

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 №17
дисциплины «Основы программной инженерии»

Выполнила:
Ламская Ксения Вячеславовна
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Лабораторная работа 2.14. Установка пакетов в Python. Виртуальные окружения.

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. Создание репозитория GitHub.

Create a new repository



A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *	Repository name *
 ksenia-lamskaya ▾	/ 17laba
 17laba is available.	

Great repository names are short and memorable. Need inspiration? How about **fantastic-waddle** ?

Description (optional)

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

- ☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Python ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание репозитория

2. Запуск Anaconda Powershell Prompt.

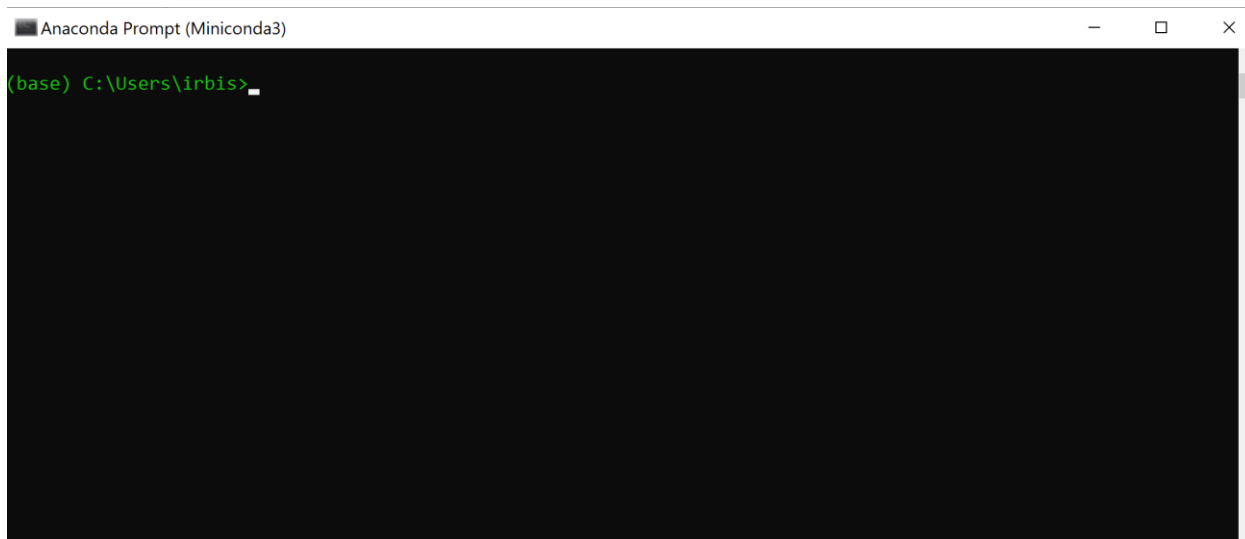


Рисунок 2.1 - Запуск Anaconda Powershell Prompt.

3. Создание виртуального окружения Anaconda с именем репозитория.

