

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»

Выполнил:
Середа Кирилл Витальевич
1 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

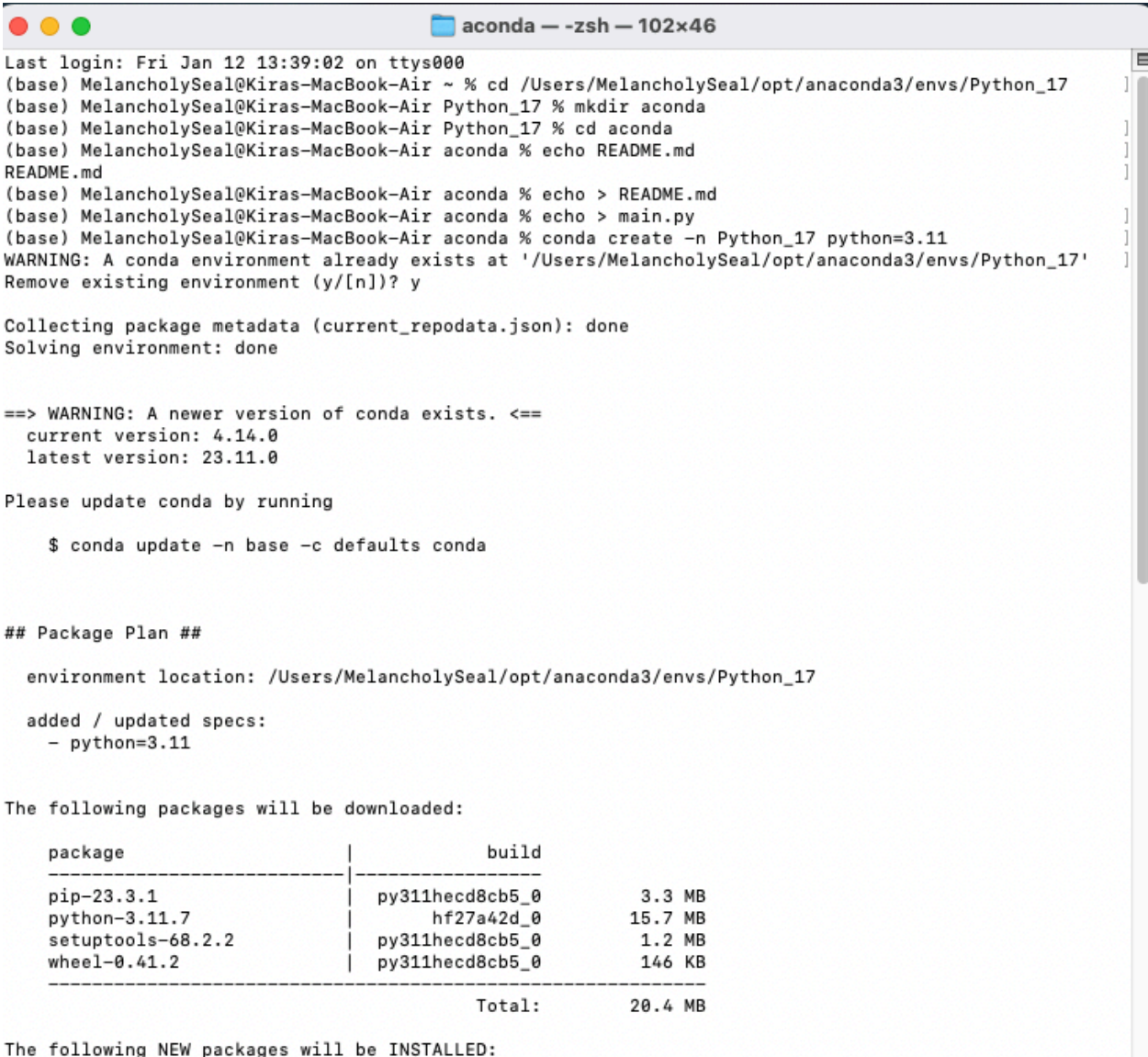
Ставрополь, 2023 г.

