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 №2.14
дисциплины
«Программирование на языке Python»

Выполнила:
Быковская Стефания Станиславовна
2 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные технологии и системы связи», очная форма обучения

_____
(подпись)

Проверил:

Воронкин Р. А., доцент кафедры инфокоммуникаций

_____
(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 работы:

Задание 1**.** Создала общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавила файл .gitignore с необходимыми правилами. Клонировала свой репозиторий на свой компьютер.

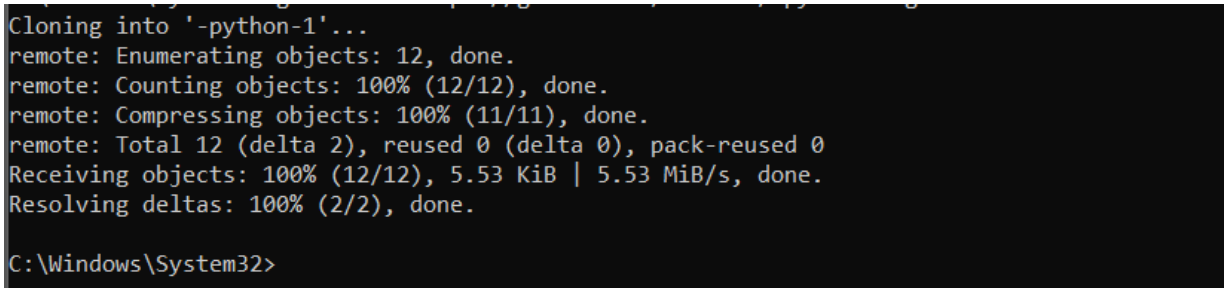


Рисунок 1. Клонирование репозитория

Задание 2. Организовала свой репозиторий в соответствие с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.

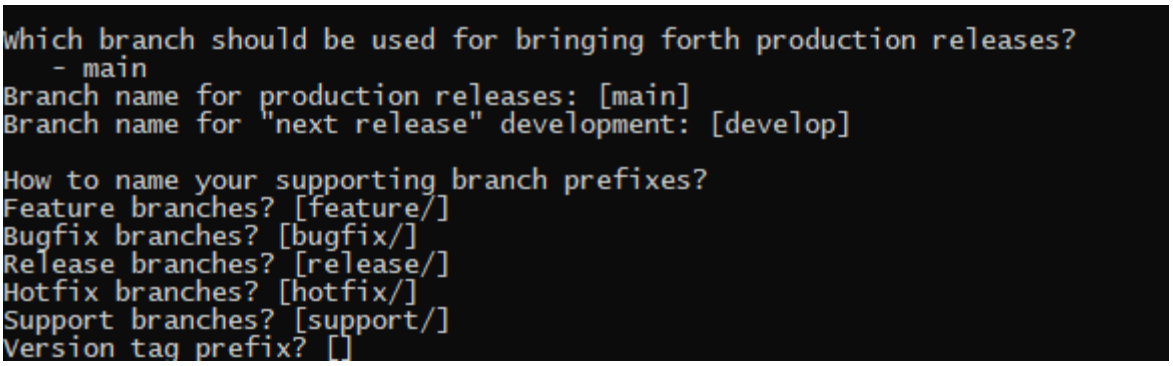


Рисунок 2. Модель ветвления git-flow

Задание 3. Создала виртуальное окружение Anaconda с именем репозитория.

