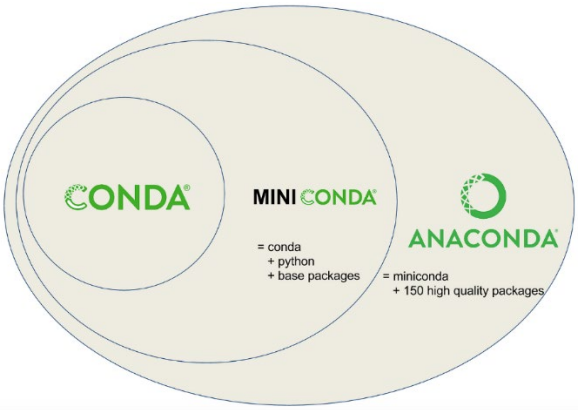
Python Setup - Conda Package Manager and Virtual Environment

The course uses Python as our language. To make software library installation and management easier, the course also uses conda. Conda is a package and environment manager developed by Anaconda, an open-source distribution platform that supports Python. It allows quick installation of packages and their dependencies.

You'll be setting up a conda environment for this course to ensure that you use the correct package versions without interfering with packages being used by your other classes or your personal projects.

If you don't already have a Conda installation on your device, please follow the instruction below to install Miniconda, a lightweight version of Anaconda.



If you already have conda installed on your device, you can skip Part 1: Conda Setup section and move directly to Part 2: Environment Setup .

We use Python 3.9 for this class. If you choose to use another version of Python, then you do so at your own risk.

Part 1: Conda Setup

Please make sure to follow the steps carefully. *Be patient when files are downloading/installing and do not exit/close terminal while it is loading. This may mess up the installation!*

On Windows

Download Miniconda

1. Download the Miniconda installer for Windows using the following link: https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Windows-x86_64.exe

2. Click on the file and agree to the license terms.
3. Choose the `Just Me` option when the installer asks for the users that it should install for. Click on `Next` to proceed.
4. Leave the destination folder unchanged. This should be `C:\Users\{your-username}\miniconda3` by default. Click on `Next` to proceed.
5. Leave both of the advanced options unchecked and click on `Install`.
6. Miniconda should now finish installation automatically.

Configure with Git Bash

We will also be using Bash scripts to run our programs from the command line. You should use Git Bash, an emulator that acts as a typical Bash command prompt that you might find on Linux.

By default, running `conda` in Git Bash will give a "command not found" error, even though conda is installed. Follow these steps to give Git Bash access to your conda installation.

1. Open Git Bash.
2. Verify that you are in the correct directory by typing the following command:

   ```
   pwd
   ```

   This should print the result `/c/Users/{your-username}`. If not, navigate to this directory using the following command:

   ```
   cd /c/Users/{your-username}
   ```

3. Enter the command:

   ```
   echo ". '${PWD}'/miniforge3/etc/profile.d/conda.sh" >> ~/.bashrc
   ```

4. Restart Git Bash. *Git Bash may give warning that it is creating a new file for you. This is ok.*
5. We will now update conda to make sure that you have the most recent version. To do this, run the following command:

   ```
   conda update conda -y
   ```

On Linux and WSL

1. Open the command-line shell.
2. Download the Miniconda installer from the Miniconda website by running the following command in the terminal:

   ```
   curl -O -L https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh
   ```

3. Install Miniconda by running its install script using the bash command.

```
bash Miniforge3-Linux-x86_64.sh -u
```

4. Accept the license agreement (scroll with ENTER) and use the default install location. When the installer asks if you want to initialize conda, choose YES.
5. Init the conda install in bash with this command.

```
miniforge3/condabin/conda init
```

6. Run the following command for the conda install to take effect

```
source ~/.bashrc
```

7. Update conda to make sure that you have the most recent version. To do this, run the following command:

```
conda update conda -y
```

8. (Optional) If you prefer Terminal not to start with conda active by default, run the following command.

```
conda config --set auto_activate_base false
```

Verifying Conda Installation

You should now be able to use the conda commands. Enter `conda --version` in your terminal. If the installation was successful, this should return `conda 23.11.0` or another version of conda (ex: `conda XX.XX.X`).

