

主讲人：正点原子团队

硬件平台：正点原子ATK-DLRV1126开发板

版权所有：广州市星翼电子科技有限公司

资料下载：www.openedv.com/docs/index.html

教学平台：www.yuanzige.com

天猫店铺：zhengdianyuanzi.tmall.com

技术论坛：www.openedv.com/forum.php

公众平台：正点原子



■ Anaconda的基本使用

- 1, conda的基本命令介绍
- 2, 进入/退出base虚拟环境
- 3, 新建/切换/退出虚拟环境
- 4, 使用conda安装/卸载包
- 5, 了解软件包的存放目录
- 6, 删除未使用的包和缓存
- 7, 使用conda设置环境变量
- 8, 配置conda的源
- 9, 克隆/复制虚拟环境
- 10, 删除虚拟环境

1、conda的基本命令介绍

查看命令帮助

```
conda --help // 查看Conda的帮助信息
```

查看某个命令帮助

```
conda install --help // 查看conda install的帮助信息
```

Conda的命令的一般用法

命令用法:

```
conda <command> [options]
```

命令介绍:

<>括号里的表示必选项, []括号里的表示可选项, 命令格式为:

```
conda 命令 选项
```

如: conda build --help

1、conda的基本命令介绍

Conda命令--包管理

命令中的env_name表示虚拟环境的名字，pkg_name表示某个包名字，“-n”可以写为“--name”，“remove”可以换为uninstall

命令	说明
conda activate env_name conda list 或 conda list -n env_name	切换到某个虚拟环境下，然后列出该虚拟环境下所有的包
conda activate env_name conda install pkg_name 或 conda install -n env_name pkg_name	给某个虚拟环境单独安装包
conda install -use-local /home/alientek /a.tar.bz2	安装本地包，例如在/home/alientek下有个包a.tar.bz2
conda remove -n env_name --all conda remove pkg_name conda remove -n env_name pkg_name conda clean	删除包，包括移除虚拟环境 删除当前环境下的包（不会删除虚拟环境） 删除某个环境下的包（不会删除虚拟环境） 删除未使用的包和缓存
conda update conda	更新Conda本身
conda update pkg_name conda update --all conda update -n env_name pkg_name conda update -n env_name --all	更新当前虚拟环境下指定的包 更新当前虚拟环境下所有包 更新指定虚拟环境下指定的包 更新指定虚拟环境下所有包
conda search pkg_name	搜索特定的包
conda update pkg_name --no-pin	防止包更新

1、conda的基本命令介绍

Conda命令--虚拟环境管理

命令中的env_name表示虚拟环境的名字，URL表示某个数据源

命令	说明
conda --version	查看Conda版本信息
conda config --show	查看Conda的配置
conda create --name env_name	新建虚拟环境（Python版本为Ubuntu自带的版本，且创建的虚拟环境没有安装什么包）
conda create -n env_name python=3.6	新建指定Python版本的虚拟环境
conda create -n env_name pandas	使用特定包创建虚拟环境（创建安装了pandas包的虚拟环境，该虚拟环境的Python版本为base虚拟环境下的Python版本）
conda activate env_name	激活某个虚拟环境
conda deactivate	退出当前虚拟环境/去激活某个虚拟环境
conda env list 或 conda info --envs	查看所有环境（当前环境用*表示）
conda create -n new_env_name --clone old_env_name	克隆/复制虚拟环境
conda config --show channels	显示目前Conda的数据源有哪些
conda config --get channels	显示已有的数据源
conda config --add channels URL	添加数据源
conda config --remove channels URL	删除数据源
conda env config vars list	查看当前的环境变量
conda env config vars set a=1	设置环境变量，如设置一个环境变量a=1
conda env config vars unset a	取消设置的环境变量a
conda env export > environment.yml	生成需要分享虚拟环境的.yml文件
conda env create -f environment.yml	使用.yml文件创建虚拟环境