Тема: Установка пакетов в Python. Виртуальные окружения

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

Ход выполнения:

1) Создание и активация conda



```
aconda — zsh — 102x46
Last login: Fri Jan 12 13:39:02 on ttys000
(base) MelancholySeal@Kiras-MacBook-Air ~ % cd /Users/MelancholySeal/opt/anaconda3/envs/Python_17
(base) MelancholySeal@Kiras-MacBook-Air Python_17 % mkdir aconda
(base) MelancholySeal@Kiras-MacBook-Air Python_17 % cd aconda
(base) MelancholySeal@Kiras-MacBook-Air aconda % echo README.md
README.md
(base) MelancholySeal@Kiras-MacBook-Air aconda % echo > README.md
(base) MelancholySeal@Kiras-MacBook-Air aconda % echo > main.py
(base) MelancholySeal@Kiras-MacBook-Air aconda % conda create -n Python_17 python=3.11
WARNING: A conda environment already exists at '/Users/MelancholySeal/opt/anaconda3/envs/Python_17'
Remove existing environment (y/[n])? y

Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.14.0
  latest version: 23.11.0

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: /Users/MelancholySeal/opt/anaconda3/envs/Python_17

  added / updated specs:
    - python=3.11

The following packages will be downloaded:

  package | build | size
  -----|-----|-----
  pip-23.3.1 | py311hecd8cb5_0 | 3.3 MB
  python-3.11.7 | hf27a42d_0 | 15.7 MB
  setuptools-68.2.2 | py311hecd8cb5_0 | 1.2 MB
  wheel-0.41.2 | py311hecd8cb5_0 | 146 KB
  -----|-----|-----
  Total: | | 20.4 MB

The following NEW packages will be INSTALLED:
```

Рисунок 1 – создание и активация conda

2) Установил пакеты `pip`, `NumPy`, `Pandas`, `SciPy`

```
Retrieving notices: ...working... done
(base) MelancholySeal@Kiras-MacBook-Air aconda % conda install pip numpy pandas scipy
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.14.0
  latest version: 23.11.0

Please update conda by running

  $ conda update -n base -c defaults conda

# All requested packages already installed.

Retrieving notices: ...working... done
(base) MelancholySeal@Kiras-MacBook-Air aconda %
```

Рисунок 2 – Установка пакетов

3) Попробовал установить менеджером пакетов conda пакет TensorFlow. При этом возникла ошибка. Проблема заключается в конфликте версий Python при установке TensorFlow. Ваша версия Python в среде Conda - 3.11, и TensorFlow ищет совместимую версию в диапазоне от 3.5 до 3.10. Поскольку TensorFlow не поддерживает Python 3.11, возникает ошибка

```
(base) MelancholySeal@Kiras-MacBook-Air aconda % conda install TensorFlow
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: /
```

