

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное автономное
образовательное учреждение высшего
образования
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Основы кроссплатформенного
программирования Отчет по лабораторной
работе №2.14**

Установка пакетов в Python. Виртуальные окружения

Выполнил студент группы
ИТС-б-о-21-1 (2)

Якупов Э.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил к.т.н., доцент

Кафедры инфокоммуникаций

Воронкин Р.А.

(подпись)

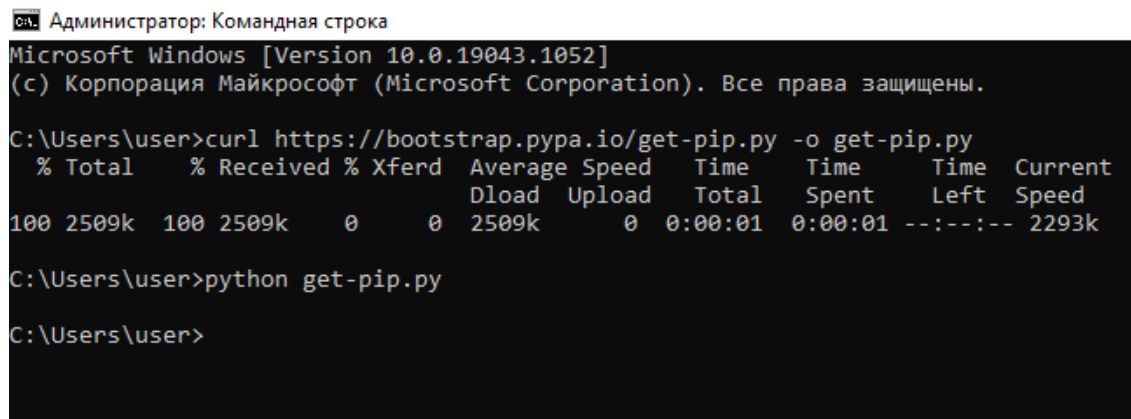
Ставрополь 2022

Цель работы: приобретение навыков по работе с менеджером пакетов `pip` и виртуальными окружениями с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

- 1) Создадим общедоступный репозиторий на GitHub (<https://github.com/Blekroyt/Fox4.git>)
- 2) Решим задачи с помощью языка программирования Python3. И отправим их на GitHub.

Установил `pip`



```
Администратор: Командная строка
Microsoft Windows [Version 10.0.19043.1052]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

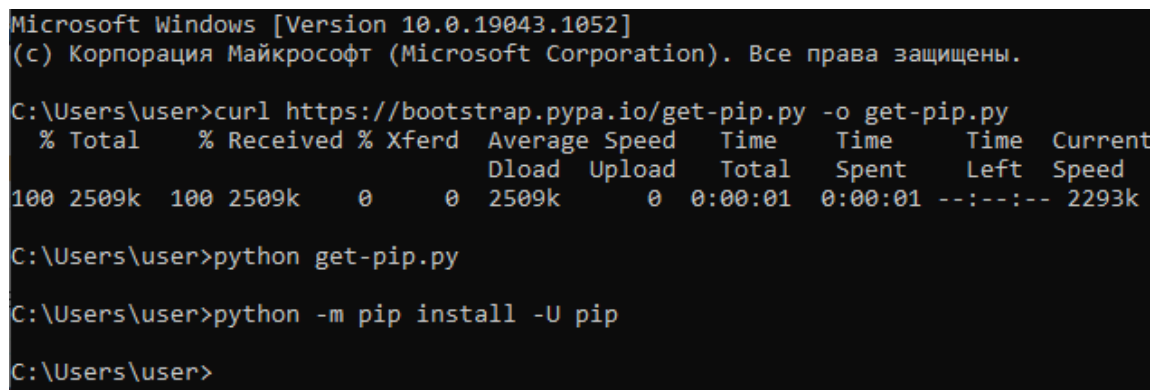
C:\Users\user>curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 2509k  100 2509k    0     0  2509k      0  0:00:01  0:00:01 --:--:-- 2293k

C:\Users\user>python get-pip.py

C:\Users\user>
```

Рисунок 1. Установка `pip`

Обновил `pip` с помощью команды `python -m pip install -U pip`



```
Microsoft Windows [Version 10.0.19043.1052]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\user>curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 2509k  100 2509k    0     0  2509k      0  0:00:01  0:00:01 --:--:-- 2293k

C:\Users\user>python get-pip.py

C:\Users\user>python -m pip install -U pip

C:\Users\user>
```

Рисунок 2. Обновление `pip`

С помощью команды `pip install ProjectName` установил последнюю версию пакета

```

Microsoft Windows [Version 10.0.19043.1052]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\user>curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 2509k  100 2509k    0     0  2509k      0  0:00:01  0:00:01 --:--:-- 2293k

C:\Users\user>python get-pip.py

C:\Users\user>python -m pip install -U pip

C:\Users\user> pip install ProjectName
"pip" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user>pip install ProjectName
"pip" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user>$ pip install ProjectName
"$" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user>pip install package-name
"pip" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

```

Рисунок 3. Установка пакета

С помощью команды `pip install --upgrade ProjectName`

```

C:\Users\user>python -m pip install -U pip

C:\Users\user> pip install ProjectName
"pip" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user>pip install ProjectName
"pip" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user>$ pip install ProjectName
"$" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user>pip install package-name
"pip" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user>pip install --upgrade ProjectName
"pip" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user>pip list
"pip" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\user>pip --version
"pip" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

```

Контрольные вопросы:

1. Каким способом можно установить пакет Python, не входящий в стандартную библиотеку?

