Hello and thank you for participating in Rice BMES's Python workshop series! This will serve as a quick tutorial for installing PyCharm, the IDE (integrated development environment) we will be using throughout the workshops. We chose PyCharm because it easily integrates Python, includes debugging features, allows for project organization, and facilitates package installation, but there are plenty of other IDEs you can use to program in Python.

Getting Started: Installation

Follow the steps in the following videos to install Python and PyCharm (thanks Gerry Jenkins <3)

Important Notes:
- Install Python version **3.8.7**! We chose version 3.8 to reduce chances of incompatibility with out-of-date libraries. If you have an older version of python 3 (or 3.8) you don't need to install a new version (we're just being wary of Python 3.9).
- Install the **Community Edition** of PyCharm
- If at any point you run into issues, feel free to reach out to the workshop instructors/TAs via email (although Google can also be a helpful tool)!

Windows:
Notes (read these before watching YouTube video below)
- You can stop watching the video at ~9:00. If you want to change your system preferences to use the system interpreter by default when you make a new project, go to 9:30. There's a short explanation of virtual environments at the end of this document.
- If you want to use terminal commands (not required), the "python" or "python3" command should work, but in some cases you might need to use "py" or "py3" instead

https://www.youtube.com/watch?v=AUiM1UaRCPc

Mac:
Notes (read these before watching YouTube video below)
- You do NOT need to do the terminal commands shown, but can if you want! (They also work on Windows).
- At around 6:59 in the video, it is optional to make the system interpreter your default. If you don't want to do that, skip to 8:33 and make a new project, click the drop down, and select "System Interpreter". Within the second drop down, look for the version of Python you just installed.

- Follow the instructions to make a project (with a virtual environment as your interpreter, don't modify any settings). Feel free to run any code you'd like!

https://www.youtube.com/watch?v=oyzH4M6X6F4

Note on Virtual Environments:

A virtual environment essentially isolates the project you're working on from the rest of your computer. So, instead of existing "on your computer" with all of the Python packages you have installed, it exists in your project folder, in its own little world without any packages that did not come by default. This can be useful to prevent different versions of packages from affecting different projects you're working on, like if something is out of date. However, our purposes, it's not really going to make a difference. The only thing you'll have to do is re-install packages for a new project if you make it in a virtual environment.