

# VIRTUAL ENVIROMENT

## Зачем нужно виртуальное окружение?

- Предотвращение конфликтов версий
- Легкость воспроизведения и установки(require.txt)
- Работает без прав администратора

## Альтернативы venv

На данный момент существует несколько альтернатив для venv:

- **pipenv** - это pipfile, pip и virtualenv в одном флаконе;
- **pyenv** - простой контроль версий Питона;
- **poetry** - новый менеджер для управления зависимостями;
- **autoenv** - среды на основе каталогов;
- **pew** - инструмент для управления несколькими виртуальными средами, написанными на чистом Python;
- **rez** - интегрированная система конфигурирования, сборки и развертывания пакетов для программного обеспечения.

## Как установить virtualenv

Необходимо, если версия python ниже 3.4 (после уже встроена)

```
pip install virtualenv
```

## Как создать виртуальное окружение

1. Откройте командную строку или терминал в нужной директории.
2. Введите следующую команду:

```
python -m venv <имя_окружения>
```

## Активация виртуального окружения

В командной строке (Windows):

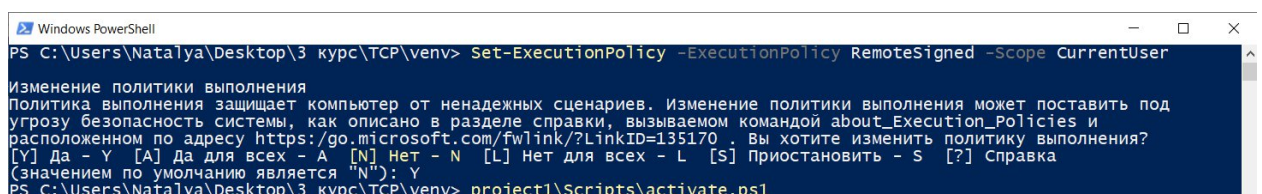
```
<имя_окружения>\Scripts\activate.bat
```

В командной строке (Windows Powershell):

```
<имя_окружения>\Scripts\activate.ps1
```

Если выходит ошибка, то нужно ввести:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```



В терминале (Mac и Linux):

```
$ source myvenv/bin/activate
```

## Деактивация виртуального окружения

По окончании работы над проектом рекомендуется деактивировать его виртуальную среду. Деактивация означает выход из данной виртуальной среды. Без деактивации виртуальной среды весь остальной выполняемый вами Python-код, даже если он находится вне каталога проекта, также будет выполняться внутри этой виртуальной среды.

Введите в терминале команду **deactivate**. (работает одинаково во всех операционных системах)

## Удаление виртуальной среды

Если вы хотите удалить эту виртуальную среду, то сначала деактивируйте ее, а затем удалите ее каталог со всем его содержимым. В Unix-подобных системах и в Windows PowerShell это можно сделать следующим образом:

```
rm -r <имя_окружения>
```

## Что находится внутри виртуальной среды?

Если заглянуть в каталог виртуальной среды, то под операционной системой Windows вы увидите примерно следующее:

```
├─ Include
├─ Lib
├─ site-packages
├─ pyvenv.cfg
└─ Scripts
    ├─ activate
    ├─ activate.bat
    ├─ Activate.ps1
    ├─ deactivate.bat
    ├─ pip3.10.exe
    ├─ pip3.exe
    ├─ pip.exe
    ├─ python.exe
    └─ pythonw.exe
```

- в операционной системе Windows двоичный файл Python копируется в каталог `scripts`
- все устанавливаемые пакеты попадают в каталог `site-packages`
- у нас есть скрипты активации для различных типов оболочек (bash, csh, fish, PowerShell)
- пакетный менеджер `pip` можно запускать посредством команд `pip` и `pip3` и даже `pip3.7`, так как на момент установки виртуальной среды у нас была версия Python 3.7.

## Установка пакетов из файла `requirements.txt`:

1. Создайте файл [requirements.txt](#) и перечислите в нем все необходимые пакеты, каждый на отдельной строке.
2. Активируйте виртуальное окружение.
3. Введите следующую команду:

[pip install -r requirements.txt](#)

