

< Teach
Me
Skills />

Занятие 13.

Экосистема python

1

Виртуальное окружение

pip

Что это такое?

- система управления пакетами

Основные команды для работы с pip

- `pip install requests` - установить библиотеку
- `pip freeze` - вывести список установленных библиотек
- `pip uninstall requests` - удалить библиотеку
- `pip freeze > requirements.txt` - записать список установленных библиотек в файл
- `pip install -r requirements.txt` - установить библиотеки из файла

Виртуальное окружение

Что это такое?

- Способ изолировать зависимости вашего приложения (когда есть несколько проектов, которые используют разные версии одной и той же библиотеки)
- Может возникнуть необходимость в зависимости от определенной версии библиотеки (чтобы обновления в библиотеке не ломали наш код)

```
- python -m venv <имя виртуального окружения> - создать виртуальное окружение  
- windows: virtual_env\Scripts\activate, linux: source virtual_env/bin/activate - активировать виртуальное окружение  
- deactivate - деактивировать виртуальное окружение
```

Файл requirements.txt

Зачем он нужен?

- нужен для того чтобы перечислить необходимые для запуска проекта библиотеки с их конкретными версиями, облегчает процесс переноса кода и деплоймента, без указания зависимостей невозможно запустить приложение без участия человека
- **ВАЖНО:**
 - не нужно хранить здесь лишние библиотеки или мешать с глобальными
 - этот файл всегда должен находиться в актуальном состоянии
 - должны быть указаны конкретные версии библиотек

Инструкция по запуску приложения

README.md

- это основной файл, из которого можно понять, что это за приложение и как его запустить, форматы тут могут быть разные
- используется язык разметки **markdown**
- что должно быть в инструкции по запуску:
 - версия python, если таковые ограничения имеются
 - какие переменные окружения нужны и за что они отвечают
 - как установить нужные зависимости
 - как запустить приложение

Переменные окружения

Что это такое:

- переменные (строки, хранящие информацию) на основе которых выполняются некоторые команды, например:
 - переменная `pwd` хранит в себе текущую рабочую директорию (`print working directory`), ее можно вывести, на основе нее можно построить абсолютный путь к какой-то дочерней директории и тд
- в python-приложении в переменные окружения помещаются:
 - логины и пароли, ключи и токены
 - ссылки на api (application programming interface)
 - все константы, которые могут изменяться со временем или в зависимости от среды

Работа с переменными окружения в python

- при помощи встроенной библиотеки os можно задавать переменные окружения
- при помощи библиотеки dotenv можно читать переменные окружения из файла

```
from dotenv import load_dotenv
import os

print(f"DOMAIN {os.environ.get('DOMAIN')}")

load_dotenv()
print(f"DOMAIN {os.environ.get('DOMAIN')}")
```


Практика

- создать в репозитории новую папку, в ней виртуальное окружение и активировать его
- установить библиотеку requests
- вывести список установленный в виртуальном окружении библиотек в файл requirements.txt
- создать файл .env и положить туда переменную окружения ENV=test
- прочитать в скрипте и вывести на экран эту переменную окружения (для этого понадобится библиотека python-dotenv)

2

Домашнее задание

Домашнее задание

Задача. Сделать get запрос на <https://jsonplaceholder.typicode.com/todos> и преобразовать данные в инстанс класса Todos, который будет содержать в себе список todo - где каждый элемент является экземпляром класса Todo. Реализовать для класса Todos поведение контейнера: сделать его итерируемым, добавить возможность получения todo по id, а также добавить возможность сериализации и десериализации (запись и чтение из json)

The background is a vibrant yellow color, covered with a repeating pattern of various geometric shapes. These shapes include circles, squares, triangles, and lines, some of which are filled with a fine grid pattern. The shapes are scattered across the entire surface, creating a dynamic and modern aesthetic.

Спасибо за внимание!