

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 «Программирование на Python»

Выполнила:
Кубанова Ксения Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: установка пакетов, виртуальные окружения

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

Порядок выполнения работы

Для начала необходимо создать окружение с таким же названием, как и репозиторий и активировать его:

```
(base) C:\Users\Сергей>conda create -n Labpy2.14
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Рисунок 1 – создание окружения

```
(base) C:\Users\Сергей>conda activate Labpy2.14
(Labpy2.14) C:\Users\Сергей>conda list
```

Рисунок 2 – активация окружения

После этого выполнить загрузку `pip`, менеджера пакетов:

```
(Labpy2.14) C:\Users\Сергей>conda install pip
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Рисунок 3 – загрузка `pip`

А после уже и остальных (`pandas`, `scipy`, `numpy` соответственно) :

```
(Labpy2.14) C:\Users\Сергей>pip install pandas
Collecting pandas
  Downloading pandas-2.1.4-cp312-cp312-win_amd64.whl.metadata (18 kB)
Collecting numpy<2,>=1.26.0 (from pandas)
  Downloading numpy-1.26.3-cp312-cp312-win_amd64.whl.metadata (61 kB)
----- 61.2/61.2 kB 1.1 MB/s eta 0:00:00
Collecting python-dateutil>=2.8.2 (from pandas)
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
----- 247.7/247.7 kB 3.0 MB/s eta 0:00:00
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.1 (from pandas)
  Downloading tzdata-2023.4-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas)
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Downloading pandas-2.1.4-cp312-cp312-win_amd64.whl (10.5 MB)
----- 10.5/10.5 MB 6.1 MB/s eta 0:00:00
Downloading numpy-1.26.3-cp312-cp312-win_amd64.whl (15.5 MB)
----- 15.5/15.5 MB 7.5 MB/s eta 0:00:00
Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
----- 502.5/502.5 kB 7.9 MB/s eta 0:00:00
Downloading tzdata-2023.4-py2.py3-none-any.whl (346 kB)
----- 346.6/346.6 kB 7.2 MB/s eta 0:00:00
Installing collected packages: pytz, tzdata, six, numpy, python-dateutil, pandas
Successfully installed numpy-1.26.3 pandas-2.1.4 python-dateutil-2.8.2 pytz-2023.3.post1 six-1.16.0 tzdata-2023.4
```

Рисунок 4 – загрузка `pandas`

```
(Labpy2.14) C:\Users\Сергей>pip install scipy
Collecting scipy
  Downloading scipy-1.11.4-cp312-cp312-win_amd64.whl.metadata (60 kB)
----- 60.4/60.4 kB 533.6 kB/s eta 0:00:00
Requirement already satisfied: numpy<1.28.0,>=1.21.6 in c:\ana\envs\labpy2.14\lib\site-packages (from scipy)
Downloading scipy-1.11.4-cp312-cp312-win_amd64.whl (43.7 MB)
----- 43.7/43.7 MB 3.8 MB/s eta 0:00:00
Installing collected packages: scipy
Successfully installed scipy-1.11.4
```

Рисунок 5 – загрузка scipy

```
(Labpy2.14) C:\Users\Сергей>pip install numpy
Requirement already satisfied: numpy in c:\ana\envs\labpy2.14\lib\site-packages (from scipy)
```

Рисунок 6 – загрузка numpy, ошибка

Пакет numpy уже был автоматически установлен вместе с пакетом pip.

Далее необходимо установить менеджер пакетов tensorflow с conda и pip:

```
(Labpy2.14) C:\Users\Сергей>conda install tensorflow
Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Solving environment: unsuccessful attempt using repodata from current_repodata.json, retrying with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Solving environment: \
Found conflicts! Looking for incompatible packages.
This can take several minutes. Press CTRL-C to abort.
failed

UnsatisfiableError: The following specifications were found
to be incompatible with the existing python installation in your environment:

Specifications:

- tensorflow -> python[version='3.10.*|3.9.*|3.8.*|3.7.*|3.6.*|3.5.*']

Your python: python=3.12

If python is on the left-most side of the chain, that's the version you've asked for.
When python appears to the right, that indicates that the thing on the left is somehow
not available for the python version you are constrained to. Note that conda will not
change your python version to a different minor version unless you explicitly specify
that.
```

Рисунок 7 – установка tensorflow conda

```
(Labpy2.14) C:\Users\Сергей>pip install tensorflow
ERROR: Could not find a version that satisfies the requirement tensorflow (from versions: none)
ERROR: No matching distribution found for tensorflow
```

Рисунок 8 – установка tensorflow pip

При попытке загрузить с помощью pip возникла ошибка, связанная с не нахождением нужной версии и с отсутствием соответствующего распределения.