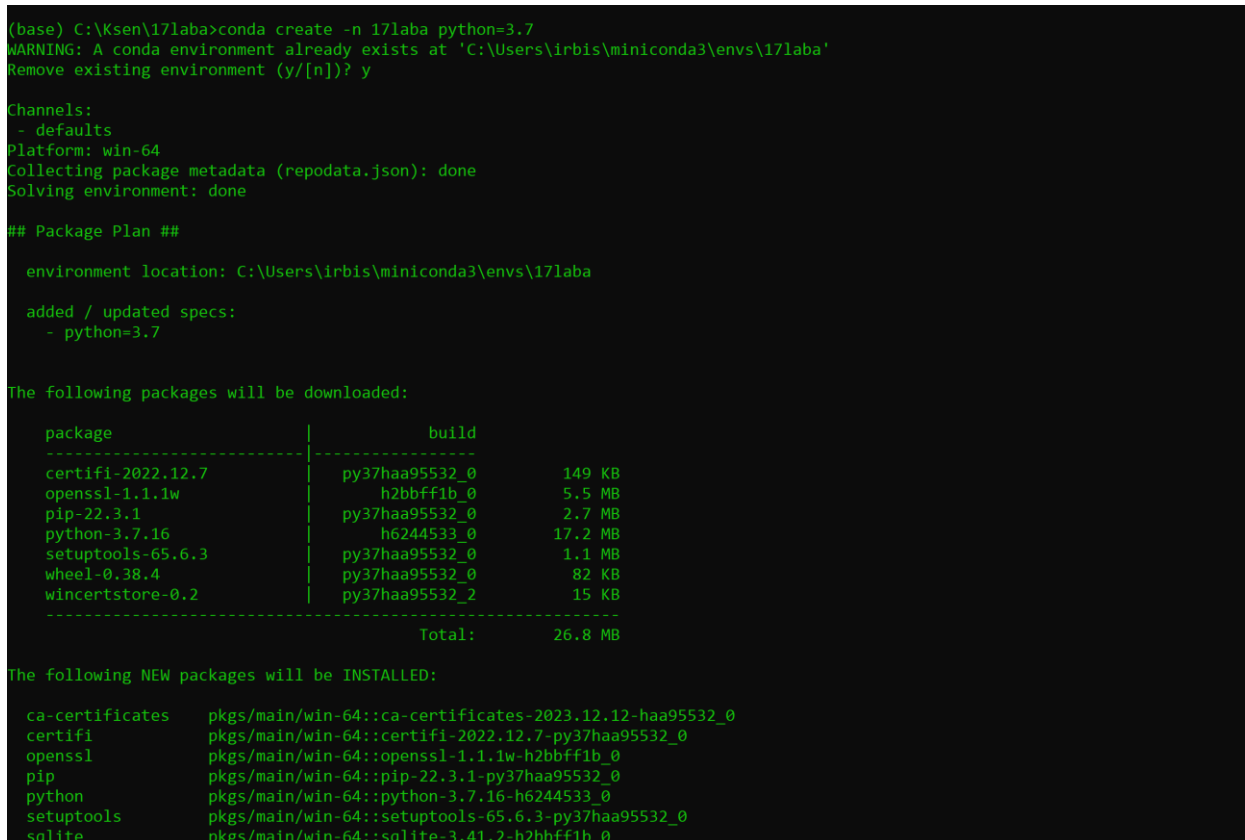


Рисунок 3.1 – Командная строка

4. Активация виртуального окружения, установка пакетов: `pip`, `NumPy`, `Pandas`, `SciPy`.

```
(base) C:\Ksen\17laba>conda activate 17laba  
(17laba) C:\Ksen\17laba>
```

Рисунок 4.1 – Активация

```
environment location: C:\Users\irbis\miniconda3\envs\17laba  
  
added / updated specs:  
- numpy  
- pandas  
- pip  
- scipy  
  
The following packages will be downloaded:
```

package	build	
blas-1.0	mk1	6 KB
bottleneck-1.3.5	py37h080aedc_0	105 KB
fftw-3.3.9	h2bbff1b_1	672 KB
icc_rt-2022.1.0	h6049295_2	6.5 MB
intel-openmp-2021.4.0	haa95532_3556	2.2 MB
mk1-2021.4.0	haa95532_640	114.9 MB
mk1-service-2.4.0	py37h2bbff1b_0	49 KB
mk1_fft-1.3.1	py37h277e83a_0	135 KB
mk1_random-1.2.2	py37hf11a4ad_0	216 KB
numexpr-2.8.4	py37h5b0cc5e_0	127 KB
numpy-1.21.5	py37h7a0a035_3	25 KB
numpy-base-1.21.5	py37hca35cd5_3	4.4 MB
packaging-22.0	py37haa95532_0	67 KB
pandas-1.3.5	py37h6214cd6_0	8.4 MB
python-3.7.1	h33f27b4_4	16.0 MB
python-dateutil-2.8.2	pyhd3eb1b0_0	233 KB
pytz-2022.7	py37haa95532_0	210 KB
scipy-1.7.3	py37h7a0a035_2	13.8 MB
six-1.16.0	pyhd3eb1b0_1	18 KB
Total:		168.0 MB