Рисунок 3 – Ошибка установки

4) Установил tensorflow с помощью pip install

```

(base) MelancholySeal@Kiras-MacBook-Air aconda % pip install tensorflow
Collecting tensorflow
  Downloading tensorflow-2.15.0-cp39-cp39-macosx_10_15_x86_64.whl (239.1 MB)
    |████████████████████████████████████████| 239.1 MB 27 kB/s
Collecting libclang<=13.0.0
  Downloading libclang-16.0.6-py2.py3-none-macosx_10_9_x86_64.whl (24.5 MB)
    |████████████████████████████████████████| 24.5 MB 9.6 MB/s
Collecting tensorboard<2.16,>=2.15
  Downloading tensorboard-2.15.1-py3-none-any.whl (5.5 MB)
    |████████████████████████████████████████| 5.5 MB 10.7 MB/s
Collecting ml-dtypes==0.2.0
  Downloading ml_dtypes-0.2.0-cp39-cp39-macosx_10_9_universal2.whl (1.2 MB)
    |████████████████████████████████████████| 1.2 MB 12.8 MB/s
Collecting tensorflow-io-gcs-filesystem>=0.23.1
  Downloading tensorflow_io_gcs_filesystem-0.35.0-cp39-cp39-macosx_10_14_x86_64.whl (1.7 MB)
    |████████████████████████████████████████| 1.7 MB 12.4 MB/s
Collecting termcolor>=1.1.0
  Downloading termcolor-2.4.0-py3-none-any.whl (7.7 kB)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /Users/MelancholySeal/opt/anaconda3/lib/python3.9/site-packages (from tensorflow) (1.42.0)
Collecting google-pasta>=0.1.1
  Downloading google_pasta-0.2.0-py3-none-any.whl (57 kB)
    |████████████████████████████████████████| 57 kB 11.1 MB/s
Collecting keras<2.16,>=2.15.0
  Downloading keras-2.15.0-py3-none-any.whl (1.7 MB)
    |████████████████████████████████████████| 1.7 MB 10.1 MB/s
Requirement already satisfied: setuptools in /Users/MelancholySeal/opt/anaconda3/lib/python3.9/site-packages (from tensorflow) (61.2.0)
Collecting gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1
  Downloading gast-0.5.4-py3-none-any.whl (19 kB)
Collecting tensorflow-estimator<2.16,>=2.15.0
  Downloading tensorflow_estimator-2.15.0-py2.py3-none-any.whl (441 kB)
    |████████████████████████████████████████| 441 kB 11.2 MB/s
Collecting astunparse>=1.6.0
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Collecting opt-einsum>=2.3.2
  Downloading opt_einsum-3.3.0-py3-none-any.whl (65 kB)
    |████████████████████████████████████████| 65 kB 6.9 MB/s
Requirement already satisfied: packaging in /Users/MelancholySeal/opt/anaconda3/lib/python3.9/site-packages (from tensorflow) (21.3)
Requirement already satisfied: six>=1.12.0 in /Users/MelancholySeal/opt/anaconda3/lib/python3.9/site-packages (from tensorflow) (1.16.0)
Collecting flatbuffers>=23.5.26
  Downloading flatbuffers-23.5.26-py2.py3-none-any.whl (26 kB)
Collecting absl-py>=1.0.0
  Downloading absl_py-2.0.0-py3-none-any.whl (130 kB)
    |████████████████████████████████████████| 130 kB 10.9 MB/s
Requirement already satisfied: h5py>=2.9.0 in /Users/MelancholySeal/opt/anaconda3/lib/python3.9/site-packages (from tensorflow) (3.6.0)

```

Рисунок 4 – Установка tensorflow

5) Сформировал файлы requirements.txt 1 environment.yml. В requirements.txt хранится список всех зависимостей проекта и их версий (установленных пакетов до выполнения команды). Environment.yml предназначен для определения и создания виртуальной среды, которая включает в себя не только зависимости Python (библиотеки), но и другие параметры окружения, такие как версия Python, пакеты системного уровня и т. Д.

```

(base) MelancholySeal@Kiras-MacBook-Air aconda % conda env export > enviroment.yml
(base) MelancholySeal@Kiras-MacBook-Air aconda % pip freeze > requirements.txt
(base) MelancholySeal@Kiras-MacBook-Air aconda %

