**Python SETUP/ Python PROJECTS**

**Setting up your Computer (MAC)**

* Install **HomeBrew** through the terminal.
  + - * Go to the link <https://brew.sh/>
      * Copy and paste the installation route into the terminal
* Run **HomeBrew** in the terminal to check the version and see if it fully installed
  + - * Check Installation / Version

~$ brew -v

* + - * Update

~$ brew update

* Install **PYTHON**
  + - * Install Python 3

~$ brew install python3

* + - * Check the version #

~$ python3 -V

**Setting up your Folder Structure (for Coding Dojo)**

* Make sure your folder for python is setup accordingly:
  + - * python\_stack

my\_environments

\_python

python\_fundamentals

OOP

django

django\_fundamentals

django\_orm

django\_fullstack

**Install DJANGO using Package Manager: pip**

* pip is the package manager to use with Python when installing third-party packages
  + - * + Install Django

~$ pip3 install django

* + - * + Uninstall Django

~$ pip3 uninstall django

Note: This can be used to install various other third-party packages just follow along like above with any other package replacing django with the package your looking to install.

**Store your environments in the “my\_environments” folder**

* Using the terminal navigate into the “my\_environments” folder
  + - * + Once inside the folder we need to store the environments using the command line below.

~$ python3 -m venv py3Env

**Activating the Virtual Environment**

* When activating a virtual environment we need to be in the “my\_environments” folder
  + - * + ~$ source py3Env/bin/activate

Note: you will know this is successful when the command line changes to

(py3Env) $

* To DEACTIVATE the virtual environment simply just type in “deactivate” into the command line.

**Creating a Django Project**

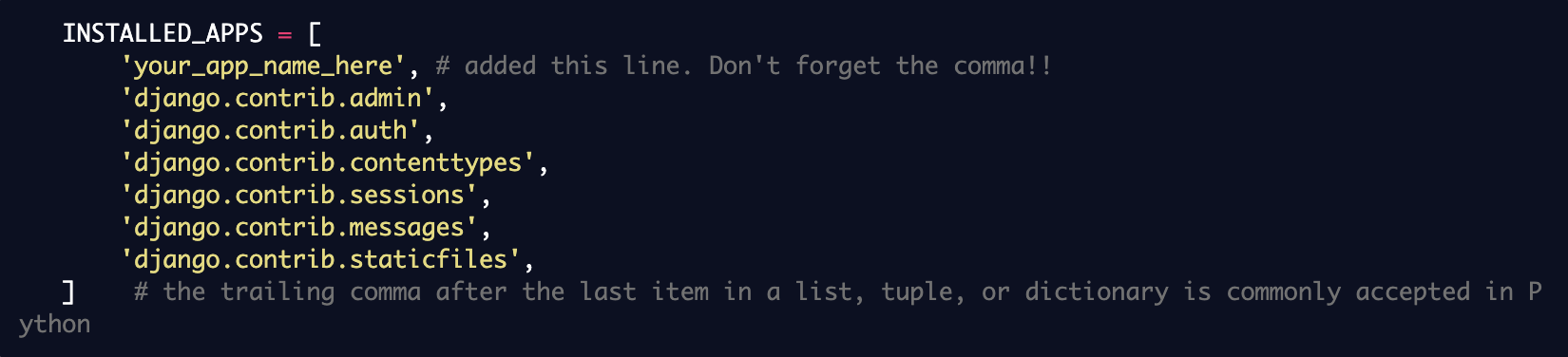
With a virtual environment activated we can create a Django project.

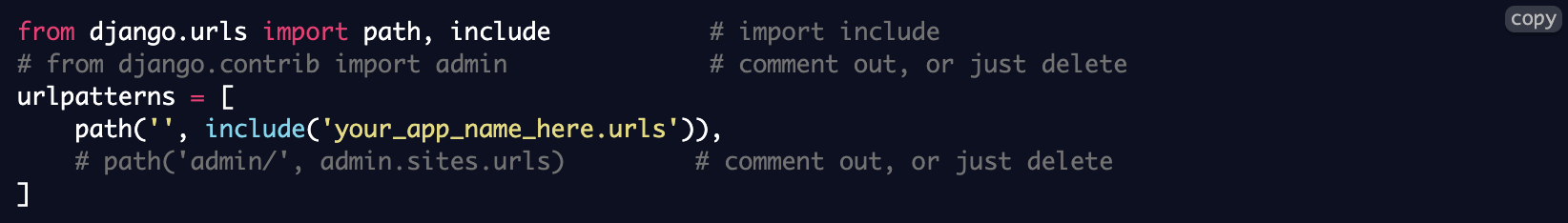
* Use the terminal to navigate to where you want the project to be saved.
  + - * + ~$ django-admin startproject your\_project\_name
* Once that command is completed in the terminal we need to navigate to the new folder that was created to run the new project
  + - * + ~$ cd your\_project\_name
* (OPTIONAL) Now that a new Django project has been created its time to check to see if it is running. Use this command to run the server
  + - * + ~$ python manage.py runserver

Note: Press ctrl-c to stop the server.

* + - * + Now go to your browser and search **localhost:8000**
* The next step is to navigate through the terminal to the your\_project\_name folder. For every app we want to add to our project we need to run this command in the terminal.
  + - * + ~$ python manage.py startapp your\_app\_name

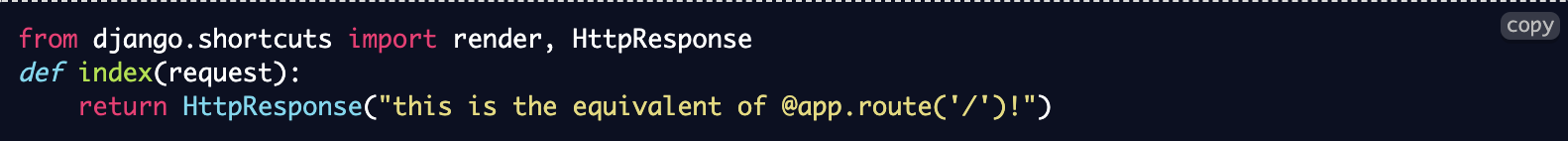
Note: the apps in a project CANNOT have the same name as the project

* Now we need to open our text editor to open the project we created through the terminal. Once that is completed we need to make some changes.
  + - * + Starting with the **settings.py** file in the project folder we need to add the app we created in the step before this.
        + In the urls.py. file we need to create a route “/“ to be associated with a specific function. This is where we add a URL pattern for our new app.

**** Note: Some lines of code need to be commented out or deleted.

* + - * + (Folder Route) **your\_project\_name/your\_project\_name/urls.py**
        + The next step we need is to add a **urls.py** file to the your\_app\_name folder. Once the file is created we need to add the code below.
        + (Folder Router) **your\_project\_name/your\_app\_name/urls.py**
        + Next we need to put a function called index in our apps **views.py** file in the your\_app\_name folder to test everything is working.

Note: Later we will replace with HTML templates.

* + - * + (Folder Route) **your\_project\_name\_here/your\_app\_name/views.py**
      * Last thing we need to do is run our app and check it out at **localhost:8000/**

**Django Routing**