Info on Pip compile: <https://github.com/jazzband/pip-tools>

Instead of installing packages individually, **pip** allows you declare all dependencies in a Requirements.txt file.

To tell your package handler to install all the packages:

py -m pip install -r requirements.txt

\*Note: The “-m” switch allows modules to be located using Python module namespace

\*Background: Package Modules (modules that contains other modules) are identifiable by either their file name (in command line) or their modules name (in Python)

If module file is fed into python interpreter, their \_\_main\_\_.py script runs

If module file is imported( import<module\_name> ), then \_\_init\_\_.py will be executed

* The “-m” tag was introduce so that you could call a module name directly from command line
  + python <filename> <args> and python -m <modulename> <args>

\*Use Cases:

Executing modules from command line without knowing their name

Execute a local package containing relative imports

“pip” is a command line program you can pass argument to.

py -m pip install SomePackage

“pip wheels” – is used to speed up installation of packages

In general you want to keep a list of dependencies your project is using (not install them on by one)

Furthermore, you want to freeze the specific dependencies of the dependencies, so it continues to function exact same way.

To do this you write down the dependencies you need in a “requirements.in” file, and then run pip compile on that file to produce requirements.txt which contains a list of all the dependencies in requirement.in, along with the **transitive dependencies** of those modules