

# Introduction of Python and Setup the Enviroment

## Mainstream Programming Language Rankings

世界上的编程语言有 600 多种，但真正大家主流在使用的最多二三十种，不同的语言有自己的特点和擅长领域，随着计算机的不断发展，新语言在不断诞生，也同时有很多老旧的语言慢慢无人使用了

有个权威的语言排名网站 [TIOBE](https://index.tiobe.com/)，可以看到主流的编程语言是哪些

### TIOBE Index for April 2020

Apr 2020	Apr 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.73%	+1.69%
2	2		C	16.72%	+2.64%
3	4	▲	Python	9.31%	+1.15%
4	3	▼	C++	6.78%	-2.06%
5	6	▲	C#	4.74%	+1.23%
6	5	▼	Visual Basic	4.72%	-1.07%
7	7		JavaScript	2.38%	-0.12%
8	9	▲	PHP	2.37%	+0.13%
9	8	▼	SQL	2.17%	-0.10%
10	16	▲▲	R	1.54%	+0.35%
11	19	▲▲	Swift	1.52%	+0.54%
12	18	▲	Go	1.36%	+0.35%
13	13		Ruby	1.25%	-0.02%
14	10	▼▼	Assembly language	1.16%	-0.55%
15	22	▲	PL/SQL	1.05%	+0.26%
16	14	▼	Perl	0.97%	-0.30%
17	11	▼▼	Objective-C	0.94%	-0.57%
18	12	▼▼	MATLAB	0.93%	-0.36%
19	17	▼	Classic Visual Basic	0.83%	-0.23%
20	27	▲	Scratch	0.77%	+0.28%

### Very Long Term History

Programming Language	2020	2015	2010	2005	2000	1995	1990	1985
Java	1	2	1	2	3	-	-	-
C	2	1	2	1	1	2	1	1
Python	3	7	6	6	23	21	-	-
C++	4	4	4	3	2	1	3	9
C#	5	5	5	7	9	-	-	-
JavaScript	6	8	8	8	6	-	-	-
PHP	7	6	3	4	25	-	-	-
SQL	8	-	-	96	-	-	-	-
Swift	9	181	-	-	-	-	-	-
Ruby	10	11	9	23	31	-	-	-
Objective-C	12	3	15	37	-	-	-	-
Lisp	27	21	14	13	8	5	6	2
Fortran	29	28	21	14	16	4	2	12
Ada	34	26	24	15	15	6	7	3
Pascal	235	13	12	53	13	3	8	5

## Programming Language Hall of Fame

Year	Winner
2019	🏆 C
2018	🏆 Python
2017	🏆 C
2016	🏆 Go
2015	🏆 Java
2014	🏆 JavaScript
2013	🏆 Transact-SQL
2012	🏆 Objective-C
2011	🏆 Objective-C
2010	🏆 Python
2009	🏆 Go
2008	🏆 C
2007	🏆 Python
2006	🏆 Ruby
2005	🏆 Java
2004	🏆 PHP
2003	🏆 C++

Python 是一门优秀的综合语言，Python 的宗旨是简明、优雅、强大，在人工智能、云计算、金融分析、大数据开发、WEB开发、自动化运维、测试等方向应用广泛，已是全球第 4 大最流行的语言

## Current Main Application Areas of Python

- **WEB 开发**，最火的 Python web 框架 Django, 支持异步高并发的 Tornado 框架，短小精悍的 Flask 等
- **网络编程**，支持高并发的 Twisted 网络框架，py3 引入的 asyncio 使异步编程变的非常简单
- **爬虫**，爬虫领域，Python 几乎是霸主地位，Scrapy\Request\BeautifulSoup\urllib 等
- **云计算**，目前最火最知名的云计算框架就是 OpenStack，Python 目前的热门，很大一部分就是因为云计算
- **自动化运维**，几乎是中国每位运维人员必备的语言
- **人工智能**，Python 是目前公认的人工智能和数据分析领域的必备语言，得益于其强大的计算库
- **金融分析**，许多趋势分析、高频交易软件都是基于 Python，到目前为止，Python 仍是金融分析、量化交易领域里使用率最高的语言
- **科学运算**，1997 年开始，NASA 就在大量使用 Python 进行各种复杂的科学运算，随着 NumPy, SciPy, Matplotlib, Enthought librarys 等众多程序库的开发，使得 Python 越来越适合于做科学计算，绘制高质量的 2D 和 3D 图像，和科学计算领域最流行的商业软件 Matlab 相比，Python 是一门通用的程序设计语言，比 Matlab 所采用的脚本语言的应用范围更广泛