2、进入/退出base虚拟环境

Anaconda自带base虚拟环境（即基础虚拟环境）

```
conda info --envs    或    conda env list    或    conda info -e    // 列出有哪些虚拟环境
conda -V    或    conda --version            // 查看conda版本
conda update conda    // 若有需要，可以执行该命令更新conda
conda activate base    // 进入/激活base虚拟环境
conda list            // 列出当前虚拟环境下有哪些包
pip -V    或    pip --version            // 查看pip版本
python -V    或    python --version        // 查看Python版本
which python          // 查看Python的路径

conda activate base    // 临时切换/退出当前终端下的base虚拟环境
// 永久退出base虚拟环境的方法
conda config --set auto_activate_base false    // 关闭自动激活Conda的base虚拟环境
conda deactivate        // 切换/关闭/退出当前虚拟环境

// 若想重新开启自动激活Anaconda的base虚拟环境，可以执行如下命令
conda config --set auto_activate_base true
conda activate          // 进入base虚拟环境
```

- 1, conda的基本命令介绍
- 2, 进入/退出base虚拟环境
- 3, 新建/切换/退出虚拟环境
- 4, 使用conda安装/卸载包
- 5, 了解软件包的存放目录
- 6, 删除未使用的包和缓存
- 7, 使用conda设置环境变量
- 8, 配置conda的源
- 9, 克隆/复制虚拟环境
- 10, 删除虚拟环境

3、新建/切换/退出虚拟环境

新建虚拟环境的命令格式：

命令格式：

```
conda create --name env_name env_link
```

命令介绍：

--name --name可以写成-n

env_name 新建的虚拟环境名字，名字可自行定义，但不能和已有的名字重复

env_link 指定虚拟环境的Python版本，如果不指定这个参数，创建的虚拟环境默认使用Ubuntu系统环境自带Python环境！

新建不指定Python版本的虚拟环境no_py（那么，新建得到的虚拟环境使用的是Ubuntu自带的Python版本）

```
conda create --name no_py            // 新建的虚拟环境名字是no_py
```

```
conda create --name no_py -y            // 以后遇到需要输入y进行确认的，如需要跳过输入y这个步骤，可以在命令后加上“-y”
```

```
conda info --envs    或    conda env list    或    conda info -e    // 列出有哪些虚拟环境，出现“*”号的表示当前处于哪个虚拟环境下
```

```
conda activate no_py            // 激活/切换进入no_py虚拟环境
```

```
which python            // 查看Python的路径
```

```
which python3            // 查看Python3的路径
```

```
python --version    或    python -V            // 查看Python版本
```

```
python3 --version    或    python3 -V            // 查看Python3版本
```

```
conda deactivate            // 切换/关闭/退出当前no_py虚拟环境（而进入到上一次所在的虚拟环境）
```

3、新建/切换/退出虚拟环境

新建指定Python版本的虚拟环境py3.6（那么，新建得到的虚拟环境会在线安装所指定Python版本的Python包和其相关的依赖项）

```
// --name可以写为-n
conda create --name py3.6 python=3.6      // 新建虚拟环境，指定Python3.6
// 以后遇到需要输入y进行确认的情况，如需要跳过输入y这个步骤，可以在命令后加上“-y”
conda create --name py3.6 python=3.6 -y

conda info --envs    或    conda env list    或    conda info -e // 列出有哪些虚拟环境，出现“*”号的表示当前处于哪个虚拟环境下
conda activate py3.6 // 激活/切换新建的虚拟环境py3.6
which python         // 查看当前虚拟环境下的Python路径
python -V            或    python --version // 查看当前虚拟环境下的Python版本
pip -V               或    pip --version    // 查看pip版本
which pip            // 查看pip的路径
conda deactivate     // 切换/关闭/退出当前py3.6虚拟环境（而进入到上一次所在的虚拟环境）
```

- 1, conda的基本命令介绍
- 2, 进入/退出base虚拟环境
- 3, 新建/切换/退出虚拟环境
- 4, 使用conda安装/卸载包**
- 5, 了解软件包的存放目录
- 6, 删除未使用的包和缓存
- 7, 使用conda设置环境变量
- 8, 配置conda的源
- 9, 克隆/复制虚拟环境
- 10, 删除虚拟环境

