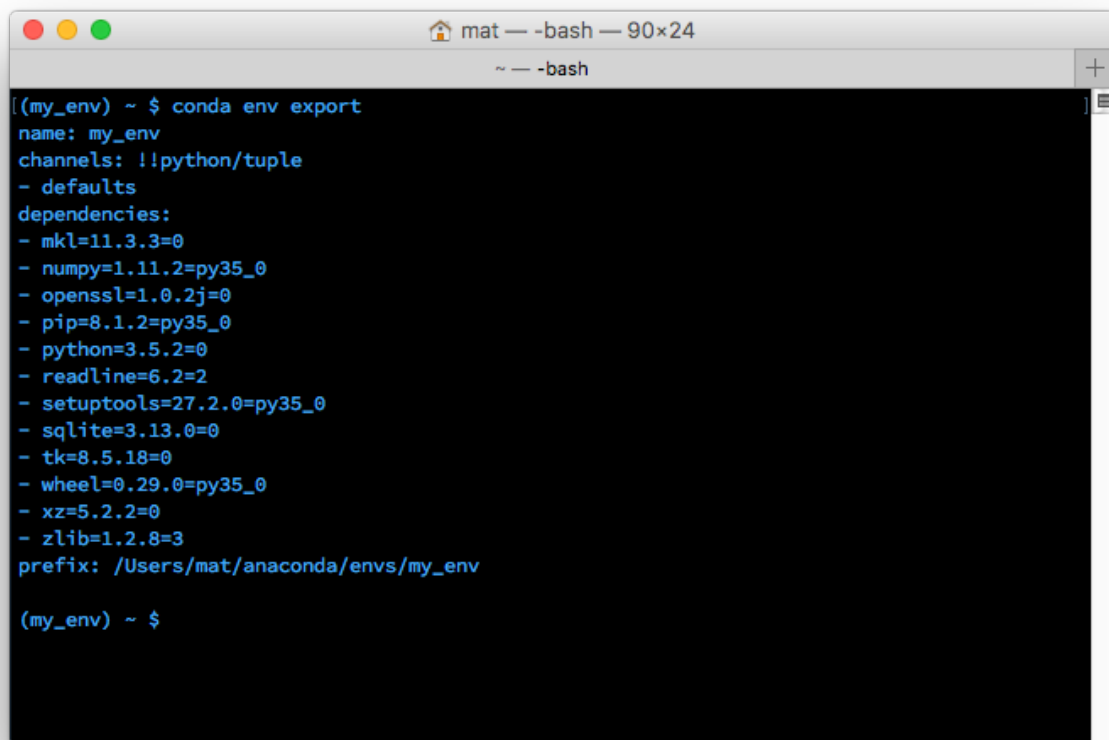




保存和加载环境

共享环境这项功能确实很有用，它能让其他人安装你的代码中使用的所有包，并确保这些包的版本正确。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）。



```
mat ~ -bash 90x24
~ -bash
(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) ~ $
```

将导出的环境输出到终端中

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



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` ）。

[下一项](#)