## Environment Configuration

### About Anaconda

Anaconda 是一个用于科学计算的 Python 发行版，支持 Linux, Mac, Windows, 包含了众多流行的科学计算、数据分析的 Python 包

Anaconda 安装包可以到 [官网](#) 或者 [清华镜像](#) 下载

**注意** 我们选择 Python 3.X 的版本（比如目前是 3.7）

**Anaconda Installers**

Windows	MacOS	Linux
<b>Python 3.7</b> <a href="#">64-Bit Graphical Installer (466 MB)</a> <a href="#">32-Bit Graphical Installer (423 MB)</a>	<b>Python 3.7</b> <a href="#">64-Bit Graphical Installer (442)</a> <a href="#">64-Bit Command Line Installer (430 MB)</a>	<b>Python 3.7</b> <a href="#">64-Bit (x86) Installer (522 MB)</a> <a href="#">64-Bit (Power8 and Power9) Installer (276 MB)</a>
<b>Python 2.7</b> <a href="#">64-Bit Graphical Installer (413 MB)</a> <a href="#">32-Bit Graphical Installer (356 MB)</a>	<b>Python 2.7</b> <a href="#">64-Bit Graphical Installer (637 MB)</a> <a href="#">64-Bit Command Line Installer (409 MB)</a>	<b>Python 2.7</b> <a href="#">64-Bit (x86) Installer (477 MB)</a> <a href="#">64-Bit (Power8 and Power9) Installer (295 MB)</a>

下载之后，除了想修改安装路径以外，基本都可以选择默认设置，一路 **Next**，不过，如果遇到如下选项，记得勾选

☒ Add Anaconda to system PATH environment variable

安装结束后，打开命令行（Mac 的 Terminal，Windows 的 Anaconda Prompt 均可），输入 `conda -version`，出现版本号表示安装成功

```
1 $ conda --version
2 conda 4.5.12
```

**Anaconda 仓库与第三方源**，各系统都可以通过修改用户目录下的 `.condarc` 文件来设置 Windows 用户无法直接创建名为 `.condarc` 的文件，可先执行 `conda config --set show_channel_urls yes` 生成该文件之后再修改

```
1 channels:
2   - defaults
3 show_channel_urls: true
4 channel_alias: https://mirrors.tuna.tsinghua.edu.cn/anaconda
5 default_channels:
6   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
7   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free
8   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
9   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/pro
10  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
11 custom_channels:
12   conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
13   msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
14   bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
15   menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
16   pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
17   simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

即可添加 Anaconda Python 免费仓库

运行 `conda clean -i` 清除索引缓存，保证用的是镜像站提供的索引

如果习惯软件式操作，可以打开 Anaconda 的软件，里面有众多应用，这里我们简单介绍其中两款

- **Spyder** 集成开发环境，类似于 Matlab
- **Jupyter** 基于网络的 Web 应用程序，可作为学习笔记本

## About Pycharm

很多语言都有比较流行的开发工具，比如 JAVA 的 Eclipse，C# 与 C++ 的 Visual Studio，Go 语言的 Goland，那 Python 的是啥呢？答案就是 Pycharm

这是 PyCharm 的下载地址 <http://www.jetbrains.com/pycharm/download/>

之后页面会根据系统自动进入相应的下载页面



Version: 2020.1  
Build: 201.6668.115  
8 April 2020

[System requirements](#)

[Installation Instructions](#)

[Other versions](#)

## Download PyCharm

Windows Mac Linux

### Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

Download

Free trial

### Community

For pure Python development

Download

Free, open-source



Get the Toolbox App to download PyCharm and its future updates with ease

Professional 表示专业版，功能会更齐全一些，Community 是社区版，推荐安装社区版，因为是免费使用的，专业版是收费的，一年一千多，土豪请随意

## Conda Virtual Environment

在 Python 开发中，很多时候我们希望每个应用有一个独立的 Python 环境（比如应用 1 需要用到 TensorFlow 1.X，而应用 2 使用 TensorFlow 2.0，再比如我的某个应用需要用到 python 的不同版本），这时，Conda 虚拟环境即可为一个应用创建一套“隔离”的 Python 运行环境，使用 Python 的包管理器 conda 即可轻松地创建 Conda 虚拟环境，常用命令如下

