Getting started with Navigator

Anaconda Navigator is a desktop application that is built on top of the conda package and environment manager. Navigator enables you to use conda to manage environments and packages, but in a graphical application instead of a command line interface (CLI).

This short guide takes you through the basic steps of using Navigator.

> **Note**
>
> Conda's getting started guide goes through some of these same instructions while using the conda CLI.

Before you start

You should have already installed Anaconda Distribution, which contains the Navigator application.

Starting Navigator

Navigator starts by default when Anaconda Distribution is first installed. If you have installed Anaconda Distribution before, open Navigator using the following instructions:

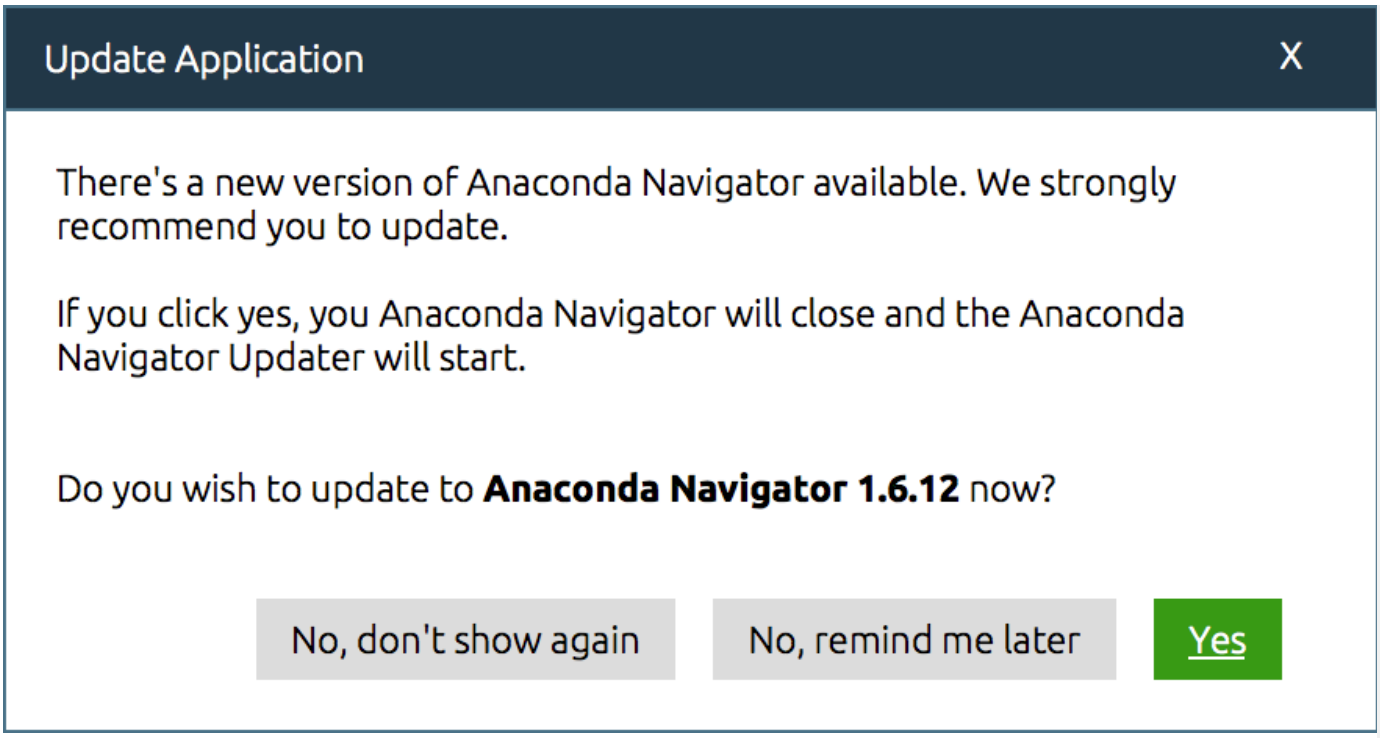
Windows | **MacOS** | **Linux**

From the Start menu, search for "Anaconda Navigator" and click to open.

Managing Navigator

> **Note**
>
> When Navigator opens, it verifies that Anaconda is installed. If Navigator does not start up properly, see the Troubleshooting page.