Существует так называемый Python Package Index (PyPI) – это репозиторий, открытый для всех Python разработчиков, в нем вы можете найти пакеты для решения практически любых задач.

2. Как осуществить установку менеджера пакетов pip?

```
python -m pip <аргументы>
```

3. Откуда менеджер пакетов pip по умолчанию устанавливает пакеты?

4. Как установить последнюю версию пакета с помощью pip?

```
python -m pip install -U pip
```

5. Как установить заданную версию пакета с помощью pip?

```
pip install ProjectName==3.2
```

6. Как установить пакет из git репозитория (в том числе GitHub) с помощью pip?

```
pip install -e git+https://gitrepo.com/ProjectName.git
```

7. Как установить пакет из локальной директории с помощью pip?

```
pip install ./dist/ProjectName.tar.gz
```

8. Как удалить установленный пакет с помощью pip?

```
pip uninstall ProjectName
```

9. Как обновить установленный пакет с помощью pip?

```
pip install --upgrade ProjectName
```

10. Как отобразить список установленных пакетов с помощью pip?

```
pip list
```

11. Каковы причины появления виртуальных окружений в языке Python?

В отдельной папке создаётся неполная копия выбранной установки Python. Эта копия является просто набором файлов (например, интерпретатора или ссылки на него), утилит для работы с собой и нескольких пакетов (в том числе pip). Стандартные пакеты при этом не копируются.

12. Каковы основные этапы работы с виртуальными окружениями?

- 1) Создаём через утилиту новое виртуальное окружение в отдельной папке для выбранной
- 2) версии интерпретатора Python.
- 3) Активируем ранее созданное виртуального окружения для работы.
- 4) Работаем в виртуальном окружении, а именно управляем пакетами используя `pip` и
- 5) запускаем выполнение кода.
- 6) Деактивируем после окончания работы виртуальное окружение.
- 7) Удаляем папку с виртуальным окружением, если оно нам больше не нужно.

13. Как осуществляется работа с виртуальными окружениями с помощью `venv`?

Создав виртуальное окружение в папке проекта. После её выполнения создастся папка `env` с виртуальным окружением. После активации приглашение консоли изменится. В его начале в круглых скобках будет отображаться имя папки с виртуальным окружением.

14. Как осуществляется работа с виртуальными окружениями с помощью `virtualenv`?

Создание в текущей папке виртуального окружения для интерпретатора доступного через команду `python3` с названием папки окружения `env`.

`freeze` - команда, используемая для получения всех установленных пакетов в формате требований. Таким образом, все пакеты, которые вы установили перед выполнением команды и предположительно использовали в каком-либо проекте, будут перечислены в файле с именем «`requirements.txt`». Кроме того, будут указаны их точные версии

15. Изучите работу с виртуальными окружениями `pipenv`. Как осуществляется работа с виртуальными окружениями `pipenv`?

При запуске проекта с `Pipenv`, он автоматически создает виртуальную среду для текущего проекта, даже если вы еще не используете ее. `Pipenv`

управляет зависимостями, отказавшись от привычного requirements.txt, и заменяя его на новый документ под названием Pipfile.

16. Каково назначение файла requirements.txt ? Как создать этот файл? Какой он имеет формат?

Способ хранения списка внешних зависимостей проекта. `pip freeze > requirements.txt . <requirement specifier>`

17. В чем преимущества пакетного менеджера conda по сравнению с пакетным менеджером pip?

Conda же способна управлять пакетами как для Python, так и для C/ C++, R, Ruby, Lua, Scala и других. Conda устанавливает двоичные файлы, поэтому работу по компиляции пакета самостоятельно выполнять не требуется (по сравнению с pip).

18. В какие дистрибутивы Python входит пакетный менеджер conda?

Anaconda

19. Как создать виртуальное окружение conda?

`conda create -n %PROJ_NAME% python=3.7`

`conda activate %PROJ_NAME%`

20. Как активировать и установить пакеты в виртуальное окружение conda?

`conda activate env`

21. Как деактивировать и удалить виртуальное окружение conda?

`conda deactivate`

22. Каково назначение файла environment.yml ? Как создать этот файл?

23. Как создать виртуальное окружение conda с помощью файла environment.yml ?

Создайте окружение из environment.yml файла Первая строка yml файла задает имя новой среды. Активируйте новую среду: `conda activate myenv` Убедитесь, что новая среда установлена правильно: `conda env list`

24. Самостоятельно изучите средства IDE PyCharm для работы с виртуальными окружениями conda. Опишите порядок работы с виртуальными окружениями conda в IDE PyCharm.

25. Почему файлы requirements.txt и environment.yml должны храниться в репозитории git?

Стандартные файлы виртуального окружения.

Вывод: приобретение навыков по работе с менеджером пакетов pip и виртуальными окружениями с помощью языка программирования Python версии 3.x.