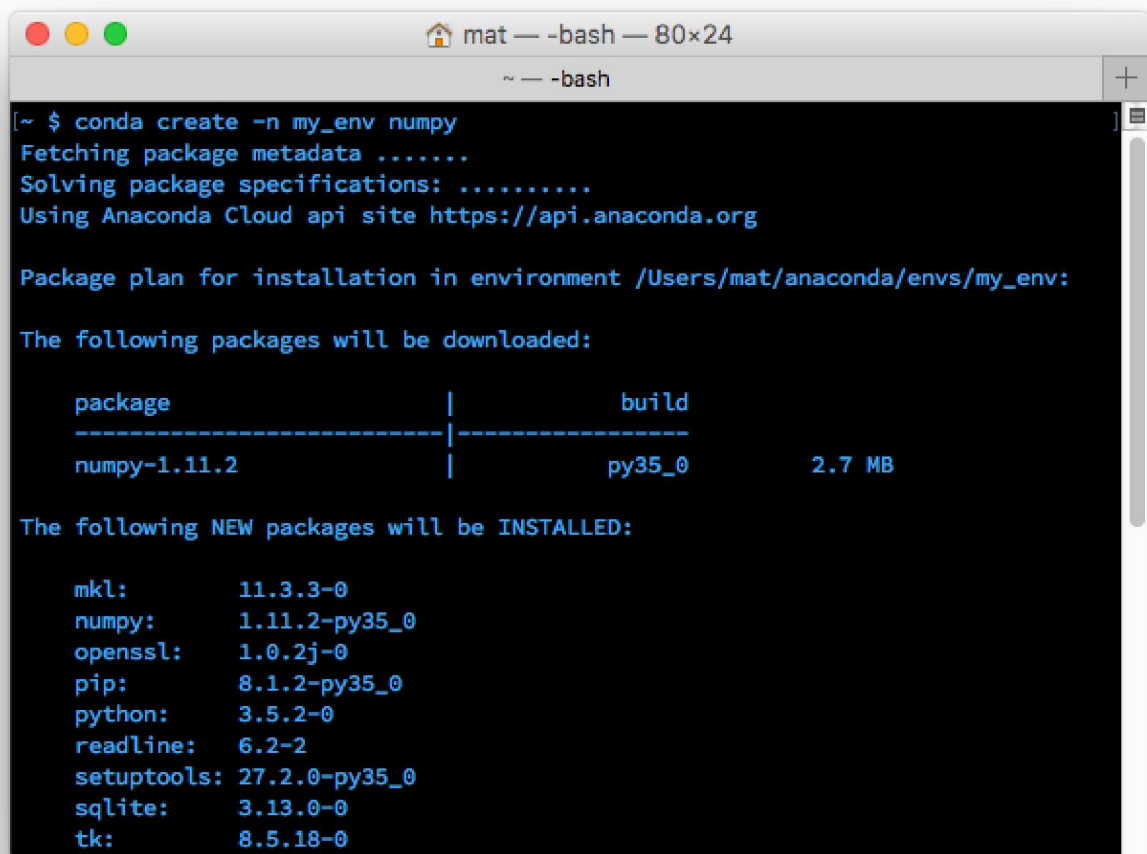


配置Anaconda

Anaconda 实际上是一个软件发行版，它附带了 conda、Python 和 150 多个科学包及其依赖项。应用程序 conda 是包和环境管理器。Anaconda 的下载文件比较大（约 500 MB），因为它附带了 Python 中最常用的数据科学包。如果只需要某些包，或者需要节省带宽或存储空间，也可以使用 Miniconda 这个较小的发行版（仅包含 conda 和 Python）。但你仍可以使用 conda 来安装任何可用的包，只是它自身没有附带这些包而已。

conda 只能通过命令行来使用。因此，如果你觉得它很难用，可以参考面向 **Windows 的命令提示符教程**，或者学习面向** OSX/Linux 用户的 Linux 命令行基础知识课程**。