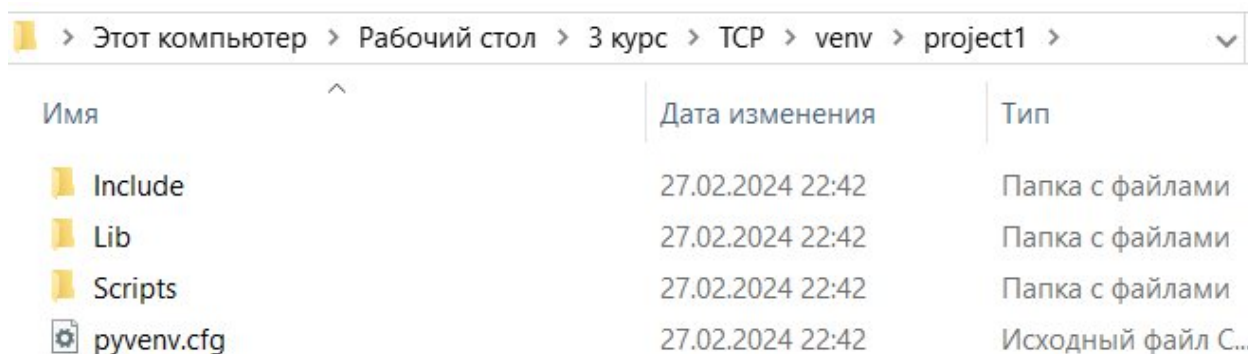
4. Pip автоматически установит все пакеты, указанные в requirements.txt.

## Пример использования venv в Powershell

1. Создание виртуальной среды:

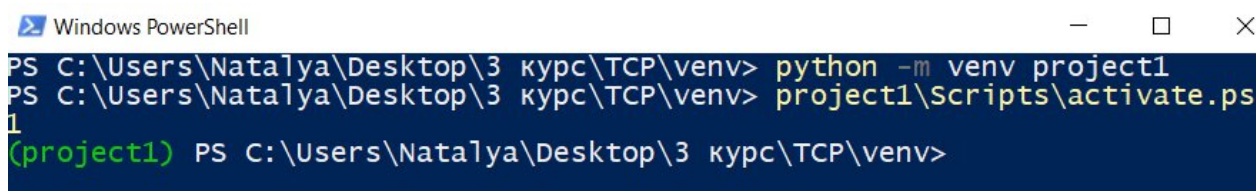


```
Windows PowerShell
PS C:\Users\Natalya\Desktop\3 курс\TCP\venv> python -m venv project1
PS C:\Users\Natalya\Desktop\3 курс\TCP\venv>
```



Имя	Дата изменения	Тип
Include	27.02.2024 22:42	Папка с файлами
Lib	27.02.2024 22:42	Папка с файлами
Scripts	27.02.2024 22:42	Папка с файлами
pyvenv.cfg	27.02.2024 22:42	Исходный файл C...

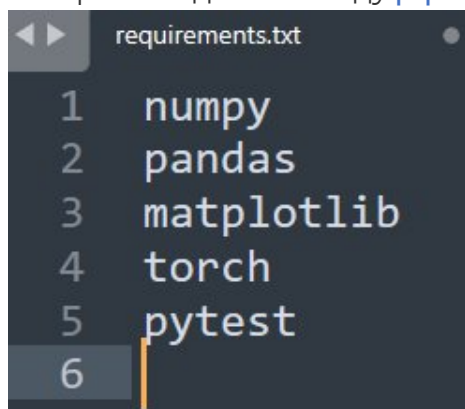
2. В той же папке, в которой создана директория с виртуальной средой активируем среду:



```
Windows PowerShell
PS C:\Users\Natalya\Desktop\3 курс\TCP\venv> python -m venv project1
PS C:\Users\Natalya\Desktop\3 курс\TCP\venv> project1\Scripts\activate.ps
1
(project1) PS C:\Users\Natalya\Desktop\3 курс\TCP\venv>
```

3. Установка библиотек:

Добавляем файл requirements.txt с нужными библиотеками в любую папку, из которой вводим команду [pip install -r requirements.txt](#):



```
requirements.txt
1 numpy
2 pandas
3 matplotlib
4 torch
5 pytest
6
```

Установка пакетов:

```
(project1) PS C:\Users\Natalya\Desktop\3 кypc\TCP\venv> pip install -r requirements.txt
>>
Collecting numpy
  Downloading numpy-1.26.4-cp311-cp311-win_amd64.whl (15.8 MB)
  ----- 15.8/15.8 MB 737.5 kB/s eta 0:00:00
Collecting pandas
  Downloading pandas-2.2.1-cp311-cp311-win_amd64.whl (11.6 MB)
  ----- 11.6/11.6 MB 892.4 kB/s eta 0:00:00
Collecting matplotlib
  Downloading matplotlib-3.8.3-cp311-cp311-win_amd64.whl (7.6 MB)
  ----- 7.6/7.6 MB 1.0 MB/s eta 0:00:00
```

4. Деактивация виртуальной среды:

 Windows PowerShell

```
(project1) PS C:\Users\Natalya\Desktop\3 кypc\TCP\venv> deactivate
>>
PS C:\Users\Natalya\Desktop\3 кypc\TCP\venv>
```

## Источники

- [pythonist.ru/python-virtual-enviroments/#why](https://pythonist.ru/python-virtual-enviroments/#why)
- [devpractice.ru/python-lesson-17-virtual-envs/](https://devpractice.ru/python-lesson-17-virtual-envs/)