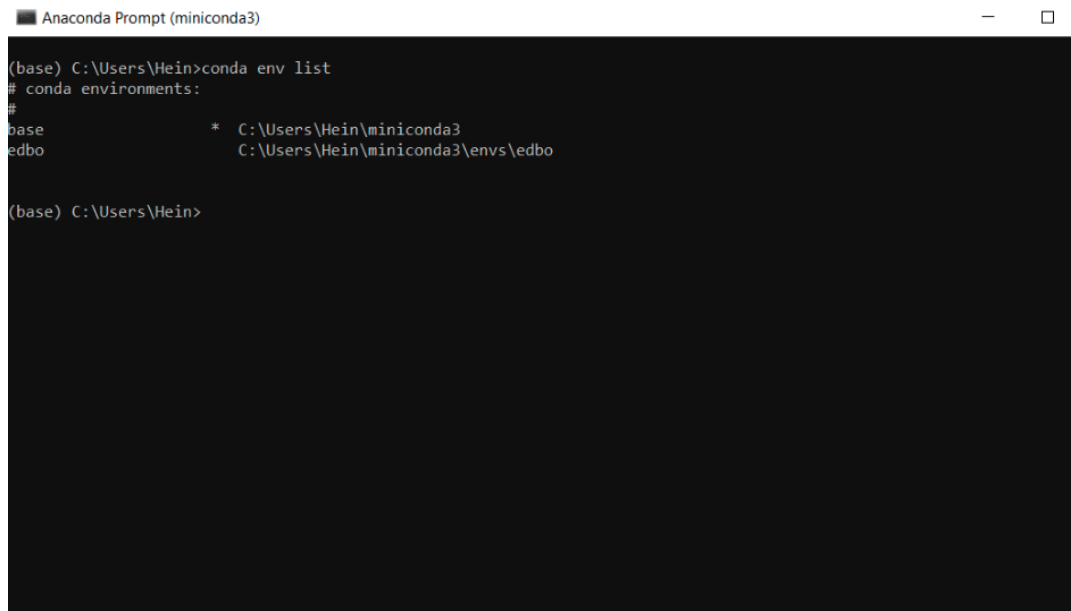


Install EDBO specific Python environment via Miniconda

There are multiple ways to install Python on Windows PCs. One of them is to do so through a Python distribution. A Python distribution can be viewed as a bundle of core Python with other added features such as pre-installed python packages, package managers and other stuff. The Python distribution used here is called Miniconda, which is a lighter version of Anaconda, and it offers all the required ingredients to start using Python on Windows.

1. **Installing Miniconda:** Visit <https://docs.conda.io/en/latest/miniconda.html> and download the Miniconda installer (such as [Miniconda3 Windows 64-bit](#)). After that, start the installer and follow each step to complete installation (leaving all settings to default). To test the installation, click on the search button (located on the bottom left), type and run Anaconda Prompt (miniconda3). Once the prompt is opened, type in '**conda list**'. A list of installed packages should pop up.
2. **Creating EDBO specific Python environment:** It is generally a good idea to create separate Python environments for different projects. To do so, in the Anaconda Prompt, type in '**conda create --name edbo python=3.7.5**' (edbo can be replaced with other names for identification of this environment). It takes a couple minutes to set up the environment. To test if the new edbo environment is installed correctly, type '**conda env list**', and a list of installed Python environments should pop up (see figure below).



```
Anaconda Prompt (miniconda3)
(base) C:\Users\Hein>conda env list
# conda environments:
#
base                * C:\Users\Hein\miniconda3
edbo                 C:\Users\Hein\miniconda3\envs\edbo

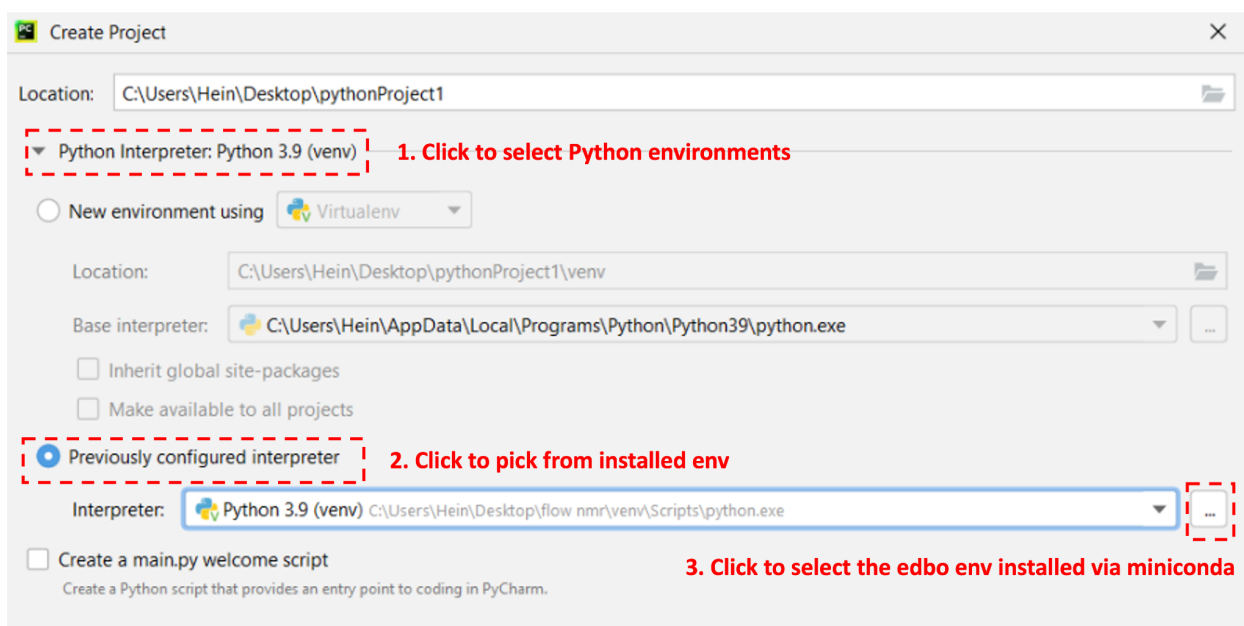
(base) C:\Users\Hein>
```

