Getting started

This 30-minute getting started procedure consists of the following exercises:

- Managing conda. Verify that Anaconda or Miniconda is installed and check that conda is updated to the current version. 3 minutes
- Managing environments. Create a few environments and then learn to move easily between them. Verify which environment you are in and make a copy of an environment as a backup. 10 minutes
- Managing Python. See which versions of Python are available to install, install another version of Python and then switch between versions. 4 minutes
- Managing packages. Work with packages:

    - List packages installed on your computer.
    - List available packages.
    - Install and remove some packages using `conda install`.
    - For packages not available using `conda install`, search on Anaconda.org.
    - For packages that are in neither location, install a package with the pip package manager. Install a free 30-day trial of Anaconda's commercial package, IOPro.

  10 minutes
- Removing packages, environments or conda. Remove 1 or more of your test packages, environments and/or conda. 3 minutes

TOTAL TIME: 30 minutes

TIP: To see the full documentation for any command, view the command-line help.

Managing conda

To manage conda versions:

Use the Terminal or an Anaconda Prompt for the following steps.

1. Verify that conda is installed:

   ```
   conda --version
   ```

Conda displays the number of the version that you have installed.

EXAMPLE: `conda 3.11.0`

NOTE: If you see an error message, verify that you are logged into the same user account that you used to install Anaconda or Miniconda and that you have closed and re-opened the Terminal window after installing it.

2. Update conda to the current version:

   ```
   conda update conda
   ```

   Conda compares versions and then displays what is available to install. It also tells you about other packages that will be automatically updated or changed with the update.

3. If a newer version of conda is available, type `y` to update:

   ```
   Proceed ([y]/n)? y
   ```

Managing environments

To create a few environments and then move between them:

Use the Terminal or an Anaconda Prompt for the following steps.

1. Create an environment with `conda create`:

   ```
   conda create --name snowflakes biopython
   ```

   This creates a new environment named “snowflakes” with the program Biopython.

   TIP: You can abbreviate many frequently used options that are preceded by 2 dashes (`--`) to just 1 dash and the first letter of the option. So `--name` and `-n` are the same, and `--envs` and `-e` are the same. For a list of abbreviations, see `conda --help` or `conda -h`.

2. To activate the new environment, run the appropriate command for your operating system:

   - Linux and macOS: `source activate snowflakes`
   - Windows: `activate snowflakes`

   TIP: By default, conda installs environments into the `envs` directory in your `conda` directory. To specify a different path, see `conda create --help`.

   TIP: Since you did not specify a version of Python, conda installs the same version used when conda was downloaded and installed.

3. Create a new environment and then install a different version of Python along with 2 packages named Astroid and Babel:

```
conda create --name bunnies python=3.5 astroid babel
```

This creates a second new environment in `/envs` named “bunnies”, with Python 3, Astroid and Babel installed.

TIP: Install all the programs you will want in this environment at the same time. Installing 1 program at a time can lead to dependency conflicts.

TIP: You can add much more to the conda create command. See `conda create --help` for details.

4. Display the environments that you have installed so far:

   ```
   conda info --envs
   ```

   A list of environments appears, similar to the following:

   ```
   conda environments:

       snowflakes    * /home/username/miniconda/envs/snowflakes
       bunnies         /home/username/miniconda/envs/bunnies
       root            /home/username/miniconda
   ```

   Conda puts an asterisk (*) in front of the active environment.

5. Verify the current environment:

   ```
   conda info --envs
   ```

   Conda displays the list of all environments, with the current environment shown in (parentheses) or [brackets] in front of your prompt:

   ```
   (snowflakes) $
   ```

6. Switch to another environment:

   - Linux, macOS: `source activate bunnies`
   - Windows: `activate bunnies`

7. Change your path from the current environment back to the root:

   - Linux, macOS: `source deactivate`
   - Windows: `deactivate`

   TIP: When the environment is deactivated, its name is no longer shown in the prompt.

8. Make a copy of the snowflakes environment by creating a clone of it called “flowers”:

   ```
   conda create --name flowers --clone snowflakes
   ```

9. Verify that the copy was made:

   ```
   conda info --envs
   ```

   The 3 environments are listed: flowers, bunnies and snowflakes.
10. Delete the flowers environment:

    ```
    conda remove --name flowers --all
    ```

11. Verify that the flowers environment has been removed:

    ```
    conda info --envs
    ```

    The flowers environment is no longer in your list, so you know it was deleted.

Managing Python

Conda treats Python the same as any other package, so it is easy to manage and update multiple installations.

To check which Python versions are available to install, in your Terminal window or an Anaconda Prompt, run:

```
conda search --full-name python
```

The `--full-name` option lists only the packages whose full name is exactly "python". To list all packages whose names contain the text "python", use `conda search python`.

To install Python 3 without overwriting your Python 2.7 environment:

Use the Terminal or an Anaconda Prompt for the following steps.

1. Create a new environment named "snakes" and install the latest version of Python 3:

   ```
   conda create --name snakes python=3
   ```

2. Activate the new environment:

   - Linux, macOS: `source activate snakes`
   - Windows: `activate snakes`
3. Verify that the snakes environment has been added:

```
   conda info --envs
   ```

   Conda displays the list of all environments, with the current environment shown in (parentheses) or [brackets] in front of your prompt:

   ```
   (snakes) $
   ```

4. Verify that the snakes environment uses Python version 3:

   ```
   python --version
   ```

5. Switch back to the default, version 2.7:

   - Linux, macOS: `source activate snowflakes`
   - Windows: `activate snowflakes`
6. Verify that the snowflakes environment uses the same Python version that was used when you installed conda:

   ```
   python --version
   ```

7. Deactivate the snowflakes environment and then revert your PATH to its previous state:

   - Linux, macOS: `source deactivate`
   - Windows: `deactivate`

Managing packages

You have already installed several packages—Astroid, Babel and a specific version of Python—when you created a new environment. In this section, you check which packages you have, check which are available and look for a specific package and install it.

