Программирование
на Python

Презентация занятия

Проект: Создание CLI приложения.

12 занятие

инжиниум®
МГТУ им. Н.Э. Баумана

2019

Тема: Проект: Создание CLI приложения.

1. CLI И GUI ПРИЛОЖЕНИЯ
1.1 CLI

Command line interface – приложения, использующие интерфейс командной строки, то есть когда мы выполняем какое-то действие с помощью терминала.

```
C:\Users\sweet>python --help
usage: python [option] ... [-c cmd | -m mod | file | -] [arg] ...
Options and arguments (and corresponding environment variables):
-b     : issue warnings about str(bytes_instance), str(bytearray_instance)
         and comparing bytes/bytearray with str. (-bb: issue errors)
-B     : don't write .pyc files on import; also PYTHONDONTWRITEBYTECODE=x
-c cmd : program passed in as string (terminates option list)
-d     : debug output from parser; also PYTHONDEBUG=x
-E     : ignore PYTHON* environment variables (such as PYTHONPATH)
-h     : print this help message and exit (also --help)
-i     : inspect interactively after running script; forces a prompt even
         if stdin does not appear to be a terminal; also PYTHONINSPECT=x
-I     : isolate Python from the user's environment (implies -E and -s)
-m mod : run library module as a script (terminates option list)
-O     : remove assert and __debug__-dependent statements; add .opt-1 before
         .pyc extension; also PYTHONOPTIMIZE=x
```

Тема: Проект: Создание CLI приложения.

1.2 GUI

Graphic User Interface – приложения, использующие графический пользовательский интерфейс, то есть когда мы выполняем какое-то действие с помощью "кнопок".

В чем главное отличие CLI от GUI?

Тема: Проект: Создание CLI приложения.

1.3 Преимущества CLI

1. Малый расход памяти
2. Скорость работы
3. Если знать команды, то их вызов быстрее
4. Просмотр содержимого консоли

Тема: Проект: Создание CLI приложения.

2. ЗАВИСИМОСТИ В СИСТЕМАХ
2.1 Системные зависимости

Системными зависимостями называют свойства и характеристики программного решения, жестко связанные с используемой ОС и ее версией.

Наличие системных зависимостей лишает приложение свойства кросс-платформенности.

Проблемы, которые могут возникнуть:

1. Различные приложения могут использовать одну и туже библиотеку, но версии могут отличаться
2. У вас просто может не быть доступа к каталогу */usr/lib/pythonXX/site-packages*

Тема: Проект: Создание CLI приложения.

2.2 Системная независимость

Системно независимые приложения могут быть установлены и использованы абсолютно на любой системе машины.

2.3 Как сделать проект независимым?

Можно вместе с проектом в распоряжение пользователю необходимые файлы зависимостей.

В python для этого существует **виртуальное окружение**.

Тема: Проект: Создание CLI приложения.

3. ВИРТУАЛЬНОЕ ОКРУЖЕНИЕ
3.1 Установка virtualenv

Виртуальное окружение - изолированное независимое окружение рабочей среды, позволяющее использовать определенные версии приложения вне зависимости от внешней ОС.

Каждый проект может иметь свои собственные зависимости, вне зависимости от того, какие зависимости у другого проекта.

Скачиваем модуль посредством утилиты pip
> pip install virtualenv

Тема: Проект: Создание CLI приложения.

3.2 Что такое pip?

pip - это система управления пакетами, которая используется для установки и управления программными пакетами, написанными на Python (так же является примером CLI приложения).

Можно запускать с параметрами:

pip help - помощь по доступным командам.
pip install package_name - установка пакета(ов).
pip uninstall package_name - удаление пакета(ов).
pip list/freeze - список установленных пакетов.
pip show package_name - показывает информацию об установленном пакете.
pip search - поиск пакетов по имени.
pip install update - обновление пакета(ов).

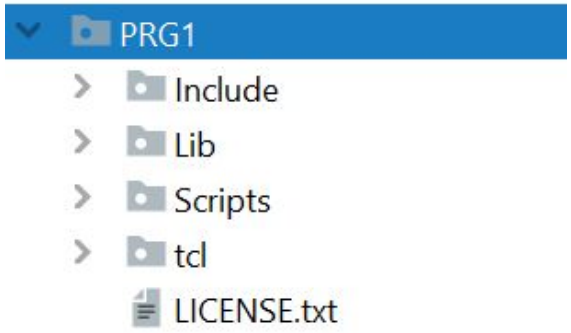
Тема: Проект: Создание CLI приложения.

