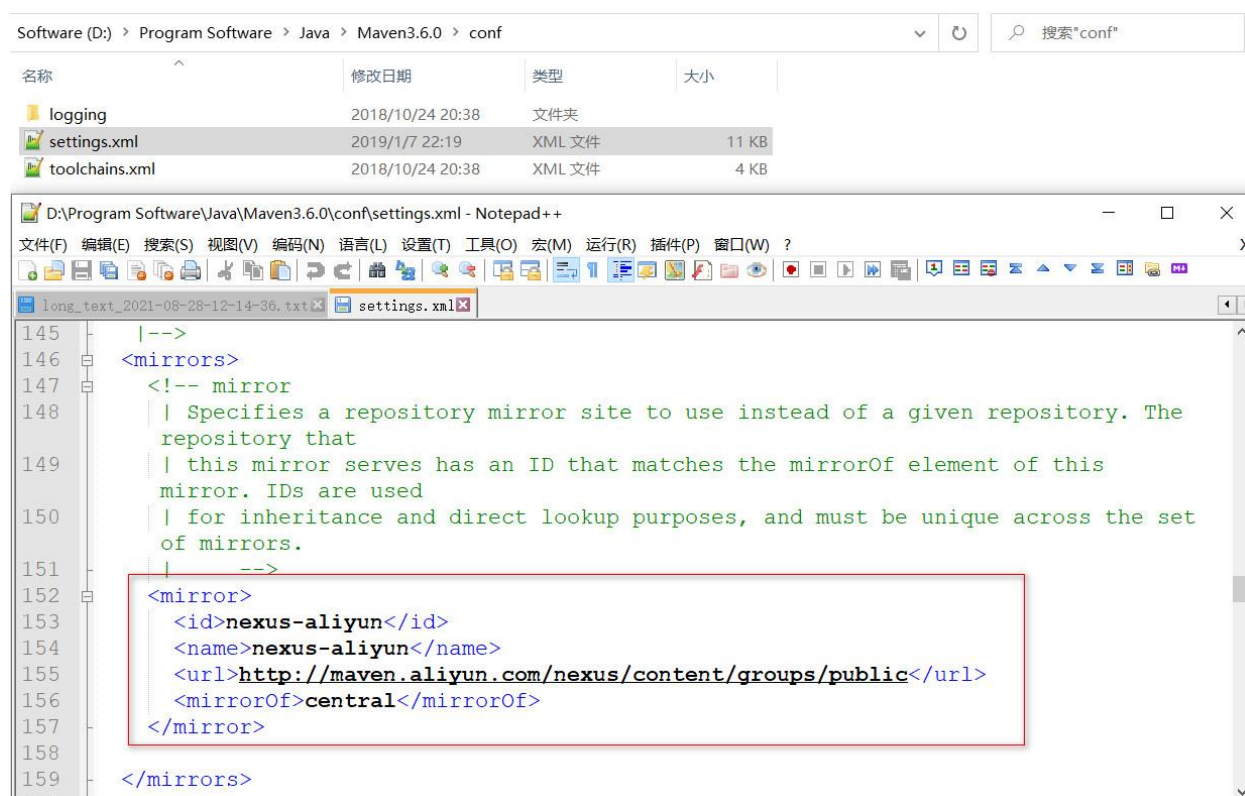


图 3-45 修改完毕 setting.xml 展示

3.5 IDEA 安装与应用

本小节我们将在 Windows 上安装 IDEA 集成开发工具，并实现基于 Maven 工具构建第一个 Scala 工程。

3.5.1 IDEA 工具介绍

IDEA 全称 IntelliJ IDEA，是 JetBrains 公司的产品，IDEA 是一款 Java 编程语言开发的集成环境。IntelliJ IDEA 在业界被公认为最好的 Java 开发工具，尤其在智能代码助手、代码自动提示、重构、JavaEE 支持、各类版本工具、JUnit、CVS 整合、代码分析等方面的功能可以说是超常的。当然 IDEA 也可以用于构建其他语言的工程，如本节介绍的 Scala 语言。

3.5.2 IDEA 下载与安装

IDEA 的安装非常简单, 我们可以通过官网 <https://www.jetbrains.com/idea/>, 单击右上角的【Download】按钮进入到下载选择页面, 如图 3-46 所示。