4、使用conda安装/卸载包

Conda命令安装包的用法如下：

```
// 在线安装包

// 命令1：在虚拟环境env_name下安装包pkg_name
conda install -n env_name pkg_name           // -n可以写为--name

// 命令2：在当前虚拟环境下安装包pkg_name（当前在哪个虚拟环境下，就将包安装在哪个虚拟环境下）
conda install pkg_name

// 命令介绍：
env_name      虚拟环境名字
pkg_name      要安装的包名，Conda所有软件名都是小写

// 以上命令是在线安装包，下面命令是离线安装包

// 命令用法：
conda install --use-local pkg_name

// 命令介绍：
--use-local    该选项表示使用本地构建的包，可写成-c local
pkg_name      包的名字，具体到包的路径
```

4、使用conda安装/卸载包

Conda命令在线安装包示例：

```
conda activate py3.6           // 先切换到py3.6虚拟环境下
conda list                     // 查看py3.6虚拟环境下有啥包
conda activate base            // 进入到base虚拟环境下（进入base虚拟环境下只是为了演示在base下可以给py3.6安装包）
conda install -n py3.6 numpy   // 给py3.6虚拟环境安装numpy
// 若提示源有问题，可以指定从哪个源下载numpy
conda install -n py3.6 -c https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main numpy //从清华源下载numpy
conda activate py3.6;conda list // 进入py3.6虚拟环境下，查看有哪些包（如果要同时执行两个命令的话，可以通过分号将两个命令隔开）
```

Conda命令离线安装包示例：

numpy包下载链接：<https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/linux-64/>

```
conda create -n py36 python=3.6 // 新建一个虚拟环境py36（不是前面的py3.6），指定Python版本为3.6
conda info --envs 或 conda env list 或 conda info -e // 列出有哪些虚拟环境，出现“*”号的表示当前处于哪个虚拟环境下
conda activate py36 // 切换进入刚新建好的虚拟环境py36下
conda list // 列出当前虚拟环境py36下有哪些包
conda install --use-local numpy-base-1.19.2-py36hfa32c7d_0.tar.bz2 // 离线安装numpy-base
conda install --use-local numpy-1.19.2-py36h6163131_0.tar.bz2 // 离线安装numpy
conda list // 列出当前虚拟环境py36下有哪些包
```

4、使用conda安装/卸载包

Conda命令卸载包的用法如下：

```
// 命令1：卸载虚拟环境env_name下的包pkg_name
conda remove -n env_name pkg_name           // remove也可以写为uninstall, -n可以写为--name

// 命令2：在当前虚拟环境下卸载包pkg_name（当前在哪个虚拟环境下，就卸载哪个虚拟环境下的包）
conda remove pkg_name                       // remove也可以写为uninstall

// 命令介绍：
env_name      虚拟环境名字
pkg_name      要卸载的包名，Conda所有软件名都是小写
```

Conda命令卸载包示例：

```
// -n可以写为--name
conda remove -n py3.6 numpy -y              // 卸载虚拟环境py3.6下的numpy包（跳过输入y的步骤，直接卸载）
conda uninstall -n py3.6 numpy -y          // 也可以将remove写为uninstall
conda activate py3.6; conda list            // 查看虚拟环境py3.6下有哪些包
conda activate py36                         // 切换到虚拟环境py36下
// 卸载当前虚拟环境py36下的numpy包（不管当前处于哪个虚拟环境下，只要有numpy包，跳过输入y的步骤，直接卸载）
conda remove numpy -y    或    conda uninstall numpy -y
```

- 1, conda的基本命令介绍
- 2, 进入/退出base虚拟环境
- 3, 新建/切换/退出虚拟环境
- 4, 使用conda安装/卸载包
- 5, 了解软件包的存放目录
- 6, 删除未使用的包和缓存
- 7, 使用conda设置环境变量
- 8, 配置conda的源
- 9, 克隆/复制虚拟环境
- 10, 删除虚拟环境