When you start Navigator, it automatically checks for a new version. If Navigator finds a new version, you will see a dialog box like this:

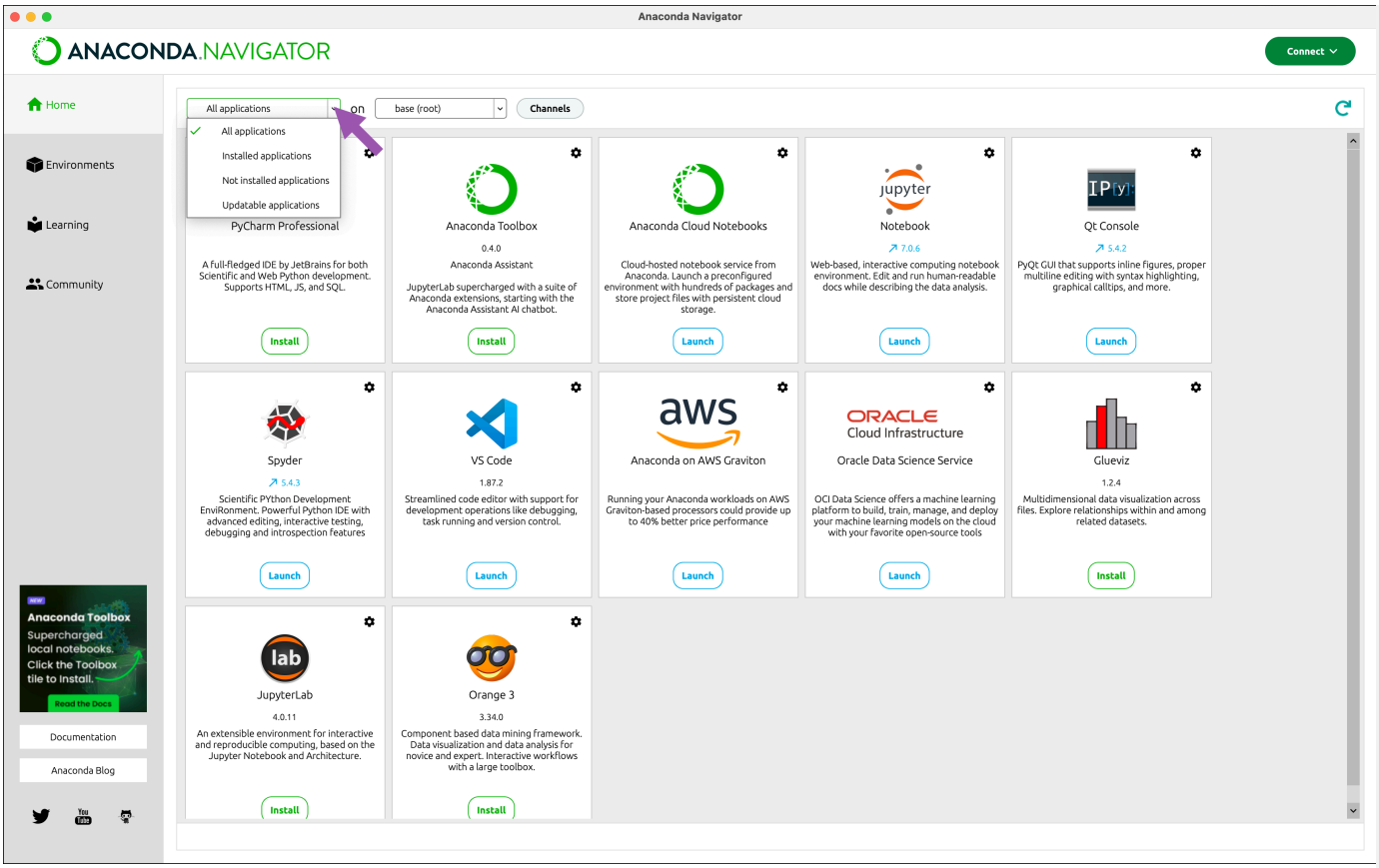


Click **Yes** to update Navigator to the current version.

> **Tip**
>
> Anaconda recommends keeping Navigator updated to the latest version.

Managing application tiles

By default, all application tiles available to launch or install within Navigator are displayed on the Home page. Filter the application tiles with the applications dropdown menu.