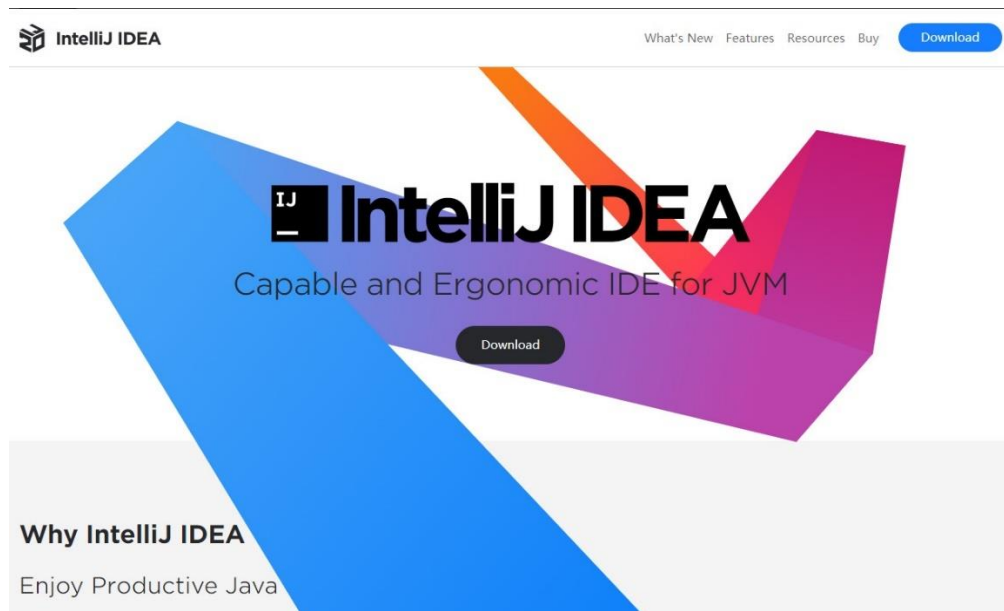


图 3-46 IDEA 官网

IDEA 分为旗舰版和社区版, 其中旗舰版相对于社区版功能更为强大, 但是旗舰版为收费软件, 这里我们选择免费的社区版即可满足绝大多数开发者的需求。默认下载安装包格式为 exe 格式, 安装方法比较简单, 只需要选择一下安装路径即可。