Далее формируются файлы environment.yml (содержит параметры окружения, используется для его воссоздания) и requirements.txt (содержит все пакеты, которые были установлены перед выполнением команды):

```
(Labpy2.14) C:\Users\Сергей>conda env export > environment.yml
(Labpy2.14) C:\Users\Сергей>pip freeze > requirements.txt
```

Рисунок 9 – формирование environment.yml и requirements.txt

```
C:\Users\Сергей>cat requirements.txt
1 numpy==1.26.3
2 pandas==2.1.4
3 python-dateutil==2.8.2
4 pytz==2023.3.post1
5 scipy==1.11.4
6 setuptools==68.2.2
7 six==1.16.0
8 tzdata==2023.4
9 wheel==0.41.2
10
```

Рисунок 10 – содержание requirements.txt

```
C:\Users\Сергей>cat environment.yml
1 name: Labpy2.14
2 channels:
3   - defaults
4 dependencies:
5   - bzip2=1.0.8=he774522_0
6   - ca-certificates=2023.12.12=haa95532_0
7   - expat=2.5.0=hd77b12b_0
8   - libffi=3.4.4=hd77b12b_0
9   - openssl=3.0.12=h2bbff1b_0
10  - pip=23.3.1=py312haa95532_0
11  - python=3.12.0=h1d929f7_0
12  - setuptools=68.2.2=py312haa95532_0
13  - sqlite=3.41.2=h2bbff1b_0
14  - tk=8.6.12=h2bbff1b_0
15  - vc=14.2=h21ff451_1
16  - vs2015_runtime=14.27.29016=h5e58377_2
17  - wheel=0.41.2=py312haa95532_0
18  - xz=5.4.5=h8cc25b3_0
19  - zlib=1.2.13=h8cc25b3_0
20  - pip:
21    - numpy==1.26.3
22    - pandas==2.1.4
23    - python-dateutil==2.8.2
24    - pytz==2023.3.post1
25    - scipy==1.11.4
26    - six==1.16.0
27    - tzdata==2023.4
28 prefix: C:\ana\envs\Labpy2.14
29
```

Рисунок 11 – содержание environment.yml

Контрольные вопросы

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

Можно использовать менеджер пакетов `pip`.

2 Как осуществить установку менеджера пакетов `pip`?

Если менеджер пакетов уже не установлен, его можно установить по инструкции для определённой ОС.

3 Откуда менеджер пакетов `pip` по умолчанию устанавливает пакеты?

По умолчанию из Python Package Index, но может и из других источников – репозитории и др.

4 Как установить последнюю версию пакета с помощью `pip`?

`Pip install --upgrade package_name`

5 Как установить заданную версию пакета с помощью `pip`?

`Pip install package_name == version number`

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

`Pip install git+http://github.com/username/repository.git`

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

`Pip install /path/to/local/directory`

8 Как удалить установленный пакет с помощью `pip`?

`Pip uninstall package_name`

9 Как обновить установленный пакет с помощью `pip`?

`Pip install --upgrade package_name`

10 Как отобразить список установленных пакетов с помощью `pip`?

`Pip list`

11 Каковы причины появления виртуальных окружений в языке Python?

Виртуальные окружения используются для изоляции проектов и их зависимостей для избежания конфликтов между различными версиями пакетов.

12 Каковы основные этапы работы с виртуальными окружениями?

Создание, активация, деактивация, удаление виртуальных окружений.

13 Как осуществляется работа с виртуальными окружениями с помощью venv?

Необходимо использовать стандартную библиотеку Python для создания и управления виртуальных окружений.

14 Как осуществляется работа с виртуальными окружениями с помощью virtualenv?

Он предоставляет инструменты для создания изолированных виртуальных окружений.

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

Представляет удобный способ управления зависимостями и виртуальными окружениями для проектов Python.

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

Используется для хранения зависимостей проекта, что позволяет легко установить их на другой системе. Создаётся вручную и содержит список пакетов и их версий.

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

Conda позволяет управлять не только пакетами, но и библиотеками, написанными на других языках. Так же умеет устанавливать библиотеки, которые содержат бинарные зависимости, что делает его более гибким по сравнению с pip.

18 В какие дистрибутивы Python входит пакетный менеджер conda?

Conda входит в дистрибутив Anaconda и Miniconda, которые предоставляют широкий выбор пакетов для научных вычислений и анализа данных.

19 Как создать виртуальное окружение conda?

Conda create –name myenv, myenv – имя окружения.

20 Как активировать и установить пакеты в виртуальное окружение conda?

Conda activate myenv

21 Как деактивировать и удалить виртуальное окружение conda?

Conda deactivate - деактивация, conda remove myenv –all – удаление.

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

Используется для написания окружения conda, включая список пакеов и их версий. Файл можно создать вручную или сгенерировать: conda env export > environment.yml.

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

Conda env create -f environment.yml.

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

В PyCharm можно работать с виртуальными окружениями conda, создавая и активируя их через интерфейс пользователя.

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

Файлы environment.yml и requirements.txt должны храниться в репозитории, чтобы другие разработчики могли легко воссоздать окружение

проекта на своих системах. Это позволяет обеспечить консистентность окружения и упростить процесс развертывания проекта.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе с менеджером пакетов `pip` и виртуальными окружениями с помощью языка программирования Python версии 3.x.