Conda Setup for XCS Courses

What is Conda?

Conda is an open source package management system and environment management system that runs on Windows, macOS, and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

Conda as a package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because conda is also an environment manager. With just a few commands, you can set up a totally separate environment to run that different version of Python, while continuing to run your usual version of Python in your normal environment.

1 Installation

The fastest way to obtain conda is to install Miniconda, a mini version of Anaconda that includes only conda and its dependencies. If you prefer to have conda plus over 7,500 open-source packages, install Anaconda.

Here are some of the system requirements necessary for installation:

- 32- or 64-bit computer

- For Miniconda - 400 MB disk space

- For Anaconda - Minimum 3GB disk space to download and install

- Windows, macOS, or Linux

For download specific instructions, please refer to this link.

For Windows: Go to your search menu on Windows after installing Miniconda, and locate the Anaconda Command Prompt to run the commands below.
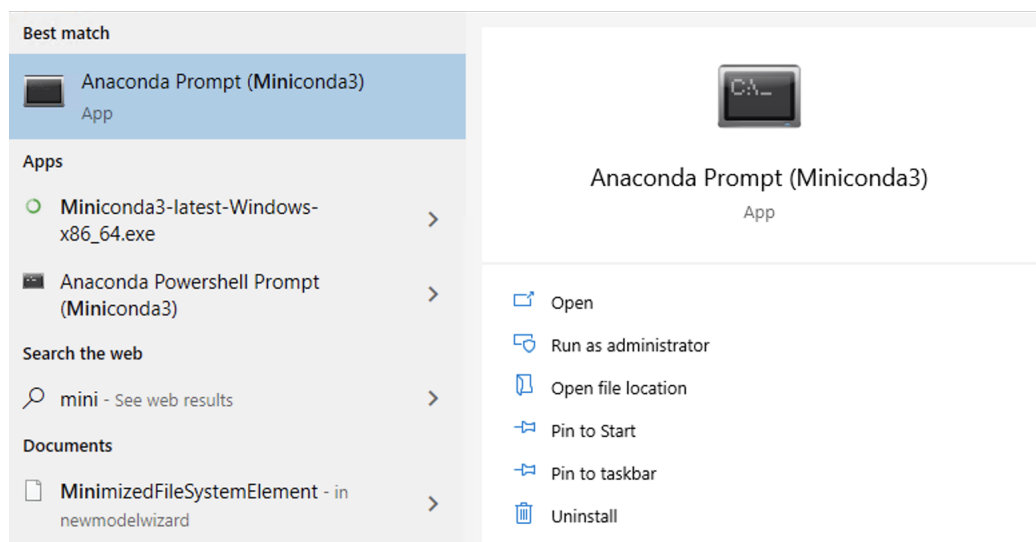
To verify the Conda download, please run:

```
conda info
```

In case you have previously installed Conda or Anaconda, be sure to update them accordingly before moving onto step 2:

```
#Update to lastest version of Conda
conda update -n base conda
#Update all packages to the latest version of Anaconda
conda update anaconda
```

2 Creating and activating environment from YAML file

After you have installed Conda, we have provided a file called **environment.yml**. This YAML file will help you create a virtual environment with

a specified name used for activation and install necessary dependencies for the coding assignments. Here is what an example environment.yml file looks like.

```
name: XCS229ii
channels:
  - defaults
dependencies:
  - python=3.7
  - notebook
  - numpy>=1.16
  - matplotlib>=3.1
  - scipy
```

Note: Although there are multiple ways of completing the homework assignments, the libraries we provide you within the YAML file should be the only ones you reference in the completion of the coding assignments. **For the remainder of this setup we will be using *your-env-name* to specify the name of your environment. You can find the name of your virutal environment by looking for the name parameter within the environment.yml file.**

To create your environment from the YAML file please run:

```
conda env create --file environment.yml
```

Now to specify that the environment was installed correctly, please run the code below and you should see an environment name that matches with the name parameter in the environment.yml file:

```
conda env list
```

Next, you will activate the created environment. To do so, you will need to to call the following command:

```
conda activate <your-env-name>
```

Now in this environment, you will have all the relevant dependencies needed to complete the assignments. If you want to deactivate your environment please run:

```
conda deactivate
```

3   Converting Environment to Jupyter Notebook

When completing the assignments, it may be helpful to use a Jupyter Notebook as a scratchpad to help you debug or formulate potential solutions. Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. The first step to creating your Jupyter Notebook for the assignments is to activate your environment as follows:

```
conda activate <your-env-name>
```

Next, in the active environment type:

```
#install ipykernel
conda install -c anaconda ipykernel
#install the new kernel
ipython kernel install --user --name=<your-env-name>
```

And last but not least, ensure to run the following command to launch your Jupyter Notebook in the active environment

```
jupyter notebook
```

That command will take you to your Jupyter Notebook directory. When you hover over the **New** button in the top right hand corner, you should see you environment present as an option.
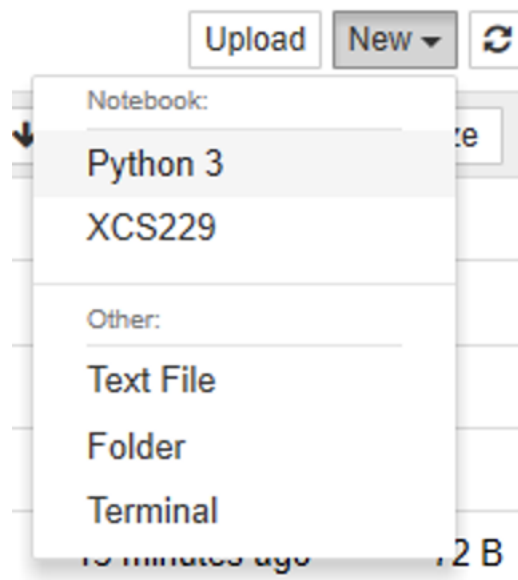
Go ahead and select it and it will create an empty Python Jupyter file where you can copy and paste assignment code or use it to formulate your thoughts.

In the event you would like to remove your environment from Jupyter, simply run the following code:

```
jupyter kernelspec uninstall <your-env-name>
```

You are now equipped with all the tools to complete the coding assignments. For a list of the most useful Conda functions, please refer to this document.

4   Updating Existing Conda Environment

If you plan on reutilizing the an existing Conda environment from a previous SCPD-AI course, we strongly recommend all students update their Conda environment upon starting or replace it with the new Conda environment as mentioned above. Below are steps to update your existing Conda environment in preparation for this course.

```
conda activate <your-env-name>
conda env update --file environment.yml
```