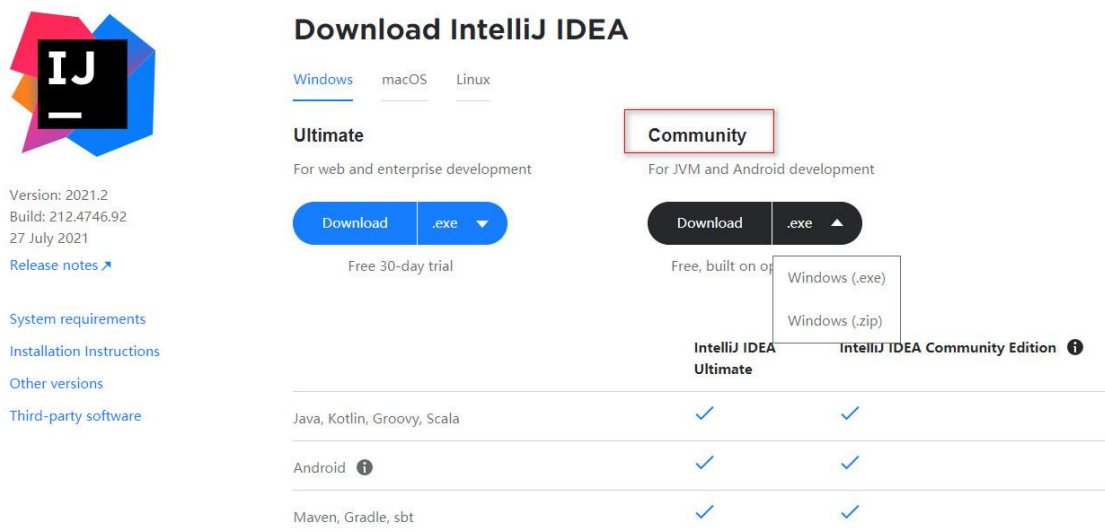


图 3-47 IDEA 下载界面

3.5.3 Hello Scala——创建第一个 Scala 程序

在完成 IDEA 的安装之后, 接下来我们将正式开启 Scala 工程之路。创建 Scala 工程的具体步骤如

下所示。

步骤 01：双击打开 IDEA 快捷方式，在弹出的窗口中单击【+Create New Project】按钮，如图 3-48 所示。

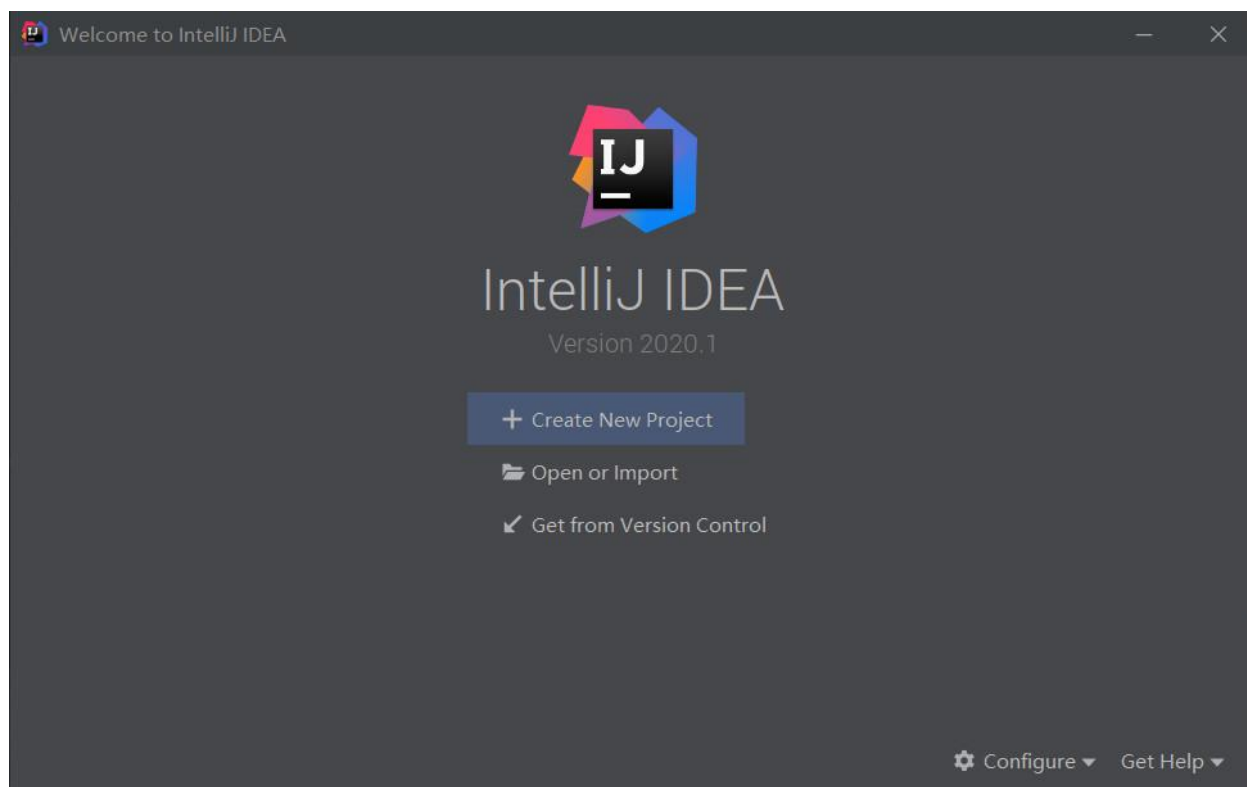


图 3-48 IDEA 界面

步骤 02：在弹出的窗口左侧，单击选择【Maven】选项，然后单击【Next】按钮，如图 3-49 所示。

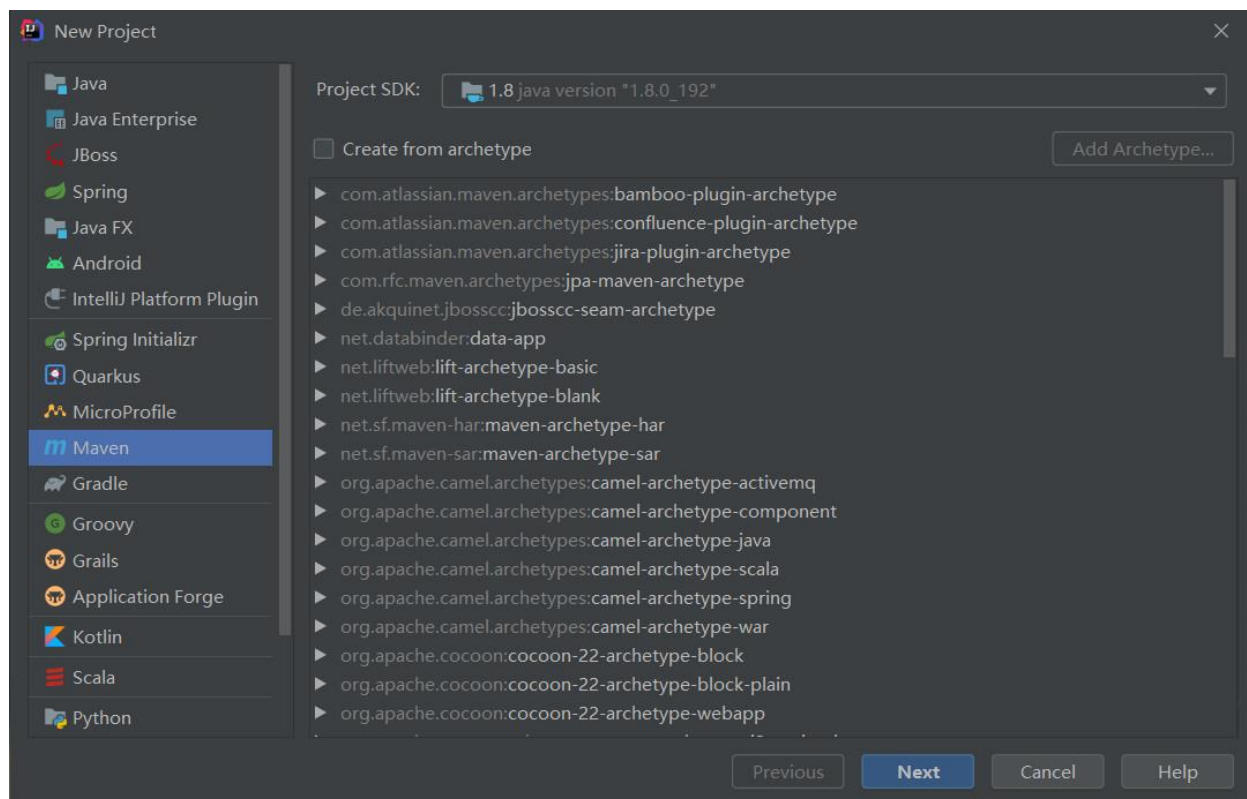


图 3-49 Maven 工程选择

步骤 03: New Project 界面中, 在“Name”一栏中输入工程的名字, 本书创建的工程名为“ScalaDemo”, 接下来在“Location”一栏选择工程存储的路径, 选择完成之后单击【Finish】按钮, 即可完成工程创建, 如图 3-50 所示。