3.3 Создание виртуального окружения

> virtualenv PRG1

PRG1 – это имя окружения.

После выполнения команды создается новый каталог с именем *PRG1*.



RPG1/Scripts– содержит скрипты для активации/деактивации окружения
PRG1/include/ и PRG1/lib/ – каталоги, содержащие библиотечные файлы окружения.

Новые пакеты будут установлены в каталог *PRG1/lib/pythonX.X/site-packages/*.

Тема: Проект: Создание CLI приложения.

3.4 Активация виртуального окружения

Для активации виртуального окружения воспользуйтесь командой:
> source PRG1/bin/activate для Linux
> PRG1\Scripts\activate.bat для Window

***Если команда выполнилась успешно**, то вы увидите в командной строке дополнительную надпись, совпадающую с именем виртуального окружения.*

Тема: Проект: Создание CLI приложения.

4. АРГУМЕНТЫ CLI
4.1 Что такое системные флаги

- необязательный аргумент, передающийся вместе с командой на выполнение определенной программы.

В зависимости от типа этого флага меняется поведение программы.

```
$ ./snippet.py --help                                          [12:55:21]
Usage: snippet.py [OPTIONS]

Options:
  -s, --service TEXT  Address of the Service.
  -u, --user TEXT     Your username of service "https://google.com".
                      [required]
  -p, --pass TEXT     Your password of service "https://google.com".
                      [required]
  --help              Show this message and exit.
```

Тема: Проект: Создание CLI приложения.

4.2 Функционал проекта «Телефонная книга»

Задача

Создать телефонную книгу с удобным интерфейсом, которая должна быть способна принимать аргументы из командной строки и в соответствии с этими аргументами выполнять действия по добавлению, удалению или обновлению контакта, а так же выводить все контакты

Тема: Проект: Создание CLI приложения.

4.2 Функционал проекта «Телефонная книга»

Добавление контакта

--add Name:telephon

Обновление контакта

--add Name:telephon

Удаление контакта по имени

--delete Name

Отобразить информацию о контакте/контактах

--show Name/all

Флаги обрабатываются при помощи специальных утилит или библиотек.
В Python есть решение — **argparse**.

Тема: Проект: Создание CLI приложения.

5. БИБЛИОТЕКА ARGPARSE

argparse - это модуль для обработки аргументов командной строки (не единственный)

Установка модуля:
pip install argparse

Тема: Проект: Создание CLI приложения.

```
import argparse
book = {"Masha": 89778881539, "Pasha": 89778081476, "Natasha": 89771231536}

parser = argparse.ArgumentParser(description='Telephon book')
parser.add_argument('-a', "--add", dest="param1")
parser.add_argument('-d', "--delete", dest="param2")
parser.add_argument('-s', "--show", dest="param3", default="all")

args = parser.parse_args()
# print(args)

# Формат запуска  python путь_к_файлу --add "Katya:81234567"
if args.param1:
    name, tele = args.param1.split(":")
    if name in book:
        book[name] = [book.get(name), int(tele)]
        print("Контакт с именем ", name, " обновлен")
        print(name, ":", book[name])
    else:
        book[name] = int(tele)
        print("Контакт с именем ", name, " добавлен")
        print(name, ":", book[name])
```

Тема: Проект: Создание CLI приложения.

Задание

Дописать флаги delete и show

Флаг delete должен:

- Удалять контакт по имени, если он существует в книге
- Иначе выводить сообщение об ошибке

Флаг show должен:

- Выводить все контакты, если аргумент равен all
- Выводить контакт и его номер телефона по имени, если он существует
- Иначе выводить сообщение об ошибке

Тема: Проект: Создание CLI приложения.

Рефлексия

1. Какие преимущества CLI приложений перед GUI?
2. Что такое системная зависимость/независимость?
3. Как сделать проект системно независимым?
4. Что такое виртуальное окружение?
5. Что такое pip?
6. Что такое системные флаги?
7. С какой библиотекой мы сегодня познакомились для работы с CLI?