

## 1. Create a new environment

Open your command prompt or terminal and navigate to the directory where you want to create the new environment.

PS: You can run terminal commands using !pip in Python if you are using a Jupyter Notebook or a similar environment. A command with ! in a Jupyter Notebook cell allows you to execute that command as if you were running it in a terminal.

```
python -m venv delta-statistic-python
```

Creating a new environment provides a controlled and isolated space for your projects, since each acts as a separate, self-contained workspace. This isolation helps avoid conflicts between different projects that may require different versions of the same package, allowing you to manage the dependencies of your project more effectively.

To activate the environment, use the following command:

### ATTENTION

You might need to manually change the Kernel

- For Windows:

```
delta-statistic-python\Scripts\activate
```

- For MacOS or Linux:

```
source delta-statistic-python/bin/activate
```

## 2. Install and import Packages that are needed in this new environment

You can now install packages and libraries specific to this environment using "pip". If you only want to run a specific part of the code, you can install it one-by-one, otherwise it's simpler to run in the terminal:

```
pip install -r requirements.txt
```

## 3. Run the web-app in your local machine

In the terminal, check if you're in the correct folder and write the following command:

```
python manage.py migrate
```

The "**migrate**" command is used to apply any pending database migrations and set up the database schema based on your Django models. You typically run this command only once, during the initial setup or whenever there are new migrations to apply. After the first time initiating the web-app, everytime you want to initiate the web-app in your local machine, while in the correct folder, you just need to type the following command in the terminal:

```
python manage.py runserver
```

The "**runserver**" command starts the development server and runs your Django web app locally. You can access your app by visiting "http://localhost:(usually 8000)" in your web browser.

- For more information check Django Documentation:  
<https://docs.djangoproject.com/en/4.2/>