Virtual environment
VENV

VENV is a module of a Python installation. It supports creating lightweight “virtual environments” each with their own independent set of Python packages installed in their site directories.

```
Python3 -m venv </path/to/new/virtual/environment>
Python3 -m venv venv
```

Here the first `venv` refers to the program we are running. The second `venv` is referring to the name I gave my virtual environment. So, instead of a second `venv` you can type in whatever you want. I find the word venv pretty easy to rememeber and to type up, so that's how I usually call my virtual environments.

```
Mac:
to activate
source venv/bin/activate

# to deactivate
Deactivate

Windows:
to activate
.\venv\scripts\activate.bat

# to deactivate
Deactivate
```

Now we can create the project once our environment is activated.

```
    Pip install Django
    Django-admin startproject <name of project> . (include period)
```

```
Python manage.py runserver
```

Again, you do not have to use a virtual environment. In fact, if you’re not experienced with them, I’d advise against it. I’m just showing you because that it how I manage my python packages and dependencies. Since all of your projects will probably be using the same dependencies for the same version of python, it won’t make any difference for you. However, if you continue to develop with Python, virtual environments will be something you probably want to look it.

But again, they are not needed for this class. I mention it because there’s always questions posted on Ed once the semester begins about virtual environments and some questions about them.