管理包

A terminal window titled 'mat - bash - 80x24' showing the output of the command 'conda create -n my_env numpy'. The output includes package metadata fetching, solving specifications, and a list of packages to be installed. A table lists the packages to be downloaded: numpy-1.11.2 (py35_0, 2.7 MB). Below, a list of new packages to be installed is shown, including mkl, numpy, openssl, pip, python, readline, setuptools, sqlite, and tk.

```
[~ $ conda create -n my_env numpy
Fetching package metadata .....
Solving package specifications: .....
Using Anaconda Cloud api site https://api.anaconda.org

Package plan for installation in environment /Users/mat/anaconda/envs/my_env:

The following packages will be downloaded:

package | build
-----|-----
numpy-1.11.2 | py35_0 2.7 MB

The following NEW packages will be INSTALLED:

mkl:      11.3.3-0
numpy:    1.11.2-py35_0
openssl:  1.0.2j-0
pip:      8.1.2-py35_0
python:   3.5.2-0
readline: 6.2-2
setuptools: 27.2.0-py35_0
sqlite:   3.13.0-0
tk:       8.5.18-0
```

安装Anaconda

Anaconda 可用于 Windows、Mac OS X 和 Linux。可以在 <https://www.anaconda.com/download/> 上找到安装程序和安装说明。

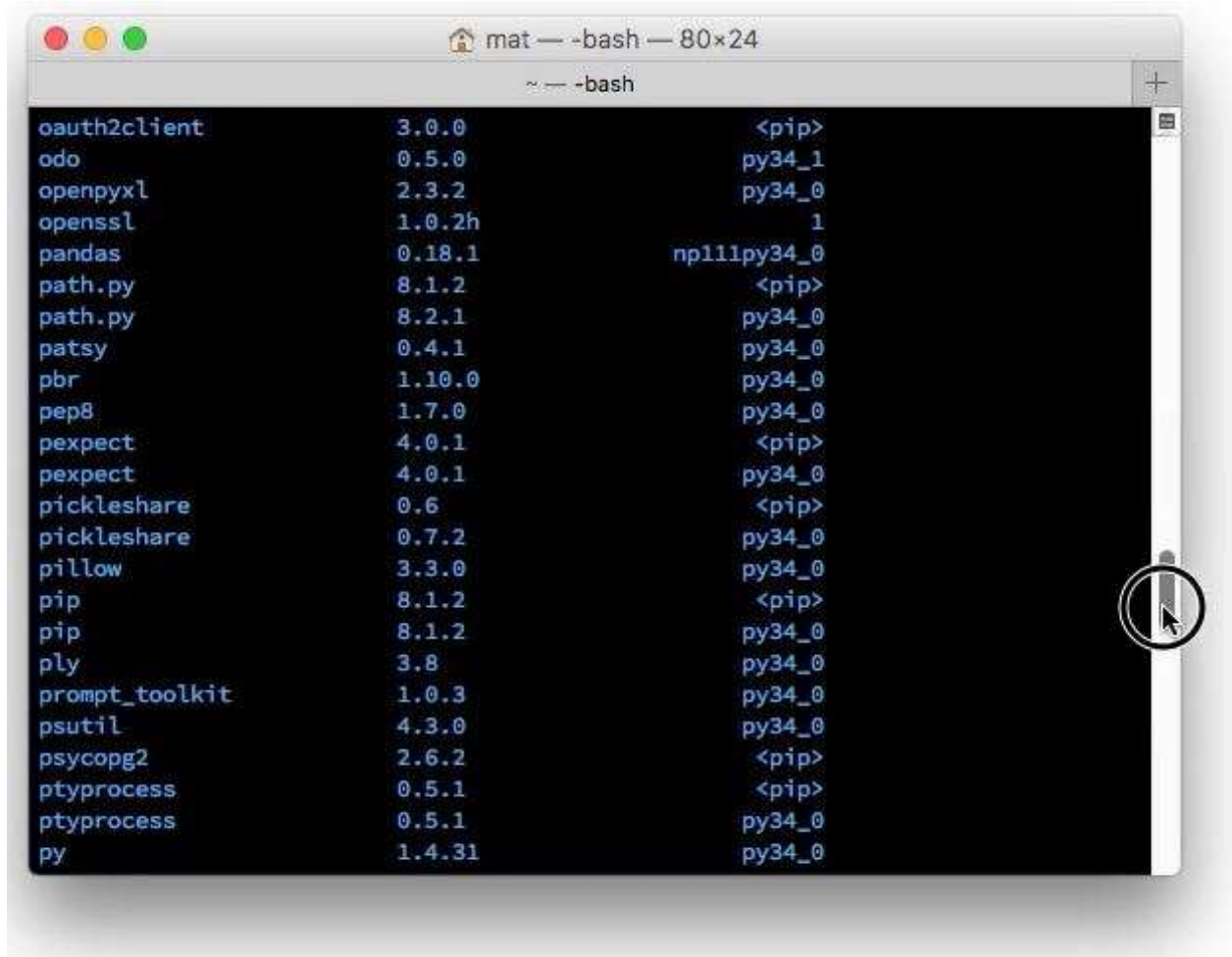
如果计算机上已经安装了 Python，这不会有任何影响。实际上，脚本和程序使用的默认 Python 是