Рисунок 4.2 – Установка пакетов

5. Установка `tensorflow` при помощи `conda` и при помощи `pip`.

```
(17laba) C:\Ksen\17laba>conda install tensorflow  
Channels:  
- defaults  
Platform: win-64  
Collecting package metadata (repodata.json): done  
Solving environment: done  
  
## Package Plan ##  
  
environment location: C:\Users\irbis\miniconda3\envs\17laba  
  
added / updated specs:  
- tensorflow  
  
The following packages will be downloaded:
```

package	build	
_tflow_select-2.2.0	eigen	3 KB
absl-py-1.3.0	py37haa95532_0	170 KB
aiohttp-3.8.3	py37h2bbff1b_0	411 KB
aiohttp-1.2.0	pyhd3eb1b0_0	12 KB
astunparse-1.6.3	py_0	17 KB
async-timeout-4.0.2	py37haa95532_0	12 KB
asynctest-0.13.0	py_0	26 KB
attrs-22.1.0	py37haa95532_0	84 KB

Рисунок 5.1 – Установка при помощи `conda`

```
(17laba) C:\Ksen\17laba>conda list tensorflow
# packages in environment at C:\Users\irbis\miniconda3\envs\17laba:
#
# Name                      Version                      Build      Channel
tensorflow                  2.10.0                      eigen_py37h0b514e4_0
tensorflow-base             2.10.0                      eigen_py37he3c91d7_0
tensorflow-estimator        2.10.0                      py37haa95532_0
```

Рисунок 5.2 – Установка при помощи conda

```
(17laba) C:\Ksen\17laba>pip install tensorflow
Requirement already satisfied: tensorflow in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (2.10.0)
Requirement already satisfied: six>=1.12.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: keras<2.11,>=2.10.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (2.10.0)
Collecting tensorflow-io-gcs-filesystem>=0.23.1
  Downloading tensorflow_io_gcs_filesystem-0.31.0-cp37-cp37m-win_amd64.whl (1.5 MB)
    ----- 1.5/1.5 MB 1.9 MB/s eta 0:00:00
Requirement already satisfied: absl-py>=1.0.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (1.3.0)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (0.4.0)
Collecting protobuf<3.20,>=3.9.2
  Downloading protobuf-3.19.6-cp37-cp37m-win_amd64.whl (896 kB)
    ----- 896.6/896.6 kB 3.0 MB/s eta 0:00:00
Requirement already satisfied: astunparse>=1.6.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: packaging in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (22.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (1.1.2)
Collecting libclang>=13.0.0
  Downloading libclang-16.0.6-py2.py3-none-win_amd64.whl (24.4 MB)
    ----- 24.4/24.4 MB 2.2 MB/s eta 0:00:00
Requirement already satisfied: wrapt>=1.11.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (1.14.1)
Requirement already satisfied: h5py>=2.9.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (3.7.0)
Requirement already satisfied: flatbuffers>=2.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (2.0)
Requirement already satisfied: numpy>=1.20 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (1.21.5)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (4.4.0)
Requirement already satisfied: setuptools in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (65.6.3)
Requirement already satisfied: tensorflow-estimator<2.11,>=2.10.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (2.10.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (1.42.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (2.1.0)
Requirement already satisfied: tensorboard<2.11,>=2.10 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorflow) (2.10.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from astunparse>=1.6.0->tensorflow) (0.38.4)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in c:\users\irbis\miniconda3\envs\17laba\lib\site-packages (from tensorboard<2.11,>=2.10->tensorflow) (0.6.0)
```

Рисунок 5.3 – Установка при помощи pip

6. Формирование файлов requirement.txt и environment.yml.

```
(17laba) C:\Ksen\17laba>pip freeze > requirements.txt
(17laba) C:\Ksen\17laba>conda env export > environment.yml
```