5、了解软件包的存放目录

在/home/alientek/anaconda3/envs/py3.6/conda-meta下，有很多配置文件：

① 有很多以.json结尾的配置文件，这些文件记录了对应软件包的：

版本信息、镜像源的路径、md5和SHA256校验码、依赖项信息、包下载后存放的目录，以及包解压的位置等等信息。

② 还有一个history文件，记录了从创建虚拟环境开始之后的所有历史操作记录。

```
conda activate py3.6           // 如果没有进入py3.6虚拟环境下的话，执行该命令切换到py3.6虚拟环境下
pip list                       // 查看虚拟环境py3.6下已安装的包（库或者模块）的名称
python -m site                 // 查看虚拟环境py3.6下第三方库（包/或者模块）的默认安装路径
pip show certify               // 查看虚拟环境py3.6下具体的模块（或者包/库）安装的路径
cat history                    // 查看虚拟环境py3.6的历史操作记录
```


6、删除未使用的包和缓存

通过conda clean --help可查询conda clean命令的具体用法，conda clean命令后面可以加选项，在选项之后可以添加控制流选项以控制是否显示进度条或者是否需要输入y确认删除等。

注意的是，conda clean是针对整个Anaconda的，**其会对所有虚拟环境下的包进行搜索然后进行删除**，达到瘦身的目的。

命令用法说明如下表所示：

选项	说明
删除选项	说明
-a 或 --all	删除索引缓存、锁定文件、未使用的缓存包和.tar
-i 或 --index-cache	删除索引缓存
-p 或 --packages	删除未使用的包
-t 或 --tarballs	删除缓存的包.tar
-f 或 --force-pkgs-dirs	删除所有可写包缓存
输出、提示和流控制选项	说明
-d 或 --dry-run	只显示会做的事情
--json	将所有输出报告为 json
-q 或 --quiet	不显示进度条
-v 或 --verbose	可以多次使用，INFO 一次，DEBUG 两次，TRACE 三次
-y 或 --yes	不要要求确认，即在指令后加-y可跳过确认选项

6、删除未使用的包和缓存

删除从未使用过的包：

```
cd anaconda3;du -sh           // 删除包之前，先计算anaconda目录的大小
conda clean --packages         // 删除从未使用过的包（删除索引缓存、锁定文件、未使用的缓存包）
conda clean --packages -y      // 如果需要跳过输入y确认以删除未使用过的包，可以使用该命令
du -sh                         // 删除包之后，再去计算 anaconda目录的大小

conda clean --tarballs         // 删除.tar压缩包
conda clean --tarballs -y      // 如果想跳过输入y确认以.tar包，可以使用该命令

// 以上的两步删除操作，可以使用如下一命令来完成，从而达到一键瘦身的目的
conda clean -a                 // 删除索引缓存、锁定文件、未使用的缓存包和.tar
conda clean -a -y              // 如果想跳过输入y确认删除的步骤，可以在命令的后面加上“-y”参数
```

- 1, conda的基本命令介绍
- 2, 进入/退出base虚拟环境
- 3, 新建/切换/退出虚拟环境
- 4, 使用conda安装/卸载包
- 5, 软件包的存放目录
- 6, 删除未使用的包和缓存
- 7, 使用conda设置环境变量
- 8, 配置conda的源
- 9, 克隆/复制虚拟环境
- 10, 删除虚拟环境

7、使用conda设置环境变量

在py3.6虚拟环境下设置环境变量a=1:

```
conda activate py3.6           // 如果没有进入py3.6虚拟环境下的话, 执行该命令切换到py3.6虚拟环境下
conda env config vars list     // 列出当前虚拟化境下的环境变量
conda env config vars set a=1  // 在当前虚拟环境下设置环境变量a=1
conda activate py3.6           // 设置完环境变量后, 可以再执行该命令去激活py3.6虚拟环境, 确保该环境变量a生效
echo $a                        // 打印当前虚拟环境py3.6下的环境变量a的值 (重新激活py3.6后, 该命令可查询到a的值)
// 若要删除前面设置的环境变量a, 可以运行以下命令
conda env config vars unset a  // 取消当前虚拟环境下设置的环境变量a
conda activate py3.6           // 取消环境变量后, 可以再执行该命令去激活py3.6虚拟环境, 确保该环境变量a已经被取消
```