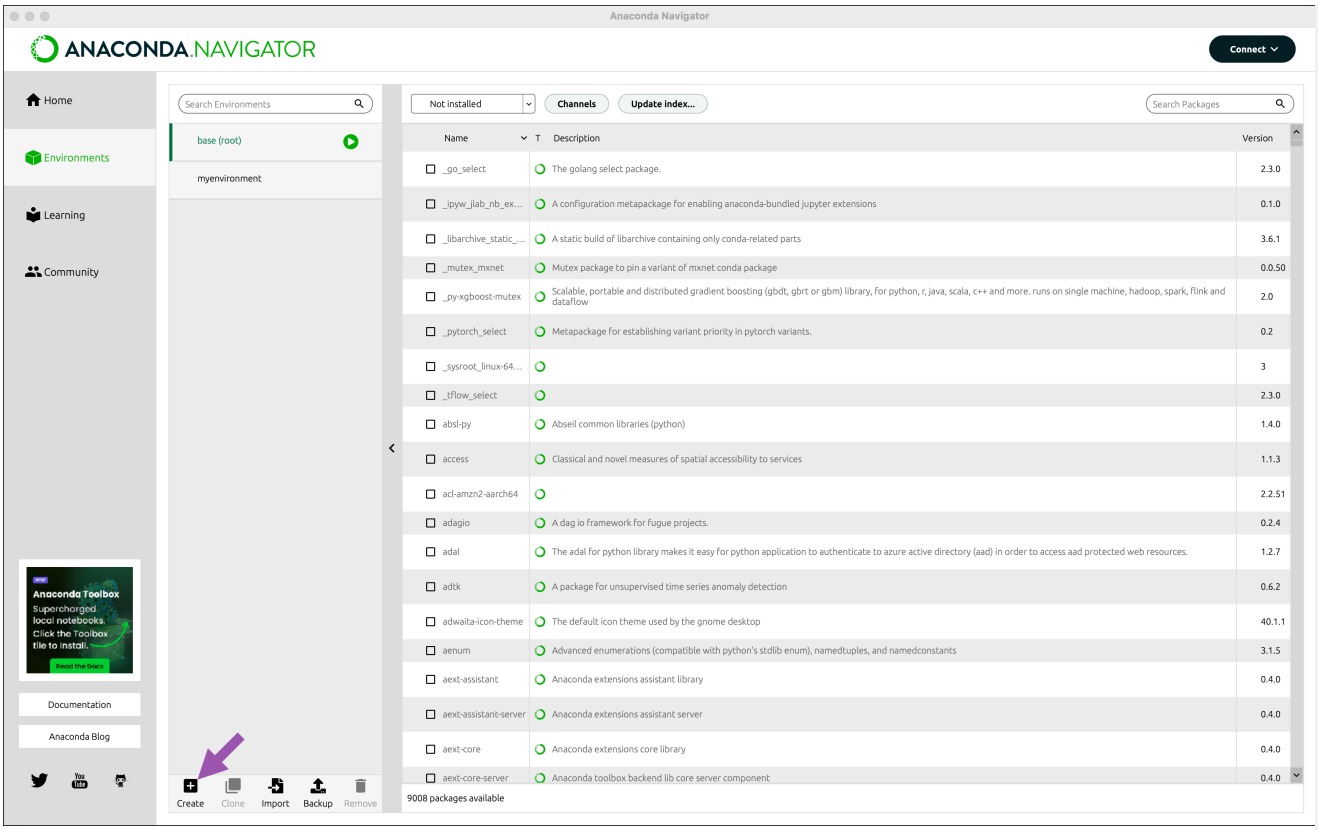
The dropdown contains filters for all applications, applications that are installed on your computer, applications that are not installed, and applications that have an update available.

Managing environments

Navigator uses the conda package and environment manager to create conda environments. Conda environments are directories within your file system that contain a specific collection of packages and package dependencies. Environments are isolated from one another and enable you to quickly switch between projects with drastically different needs.

In this example, we will create a new environment named `myenvironment`:

1. On the **Environments** page, click **Create**.



2. In the **Environment** name field, type a descriptive name for your environment.
3. Click **Create**. Navigator creates the new environment and activates it.
4. Now you have two environments: the default environment `base (root)` and `myenvironment`.

> **Caution**
>
> Anaconda does not recommend working in your base environment.

5. Switch between environments by clicking the name of the environment you want to use. This will deactivate the first environment and activate the environment you just selected.

> **Tip**
>
> The active environment is the one with the arrow beside its name.