如本计算例上已经安装 Python，这个云有任的影响。实际上，脚本和性序使用的默认 Python 是 Anaconda 附带的 Python。

选择 Python 3.6 版本（你也可以根据具体的需要选择 Python 2 的版本）。此外，如果是 64 位操作系统，则选择 64 位安装程序，否则选择 32 位安装程序。选择下载合适的版本，并继续进行安装！

完成安装后，会自动进入默认的 conda 环境，而且所有包均已安装完毕，如下面所示。可以在终端或命令提示符中键入

`conda list`，以查看你安装的内容。



在Windows上，会随Anaconda一起安装一批应用程序：

- Anaconda Navigator,它是用于管理环境和包的GUI
- Anaconda Prompt终端，它可让你使用命令行界面来管理环境和包
- Spyder,它是面向科学开发的IDE

为了避免报错，我推荐在默认环境下更新所有的包。打开 Anaconda Prompt（或者 Mac 下的终端），键入：

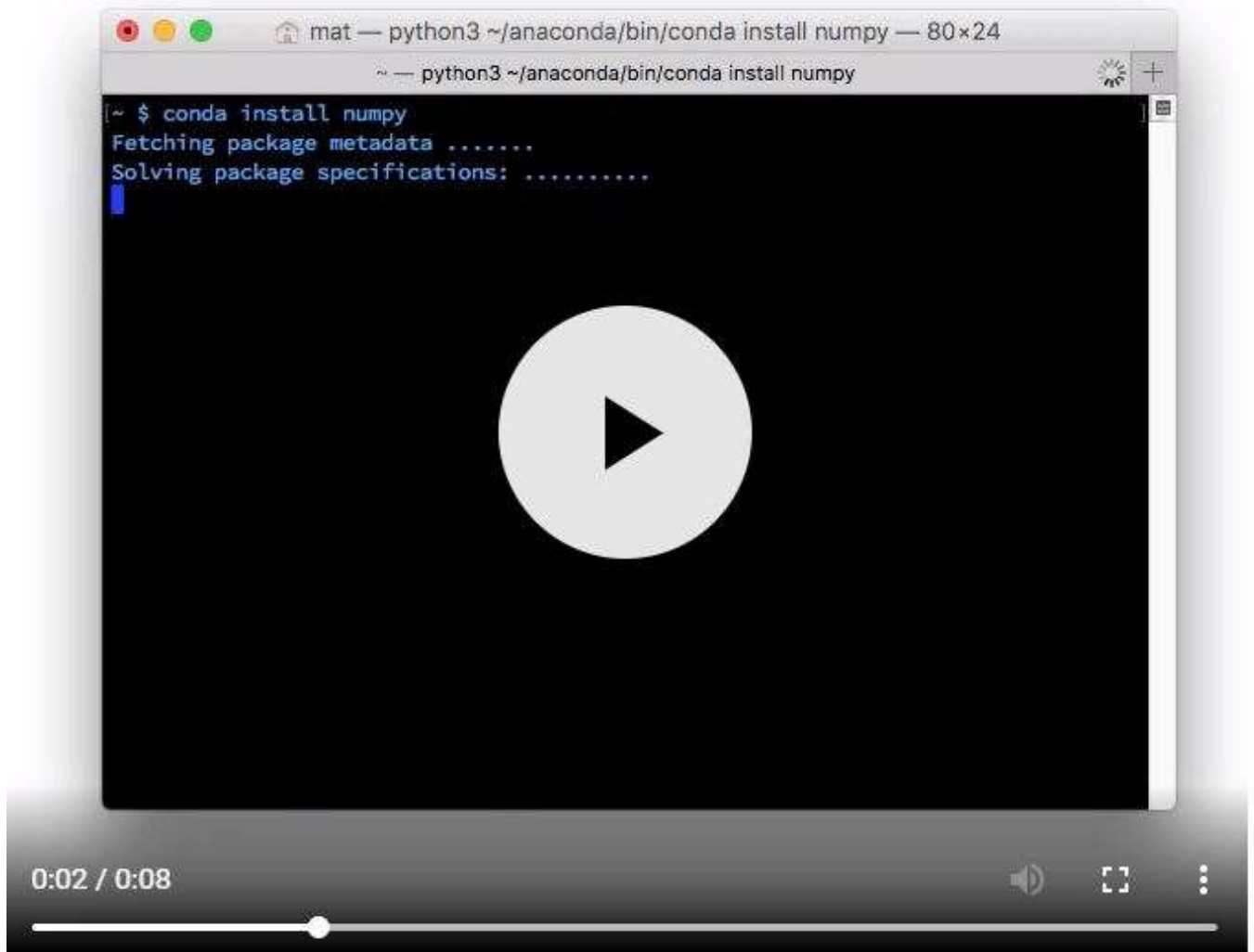
```
conda upgrade --all
```

并在提示是否更新的时候输入 y（Yes）以便让更新继续。初次安装下的软件包版本一般都比较老旧，因此提前更新可以避免未来不必要的问题。

管理包

安装了 Anaconda 之后，管理包是相当简单的。要安装包，请在终端中键入 `conda install package_name`。例

义表示 Anaconda 窗口，蓝色表示正在运行的。又如表示，请在终端中输入 `conda install package_name`。例如，要安装 `numpy`，请键入 `conda install numpy`。



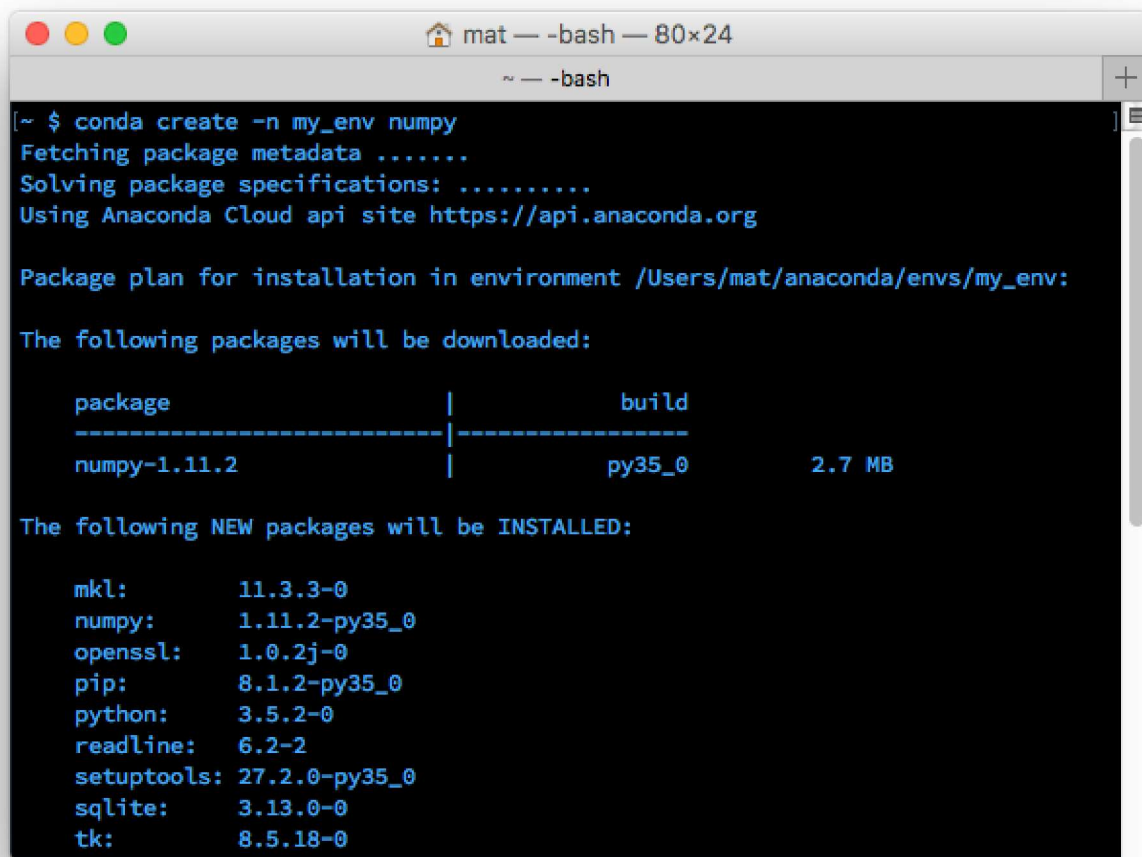
还可以同时安装多个包，类似 `conda install numpy scipy pandas` 的命令会同时安装所有这些包。还可以通过添加版本号(e.g `conda install numpy=1.10`)来指定所需的包版本。