在其它虚拟环境下给虚拟环境py3.6设置环境变量a=1:

```
conda activate base            // 先切换到base虚拟环境
conda env config vars set a=1 --name py3.6 // 设置py3.6虚拟环境下的环境变量a的值为1
conda env config vars list --name py3.6    // 列出py3.6虚拟环境base下的环境变量
conda env config vars list        // 列出当前虚拟环境 (base) 下的环境变量 (base下是没有环境变量的)
// 打开另外一个终端, 执行两次conda activate py3.6命令, 就可以查询到设置的环境变量了
conda activate py3.6; conda activate py3.6; conda env config vars list; echo $a
conda env config vars unset a --name py3.6 // 删除py3.6虚拟环境下的环境变量a
conda env config vars list --name py3.6    // 列出py3.6虚拟环境下的环境变量
```

- 1, conda的基本命令介绍
- 2, 进入/退出base虚拟环境
- 3, 新建/切换/退出虚拟环境
- 4, 使用conda安装/卸载包
- 5, 软件包的存放目录
- 6, 删除未使用的包和缓存
- 7, 使用conda设置环境变量
- 8, 配置conda的源**
- 9, 克隆/复制虚拟环境
- 10, 删除虚拟环境

8、配置conda的源

国内一些源（channel）：

清华大学开源软件镜像站<https://mirrors.tuna.tsinghua.edu.cn/help/anaconda/>

① 清华大学的conda源配置参考（.condarc文件配置）：

```
channels:
- defaults
show_channel_urls: true
default_channels:
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
custom_channels:
conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
pytorch-lts: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
deepmodeling: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

国内一些源（channel）：

北京外国语大学开源软件镜像站<https://mirrors.bfsu.edu.cn/>

② 北京外国语大学的conda源配置参考（.condarc文件配置）：

```
channels:
- defaults
show_channel_urls: true
channel_alias: https://mirrors.bfsu.edu.cn/anaconda
default_channels:
- https://mirrors.bfsu.edu.cn/anaconda/pkgs/main
- https://mirrors.bfsu.edu.cn/anaconda/pkgs/free
- https://mirrors.bfsu.edu.cn/anaconda/pkgs/r
- https://mirrors.bfsu.edu.cn/anaconda/pkgs/pro
- https://mirrors.bfsu.edu.cn/anaconda/pkgs/msys2
custom_channels:
conda-forge: https://mirrors.bfsu.edu.cn/anaconda/cloud
msys2: https://mirrors.bfsu.edu.cn/anaconda/cloud
bioconda: https://mirrors.bfsu.edu.cn/anaconda/cloud
menpo: https://mirrors.bfsu.edu.cn/anaconda/cloud
pytorch: https://mirrors.bfsu.edu.cn/anaconda/cloud
simpleitk: https://mirrors.bfsu.edu.cn/anaconda/cloud
```

8、配置conda的源

国内一些源（channel）：

浙江大学开源软件镜像站<http://mirrors.zju.edu.cn/>

③ 浙江大学的conda源配置参考（.condarc文件配置）：

channels:

- defaults

show_channel_urls: true

default_channels:

- <https://mirrors.zju.edu.cn/anaconda/pkg/main>
- <https://mirrors.zju.edu.cn/anaconda/pkg/r>
- <https://mirrors.zju.edu.cn/anaconda/pkg/msys2>

custom_channels:

conda-forge: <https://mirrors.zju.edu.cn/anaconda/cloud>

msys2: <https://mirrors.zju.edu.cn/anaconda/cloud>

bioconda: <https://mirrors.zju.edu.cn/anaconda/cloud>

menpo: <https://mirrors.zju.edu.cn/anaconda/cloud>

pytorch: <https://mirrors.zju.edu.cn/anaconda/cloud>

pytorch-lts: <https://mirrors.zju.edu.cn/anaconda/cloud>

simpleitk: <https://mirrors.zju.edu.cn/anaconda/cloud>

国内一些源（channel）：

阿里巴巴<https://developer.aliyun.com/mirror/anaconda>

④ 阿里巴巴的conda源配置参考（.condarc文件配置）：

channels:

- defaults

show_channel_urls: true

default_channels:

- <http://mirrors.aliyun.com/anaconda/pkg/main>
- <http://mirrors.aliyun.com/anaconda/pkg/r>
- <http://mirrors.aliyun.com/anaconda/pkg/msys2>

custom_channels:

conda-forge: <http://mirrors.aliyun.com/anaconda/cloud>

msys2: <http://mirrors.aliyun.com/anaconda/cloud>

bioconda: <http://mirrors.aliyun.com/anaconda/cloud>

menpo: <http://mirrors.aliyun.com/anaconda/cloud>

pytorch: <http://mirrors.aliyun.com/anaconda/cloud>

simpleitk: <http://mirrors.aliyun.com/anaconda/cloud>

8、配置conda的源

其它国内一些源（channel）详情链接：

西安交通大学软件镜像站<https://mirrors.xjtu.edu.cn/>

兰州大学<https://mirror.lzu.edu.cn/help/#/docs/anaconda>

上海交通大学<https://mirror.sjtu.edu.cn/docs/anaconda>

南京大学<https://mirror.nju.edu.cn/mirrorz-help/anaconda/?mirror=NJU>

南京工业大学 <https://mirrors.njtech.edu.cn/docs/anaconda>

上海科技大学<https://mirrors.shanghaitech.edu.cn/>

南方科技大学<https://mirrors.sustech.edu.cn/help/anaconda.html>

阿里巴巴<https://developer.aliyun.com/mirror/anaconda>

哈尔滨工业大学开源镜像站<https://mirrors.hit.edu.cn/#/home>

8、配置conda的源

查看conda的配置和源（channel）：