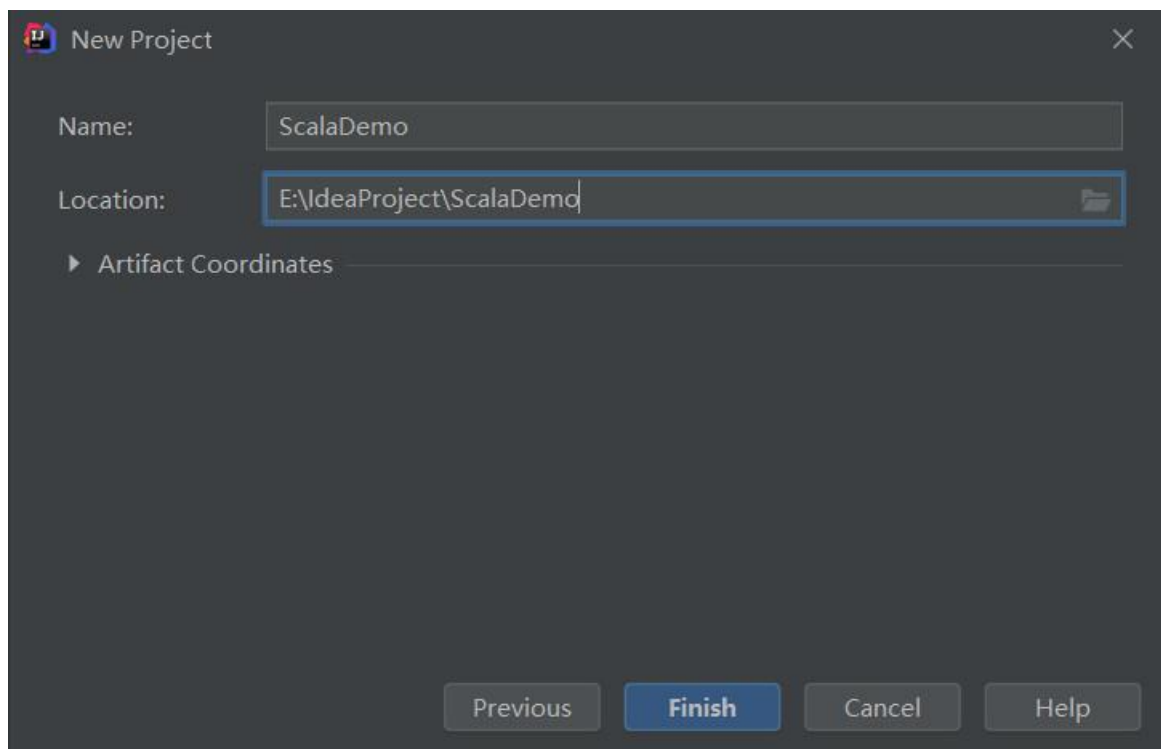


图 3-50 新建 scala 项目

步骤 04: 创建完成工程之后, 我们需要对工程进行简单设置, 单击左上角【File】菜单, 在下拉菜单中选择【Project Structure...】选项, 打开工程设置窗口, 如图 3-51 所示。

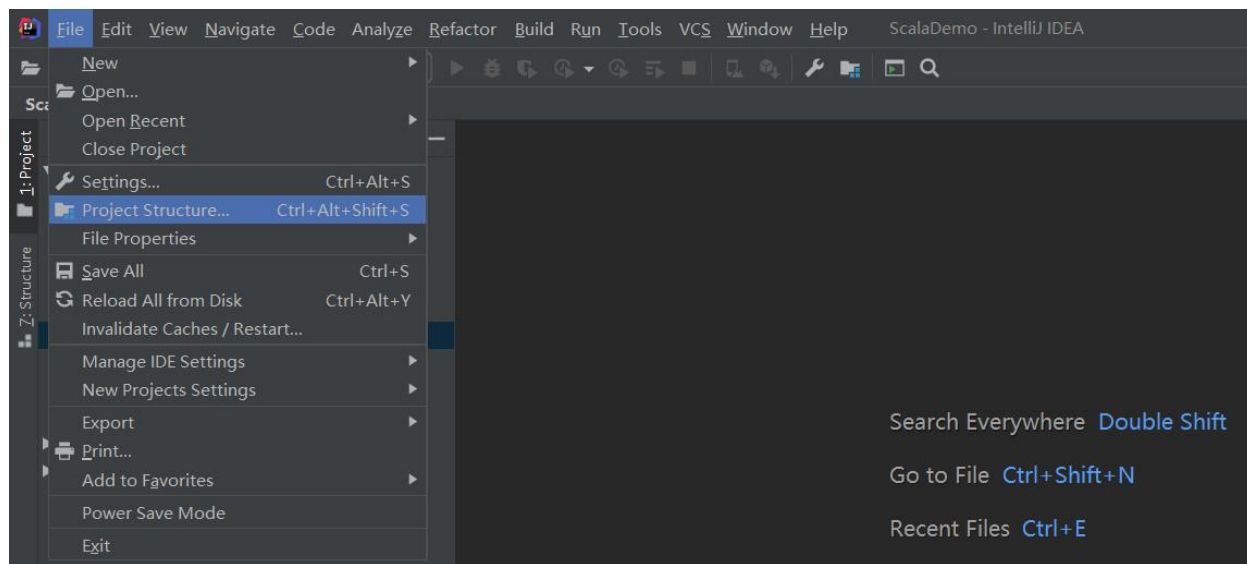


图 3-51 Project Structure

步骤 05: 单击选择【Libraries】, 并在右侧单击【+】按钮, 在弹出的【New Project Library】下拉菜单中单击【Scala SDK】, 如图 3-52 所示。在弹出的窗口中选择 Version 为 2.12.10 的一栏, 如图 3-53 所示。然后单击【OK】按钮即可, 如图 3-54 所示。