Рисунок 6.1 – Формирование файлов

! environment.yml X

C: > Ksen > 17laba > ! environment.yml

```
1 name: 17laba
2 channels:
3   - defaults
4 dependencies:
5   - _tfflow_select=2.2.0=eigen
6   - absl-py=1.3.0=py37haa95532_0
7   - aiohttp=3.8.3=py37h2bbff1b_0
8   - aiosignal=1.2.0=pyhd3eb1b0_0
9   - astunparse=1.6.3=py_0
10  - async-timeout=4.0.2=py37haa95532_0
11  - asyncctest=0.13.0=py_0
12  - attrs=22.1.0=py37haa95532_0
13  - blas=1.0=mkl
14  - blinker=1.4=py37haa95532_0
15  - bottleneck=1.3.5=py37h080aedc_0
16  - brotli=0.7.0=py37h2bbff1b_1003
17  - ca-certificates=2023.12.12=haa95532_0
18  - cachetools=4.2.2=pyhd3eb1b0_0
19  - certifi=2022.12.7=py37haa95532_0
20  - cffi=1.15.1=py37h2bbff1b_3
21  - charset-normalizer=2.0.4=pyhd3eb1b0_0
22  - click=8.0.4=py37haa95532_0
23  - colorama=0.4.6=py37haa95532_0
24  - cryptography=39.0.1=py37h21b164f_0
25  - fftw=3.3.9=h2bbff1b_1
26  - flatbuffers=2.0.0=h6c2663c_0
27  - flit-core=3.6.0=pyhd3eb1b0_0
28  - frozenlist=1.3.3=py37h2bbff1b_0
29  - gast=0.4.0=pyhd3eb1b0_0
30  - giflib=5.2.1=h8cc25b3_3
31  - google-auth=2.6.0=pyhd3eb1b0_0
32  - google-auth-oauthlib=0.4.4=pyhd3eb1b0_0
33  - google-pasta=0.2.0=pyhd3eb1b0_0
```

Рисунок 6.2 – Файл environment.yml

```
! environment.yml requirements.txt X
C: > Ksen > 17laba > requirements.txt
1  absl-py @ file:///C:/b/abs_5babsu7y5x/croot/absl-py_1666362945682/work
2  aiohttp @ file:///C:/b/abs_c4zmy2l696/croot/aiohttp_1670009573673/work
3  aiosignal @ file:///tmp/build/80754af9/aiosignal_1637843061372/work
4  astunparse==1.6.3
5  async-timeout @ file:///C:/b/abs_43ozhz2a8g/croots/recipe/async-timeout_1664876362767/work
6  asynctest==0.13.0
7  attrs @ file:///C:/b/abs_09s3y775ra/croot/attrs_1668696195628/work
8  blinker==1.4
9  Bottleneck @ file:///C:/Windows/Temp/abs_3198ca53-903d-42fd-87b4-03e6d03a8381yfwsuve8/croots/recipe/bottleneck_1670009573673/work
10 brotli==1.0.9
11 cachetools @ file:///tmp/build/80754af9/cachetools_1619597386817/work
12 certifi @ file:///C:/b/abs_85o_6fm0se/croot/certifi_1671487778835/work/certifi
13 cffi @ file:///C:/b/abs_49n3v2hyhr/croot/cffi_1670423218144/work
14 charset-normalizer @ file:///tmp/build/80754af9/charset-normalizer_1630003229654/work
15 click @ file:///C:/ci/click_1646038601470/work
16 colorama @ file:///C:/b/abs_a9ozq0l032/croot/colorama_1672387194846/work
17 cryptography @ file:///C:/b/abs_8ecplyc3n2/croot/cryptography_1677533105000/work
18 flatbuffers @ file:///home/ktietz/cip/python-flatbuffers_1634039120618/work
19 flit_core @ file:///opt/conda/conda-bld/flit-core_1644941570762/work/source/flit_core
20 frozenlist @ file:///C:/b/abs_2bb5uzghsi/croot/frozenlist_1670004511812/work
21 gast @ file:///Users/ktietz/demo/mc3/conda-bld/gast_1628588903283/work
22 google-auth @ file:///opt/conda/conda-bld/google-auth_1646735974934/work
23 google-auth-oauthlib @ file:///tmp/build/80754af9/google-auth-oauthlib_1617120569401/work
24 google-pasta @ file:///Users/ktietz/demo/mc3/conda-bld/google-pasta_1630577991354/work
25 grpcio @ file:///C:/ci/grpcio_1637590993074/work
26 h5py @ file:///C:/ci/h5py_1659089886851/work
27 idna @ file:///C:/b/abs_bdhibebrio/croot/idna_1666125572046/work
28 importlib-metadata @ file:///C:/ci/importlib-metadata_1648562631189/work
29 keras @ file:///C:/Users/builder/adipietro/mc3/tf210/conda-bld/keras_1669760570649/work/keras-2.10.0-py2.py3-none-any.whl
```