```
conda activate base           // 切换到base虚拟环境下（先查看base虚拟环境下的配置）
conda config --show           // 打印conda的所有配置（包括源）
conda info                   // 查看base虚拟环境的信息
conda config --show-sources   // 查看conda的配置文件位置以及变量的值
```

defaults是Anaconda官方维护的用于下载包的通道，defaults的源是：

<https://repo.anaconda.com/pkgs/main>

<https://repo.anaconda.com/pkgs/r>

conda-forge算是第三方源，conda-forge官网：

<https://conda-forge.org/>

conda的配置文件是/home/alientek/.condarc

单独查看conda的源（channel）：

```
conda config --show channels   // 查看conda的源
conda config --get channels    // 查看conda的源
```

8、配置conda的源

conda的配置文件:

当第一次执行完conda config命令后，会在用户的家目录下生成一个.condarc文件，该文件就是conda应用程序的配置文件（如果没有.condarc文件，用户也可以手动新建该文件），该文件保存了conda的一些配置，可以在该文件中添加源。

```
vi .condarc
```

```
// 打开conda的配置文件
```

```
auto_activate_base: false
channels:
  - defaults
~
~
```

auto_activate_base表示是否开启base的自动激活Conda的base虚拟环境功能，若为true，表示开启自动激活功能，若为false，表示关闭自动激活功能。

channels就是conda源的配置。

8、配置conda的源

给conda添加源（通过命令的方式添加）：

命令用法，`new_channel`表示具体的源地址：

```
conda config --add channels new_channel           // 添加新的源，默认最高优先级
conda config --prepend channels new_channel        // 添加新的源到顶部，最高优先级
conda config --append channels new_channel         // 添加新的源到底部，最低优先级
conda config --add default_channels new_channe    // 添加default_channels的源
```

// 添加清华源 的具体命令如下

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
```

// 查看源等配置信息

```
conda config --show-sources           // 查看conda的配置文件位置以及变量的值
conda info                           // 查看base虚拟环境的信息
conda config --get channels           // 查看conda的源
conda config --show channels          // 查看conda的源
conda config --show                   // 打印conda的所有配置（包括源）
```

8、配置conda的源

给conda删除源（通过命令的方式删除）：

```
conda config --show-sources // 查看conda的配置文件位置以及变量的值
```

// 删除源的命令用法，URL表示源地址：

```
conda config --remove channels URL // 删除channels的源（URL表示源地址）
```

```
conda config --remove default_channels URL // 删除default_channels 的源
```

