# Настройка виртуального окружения

Использование одного и того же окружения в нескольких проектах может привести к ошибкам, например к несовместимости версий библиотек. Чтобы предотвратить подобные конфликты, используют механизм **виртуального окружения**.

Для этого в Python существует специальный модуль venv, который позволяет работать с виртуальным окружением.

Рассмотрим основные операции подробнее.

|  |  |  |
| --- | --- | --- |
| **Операция \ ОС** | **Windows** | **Linux / macOS** |
| Создание | python -m venv .venv | python3 -m venv .venv |
| Активация | .\.venv\Scripts\activate.bat | source ./.venv/bin/activate |
| Деактивация | deactivate | deactivate |

# **Создание виртуального окружения**

Модуль venv обычно поставляется вместе с интерпретатором Python на macOS и Windows. На Ubuntu может потребоваться отдельная установка:

$ sudo apt install python3-venv

Чтобы проверить, что модуль установлен, напишите следующую команду:

$ python -m venv *--help*  
*usage: venv [-h] [--system-site-packages] [--symlinks | --copies] [--clear] [--upgrade] [--without-pip] [--prompt PROMPT] [--upgrade-deps] ENV\_DIR [ENV\_DIR ...]*

Для создания виртуального окружения перейдите в директорию с проектом и воспользуйтесь следующей командой:

$ python -m venv .venv

В текущей рабочей директории будет создана папка .venv.

Активация виртуального окружения

Для активации воспользуйтесь следующей командой:

$ source ./.venv/bin/activate

или в Windows в консоли cmd:

> .\.venv\Scripts\activate.bat

После активации должен появиться специальный маркер такого вида:

(.venv) $

Теперь можно устанавливать библиотеки, которые будут храниться в этом виртуальном окружении.

(.venv) $ pip install Flask

Хорошая практика — указывать версию устанавливаемой библиотеки. Их разработчики тоже могут допускать ошибки, поэтому лучше довериться стабильному и проверенному временем варианту. Например, так можно установить версию 2.2.3:

(.venv) $ pip install Flask==2.2.3

# **Список установленных библиотек**

Для получения всех установленных библиотек в этом окружении можно использовать следующую команду:

(.venv) $ pip freeze  
click==8.1.3  
Flask==2.2.3  
itsdangerous==2.1.2  
Jinja2==3.1.2  
MarkupSafe==2.1.2  
Werkzeug==2.2.3

Библиотеки можно перенаправить в файл requirements.txt, используемый для хранения этих зависимостей.

(.venv) $ pip freeze > requirements.txt

С помощью знака > мы перенаправляем результат команды pip freeze в файл requirements.txt.

# **Установка зависимостей**

Если вы клонировали чужой проект, то все зависимости из файла requirements.txt можно установить одной командой:

(.venv) $ pip install -r requirements.txt

С помощью параметра -r мы говорим, что библиотеки, которые нужно установить, находятся в файле requirements.txt.

# **Деактивация виртуального окружения**

Чтобы выйти из виртуального окружения, достаточно ввести следующую команду:

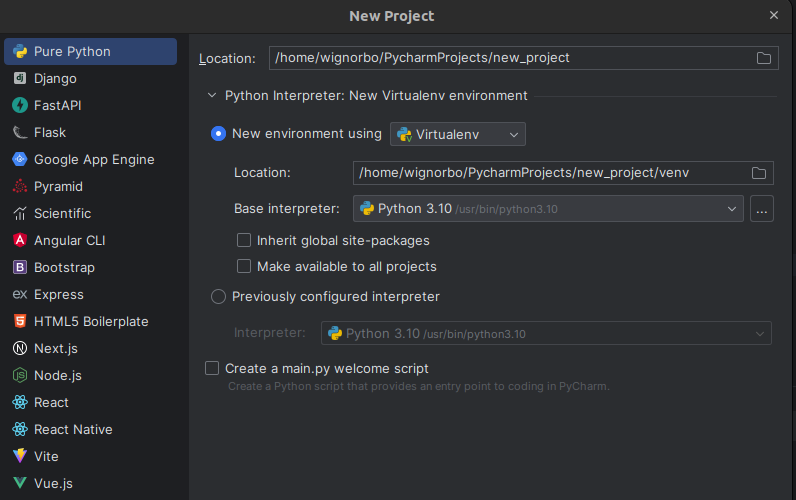
(.venv) $ deactivate

После исполнения должен пропасть маркер с названием виртуального окружения — это значит, что вы успешно вышли из него.

# **Работа с виртуальным окружением в PyCharm**

PyCharm позволяет упростить взаимодействие с виртуальным окружением.

При создании проекта вам сразу предложат выбрать интерпретатор:



Чтобы настроить виртуальное окружение в имеющемся проекте, нажмите File → Settings → Project Settings → Python Interpreter. Должно появиться такое окно:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Далее нажмите Add Interpreter → Add Local Interpreter. Появится следующее окно:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

В поле Location вы указываете путь к виртуальному окружению. В этом случае оно будет находиться в корне проекта и называться venv.

В поле Base interpreter вы выбираете интерпретатор, который будет лежать в основе вашего виртуального окружения. В данном случае — Python 3.10.

Нажмите OK, а затем Apply. Теперь все библиотеки будут устанавливаться в указанное виртуальное окружение.

Открыв терминал, вы увидите маркер, который указывает, что виртуальное окружение активировано:

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

# **Как запустить приложение Flask на операционной системе Windows:**

1. Используя консоль cmd создайте виртуальное окружение с помощью команды python -m venv venv
2. Активируйте виртуальное окружение c помощью команды venv\Scripts\activate.bat
3. Установите Flask с помощью команды pip install Flask
4. Проверьте, что Flask действительно установлен в системе с помощью команды pip freeze | findstr /i flask
5. Перейдите в ваш личный репозиторий с кодом с помощью кнопки «Перейти в GitLab» в практической работе.
6. Клонируйте репозиторий с кодом себе на компьютер с помощью команды git clone <https://gitlab.skillbox.ru/><ВАШ\_НИКНЕЙМ>/python\_advanced.git
7. Перейдите в директорию first\_application с кодом Flask приложения с помощью команды cd module\_01\_flask\materials\first\_application
8. С помощью setx добавьте переменные окружения \*FLASK\_APP и FLASK\_DEBUG с помощью команд  
   setx FLASK\_APP "app.py"  
   setx FLASK\_DEBUG 1
9. Запустите Flask с помощью команды python -m flask run --port=5555
10. Перейдите на<http://127.0.0.1:5555/test>