Then you look for specific packages on the Anaconda.org repository, install packages from Anaconda.org, install more packages using `pip` install instead of `conda install` and then install a commercial package.

To find a package:

Use the Terminal or an Anaconda Prompt for the following steps.

1. To confirm that a package has been added or removed, view a list of packages and versions installed in an environment:

```
conda list
```

2. View a list of packages available for `conda install`, sorted by Python version, at http://docs.continuum.io/anaconda/pkg-docs.html
3. Check to see if a package called “beautifulsoup4” is available for conda to install:

   ```
   conda search beautifulsoup4
   ```

   This displays the package, so we know it is available.

To install the package:

Use the Terminal or an Anaconda Prompt for the following steps.

1. Install beautifulsoup4 into the current environment:

   ```
   conda install --name bunnies beautifulsoup4
   ```

   NOTE: If you don’t specify the name of the environment, as in `--name bunnies`, conda installs into the current environment.
2. Activate the bunnies environment:
   - Linux, macOS: `source activate bunnies`
   - Windows: `activate bunnies`
3. List the newly installed program:

   ```
   conda list
   ```

Installing packages from Anaconda.org

For packages that are not available using `conda install`, look on Anaconda.org, a package management service for both public and private package repositories. Like Anaconda and Miniconda, Anaconda.org is an Anaconda product.

TIP: You are not required to register with Anaconda.org to download files.

To download into the current environment from Anaconda.org, you need to specify Anaconda.org as the channel by typing the full URL to the package that you want. To find this URL:

1. In a browser, go to http://anaconda.org.

2. Look for a package named “bottleneck”:

    1. In the top-left corner of the screen, in the Search Anaconda Cloud box, type `bottleneck`.
    2. Click the Search button.

    There are more than a dozen copies of bottleneck available on Anaconda.org, but you want the most frequently downloaded copy.
3. Click the Downloads column heading to sort the results by number of downloads.
4. Click the package name of the version that has the most downloads.

    The Anaconda.org detail page appears, showing the command to use to download the package:

    ```
    conda install --channel https://conda.anaconda.org/pandas bottleneck
    ```

5. Run the displayed command.
6. To check that the package downloaded, in the Terminal or an Anaconda Prompt, run:

    ```
    conda list
    ```

Installing a package with pip

For packages that are not available from conda or Anaconda.org, you can often install the package with pip, which stands for “pip installs packages.”

TIP: Pip is only a package manager, so it cannot manage environments for you. Pip cannot even update Python, because unlike conda, it does not consider Python a package. But it does install some things that conda does not, and vice versa. Both pip and conda are included in Anaconda and Miniconda.

Use the Terminal or an Anaconda Prompt for the following steps.

1. Activate the environment where you want to put the program, such as bunnies:

    - macOS and Linux—`source activate bunnies`
    - Windows—`activate bunnies`
2. Install a program named “see”:

    ```
    pip install see
    ```

3. Verify that see was installed:

    ```
    conda list
    ```

Installing commercial packages

Installing commercial packages is the same as installing any other package with conda.

EXAMPLE: To install a free trial of one of Anaconda's commercial packages, IOPro, which can speed up your Python processing, in your Terminal window or an Anaconda Prompt, run:

```
conda install iopro
```

TIP: Except for academic use, this free trial expires after 30 days.

Removing packages, environments, or conda

To remove 1 or more of your test packages, environments, and/or conda:

1. To remove the commercial package IOPro from the bunnies environment, in your Terminal window or an Anaconda Prompt, run:

   ```
   conda remove --name bunnies iopro
   ```

2. To confirm that IOPro has been removed, in your Terminal window or an Anaconda Prompt, run:

   ```
   conda list
   ```

3. To remove the snakes environment, in your Terminal window or an Anaconda Prompt, run:

   ```
   conda remove --name snakes --all
   ```

4. To verify that the snakes environment has been removed, in your Terminal window or an Anaconda Prompt, run:

   ```
   conda info --envs
   ```

   You know that snakes was deleted because it no longer appears in the environment list.
5. Remove conda:

   - For Linux and macOS, remove the Anaconda or Miniconda install directory:

     ```
     rm -rf ~/miniconda
     ```

or:

```
rm -rf ~/anaconda
```

- For Windows: In Control Panel, select Add or Remove Programs, select Python X.X (Anaconda) or Python X.X (Miniconda) and then click Remove Program.

  NOTE: Replace X.X with your version of Python.

  NOTE: Instructions are different for Windows 10.

More information

- Full documentation—https://conda.io/docs/.
- Cheat sheet—Cheat sheet.
- FAQs—http://docs.continuum.io/anaconda/faq.html.
- Free community support—https://groups.google.com/a/anaconda.com/forum/#!forum/anaconda.
- Paid support options—https://www.anaconda.com/support/.
- Training—https://www.anaconda.com/training/.
- Consulting—https://www.anaconda.com/consulting/.