

Министерство науки и высшего образования Российской
Федерации Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4
Установка пакетов в Python. Виртуальные окружения

Выполнил:

Боженко Александр Иванович

2 курс, группа ИТС-б-о-21-1,

11.03.02 «Инфокоммуникационные

технологии и системы связи»,

направленность (профиль)

«Инфокоммуникационные системы и сети»,

очная форма обучения

(подпись)

Руководитель практики:

Воронкин Р.А, канд. техн. наук, доцент

кафедры инфокоммуникаций

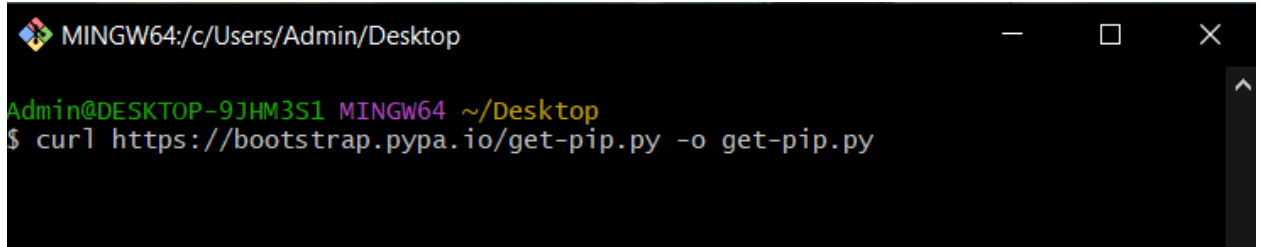
(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

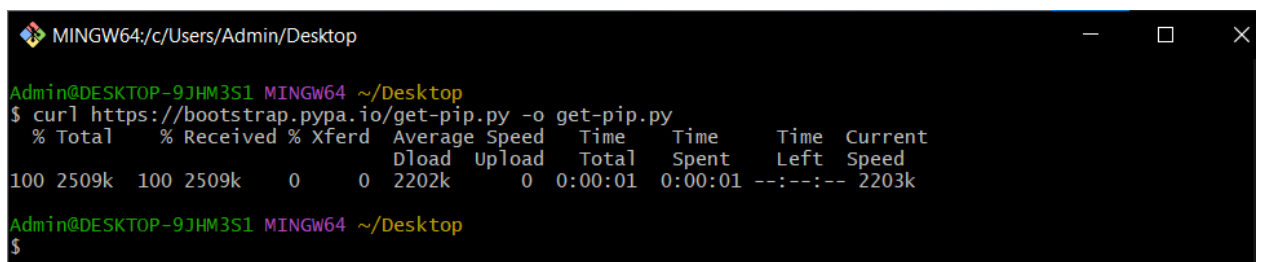
Цель работы: приобретение навыков по работе с менеджером пакетов `pip` и виртуальными окружениями с помощью языка программирования Python версии 3.x

Установка `pip`



```
MINGW64:/c/Users/Admin/Desktop
Admin@DESKTOP-9JHM3S1 MINGW64 ~/Desktop
$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

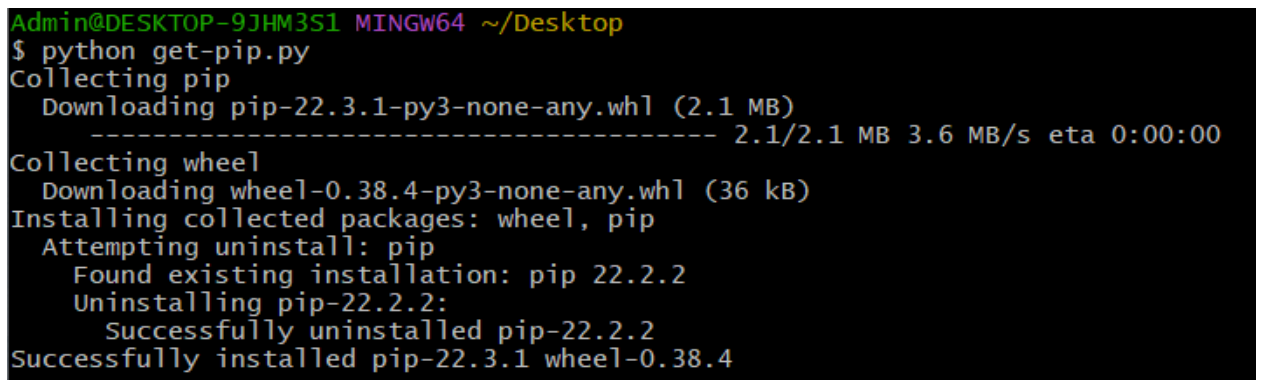
Рисунок 1. Ввод команды для установки `pip`