Conda还会自动为你安装依赖项。例如，`scipy` 依赖于 `numpy`，因为它使用并需要 `numpy`。如果你只安装 `scipy` (`conda install scipy`)，则conda还会安装 `numpy` (如果尚未安装的话)

卸载包，请使用 `conda remove package_name`；要是更新包，请使用 `conda update package_name`；如果想更新环境中的所有包,请使用 `conda update --all`；要列出已安装的包，请使用 `conda list`；如果不知道要找的包的确切名称，可以尝试使用 `conda search search-term` 进行搜索。

管理环境

你可以使用 conda 创建环境。要创建环境，请在终端下使用 `conda create -n env_name list of packages`。在这里：`-n env_name` 设置环境的名称(`-n` 是指名称)
`list of packages` 是要安装在环境中的包的列表。例如：要创建名为 `my_env` 的环境并在其中安装 `numpy`，请键入 `conda create -n my_env numpy`



```
mat ~ -bash — 80x24
~ -bash
[~ $ conda create -n my_env numpy
Fetching package metadata .....
Solving package specifications: .....
Using Anaconda Cloud api site https://api.anaconda.org

Package plan for installation in environment /Users/mat/anaconda/envs/my_env:

The following packages will be downloaded:

package | build
-----|-----
numpy-1.11.2 | py35_0 2.7 MB

The following NEW packages will be INSTALLED:

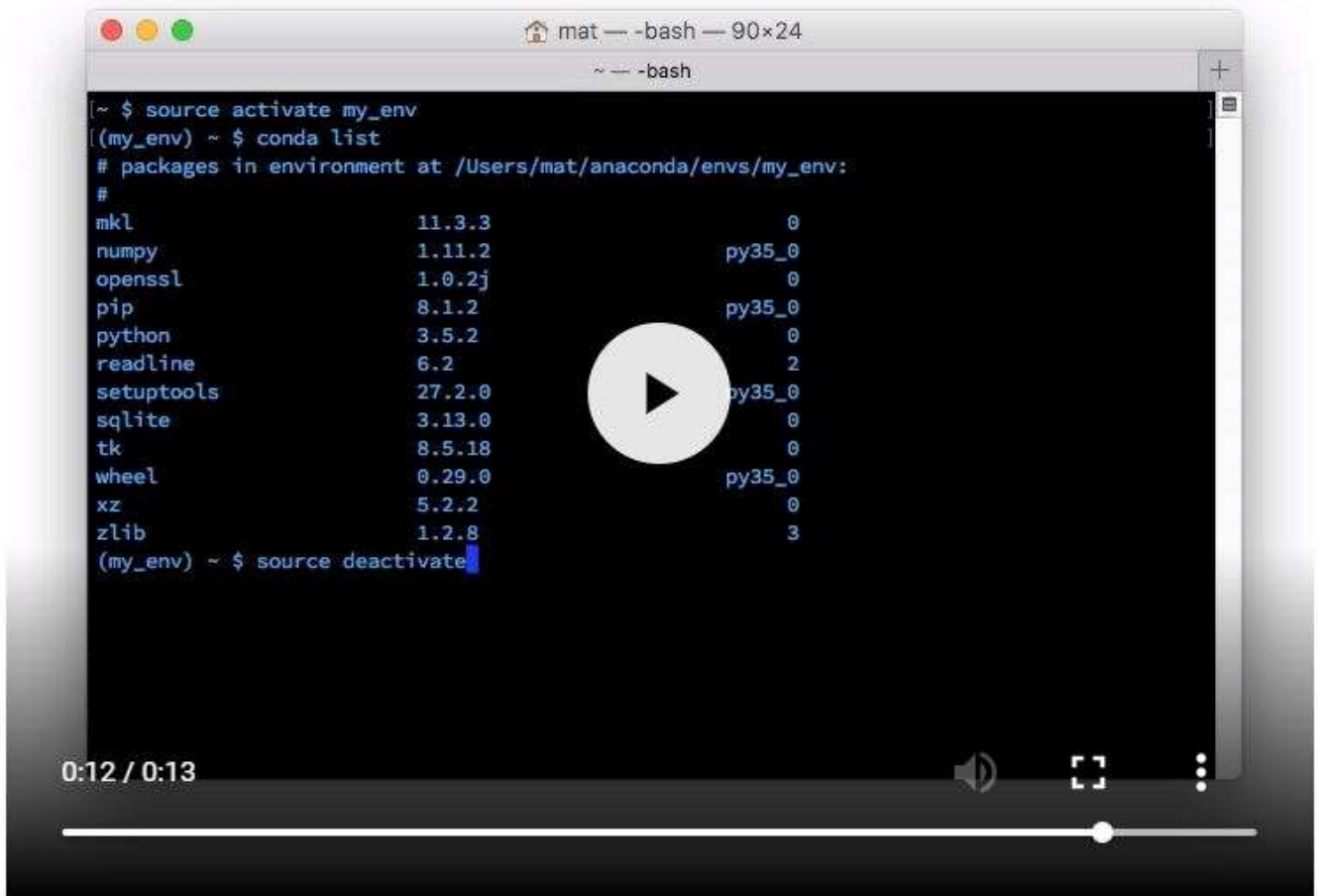
mkl:      11.3.3-0
numpy:    1.11.2-py35_0
openssl:  1.0.2j-0
pip:      8.1.2-py35_0
python:   3.5.2-0
readline: 6.2-2
setuptools: 27.2.0-py35_0
sqlite:   3.13.0-0
tk:       8.5.18-0
```

