

# Preparing for Materials Informatics

---

This course will require use of the python programming language. While prior programming experience is a pre-requisite for this course, we understand some people's python skills might be a little shaky due to a gap in usage. This document will help you re-install python (if you don't have it) and set you up to hit the ground running when the course starts.

## Package Managers

In this class we will take advantage of several python libraries that are designed for machine learning and materials science. While these libraries can be manually installed on your machine, it is common practice to use a package manager to simplify the install process and help manage package [dependencies](#). While many exist, we recommend using Miniconda, which is a stripped down version of the Anaconda package manager you might be familiar with.

What's the difference? Anaconda includes a number of data science libraries by default, which, while convenient, result in bloated virtual environments, slow dependency resolutions, and poor programming practice. If you already have Anaconda installed on your machine, you are welcome to continue using it and can safely skip the remainder of this section; however, if you don't have a package manager installed or are looking for a fresh start, Miniconda is the way to go.

## Installing Miniconda

Go to the [Miniconda Install Page](#) and follow the instructions (you can safely ignore the *optional* step 2) specific to your machine type (windows vs. mac). If everything goes smoothly, you will be ready to move onto the next step. If everything does not go smoothly, you can reach out to us and we can try to troubleshoot the issue for you.

## Virtual Environments

To access Miniconda you can either open your terminal application if you are on Mac or the Anaconda Prompt application if you are on windows. Regardless, you should see a command line with some information about your machine, but most importantly you should see `(base)` somewhere on the line. This tells you that you are in your base conda environment. This is the foundation from which we will build other conda environments.

But what is a conda environment? An environment is like a separate, isolated room in a large house, where you can install and use different tools and furniture (software and libraries) without affecting other rooms (environments). Our kitchen is separate from our bathroom so that we can cook without interfering with our ability to use the bathroom. In the world of programming, sometimes you need different versions of software and libraries for different projects. A conda environment allows you to set up these separate “rooms” on your computer, so you can have the specific versions of software you need for a project, without affecting other projects that might need different versions. This way, you can easily switch between projects (or rooms) without any conflicts or issues.

You can think of your `(base)` environment as the essential components of a room (floor, walls, doorway, etc.). When we create a new conda environment, we start with the bare essentials included in the `(base)` environment and add other libraries as needed. As such, you should **never modify the `(base)` environment!** Adding a library to the `(base)` environment would be the equivalent of mandating that a bed be included by default in every single room of a house. Instead, when embarking on a new project, we create a new conda environment and install necessary packages there.

## Creating a Conda Environment For This Class

To create a generic conda environment you can type the following command into your terminal or Anaconda prompt:

```
conda create --name my_first_environment python=3.10
```

To install a package into that environment you need to “activate” it, you can do that through:

```
conda activate my_first_environment
```

If the activation is successful, you should see `(my_first_environment)` (or whatever you named it) appear in your command line.

You can then install packages to this environment using the following pattern:

```
conda install [package name]
```

In the interest of avoiding inconsistencies, we ask that you install the following virtual environment ahead of class:

```
conda create --name MatInformatics python=3.10  
conda activate MatInformatics  
conda install pandas numpy matplotlib scikit-learn statsmodels
```

Once that has completed, you should be ready to go!

## Installing VS Code

An IDE simplifies the process of editing and running python code on your computer. While there are many to choose from (Spyder, PyCharm, Atom), we recommend [VS Code](#). VS Code is fast, well maintained, and offers a number of nice features and additions that will make your machine learning adventures a lot easier.

Once installed, you can install a number of helpful extensions through the application. We recommend the following as a starting point, but feel free explore more.

- [Python](#)
- [Pylance](#)
- [Rainbow CSV](#)
- [Data Wrangler](#)
- [Jupyter](#)

---

If you followed this correctly, you should be ready to dive into the world of materials informatics.