Installing Peter Corke's Robotics Toolbox (Python)

August 2025

Why do I need this guide?

Peter Corke's `roboticstoolbox-python` package is only supported up to **Python 3.11**. If you have installed Python recently, you most likely have a newer version. The Robotics Toolbox does not yet support versions newer than 3.11. If you try to install it directly in a newer version, the installation will fail or you may run into strange errors later.

To solve this, we use **Conda**. Conda allows us to create a separate "environment" where we can control exactly which Python version and packages are used. This way we can make sure that we are always running Python 3.11 together with the toolbox, without interfering with the rest of your computer.

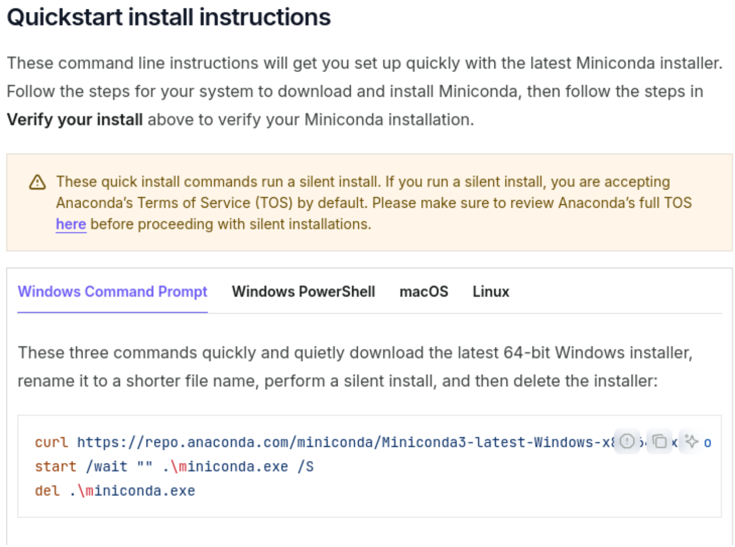
Even when using Conda there can be some small issues the first time you set it up, but by following these instructions step by step, Conda will handle most of the installation details for you and make sure everything works smoothly.

Step-by-step instructions

- **Step 1: Install Miniconda**

  Go to the official Miniconda page: `https://www.anaconda.com/docs/getting-started/miniconda/install#quickstart-install-instructions`

  You should see a page that looks like in Figure 1:



Quickstart install instructions

These command line instructions will get you set up quickly with the latest Miniconda installer. Follow the steps for your system to download and install Miniconda, then follow the steps in **Verify your install** above to verify your Miniconda installation.

⚠ These quick install commands run a silent install. If you run a silent install, you are accepting Anaconda's Terms of Service (TOS) by default. Please make sure to review Anaconda's full TOS here before proceeding with silent installations.

These three commands quickly and quietly download the latest 64-bit Windows installer, rename it to a shorter file name, perform a silent install, and then delete the installer:

```
curl https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe -o
start /wait "" .\miniconda.exe /S
del .\miniconda.exe
```

Figure 1: The view you should see when following the link above.

Now do the following:

1. Choose the tab corresponding to you setup (Windows Command Prompt, Windows PowerShell, macOS, or Linux).
2. Copy the lines of code shown there.
3. Open a terminal
4. Paste the three lines one after the other and press ENTER after each.

This will download the Miniconda installer, run it automatically ("silent install"), and then delete the installer file.

> **Note for Windows Users**
>
> After installing Miniconda, **close your terminal**. On Windows, search for and open **Anaconda Prompt** and use it for the rest of this guide. This ensures Conda is available and on your PATH.

- **Step 2: Create a new Python environment** In your terminal, type the following line and press ENTER:

```
conda create -n rtb -c conda-forge python=3.11 roboticstoolbox-python -y
```

This command does the following:

  - `conda create` = makes a new environment.
  - `-n rtb` = names the environment "rtb".
  - `-c conda-forge` = tells conda to download packages from conda-forge.
  - `python=3.11` = installs Python version 3.11.
  - `roboticstoolbox-python` = installs Peter Corke's Robotics Toolbox.
  - `-y` = automatically says YES to prompts.

> **Tip**
>
> When copy-pasting commands into any terminal, make sure they do **not** break across lines or include stray spaces. If a command fails unexpectedly, double-check for extra spaces or line breaks and try again.

- **Step 3: If you get errors** Sometimes the first time you run this command, it asks you to accept Terms of Service. If it does, read the message and copy-paste the command it gives you to accept. After that, run the same command again:

```
conda create -n rtb -c conda-forge python=3.11 roboticstoolbox-python -y
```

- **Step 4: Activate the environment** Once the installation finishes, you need to **activate** it every time before using it:

```
conda activate rtb
```

If this works, your terminal prompt will now start with `(rtb)` and running

```
python --version
```

should show `Python 3.11.xx`. If you don't see this, the environment is not active.

> **Note for Windows Users**
>
> If you plan to use VS Code, it can help to start it **directly from the Anaconda Prompt** (with `(rtb)` active) by running:
>
> ```
> code .
> ```
>
> This ensures VS Code inherits the correct Conda environment.

At this point the toolbox will work when running Python directly from the terminal. However, most people will want to use an IDE, so next we explain how to use this Python version inside VS Code.

