致Python初学者：Anaconda入门使用指南

<http://python.jobbole.com/87522/?utm_source=blog.jobbole.com&utm_medium=relatedPosts>

打算学习 Python 来做数据分析的你，是不是在开始时就遇到各种麻烦呢？

* 到底该装 Python2 呢还是 Python3 ？
* 为什么安装 Python 时总是出错？
* 怎么安装工具包呢？
* 为什么提示说在安装这个工具前必须先安装一堆其他不明所以的工具？

相信大多数 Python 的初学者们都曾为环境问题而头疼不已，但你并不孤独，大家都是这么折腾过来的。为了在入门时少走弯路，并且让高涨的积极性不至于太受打击，这里推荐使用 Anaconda 来管理你的安装环境和各种工具包。

本文介绍了Anaconda的使用，全文大纲如下：

* 为什么选择 Anaconda
  + 什么是 Anaconda
  + 什么是 conda
  + Anaconda 的优点
* 如何安装 Anaconda
* 如何管理 Python 包
* 如何管理 Python 环境

## 一、为什么选择Anaconda？

### 1.1 什么是 Anaconda？

Anaconda是专注于数据分析的Python发行版本，包含了conda、Python等190多个科学包及其依赖项。作为好奇宝宝的你是不是发现了一个新名词 conda，那么你一定会问 conda 又是什么呢？

### 1.2 什么是 conda ？

conda 是开源包（packages）和虚拟环境（environment）的管理系统。

* **packages 管理：** 可以使用 conda 来安装、更新 、卸载工具包 ，并且它更关注于数据科学相关的工具包。在安装 anaconda 时就预先集成了像 Numpy、Scipy、 pandas、Scikit-learn 这些在数据分析中常用的包。另外值得一提的是，conda 并不仅仅管理Python的工具包，它也能安装非python的包。比如在新版的 Anaconda 中就可以安装R语言的集成开发环境 Rstudio。
* **虚拟环境管理：** 在conda中可以建立多个虚拟环境，用于隔离不同项目所需的不同版本的工具包，以防止版本上的冲突。对纠结于 Python 版本的同学们，我们也可以建立 Python2 和 Python3 两个环境，来分别运行不同版本的 Python 代码。

知道 **是什么（what）** 的同时，我们也需要问一问 **为什么（why）**。那么，为什么要选择用Anaconda呢？

### 1.3 Anaconda 的优点？

Anaconda的优点总结起来就八个字：省时省心、分析利器。

* **省时省心：** Anaconda通过管理工具包、开发环境、Python版本，大大简化了你的工作流程。不仅可以方便地安装、更新、卸载工具包，而且安装时能自动安装相应的依赖包，同时还能使用不同的虚拟环境隔离不同要求的项目。
* **分析利器：** 在 [Anaconda](https://www.continuum.io/why-anaconda) 官网中是这么宣传自己的：适用于企业级大数据分析的Python工具。其包含了720多个数据科学相关的开源包，在数据可视化、机器学习、深度学习等多方面都有涉及。不仅可以做数据分析，甚至可以用在大数据和人工智能领域。

解决了 **是什么** 以及 **为什么** 的问题后，下面让我们看一下 **怎么做（How）**。

## 二、如何安装Anaconda？

可以从[这里](https://www.continuum.io/downloads)下载 Anaconda 的安装程序以及查看安装说明。无论是 Windows、Linux 还是 MAC 的 OSX 系统，都可以找到对应的安装软件。如果你的电脑是64位则尽量选64位版本。至于 Python 的版本是 2.7 还是 3.x，这里推荐你使用 Python3，因为 Python2 终将[停止维护](https://pythonclock.org/" \t "_blank)。可能目前市面上大多数教程使用的都还是 Python2，这也不用着急，因为在 Anaconda 中可以同时管理两个 Python 版本的环境。