```
1 conda create --name [env-name] # 建立名为[env-name]的Conda虚拟环境
2 conda activate [env-name] # 进入名为[env-name]的Conda虚拟环境
3 conda deactivate # 退出当前的Conda虚拟环境
4 conda env remove --name [env-name] # 删除名为[env-name]的Conda虚拟环境
5 conda env list # 列出所有Conda虚拟环境
```

以下命令可以对 conda 以及 anaconda 更新，但不建议频繁使用

```
1 conda update conda
2 conda update anaconda # 需要先更新 conda
```

以下命令可以对 python 更新，但不建议频繁使用

```
1 conda install python=3.7 # 3.7 为示例
```

此外虚拟环境还可以使用 virtualenv 等，这里不再展开

## pip and conda Package Managers

pip 是最为广泛使用的 Python 包管理器，可以帮助我们获得最新的 Python 包并进行管理

## pypi Image Help

pypi 镜像每 5 分钟同步一次

### Temporary Use

```
1 | pip install -i https://pypi.tuna.tsinghua.edu.cn/simple some-package
```

注意, `simple` 不能少, 是 `https` 而不是 `http`

### Set as Default

升级 pip 到最新的版本 (`>=10.0.0`) 后进行配置

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

如果您到 pip 默认源的网络连接较差, 临时使用本镜像站来升级 pip

```
1 | pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pip -U
```

## pip Common Commands

```
1 | pip install [package-name]           # 安装名为[package-name]的包
2 | pip install [package-name]==X.X      # 安装名为[package-name]的包并指定版本
   X.X
3 | pip install [package-name] --proxy=代理服务器IP:端口号      # 使用代理服务器
   安装
4 | pip install [package-name] --upgrade  # 更新名为[package-name]的包
5 | pip uninstall [package-name]         # 删除名为[package-name]的包
6 | pip list                             # 列出当前环境下已安装的所有包
```

## conda Package Manager

conda 包管理器是 Anaconda 自带的包管理器, 可以帮助我们在 conda 环境下轻松地安装各种包, 相较于 pip 而言, conda 的通用性更强 (不仅是 Python 包, 其他包如 CUDA Toolkit 和 cuDNN 也可以安装), 但 conda 源的版本更新往往较慢, 常用命令如下

```
1 | conda install [package-name]         # 安装名为[package-name]的包
2 | conda install [package-name]=X.X     # 安装名为[package-name]的包并指定版本X.X
3 | conda update [package-name]          # 更新名为[package-name]的包
4 | conda remove [package-name]          # 删除名为[package-name]的包
5 | conda list                            # 列出当前环境下已安装的所有包
6 | conda search [package-name]          # 列出名为[package-name]的包在conda源中的所
   有可用版本
```

conda 中配置代理, 在用户目录下的 `.condarc` 文件中添加以下内容

```
1 proxy_servers:
2     http: http://代理服务器IP:端口号
```

## About Jupyter Notebook

Jupyter Notebook 是基于网页的用于交互计算的应用程序，其可被应用于全过程计算，开发、文档编写、运行代码和展示结果 —— [Jupyter Notebook 官方介绍](#)

简而言之，Jupyter Notebook 是以网页的形式打开，可以在网页页面中直接编写代码和运行代码，代码的运行结果也会直接在代码块下显示，如在编程过程中需要编写说明文档，可在同一个页面中直接编写，便于作及时的说明和解释

## Installation of Jupyter Notebook

### Install using Anaconda

我们可以通过安装 Anaconda 来解决 Jupyter Notebook 的安装问题，因为 Anaconda 已经自动为你安装了 Jupyter Notebook 及其他工具，还有 python 中超过 180 个科学包及其依赖项

常规来说，安装了 Anaconda 发行版时已经自动为你安装了 Jupyter Notebook 的，但如果没有自动安装，那么就在终端（Linux 或 macOS 的 Terminal，Windows 的 Anaconda Prompt，以下均简称 终端）中输入以下命令安装