```

Рисунок 5 – Создание файлов

```
1 hbsl-py==2.0.0
2 aiohttp @ file:///opt/concourse/worker/volumes/live/ba252660-e2b9-4a6d-6906-9be4ed7deda4/volume/aiohttp_1646806382603/work
3 aiosignal @ file:///tmp/build/80754af9/aiosignal_1637843061372/work
4 alabaster @ file:///home/ktietz/src/ci/alabaster_1611921544520/work
5 anaconda-client @ file:///opt/concourse/worker/volumes/live/866d4dd0-ffb5-4d0b-718d-0267a3b10e06/volume/anaconda-client_1635342573767/work
6 anaconda-navigator==2.3.0
7 anaconda-project @ file:///tmp/build/80754af9/anaconda-project_1637161053845/work
8 anyio @ file:///opt/concourse/worker/volumes/live/485b0f52-1188-482a-6285-65a36c8fa8a6/volume/anyio_1644481714856/work/dist
9 appdirs==1.4.4
10 applaunchservices @ file:///Users/ktietz/demo/mc3/conda-bld/applaunchservices_1630511705208/work
11 appnope @ file:///opt/concourse/worker/volumes/live/6ca6f098-d773-4461-5c91-a24a17435bda/volume/appnope_1606859448531/work
12 appscript @ file:///opt/concourse/worker/volumes/live/00049ed6-6263-4a6e-72b9-9d990f6e2f07/volume/appscript_1611427000595/work
13 argon2-cffi @ file:///opt/conda/conda-bld/argon2-cffi_1645000214183/work
14 argon2-cffi-bindings @ file:///opt/concourse/worker/volumes/live/42cf1b28-e71f-45ed-47b2-50f828088636/volume/argon2-cffi-bindings_1644569709119/work
15 arrow @ file:///opt/conda/conda-bld/arrow_1649166651673/work
16 astroid @ file:///opt/concourse/worker/volumes/live/5aff3c6b-d8ac-4e74-4846-0f446794397d/volume/astroid_1628063157520/work
17 astropy @ file:///private/var/folders/sy/f16zz6x50xz3113nwtb9bvq00000gp/T/abs_906836d0-ceb2-4914-9bb6-ba833df0befasz7z38b2/croots/recipe/astropy_1650891084492/work
18 asttokens @ file:///opt/conda/conda-bld/asttokens_1646925590279/work
19 astunparse==1.6.3
20 async-timeout @ file:///tmp/build/80754af9/async-timeout_1637851218186/work
21 atomicwrites==1.4.0
22 attrs @ file:///opt/conda/conda-bld/attrs_1642510447205/work
23 Automat @ file:///tmp/build/80754af9/automat_1600298431173/work
24 autopep8 @ file:///opt/conda/conda-bld/autopep8_1639166893812/work
25 Babel @ file:///tmp/build/80754af9/babel_1620871417480/work
26 backcall @ file:///home/ktietz/src/ci/backcall_1611930011877/work
27 backports.functiontools-lru-cache @ file:///tmp/build/80754af9/backports.functiontools_lru_cache_1618170165463/work
28 backports.tempfile @ file:///home/linux1/recipes/ci/backports.tempfile_1610991236607/work
29 backports.weakref==1.0.post1
30 bcrypt @ file:///opt/concourse/worker/volumes/live/7540a0c1-471d-4a61-5243-0bfec3601197/volume/bcrypt_1607022666156/work
31 beautifulsoup4 @ file:///private/var/folders/sy/f16zz6x50xz3113nwtb9bvq00000gp/T/abs_croot-cdiuinh5/beautifulsoup4_1650462164803/work
32 binaryornot @ file:///tmp/build/80754af9/binaryornot_1617751525010/work
33 bitarray @ file:///opt/concourse/worker/volumes/live/39912a06-1fdd-4d60-40a6-aad1836148be/volume/bitarray_1648739520150/work
34 bkcharts==0.2
35 black==19.10b0
36 bleach @ file:///opt/conda/conda-bld/bleach_1641577558959/work
37 bokeh @ file:///opt/concourse/worker/volumes/live/daea12c5-5a78-4471-406c-958ab70942ea/volume/bokeh_1638362839809/work
38 boto3 @ file:///opt/conda/conda-bld/boto3_1649078879353/work
39
```

Рисунок 6 – Содержимое файла requirements.txt

```
1 name: base
2 channels:
3   - defaults
4 dependencies:
5   - ipyw_jlab_nb_ext_conf=0.1.0=py39hcd8cb5_1
6   - aiohttp=3.8.1=py39hca72f7f_1
7   - aiosignal=1.2.0=pyhd3eb1b0_0
8   - alabaster=0.7.12=pyhd3eb1b0_0
9   - anaconda=2022.05=py39_0
10  - anaconda-client=1.9.0=py39hcd8cb5_0
11  - anaconda-navigator=2.3.0=py39hcd8cb5_0
12  - anaconda-project=0.10.2=pyhd3eb1b0_0
13  - anyio=3.5.0=py39hcd8cb5_0
14  - appdirs=1.4.4=pyhd3eb1b0_0
15  - applaunchservices=0.2.1=pyhd3eb1b0_0
16  - appnope=0.1.2=py39hcd8cb5_1001
17  - appscript=1.1.2=py39h9ed2024_0
18  - argon2-cffi=21.3.0=pyhd3eb1b0_0
19  - argon2-cffi-bindings=21.2.0=py39hca72f7f_0
20  - arrow=1.2.2=pyhd3eb1b0_0
21  - astroid=2.6.6=py39hcd8cb5_0
22  - astropy=5.0.4=py39h67323c0_0
23  - asttokens=2.0.5=pyhd3eb1b0_0
24  - async-timeout=4.0.1=pyhd3eb1b0_0
25  - atomicwrites=1.4.0=py_0
26  - attrs=21.4.0=pyhd3eb1b0_0
27  - automat=20.2.0=py_0
28  - autopep8=1.6.0=pyhd3eb1b0_0
29  - babel=2.9.1=pyhd3eb1b0_0
30  - backcall=0.2.0=pyhd3eb1b0_0
31  - backports=1.1=pyhd3eb1b0_0
32  - backports.functiontools_lru_cache=1.6.4=pyhd3eb1b0_0
33  - backports.tempfile=1.0=pyhd3eb1b0_1
34  - backports.weakref=1.0.post1=py_1
35  - bcrypt=3.2.0=py39h9ed2024_0
36  - beautifulsoup4=4.11.1=py39hcd8cb5_0
37  - binaryornot=0.4.4=pyhd3eb1b0_1
38  - bitarray=2.4.1=py39hca72f7f_0
39  - bkcharts=0.2=env39hcd8cb5_0
```