// 挨个删除之前添加的源：

```
conda config --remove channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
```

```
conda config --remove channels https://mirrors.hit.edu.cn/anaconda/cloud/
```

// 删完再查看源，核实一下是否已经删除成功

```
conda config --show-sources // 查看conda的配置文件位置以及变量的值
```

8、配置conda的源

给conda添加源（通过修改.condarc的方式添加，该方式操作起来更方便）：

```
vi .condarc
```

.condarc文件配置如下：

```
channels:
- defaults
show_channel_urls: true
default_channels:
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
- https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
custom_channels:
conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
pytorch-lts: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
deepmodeling: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/
```

show_channel_urls: true表示开启显示源的URL，即下载包的时候会显示从哪个URL下载。也可以通过命令行的方式开启或者关闭该选项（配置为false为关闭）：
conda config --set show_channel_urls true // 下载时显示源地址（即URL）

每次在配置好后，建议执行如下命令清除索引缓存，以保证使用的是镜像站提供的索引：

```
conda clean -i
```

然后再使用conda命令去安装包，如：

```
conda install numpy
```

```
conda install pytorch -c pytorch
```

8、配置conda的源

换回默认的源:

- ① 可以通过修改.condarc文件来实现
- ② 可以通过执行命令行来实现

```
conda config --remove-key custom_channels // 删除custom_channels下的源
conda config --remove-key default_channels // 删除default_channels下的源
conda clean -i // 清除缓存索引
conda config --show channels // 查看conda的源
```

临时指定源来安装包:

以在指令后面通过“-c”选项来指定从哪个源下载安装包,“-c”即channel(源)。

```
conda activate py3.6 // 若在py3.6虚拟环境下安装的话,可以切换到py3.6虚拟环境下
conda install -c https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main numpy // 从清华源下载Numpy
```

8、配置conda的源

再谈源的优先级:

```
conda config --show channel_priority      // 查看channel_priority的值
conda config --describe channel_priority  // 查看channel_priority的说明
```

channel_priority的值可以是strict（严格）、flexible（灵活）和disabled（禁用），默认的值是flexible，默认情况下我们采用flexible即可：

- ① 如果channel_priority配置为strict，则表示如果具有相同名称的包出现在较高优先级的channel中时，则不考虑较低优先级channel中的包。strict的通道优先级可以显著加快 Conda 操作，不过要注意的是，严格的通道优先级可能会筛除满足所需的包，所以默认情况下先将channel_priority配置为flexible。
- ② 如果channel_priority配置为flexible，则表示可以进入优先级较低的通道查找包，查找的包尽量满足依赖关系。
- ③ 如果channel_priority配置为disabled，表示禁用通道优先级，此时，软件包版本优先，配置的通道优先级仅用于断开连接。

```
// 假设需要将channel_priority的值修改为strict，可通过执行如下命令来配置（但是不建议修改，否则可能会导致网络不通，建议采用默认的flexible）
// 改完以后，记得最好重启一下Ubuntu以使配置生效，要不然可能后面在线安装包时提示网络不行。
```

```
conda config --set channel_priority strict      // 配置channel_priority的值为strict
conda config --show channel_priority           // 查看channel_priority的值
conda config --set channel_priority flexible    // 配置channel_priority的值为flexible
```

8、配置conda的源

查找包版本和源:

```
命令用法:
// 若要精确查找包, 则可采用如下命令, pkg_name必须是包的全名
conda install --full-name pkg_name

// 若要模糊查找, 可使用如下命令, 查找的是包含字段pkg_name的包
conda search pkg_name

// 执行如下命令搜索pillow
conda search pillow      // 搜索pillow

// 搜索到包以后, 可以使用如下命令来安装包
命令用法:
conda install pkg_name=pkg_version

命令介绍:
pkg_name           包的名字
pkg_version        包的版本

// 安装5.3.0版本的Pillow
conda install pillow=5.3.0

conda list | grep pillow      // 查询是否有Pillow这个包 (可以通过-c来指定源安装)
```