```
1 conda install jupyter notebook
```

### Install using pip

如果你是有经验的 Python 玩家，可以用 `pip` 命令来安装 Jupyter Notebook，依次执行以下步骤即可

```
1 pip install --upgrade pip
```

**注意**，老版本的 `pip` 在安装 Jupyter Notebook 过程中或面临依赖项无法同步安装的问题，因此强烈建议先把 `pip` 升级到最新版本

再安装 Jupyter Notebook

```
1 pip install jupyter
```

## Run Jupyter Notebook

如果你有任何 Jupyter Notebook 命令的疑问，可以考虑查看官方帮助文档，命令如下

```
1 jupyter notebook --help
```

或

```
1 jupyter notebook -h
```



在终端中输入以下命令，即可 启动 Jupyter

```
1 | jupyter notebook
```

执行命令之后，在终端中将会显示一系列 Notebook 的服务器信息，同时浏览器将会自动启动 Jupyter Notebook

启动过程中终端显示内容如下

```
1 [I 16:57:25.472 NotebookApp] 启动notebooks 在本地路径：
  *****
2 [I 16:57:25.473 NotebookApp] 本程序运行在：
  http://localhost:8888/*****
3 [I 16:57:25.473 NotebookApp] or
  http://127.0.0.1:8888/*****
4 [I 16:57:25.473 NotebookApp] 使用control-c停止此服务器并关闭所有内核(两次跳过确认)。
```

注意，之后在 Jupyter Notebook 的所有操作，都请保持终端不要关闭，因为一旦关闭终端，就会断开与本地服务器的链接，你将无法在 Jupyter Notebook 中进行其他操作

浏览器地址栏中默认地将会显示：<http://localhost:8888>，其中，**localhost** 指的是本机，**8888** 则是端口号

如果你同时启动了多个 Jupyter Notebook，由于默认端口 8888 被占用，因此地址栏中的数字将从 8888 起，每多启动一个 Jupyter Notebook 数字就加 1，如 8889, 8890, ...

如果你想自定义端口号来启动 Jupyter Notebook，可以在终端中输入以下命令

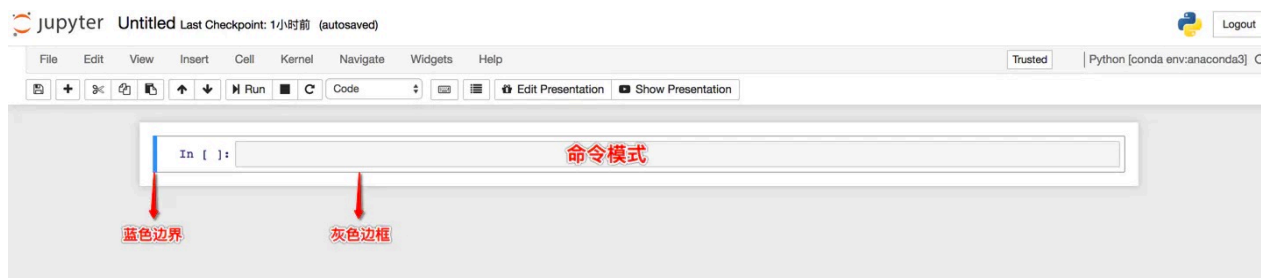
```
1 | jupyter notebook --port <port_number>
```

其中，`<port_number>` 是自定义端口号，直接以数字的形式写在命令当中

## Two modes of the Jupyter Notebook

### Command Mode

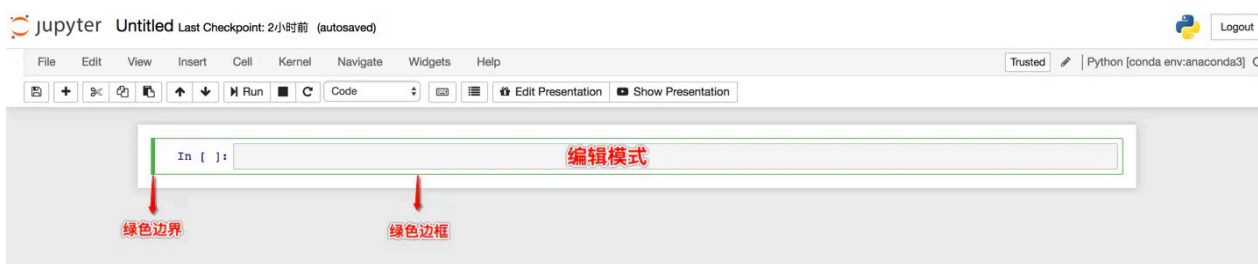
- 命令模式将键盘命令与 Jupyter Notebook 笔记本命令相结合，可以通过键盘不同键的组合运行笔记本的命令