For more documentation on environments, see Managing environments in the Tutorials section.

Managing Python

When you create a new environment, Navigator installs Python 3.11 by default. To use a different version of Python in a given environment:

- specify when you create the environment
- change the Python package version in an existing environment

Changing the Python version when creating an environment

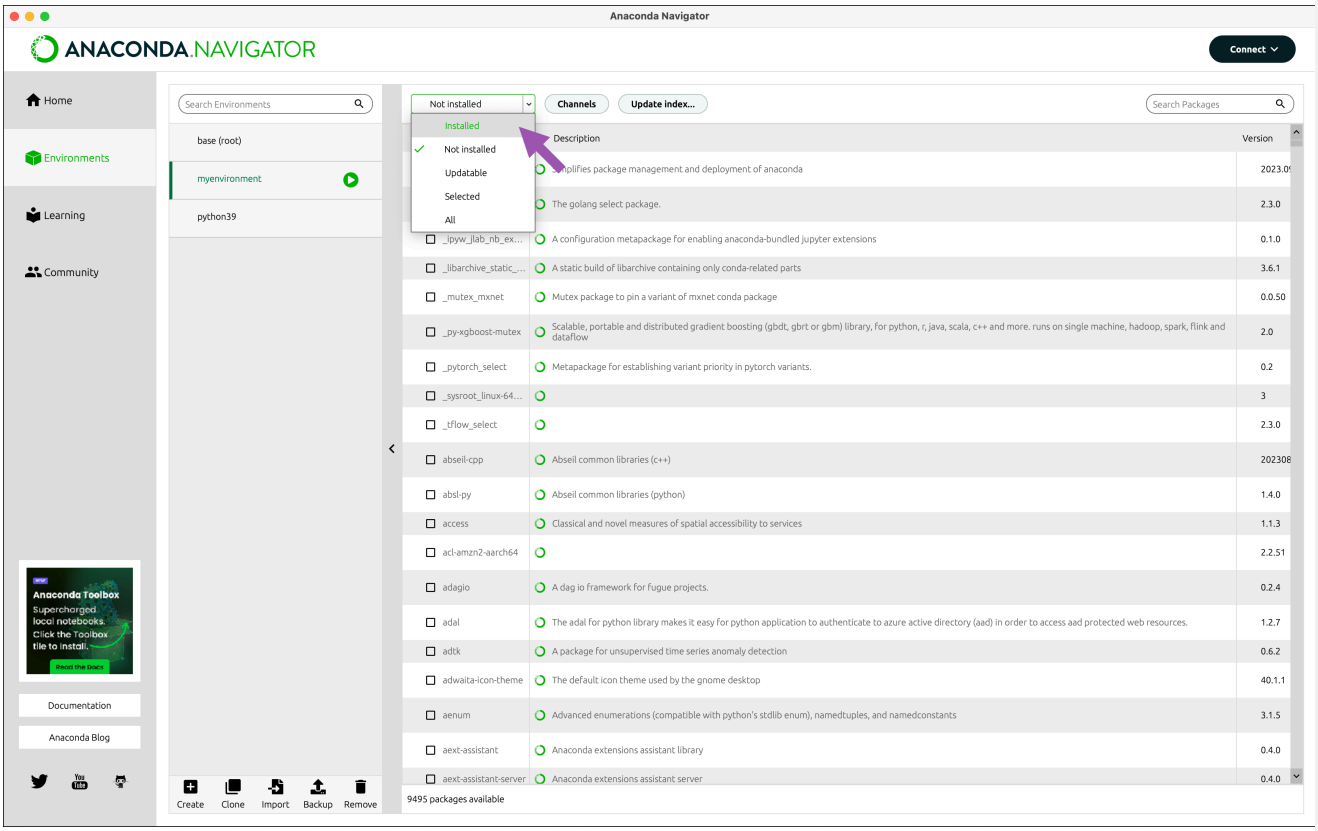
In this example, we will create a new environment named `python39` that contains Python 3.9.18:

1. On the **Environments** page, click **Create**.
2. In the **Environment name** field, type the descriptive name “python39”.
3. Select *3.9.18* from the **Python** dropdown.
4. Click **Create**.

Changing the Python version in an existing environment

In this example, we will change the Python version of `myenvironment` to Python 3.12.0:

1. On the **Environments** page, select `myenvironment` (which we created in the Managing environments section above) from the environment list to activate it.
2. Select *Installed* from the package filter dropdown.