创建环境时，可以指定要安装在环境中的 Python 版本。这在你同时使用 Python 2.x 和 Python 3.x 中的代码时很有用。要创建具有特定 Python 版本的环境，请键入类似于 `conda create -n py3 python=3` 或 `conda create -n py2 python=2` 的命令。实际上，我在我的个人计算机上创建了这两个环境。我将它们用作与任何特定项目均无关的通用环境，以处理普通的工作（可轻松使用每个 Python 版本）。这些命令将分别安装 Python 3 和 Python 2 的最新版本。要安装特定版本（例如 Python 3.3），请使用 `conda create -n py python=3.3`。

进入环境

创建了环境后，在 OSX/Linux 上使用 `source activate my_env` 进入环境。在 Windows 上，请使用 `activate`

创建环境后，在 OSX/Linux 上使用 `source activate my_env` 进入环境。在 Windows 上，请使用 `activate my_env`。



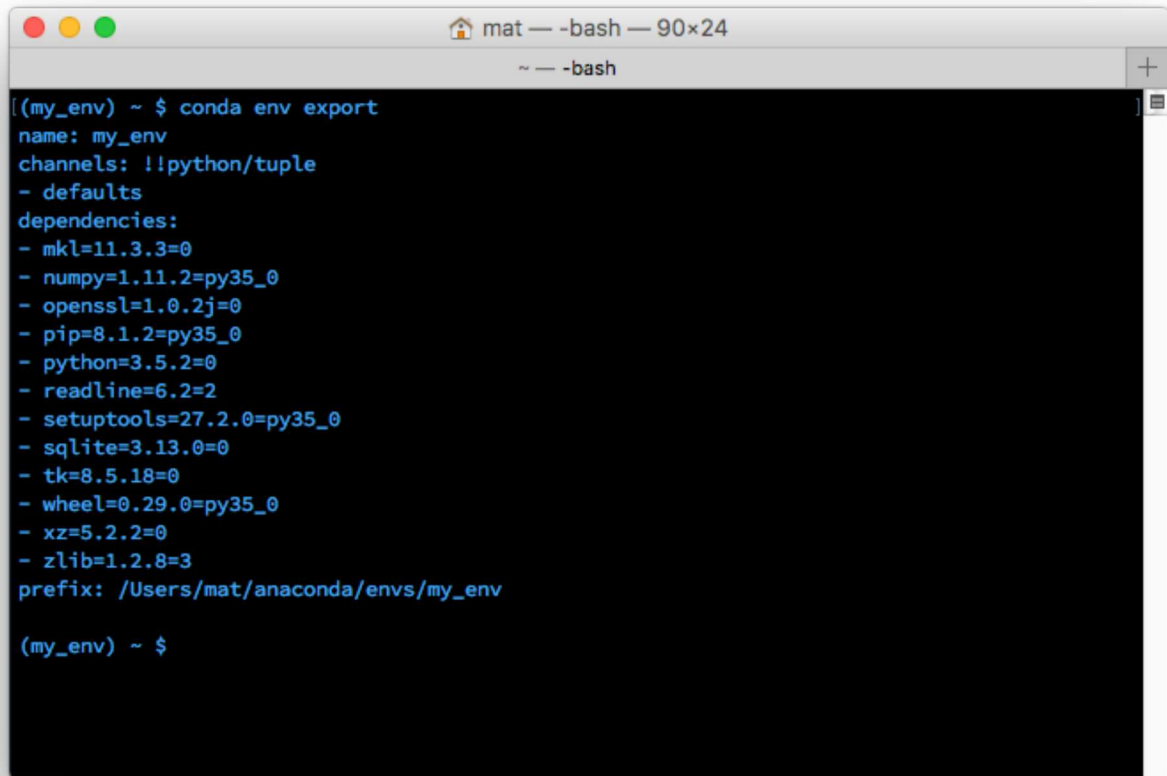
进入环境后，你会在终端提示符中看到环境名称，它类似于

```
(my_env) ~ $。
```