Install Python Integrated Development Environment (IDE) PyCharm

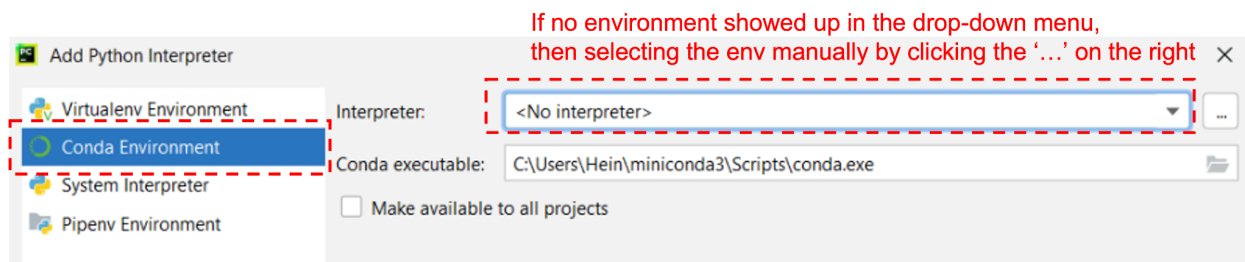
It is possible to work with the previously installed Python environment directly in the miniconda command prompt, but most of the time, one would choose to develop their Python projects in a Integrated Development Environment (IDE) that makes coding and project management much easier. The IDE we will use here is called PyCharm. To download PyCharm, visit <https://www.jetbrains.com/pycharm/download/#section=windows> and pick the community version. After that, simply follow instructions for installation.

Set up project in PyCharm

After installing PyCharm, we want to run the code in the correct Python environment (the one we just created that's called edbo). To do so, open PyCharm, and click New Project. In the popped up window, first choose the location where you want this project to be. Then click on the Python interpreter, next select Previously configured interpreter, and click on the '...' on the right (see figure below).



In the popped up window, click on Conda Environment, and in the interpreter setting, bring up the drop down menu and select the previously installed edbo environment. If no env is shown, then we need to manually select it by clicking the '...' on the right (see figure below).



The path to the desired env should be: `...\miniconda3\python.exe` where '`...`' is the path to miniconda3. After selecting the environment, click create, and a new PyCharm window should appear. The left panel shows files contained in the project folder, and the right panel is where we code. To check whether the correct environment is selected, go to the bottom right, and you should see 'Python 3.7 (base)' listed there.

Run flow optimization codes in PyCharm

First, unzip the files that contain all the codes. Then transfer the two folders contained in the resulting folder (edbo and utils) into the PyCharm projected fold that just got created. The python script that runs the whole closed-loop optimization and resulting data will be stored in the edbo folder, and all helper scripts will be stored in the util folder.

Currently, the main python script cannot be executed, simply because we haven't installed the required packages. The main package we need here is of course 'edbo'. There are a couple ways one can install python packages in PyCharm, But I have noticed, especially for the package edbo, installing stuff in Pycharm can lead to error messages. So we are going to install the packages using the miniconda command prompt.

First, open up the miniconda command prompt as we did before. Currently, we are in the default python environment, but we want to switch to the edbo environment. To do so, type in 'conda activate edbo' and you should see the name inside the bracket change to edbo.

```
(base) C:\Users\Hein>conda activate edbo
```

Once we are in the edbo python environment, we need to install a series of packages before installing the actual edbo package. type and run the following command one after another.

- **conda install -c rdkit rdkit**
- **conda install -c rdkit -c mordred-descriptor mordred**
- **conda install -c pytorch pytorch=1.3.1**
- **pip install edbo**

After installing edbo, we still need two other major parts to be able to run the closed-loop optimization code. The first major part is the vapourtec API code. Those codes cannot be installed the normal way, but we need to manually copy the folder 'rseriesopc' into the following folder: '`...\miniconda3\env\edbo\Lib`'. The second major part is all other packages that are

required by the helper codes. To install them all at once, return to the command prompt (make sure we are in the edbo Python env), then type in 'pip install copy datetime watchdog'.

After that, the main optimization script should be ready to run. To run it, double click this python file in the left panel so it shows up on the right. Then right click its tab and select 'Run selfOptimizeFlow'. If an error message pops up regarding missing certain packages, simply pip install that package like we did before in the command prompt.

Setting up python env offline

Those previous steps might not work on company computers, so in the case where we cannot download stuff using miniconda from the internet, the python env can still be set up in offline mode. To do so, in a pc that has full internet access, install miniconda as detailed in the first section. After that, open up the miniconda command prompt, and type '**pip download -r requirements.txt**', which will download all required packages in the txt file without installing them. Zip those files with the requirements.txt and transfer them to the PC that we want to run EDBO. On that PC, open up the miniconda command prompt, and instead of creating a project specific env (which requires internet access), we will work with the base env. Navigate to the folder that we just unzipped (by using 'cd' command, lookup online if not sure how to do i), then type '**pip install --no-index --find-links ./ -r requirement.txt**'. This will take a while to finish, if after waiting for a long period of time, the command prompt still shows it is installing, try hit 'enter' (the installation process might be finished already, just didn't show that on the screen).

After that, follow the **Setup project in the PyCharm** section to create projects in pycharm.