
Виртуальное окружение - изолированная среда, в которой устанавливаются библиотеки и зависимости для конкретного проекта, не затрагивая глобальную установку Python или другие проекты. Грубо говоря - отдельная "песочница" для проекта

Зачем нужны виртуальные окружения?

1. **Изоляция зависимостей:** разные проекты могут требовать разные версии одной и той же библиотеки (например, `requests==2.25.1` для одного проекта и `requests==2.28.0` для другого). Виртуальное окружение позволяет это сделать без каких-либо конфликтов
2. **Чистота системы:** установка библиотек только в окружении, а не глобально, предотвращает "захламление" основной системы
3. **Легкость воспроизведения:** можно легко передать проект другому сотруднику, указав зависимости (например: в файле `requirements.txt`)
4. **Тестирование:** для автоматизации тестирования можно тестировать код с разными версиями библиотек в разных окружениях

Основы работы с виртуальным окружением

Python имеет встроенный модуль `venv` для создания виртуальных окружений.

Команда для создания виртуального окружения

```
python -m venv <наименование_окружения>
```

```
# Пример фактического использования
```

```
python -m venv venv
```

1. Второе упоминание `venv` - имя папки, где будет создано виртуальное окружение. Обычно окружение называют `venv` или `.venv`
2. После выполнения команды в папке проекта появится директория

Установка библиотек

В окружении можно устанавливать библиотеки с помощью `pip`

```
pip install requests
```

Установленные библиотеки будут находится только в виртуальном окружении.

Сохранение и импорт зависимостей проекта

1. Сохранение зависимостей. Чтобы зафиксировать все установленные библиотеки с проекта, необходимо создать вручную файл `requirements.txt` , а затем прописать следующую команду

```
pip freeze > requirements.txt
```

Этот файл будет содержать список библиотек и их версий

2. Импорт/Установка зависимостей. Чтобы установить библиотеки из `requirements.txt` нужно прописать

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