环境中只安装了几个默认的包，以及你在创建它时安装的包。你可以使用 `conda list` 检查这一点。在环境中安装包的命令与前面一样：`conda install package_name`。不过，这次你安装的特定包仅在你进入环境后才可用。要离开环境，请键入 `source deactivate`（在 OSX/Linux 上）。在 Windows 上，请使用 `deactivate`。

保存和加载环境

于子环境必须能够调用任何包，它能让其他人又表示你何时使用的所有包，并删除这些包的版本控制。你可以使用 `conda env export > environment.yaml` 将包保存为 YAML。命令的第一部分 `conda env export` 用于输出环境中的所有包的名称（包括 Python 版本）。

A terminal window titled 'mat - bash - 90x24' showing the output of the 'conda env export' command. The output is a YAML file content. The terminal text is as follows:

```
[(my_env) ~ $ conda env export
name: my_env
channels: !!python/tuple
- defaults
dependencies:
- mkl=11.3.3=0
- numpy=1.11.2=py35_0
- openssl=1.0.2j=0
- pip=8.1.2=py35_0
- python=3.5.2=0
- readline=6.2=2
- setuptools=27.2.0=py35_0
- sqlite=3.13.0=0
- tk=8.5.18=0
- wheel=0.29.0=py35_0
- xz=5.2.2=0
- zlib=1.2.8=3
prefix: /Users/mat/anaconda/envs/my_env

(my_env) ~ $
```

上图中，你可以看到环境的名称和所有依赖项及其版本。导出命令的第二部分 `> environment.yaml` 将导出的文本写入到 YAML 文件 `environment.yaml` 中。现在可以共享此文件，而且其他人能够用于创建和你项目相同的环境。

要通过环境文件创建环境，请使用 `conda env create -f environment.yaml`。这会创建一个新环境，而且它具有同样的在 `environment.yaml` 中列出的库。

列出环境

如果忘记了环境的名称（我有时会这样），可以使用 `conda env list` 列出你创建的所有环境。你会看到环境的列表，而且你当前所在环境的旁边会有一个星号。默认的环境（即当你不在选定环境中时使用的环境）名为 `root`。

删除环境

如果你不再使用某些环境，可以使用 `conda env remove -n env_name` 删除指定的环境（在这里名为 `env_name`）。

使用环境

Python 2 和 Python 3 具有独立的环境。我使用了 `conda create -n py2 python=2` 和 `conda create -n py3`

Python 2 和 Python 3 具有独立的环境。我使用了 `conda create -n py2 python=2` 和 `conda create -n py3 python=3` 创建两个独立的环境，即 py2 和 py3。现在，我的每个 Python 版本都有一个通用环境。在所有这些环境中，我都安装了大多数常用的标准数据科学包（numpy、scipy、pandas 等）。

共享环境

在 GitHub 上共享代码时，最好同样创建环境文件并将其包括在代码库中。这能让其他人更轻松地安装你的代码的所有依赖项。对于不使用 conda 的用户，我通常还会使用 `pip freeze`（[在此处了解详情](#)）将一个 `pip requirements.txt` 文件导出并包括在其中。