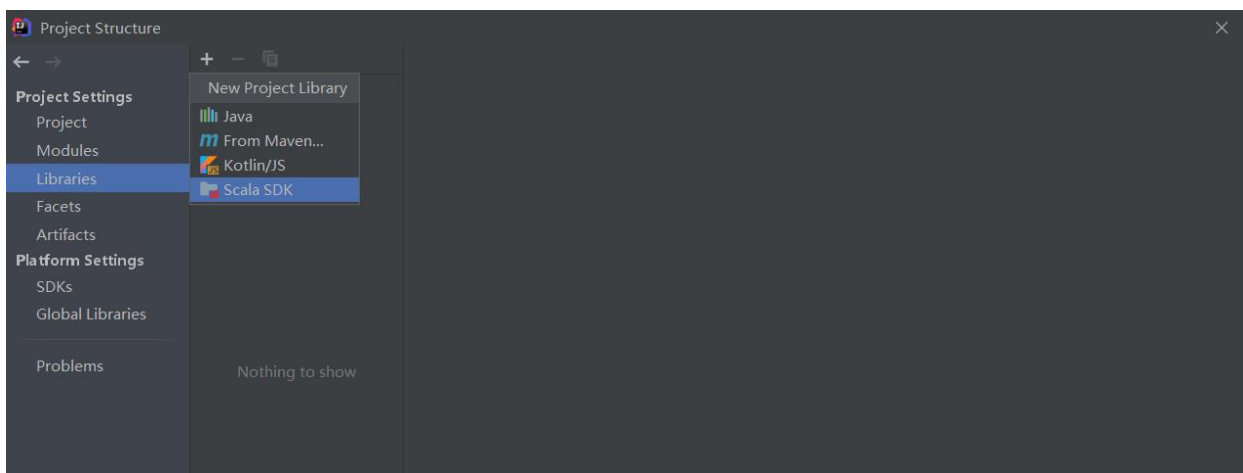


图 3-52 New Project Library

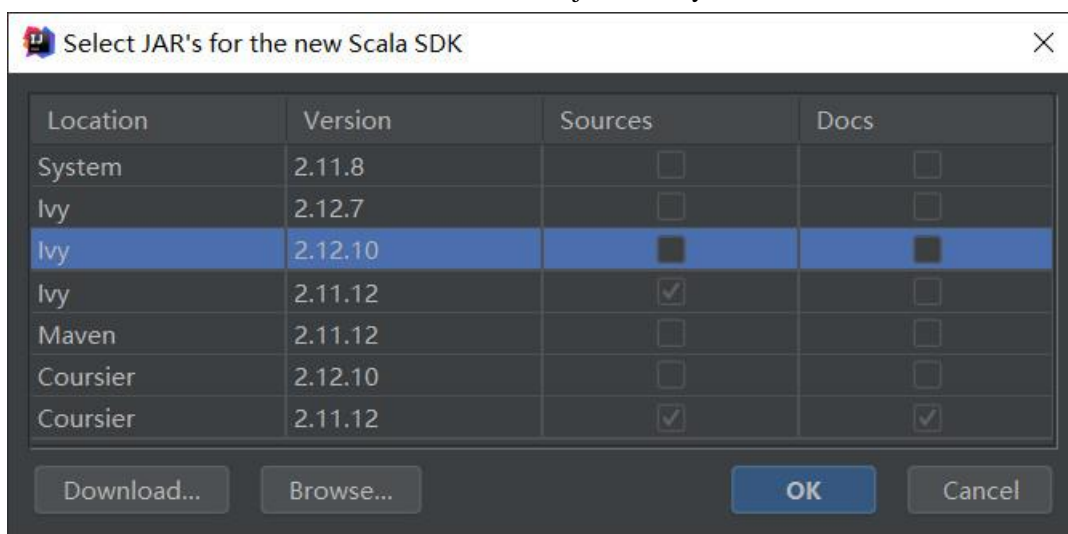


图 3-53new Scala SDK 选择

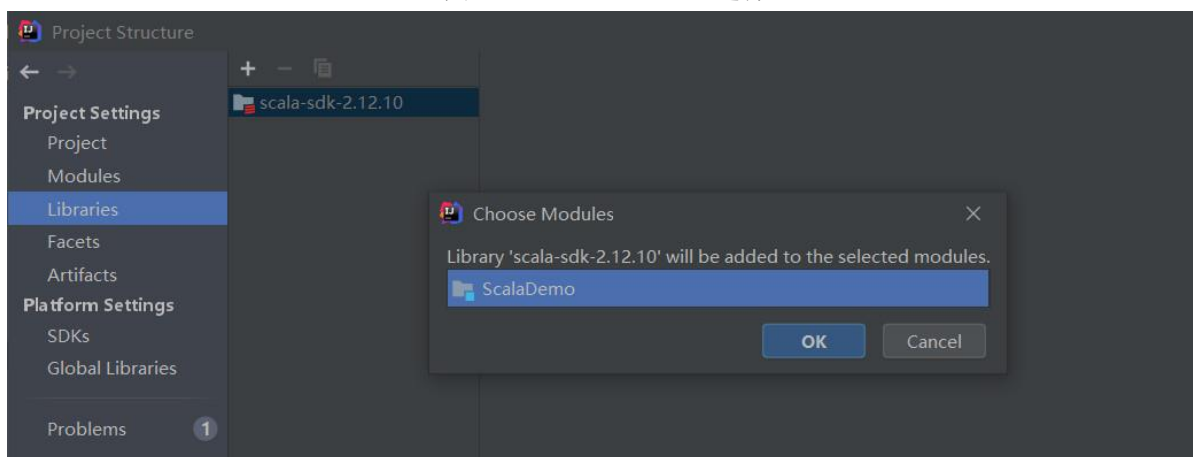


图 3-54 选择工程 Modules

步骤 06: 回到工程界面之后, 右键单击 `main` 文件夹, 在弹出的下拉菜单中依次选择 **【New】** -> **【Dictionary】** 创建名为 `scala` 的文件夹, 如图 3-55 所示。在完成 `scala` 文件夹创建之后, 右键单击 `scala` 文件夹, 在弹出的下拉菜单中依次选择 **【Mark Dictionary as】** -> **【Source Root】**, 如图 3-56 所示。