- 按 Esc 键进入命令模式
- 命令模式下，单元格边框为灰色，且左侧边框线为蓝色粗线条



### Edit Mode



- 编辑模式使用户可以在单元格内编辑代码或文档
- 按 Enter 或 Return 键进入编辑模式
- 编辑模式下，单元格边框和左侧边框线均为绿色



## Shortcut for Jupyter

### Mac and Windows Special Keys Comparison Chart

符号	Mac 按键	Windows 按键
⌘	command	None
⌃	control	ctrl
⌥	option	alt
⇧	shift	shift
↵	return	return
␣	space	space
→	tab	tab
⌫	delete	backspace
⌘⌫	fn delete	delete

### Shortcut under Command Mode

快捷键	用途
F	查找和替换
↵	进入编辑模式
⌘⇧F	打开命令选项板
⌘⇧P	打开命令选项板
P	打开命令选项板
⇧↵	运行当前单元格并选中下一个单元格
⌃↵	运行选中单元格
⌥↵	运行当前单元格并且在下方插入单元格

Y	将单元格切换至 <b>Code</b> 状态
M	将单元格切换至 <b>Markdown</b> 状态
R	将单元格切换至 Raw 状态
1	将单元格设定一级标题
2	将单元格设定二级标题
3	将单元格设定三级标题
4	将单元格设定四级标题
5	将单元格设定五级标题
6	将单元格设定六级标题
↑	选中上方单元格
K	选中上方单元格
↓	选中下方单元格
J	选中下方单元格
⇧K	向上多选单元格
⇧↑	向上多选单元格
⇩J	向下多选单元格
⇩↓	向下多选单元格
A	在上方插入单元格
B	在下方插入单元格
X	剪切选中单元格
C	复制选中单元格
⇧V	粘贴到上方单元格
V	粘贴到下方单元格
Z	撤销删除
D, D	删除选中单元格
⇧M	合并选中单元格，若直选中一个则与下一个单元格合并
⌘S	保存
S	保存

L	转换行号
O	转换输出
⇧O	转换滚动输出
H	显示快捷键帮助
I, I	中断 Notebook 内核
O, O	重启 Notebook 内核
⇧L	转换所有单元格行号且设置持续有效
⇧⏮	向上滚动
⏭	向下滚动

### Shortcut under Edit Mode

Mac 快捷键	Windows 快捷键	用途
→	→	代码补全或缩进
⇧→	⇧→	提示
⌘]	^]	向后缩进
⌘[	^[	向前缩进
⌘A	^A	全选
⌘Z	^Z	撤销
⌘/		注释
⌘D		删除该行内容
⌘U		撤销
⌘↑	^↑	光标跳转至单元格起始位置
⌘↓	^↓	光标跳转至单元格最终位置
⌘←	^←	光标位置左移一个单词
⌘→	^→	光标位置右移一个单词
⌘⏮	^⏮	删除前边一个单词
⌘⏭	^⏭	删除后边一个单词
⌘⇧Z	^Y	重做
⌘⇧U	^⇧Z	重做

⌘⌫	^⌫	删除该行光标左边内容
⌘⇧⌫	^⇧⌫	删除该行光标右边内容
^M	^M	进入命令模式
esc	esc	进入命令模式
⌘⇧F		打开命令选项板
⌘⇧P		打开命令选项板
⇧↵	⇧↵	运行当前单元格并选中下一个单元格
^↵	^↵	运行选中单元格
⌘↵	⌘↵	运行当前单元格并且在下方插入单元格
^⇧-	^⇧-	以光标所在位置分割单元格
⌘S	^S	保存
↓	↓	下移光标
↑	↑	上移光标