On MacOS (M1 or M2 macs)

1. Open a terminal window.
2. Download the Miniconda installer using the following command:

```
curl -O -L https://github.com/conda-
forge/miniforge/releases/latest/download/Miniforge3-MacOSX-arm64.sh
```

3. Install Miniconda by running its install script using the bash command.

```
bash Miniforge3-MacOSX-arm64.sh
```

4. Agree to the prompts and leave the default conda settings alone. In order to do this:
    1. Hit `enter` to agree to the installation.
    2. Press `q` to exit the license agreement.
    3. Type `yes` and hit `enter` to agree to the license agreement.
    4. Hit `enter` to agree to the default installation location.
    5. Type `yes` and hit `enter` to agree to the use of conda. Do not quit or close terminal until it shows the prompt on the next line so that you can type commands.

5. Quit and reopen terminal to apply the changes.
6. Update conda to make sure that you have the most recent version. To do this, run the following command:

```
conda update conda -y
```

7. (Optional) If you prefer terminal not to start with conda active by default, run the following command.

```
conda config --set auto_activate_base false
```

On MacOS (excluding M1 and M2 macs)

1. Open a new terminal window.
2. Download the Miniconda installer by running the following command in the terminal:

```
curl -O -L https://github.com/conda-
forge/miniforge/releases/latest/download/Miniforge3-MacOSX-x86_64.sh
```

3. Run the installation script using this bash command:

```
bash Miniforge3-MacOSX-x86_64.sh
```

4. Agree to the prompts and leave the default conda settings alone. In order to do this:
    1. Hit `enter` to agree to the installation.
    2. Press `q` to exit the license agreement.
    3. Type `yes` and hit `enter` to agree to the license agreement.
    4. Hit `enter` to agree to the default installation location.
    5. Type `yes` and hit `enter` to agree to the use of conda. Do not quit or close terminal until it shows the prompt on the next line so that you can type commands.
5. Close and reopen terminal to apply the changes.
6. Update conda to make sure that you have the most recent version. To do this, run the following command:

```
conda update conda -y
```

(Optional) If you prefer Terminal not to start with conda active by default, run the following command.

```
conda config --set auto_activate_base false
```

Part 2: CS1430 Environment Setup

After you've finished the conda installation, you can create the conda environment for this course. All the code for this class should be executed inside this conda environment.

On Windows and Linux

1. Open Terminal (on Linux) or Git Bash (on Windows).
2. Download the conda environment configuration file from the course website using the curl command:

```
curl -O https://raw.githubusercontent.com/ariarobotics/cv/main/cv-environment.yml
```

3. If conda is already active, the terminal prompt has `(base)` at the start of command line. Deactivate the current environment by running the following command:

```
conda deactivate
```

4. We will create a new conda environment by running the following command. This environment will contain all of the dependencies we need for the class.

```
conda env create -f cv-environment.yml -n cv
```

5. Now, you can activate the virtual environment by running:

```
conda activate cv
```

6. Make sure the virtual environment is active whenever you run the code for this course. In other words, the prompt in terminal should always have `(cv)` at the start of the line when you are working on projects.

On MacOS

1. If conda is already active, the terminal prompt will have `(base)` at the start of command line. Deactivate the current environment by running the following command:

```
conda deactivate
```

2. Download the conda environment configuration file from the course website using this curl command.

```
curl -O https://raw.githubusercontent.com/ariarobotics/cv/main/cv-mac-
environment.yml
```

3. We will create a new conda environment by running the following command. This environment will contain all of the dependencies we need for the class.

```
conda env create -f cv-mac-environment.yml -n cv
```

4. Now, you can activate the virtual environment by running:

```
conda activate cv
```

5. Make sure the virtual environment is active whenever you run the code for this course. In other words, the prompt in terminal should always have `(cv)` at the start of the line when you are working on projects.

Made it this far?

Learn how to set up your VSCode environment!