3. Select the Python package checkbox and hover over *Mark for specific version installation*.
4. Select *3.12.0* from the dropdown that appears.
5. Click **Apply**.
6. A dialog listing all the packages that will be modified appears. Click **Apply**.

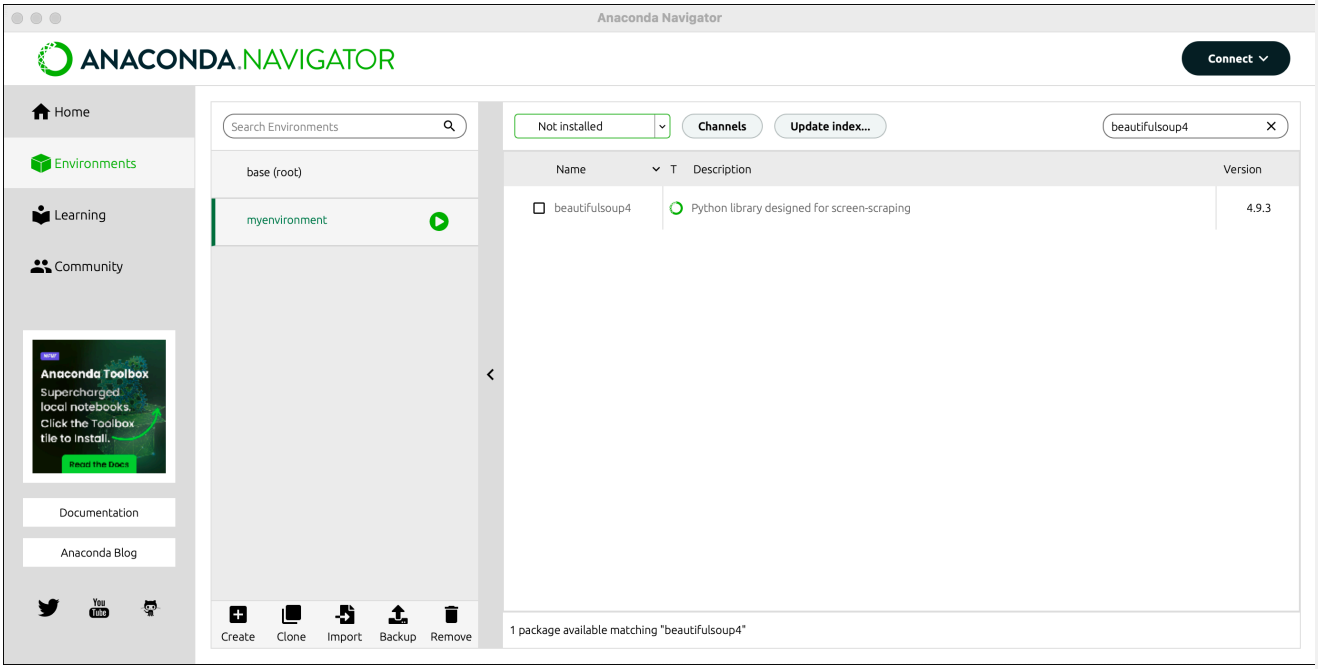
Managing packages

On the **Environments** page, you can:

- View which packages are in each environment
- Search for and install new packages
- Upgrade existing packages
- Delete packages

In this example, we will install a package called `beautifulsoup4` into `myenvironment`.

1. On the **Environments** page, select `myenvironment` from the environment list to activate it.
2. Open the package filter dropdown and select *Not installed* to view a list of packages that are available to install from the channels you are connected to.
3. Type “beautifulsoup4” into the **Search packages** box.



4. Select the checkbox beside `beautifulsoup4` in the package list.
5. Click **Apply**.
6. A dialog appears that lists all packages that will be installed. Click **Apply**.
7. Open the package filter dropdown and select *Installed*.

The newly-installed `beautifulsoup4` package is now displayed in the list of installed programs in the current environment.

For more documentation on packages, see Managing packages in the Tutorials section.

Managing channels

By default, Navigator searches in and installs packages from Anaconda's free public repositories, also known as `defaults`. This contains the package repositories `main`, `r`, and `msys2` (Windows only), which are hosted at repo.anaconda.com/pkgs/. There are many more package channels available, hosted on Anaconda.org, in Anaconda Cloud, or in an on-