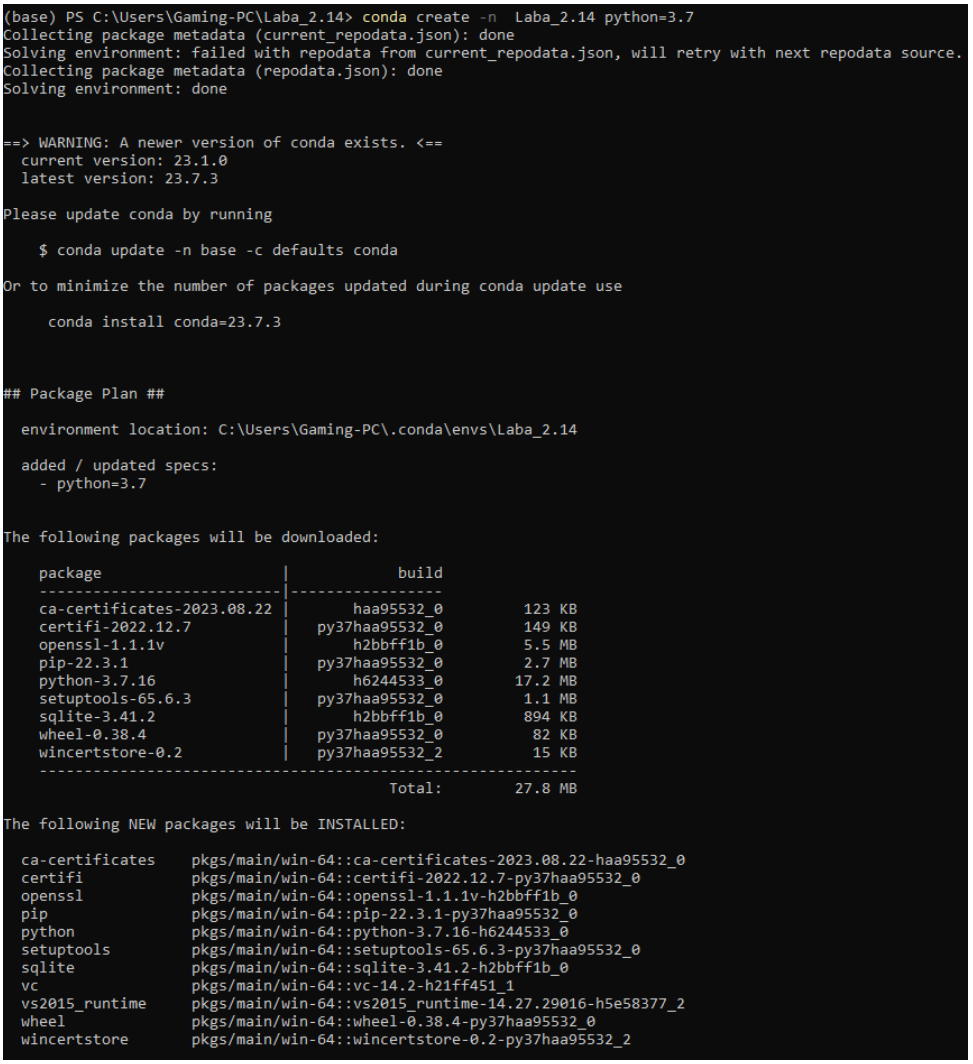


Рисунок 3. Виртуальное окружение

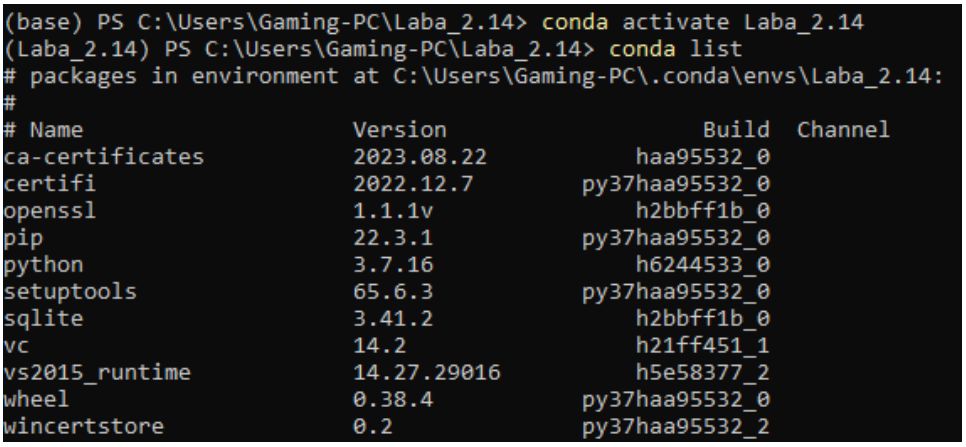


Рисунок 4. Активация окружения

Задание 4. Установила в виртуальное окружение следующие пакеты: pip, NumPy, Pandas, SciPy.