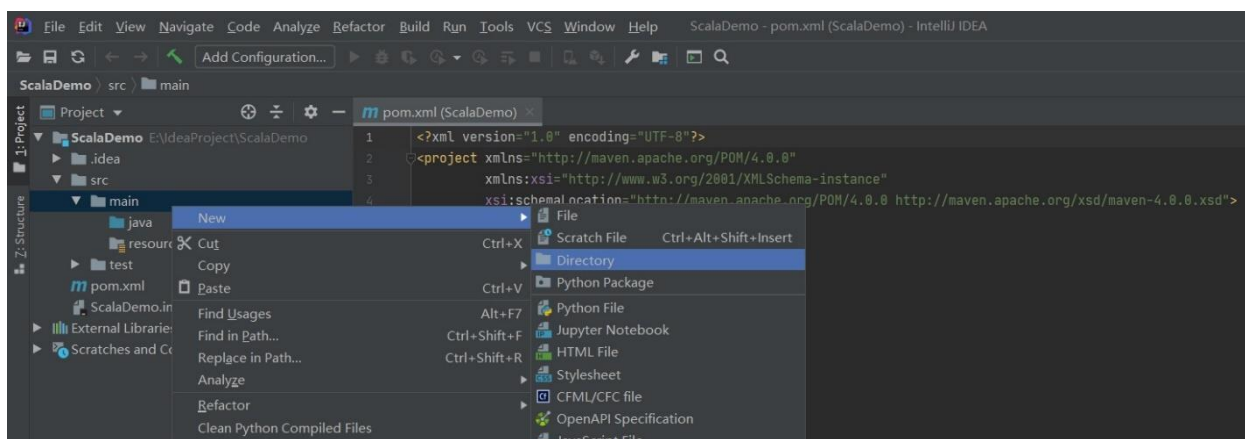


图 3-55 创建文件夹

选中“scala”文件夹，右键，在弹出的选项菜单中选择“Mark Directory as”，并在下一级选项菜单中选择“Sources Root”，完成将“scala”文件设置问代码文件夹。

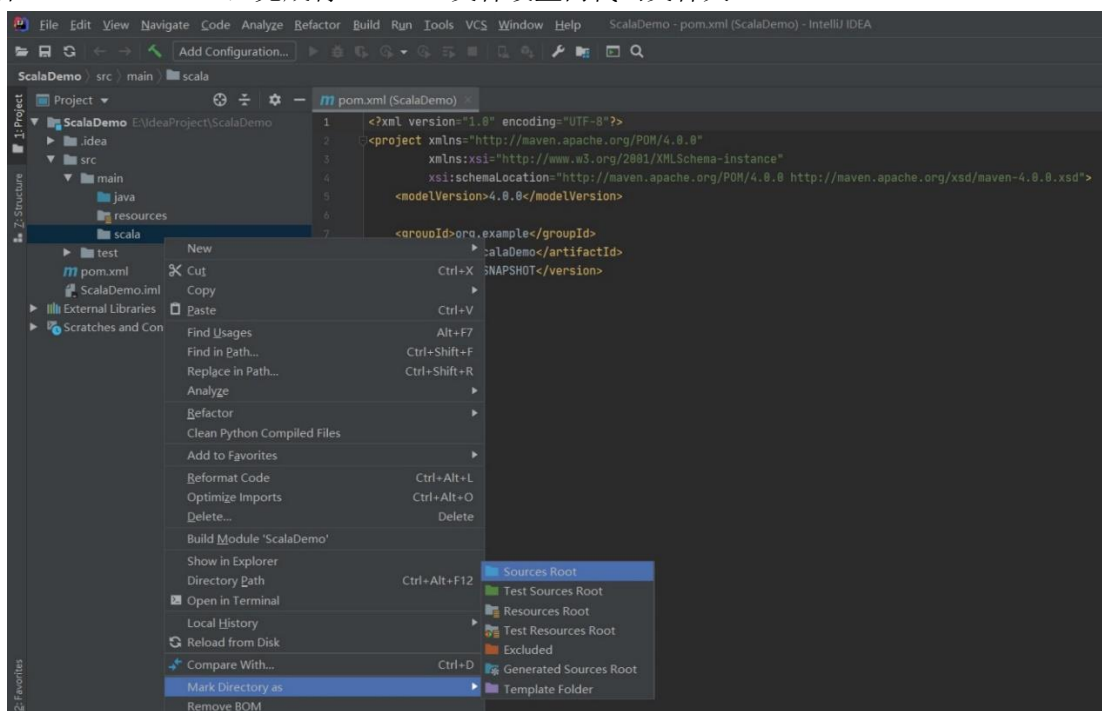


图 3-56 设置为 Sources Root

步骤 07: 工程设置相关的工作完成后，接下来我们创建 scala 代码文件，并编写代码，选中“scala”文件夹，右键，选择“New”，在弹出的选项菜单中选择“Scala Class”，如图 3-57 所示。