- 1, conda的基本命令介绍
- 2, 进入/退出base虚拟环境
- 3, 新建/切换/退出虚拟环境
- 4, 使用conda安装/卸载包
- 5, 软件包的存放目录
- 6, 删除未使用的包和缓存
- 7, 使用conda设置环境变量
- 8, 配置conda的源
- 9, 克隆/复制虚拟环境
- 10, 删除虚拟环境

9、克隆/复制虚拟环境

要求Ubuntu能联网，能在线安装包

在本地计算机（Ubuntu）上克隆/复制虚拟环境

命令格式：

```
conda create --name newname --clone oldname
```

命令介绍：

newname 新的虚拟环境，即目标虚拟环境

oldname 旧的虚拟环境，即源虚拟环境

```
conda create --name clone.py --clone py3.6
```

// 将py3.6的虚拟环境配置克隆到clone.py虚拟环境下

```
conda env list
```

// 列出所有的虚拟环境

```
conda activate clone.py
```

// 进入虚拟环境clone.py下

```
conda info
```

// 查看虚拟环境clone.py的配置

```
conda list
```

// 查看虚拟环境clone.py装了哪些包

```
conda activate py3.6
```

// 进入虚拟环境py3.6下

```
conda info
```

// 查看虚拟环境py3.6的配置，可以和clone.py的配置进行对比

```
conda list
```

// 查看虚拟环境py3.6安装了哪些包

要求Ubuntu能联网，能在线安装包

9、克隆/复制虚拟环境

通过.yml文件克隆虚拟环境（利用得到的.yml文件，可以实现在本地/非本地计算机上克隆/复制一个一模一样的虚拟环境）

```
conda activate py3.6                // 激活/进入/切换虚拟环境py3.6
conda env export > py3.6.yml         // 导出虚拟环境py3.6的配置到py3.6.yml文件中

// 若要安装的Ubuntu/计算机下已有和py3.6同名的虚拟环境，请修改导出的py3.6.yml 文件中的name属性为其它名字后再进行后续的操作
// （笔者改为copy_from_py3.6 了）

conda env create --file py3.6.yml    // 从py3.6.yml文件中导入虚拟环境
conda info --envs                   // 列出所有的虚拟环境
conda activate copy_from_py3.6      // 进入克隆/复制得到的虚拟环境copy_from_py3.6下
conda info                          // 查看虚拟环境copy_from_py3.6的配置信息
conda list                          // 列出虚拟环境copy_from_py3.6安装了哪些包
```

注意：

- ① Ubuntu要能联网，能在线下载和安装包；
- ② 在同一个Anaconda目录下，不能有同名的虚拟环境，若有同名的虚拟环境，可通过修改py3.6.yml 文件的name属性，改一下虚拟环境的名字即可

- 1, conda的基本命令介绍
- 2, 进入/退出base虚拟环境
- 3, 新建/切换/退出虚拟环境
- 4, 使用conda安装/卸载包
- 5, 软件包的存放目录
- 6, 删除未使用的包和缓存
- 7, 使用conda设置环境变量
- 8, 配置conda的源
- 9, 克隆/复制虚拟环境
- 10, 删除虚拟环境

10、删除虚拟环境

如果要移除新建的虚拟环境，可执行如下命令：

命令格式：

```
conda remove --name env_name --all
```

命令介绍：

env_name 是新建的虚拟环境名字，自行定义。指令中的remove也可以换成uninstall，即：

```
conda uninstall --name env_name --all
```

// 具体命令如下

```
conda info --envs
```

// 查看当前有哪些虚拟环境

```
conda deactivate copy_from_py3.6
```

// 确保已经退出虚拟环境copy_from_py3.6以后，再去删除它

```
conda remove --name copy_from_py3.6 --all
```

// 卸载/删除虚拟环境copy_from_py3.6

```
conda info --envs
```

// 查看当前有哪些虚拟环境

注意：

如果此时已经激活/进入要删除的虚拟环境，应该先退出要删除的虚拟环境以后，再去删除虚拟环境



版权所有：广州市星翼电子科技有限公司
天猫店铺：<https://zhengdianyuanyi.tmall.com>