```
MINGW64:/c/Users/Admin/Desktop
Admin@DESKTOP-9JHM3S1 MINGW64 ~/Desktop
$ 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 2202k      0  0:00:01  0:00:01 --:--:-- 2203k
Admin@DESKTOP-9JHM3S1 MINGW64 ~/Desktop
$
```

Рисунок 2. Установка `pip`

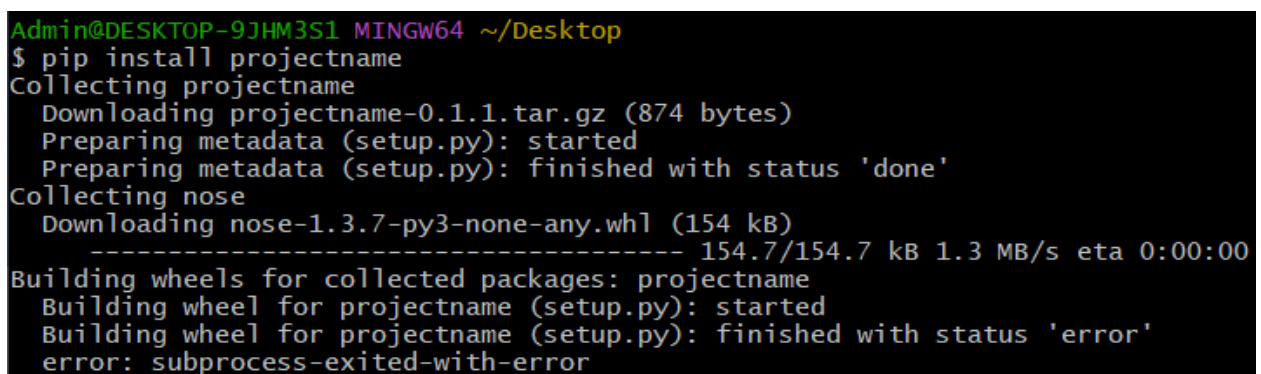
Обновил `pip` с помощью команды `python get-pip.py`



```
Admin@DESKTOP-9JHM3S1 MINGW64 ~/Desktop
$ python get-pip.py
Collecting pip
  Downloading pip-22.3.1-py3-none-any.whl (2.1 MB)
  ----- 2.1/2.1 MB 3.6 MB/s eta 0:00:00
Collecting wheel
  Downloading wheel-0.38.4-py3-none-any.whl (36 kB)
Installing collected packages: wheel, pip
  Attempting uninstall: pip
    Found existing installation: pip 22.2.2
    Uninstalling pip-22.2.2:
      Successfully uninstalled pip-22.2.2
Successfully installed pip-22.3.1 wheel-0.38.4
```

Рисунок 3. Обновление `pip`

Установка пакета



```
Admin@DESKTOP-9JHM3S1 MINGW64 ~/Desktop
$ pip install projectname
Collecting projectname
  Downloading projectname-0.1.1.tar.gz (874 bytes)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Collecting nose
  Downloading nose-1.3.7-py3-none-any.whl (154 kB)
  ----- 154.7/154.7 kB 1.3 MB/s eta 0:00:00
Building wheels for collected packages: projectname
  Building wheel for projectname (setup.py): started
  Building wheel for projectname (setup.py): finished with status 'error'
error: subprocess-exited-with-error
```

Рисунок 4.

Просмотр установленных пакетов

```
Admin@DESKTOP-9JHM3S1 MINGW64 ~/Desktop
$ pip list
Package      Version
-----
nose         1.3.7
pip          22.3.1
setuptools   63.2.0
wheel        0.38.4
```

Рисунок 5.

файлы requirements.txt

```
F:\2 курс\языки программирования\4>pip install -r requirements.txt
Requirement already satisfied: nose==1.3.7 in c:\users\admin\appdata\local\
\site-packages (from -r requirements.txt (line 1)) (1.3.7)

F:\2 курс\языки программирования\4>pip freeze
nose==1.3.7
```

Рисунок 5.

файлы environment.yml

```
PS F:\Универ\Воронкин\14 лаба> conda env create -f environment.yml
Collecting package metadata (repodata.json): done
Solving environment: done

Downloading and Extracting Packages
tornado-6.2 | 609 KB | ##### | 100%
nodejs-16.13.1 | 18.9 MB | ##### | 100%
certifi-2022.9.24 | 154 KB | ##### | 100%
bokeh-2.4.2 | 7.6 MB | ##### | 100%
```

Контрольные вопросы:

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

Существует так называемый PythonPackageIndex (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 виртуальное окружение 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 проект 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. `pipfreeze>`
`requirements.txt . <requirementspecifier>`

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 my env`. 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.