- **Step 5: Set up VS Code**

  1. Open Visual Studio Code.
  2. Make a new empty file and save it as `test.py`.
  3. In the file, type:

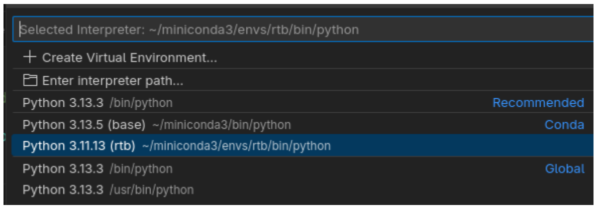
     ```
     print(1+1)
     ```

     and run the file (press the green triangle at the top right, or right-click → "Run Python File") just to check that things are working.
  4. Now, **look in the bottom right corner of VS Code**. It should show something like: `Python 3.11.xx ('rtb')`

     If it does **not** say `('rtb')` after the Python version, you are in the wrong Python environment!
  5. To fix this:

     (a) Click on the Python version in the bottom-right corner.

     (b) A list of environments will appear, as in Figure 2.



Selected Interpreter: ~/miniconda3/envs/rtb/bin/python

+ Create Virtual Environment...

Enter interpreter path...

Python 3.13.3 /bin/python Recommended

Python 3.13.5 (base) ~/miniconda3/bin/python Conda

Python 3.11.13 (rtb) ~/miniconda3/envs/rtb/bin/python

Python 3.13.3 /bin/python Global

Python 3.13.3 /usr/bin/python

Figure 2: A list of the different Python environments you have installed.

     (c) Pick the one called `rtb`.
  6. Run the file again. Now VS Code is using the correct environment.

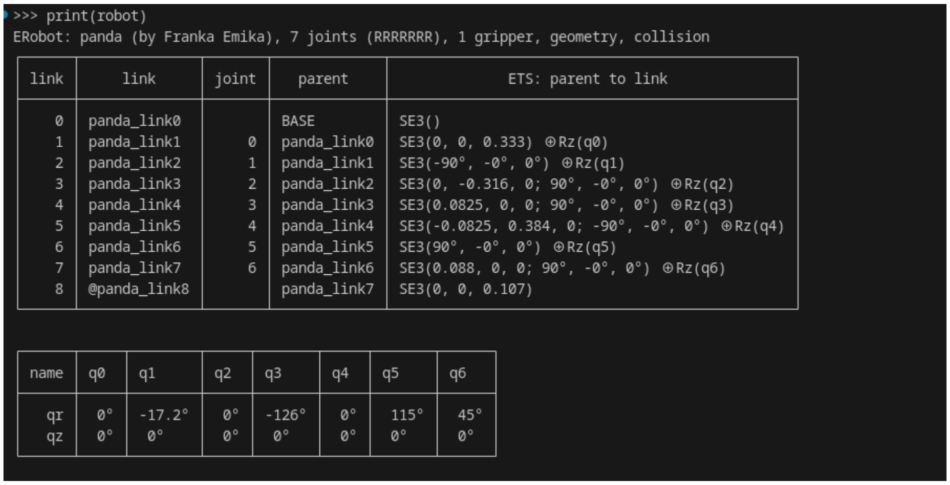
- **Step 6: Test the Robotics Toolbox** In your Python file, type:

```
import roboticstoolbox as rtb
robot = rtb.models.Panda()
print(robot)
```

> **Tip**
>
> The first time you run `import roboticstoolbox as rtb` it can take quite a while. This is normal on first import while packages and caches are prepared. Subsequent runs are faster.

- **Step 7: Check the output** If it worked, you should see something like in Figure 3:

- **Step 8: Celebrate** If you see the above, congratulations — it is working!



```
>>> print(robot)
ERobot: panda (by Franka Emika), 7 joints (RRRRRRR), 1 gripper, geometry, collision
┌──────┬──────────────┬───────┬─────────────┬────────────────────────────────────────────────┐
│ link │     link     │ joint │   parent    │              ETS: parent to link               │
├──────┼──────────────┼───────┼─────────────┼────────────────────────────────────────────────┤
│    0 │ panda_link0  │       │ BASE        │ SE3()                                          │
│    1 │ panda_link1  │     0 │ panda_link0 │ SE3(0, 0, 0.333) ⊕ Rz(q0)                      │
│    2 │ panda_link2  │     1 │ panda_link1 │ SE3(-90°, -0°, 0°) ⊕ Rz(q1)                    │
│    3 │ panda_link3  │     2 │ panda_link2 │ SE3(0, -0.316, 0; 90°, -0°, 0°) ⊕ Rz(q2)       │
│    4 │ panda_link4  │     3 │ panda_link3 │ SE3(0.0825, 0, 0; 90°, -0°, 0°) ⊕ Rz(q3)       │
│    5 │ panda_link5  │     4 │ panda_link4 │ SE3(-0.0825, 0.384, 0; -90°, -0°, 0°) ⊕ Rz(q4) │
│    6 │ panda_link6  │     5 │ panda_link5 │ SE3(90°, -0°, 0°) ⊕ Rz(q5)                     │
│    7 │ panda_link7  │     6 │ panda_link6 │ SE3(0.088, 0, 0; 90°, -0°, 0°) ⊕ Rz(q6)        │
│    8 │ @panda_link8 │       │ panda_link7 │ SE3(0, 0, 0.107)                               │
└──────┴──────────────┴───────┴─────────────┴────────────────────────────────────────────────┘

┌──────┬─────┬────────┬─────┬───────┬─────┬───────┬──────┐
│ name │ q0  │ q1     │ q2  │ q3    │ q4  │ q5    │ q6   │
├──────┼─────┼────────┼─────┼───────┼─────┼───────┼──────┤
│   qr │  0° │ -17.2° │  0° │ -126° │  0° │  115° │  45° │
│   qz │  0° │  0°    │  0° │  0°   │  0° │ 0°    │ 0°   │
└──────┴─────┴────────┴─────┴───────┴─────┴───────┴──────┘
```

Figure 3: The expected output!