根据提示进行安装，完成后你大概会惊讶地发现电脑中多了好多应用，不用担心，我们一项项来看：

* **Anaconda Navigtor** ：用于管理工具包和环境的图形用户界面，后续涉及的众多管理命令也可以在 Navigator 中手工实现。
* **Jupyter notebook** ：基于web的交互式计算环境，可以编辑易于人们阅读的文档，用于展示数据分析的过程。
* **qtconsole** ：一个可执行 IPython 的仿终端图形界面程序，相比 Python Shell 界面，qtconsole 可以直接显示代码生成的图形，实现多行代码输入执行，以及内置许多有用的功能和函数。
* **spyder** ：一个使用Python语言、跨平台的、科学运算集成开发环境。

安装完成后，我们还需要对所有工具包进行升级，以避免可能发生的错误。打开你电脑的终端，在命令行中输入：



|  |  |
| --- | --- |
| 1 | conda upgrade --all |

在终端询问是否安装如下升级版本时，输入 y。

有的情况下，你可能会遇到找不到 conda 命令的错误提示，这很可能是环境路径设置的问题，需要添加conda环境变量：export PATH=xxx/anaconda/bin:$PATH, 其中xxx替换成anaconda的安装路径。

至此，安装完成，下面让我们看一下如何用 Anaconda 管理工具包和环境。

## 三、如何管理Python包？

安装一个 package：



|  |  |
| --- | --- |
| 1 | conda install package\_name |

这里 package\_name 是需要安装包的名称。你也可以同时安装多个包，比如同时安装numpy 、scipy 和 pandas，则执行如下命令：



|  |  |
| --- | --- |
| 1 | conda install numpy scipy pandas |

你也可以指定安装的版本，比如安装 1.1 版本的 numpy ：



|  |  |
| --- | --- |
| 1 | conda install numpy=1.10 |

移除一个 package：



|  |  |
| --- | --- |
| 1 | conda remove package\_name |

升级 package 版本：



|  |  |
| --- | --- |
| 1 | conda update package\_name |

查看所有的 packages：



|  |  |
| --- | --- |
| 1 | conda list |

如果你记不清 package 的具体名称，也可以进行模糊查询：



|  |  |
| --- | --- |
| 1 | conda  search search\_term |

## 四、如何管理Python环境？

默认的环境是 root，你也可以创建一个新环境：



|  |  |
| --- | --- |
| 1 | conda create -n env\_name  list of packages |

其中 -n 代表 name，env\_name 是需要创建的环境名称，list of packages 则是列出在新环境中需要安装的工具包。

例如，当我安装了 Python3 版本的 Anaconda 后，默认的 root 环境自然是 Python3，但是我还需要创建一个 Python 2 的环境来运行旧版本的 Python 代码，最好还安装了 pandas 包，于是我们运行以下命令来创建：



|  |  |
| --- | --- |
| 1 | conda create -n py2 python=2.7 pandas |

细心的你一定会发现，py2 环境中不仅安装了 pandas，还安装了 numpy 等一系列 packages，这就是使用 conda 的方便之处，它会自动为你安装相应的依赖包，而不需要你一个个手动安装。

进入名为 env\_name 的环境：



|  |  |
| --- | --- |
| 1 | source activate env\_name |

退出当前环境：



|  |  |
| --- | --- |
| 1 | source deactivate |

另外注意，在 Windows 系统中，使用 activate env\_name 和 deactivate 来进入和退出某个环境。

删除名为 env\_name 的环境：



|  |  |
| --- | --- |
| 1 | conda env remove -n env\_name |

显示所有的环境：



|  |  |
| --- | --- |
| 1 | conda env list |

当分享代码的时候，同时也需要将运行环境分享给大家，执行如下命令可以将当前环境下的 package 信息存入名为 environment 的 YAML 文件中。



|  |  |
| --- | --- |
| 1 | conda env export > environment.yaml |

同样，当执行他人的代码时，也需要配置相应的环境。这时你可以用对方分享的 YAML 文件来创建一摸一样的运行环境。



|  |  |
| --- | --- |
| 1 | conda env create -f environment.yaml |

至此，你已跨入 Anaconda 的大门，后续就可以徜徉在 Python 的海洋中了。

祝学习愉快！

注：本文代码示例参考自Udacity数据分析课程之Anaconda章节。