Рисунок 7 – Содержимое файла environment.yml

Ответы на вопросы:

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

Для установки пакета Python, не входящего в стандартную библиотеку, можно воспользоваться менеджером пакетов `pip`. Например, для установки пакета "requests" выполните команду: `pip install requests`

2. Как осуществить установку менеджера пакетов `pip`?

Для установки менеджера пакетов `pip`, обычно он устанавливается вместе с Python. Если он не установлен, можно воспользоваться инструкцией по установке `pip` для вашей операционной системы.

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

По умолчанию `pip` устанавливает пакеты из Python Package Index (PyPI), но также может устанавливать их из других источников, таких как Git репозитории.

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

Для установки последней версии пакета с помощью `pip`, используйте команду: `pip install --upgrade package_name`

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

Для установки заданной версии пакета с помощью `pip`, используйте команду: `pip install package_name==version_number`

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

Для установки пакета из git репозитория с помощью `pip`, используйте команду: `pip install git+https://github.com/username/repository.git`

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

Для установки пакета из локальной директории с помощью `pip`, используйте команду: `pip install /path/to/local/directory`

8. Как удалить установленный пакет с помощью `pip`?

Для удаления установленного пакета с помощью `pip`, используйте команду: `pip uninstall package_name`

9. Как обновить установленный пакет с помощью `pip`?

Для обновления установленного пакета с помощью `pip`, используйте команду: `pip install --upgrade package_name`

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

Для отображения списка установленных пакетов с помощью `pip`, используйте команду: `pip list`

11. Каковы причины появления виртуальных окружений в языке Python?

Виртуальные окружения в Python используются для изоляции проектов и их зависимостей, чтобы избежать конфликтов между различными версиями пакетов.

12. Каковы основные этапы работы с виртуальными окружениями?

Основные этапы работы с виртуальными окружениями включают создание, активацию, деактивацию и удаление виртуальных окружений.

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

Для работы с виртуальными окружениями с помощью `venv`, используйте стандартную библиотеку Python для создания и управления виртуальными окружениями.

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

`Virtualenv` предоставляет инструменты для создания изолированных виртуальных окружений Python.

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

`Pipenv` предоставляет удобный способ управления зависимостями и виртуальными окружениями для проектов Python.

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

Файл `requirements.txt` используется для хранения списка зависимостей проекта, что позволяет легко установить их на другой системе. Файл создается 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 Python-пакетами, но и библиотеками, написанными на других языках. Он также 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?

Для создания виртуального окружения с помощью conda используется команда `conda create --name myenv` для создания нового окружения с именем "myenv".

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

Для активации виртуального окружения conda используйте команду `conda activate myenv`, а для установки пакетов в это окружение используйте `conda install package_name`.

21. Как деактивировать и удалить виртуальное окружение conda?

Для деактивации виртуального окружения conda используйте команду `conda deactivate`, а для удаления окружения используйте `conda remove --name myenv --all`.

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

Файл `environment.yml` используется для описания окружения conda, включая список пакетов и их версий. Этот файл можно создать вручную, указав необходимые пакеты и их версии, или сгенерировать автоматически с помощью команды `conda env export > environment.yml`.

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

Для создания виртуального окружения conda с использованием файла `environment.yml`, выполните команду `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. Для этого необходимо установить плагин Conda, после чего можно создавать, активировать и управлять виртуальными окружениями через интерфейс PyCharm.

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

Файлы `requirements.txt` и `environment.yml` должны храниться в репозитории git, 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 работы были приобретены навыки по работе с менеджером пакетов `pip` и виртуальными окружениями с помощью языка программирования Python версии 3.x