Рисунок 6.3 – Файл requirements.txt

Контрольные вопросы:

1. Если вы хотите добавить новый пакет Python, который не входит в стандартную библиотеку, то просто используйте менеджер пакетов `pip`. Просто запустите команду `pip install имя_пакета`, и пакет будет установлен.
2. Для того чтобы получить менеджер пакетов `pip`, просто загрузите скрипт `get-pip.py` и выполните его с помощью Python. Просто запустите `python get-pip.py`.
3. По умолчанию менеджер пакетов `pip` загружает пакеты из Python Package Index (PyPI), что делает процесс установки пакетов очень простым.
4. Если вам нужна последняя версия пакета, то можно воспользоваться командой `pip install имя_пакета`.
5. Для установки конкретной версии пакета, используйте команду `pip install имя_пакета==версия`.
6. Установить пакет из `git` репозитория можно с помощью команды `pip install git+https://github.com/пользователь/репозиторий.git`.

7.Если вам нужно установить пакет из локальной директории, просто выполните `pip install ./директория`.

8.Чтобы удалить установленный пакет, просто выполните `pip uninstall имя_пакета`.

9.Для обновления установленного пакета, используйте команду `pip install --upgrade имя_пакета`.

10.Список всех установленных пакетов можно увидеть с помощью команды `pip list`.

11.Виртуальные окружения в Python позволяют изолировать зависимости проектов и предотвращать конфликты между разными версиями пакетов.

12.Создание, активация, установка зависимостей и деактивация виртуальных окружений - основные этапы работы с ними.

13.Для работы с виртуальными окружениями через `venv`, используйте команды: создание - `python -m venv имя_окружения`, активация - `source имя_окружения/bin/activate` или `имя_окружения\Scripts\activate`, установка и деактивация.

14.Работа с виртуальными окружениями через `virtualenv` аналогична, но требует предварительной установки `virtualenv`.

15.Виртуальные окружения `pipenv` включают создание, активацию, установку зависимостей через `pipenv install`, активацию через `pipenv shell` и установку зависимостей из `Pipfile`.

16.Файл `requirements.txt` используется для определения зависимостей проекта с указанием версий пакетов.

17.Преимущества `conda` включают управление не только Pythonпакетами, но и бинарными зависимостями и окружениями.

18.Пакетный менеджер `conda` поставляется в дистрибутивах Python Anaconda и Miniconda.

19.Виртуальное окружение `conda` создается командой `conda create --name имя_окружения пакеты`.

20.Для активации виртуального окружения conda используйте `conda activate имя_окружения`, для установки пакетов - `conda install пакеты`.

21.Для деактивации виртуального окружения conda используйте `conda deactivate`, для удаления - `conda remove --name имя_окружения --all`.

22.Файл `environment.yml` используется для определения окружения conda и его зависимостей.

23.Создать виртуальное окружение conda из файла `environment.yml` можно с помощью `conda env create -f environment.yml`.

24.В PyCharm работа с виртуальными окружениями conda включает выбор окружения при настройке проекта и управление зависимостями.

25.Файлы `requirements.txt` и `environment.yml` необходимы для воспроизводимости среды разработки и зависимостей проекта среди разработчиков.