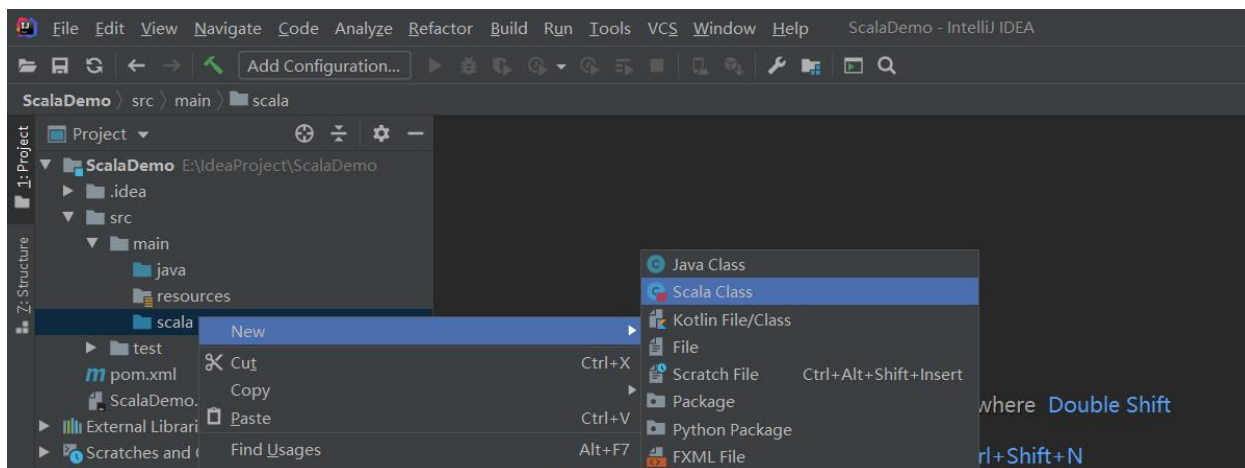


图 3-57 创建 Scala Class

此时会弹出“Create New Scala Class”菜单，提示选择何种类型的 Scala 文件，这里选择“Object”。由于本书使用的主要编程语言为 Python，这里不对 Scala 语言展开介绍，只对 Class 和 Object 的区别进行简单介绍。

Scala 是一种面向对象的编程语言，常量/变量/方法等必须要定义在 Class 或 Object 里面才可以，在其他之外的地方是不能被定义的。也就是说只有 Class 对象和 Object 对象类是定义其成员的地方。

对于一个 Class 来说，所有的方法和成员变量在实例被 new 出来之前都是无法访问的，因此 class 文件中的 main 方法也就没用了，Object 中所有成员变量和方法默认都是 static 静态的，所以可以直接访问 main 方法。

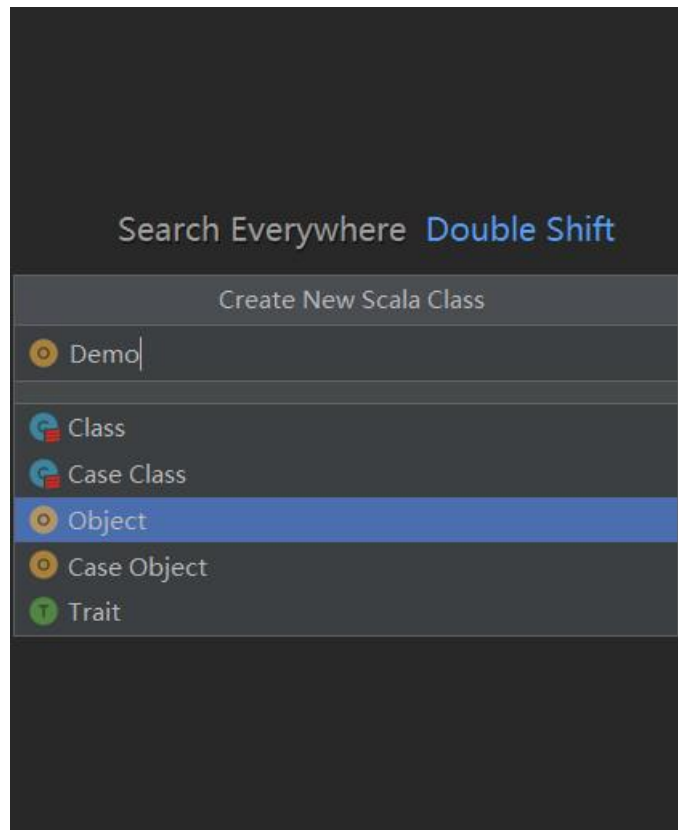


图 3-58 创建 Demo Object 文件

在创建的 Demo.scala 文件中，输入以下代码。

```
object Demo {  
  
    def main(args: Array[String]): Unit = {  
        println("Hello Scala!")  
    }  
  
}
```

编写完代码之后，单击 main 函数前方的绿色箭头，在弹出的下拉菜单中单击【Run ‘Demo’】即可开始运行程序，程序运行输出“Hello Scala!”，如图 3-59 所示。

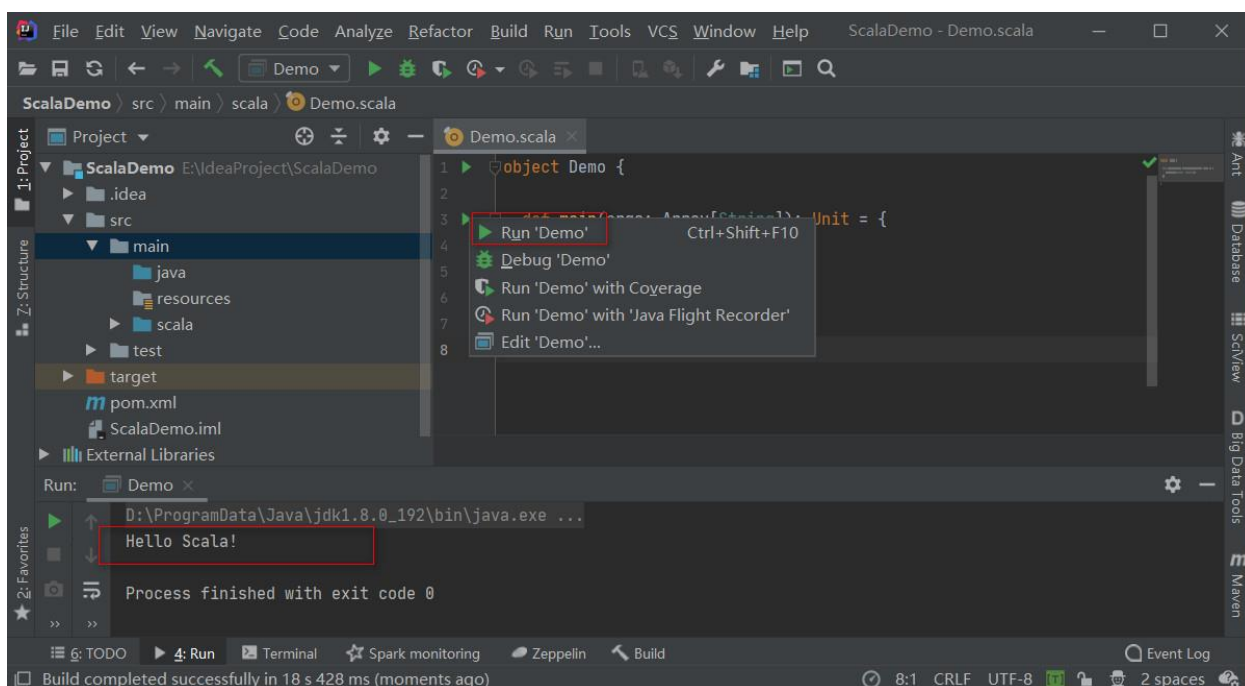


图 3-59 运行 Demo 程序

至此，我们完成了第一个 Scala 工程的创建及代码的编写，初步认识了 IDEA 工具的使用和 Scala 工程的创建方法。

★回顾思考★

01 Jupyter Notebook 常用的快捷键有哪些？

答：包括如下：

- (1) 按“Ctrl+A”，在当前行上方插入一行；
- (2) 按“Ctrl+B”，在当前行下方插入一行；
- (3) 按“Shift+Enter”，执行当前行代码；
- (4) 单击选中当前行，按“Y”，切换当前行为代码编辑格式；
- (5) 单击选中当前行，按“M”，切换当前行为 Markdown 编辑格式；

★练习★

一、选择题

1. Scala 定义一个字符串正确的是（ ）
A. val x = sbd

- B. `val x = hello`
- C. `val x = "123"`
- D. `val x = "sbd"`

2. Scala 定义可变类型变量的关键字是 ()

- A. `public int`
- B. `val`
- C. `var`
- D. `char`

3. Python 创建一个字符串正确的是 ()

- A. `x = "jeck"`
- B. `x = "heck"`
- C. `x = jack`
- D. `x = 3.69`

二、判断题

- 1. JDK 需要在 Hadoop 集群每个节点机器安装,但 Scala 只需要在 Master 节点安装。 ()
- 2. Anaconda 是 Scala 的集成环境安装包。 ()

三、实战练习

- 1. 使用 Jupyter Notebook 创建一个斐波那契函数。
- 2. 使用 IDEA 工具创建 Scala 工程并运行。

本章小结

本章首先简单介绍了 Python 语言和 Scala 语言,其中 Python 语言为本书开发的核心使用语言,Spark 是基于 Scala 语言进行开发的,所以同样进行了简单介绍。其次,本章详细介绍了大数据分析处理开发工具的安装和使用,如 Anaconda、Jupyter Notebook、Maven 以及 IDEA 等软件的详细安装方法。Jupyter Notebook 是本书所使用最多的开发工具,所以对使用方法进行了详细的介绍和充足的示例。最后,介绍了创建基于 Maven 项目的 Scala 工程,并且编写运行了第一个 Scala 工程,感兴趣的读者可以自行进一步学习了解 Scala 语言,后续章节将不再过多介绍 Scala 语言的开发内容。