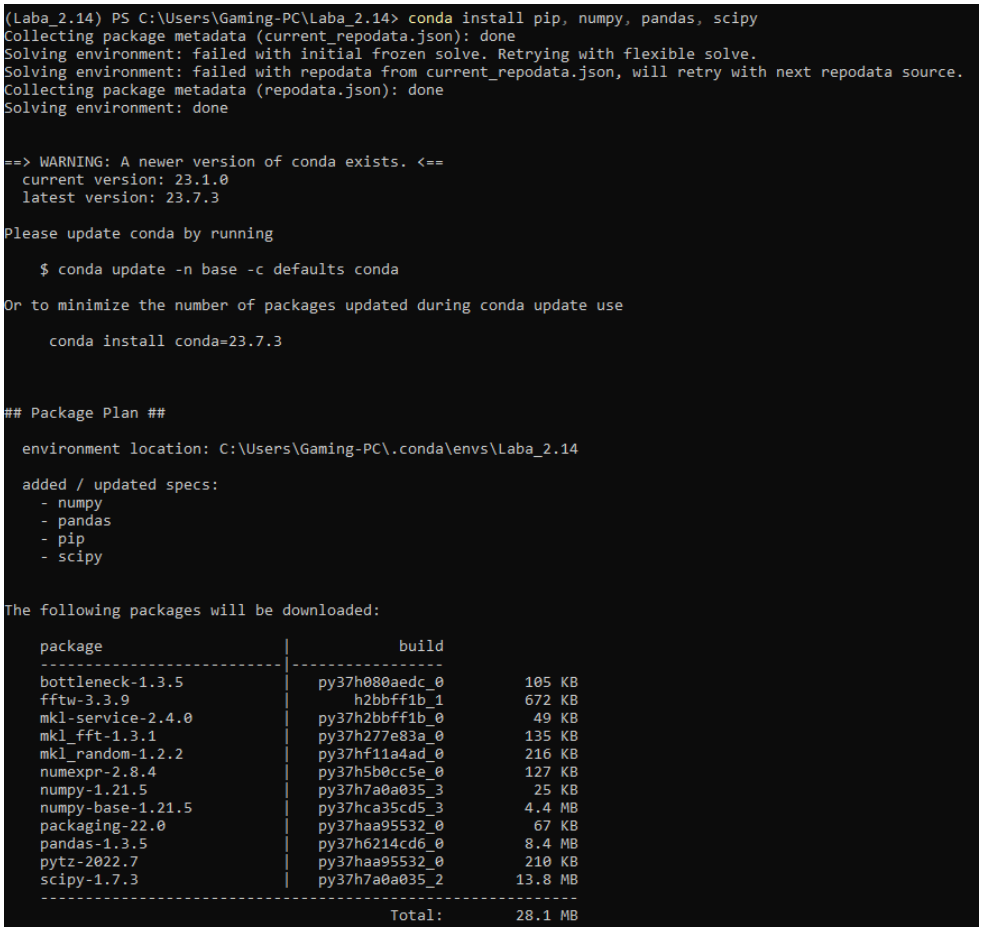


Рисунок 5. Установка пакетов

Задание 5. Установить менеджером пакетов conda пакет TensorFlow. В результате выдало предупреждение о разных версиях.

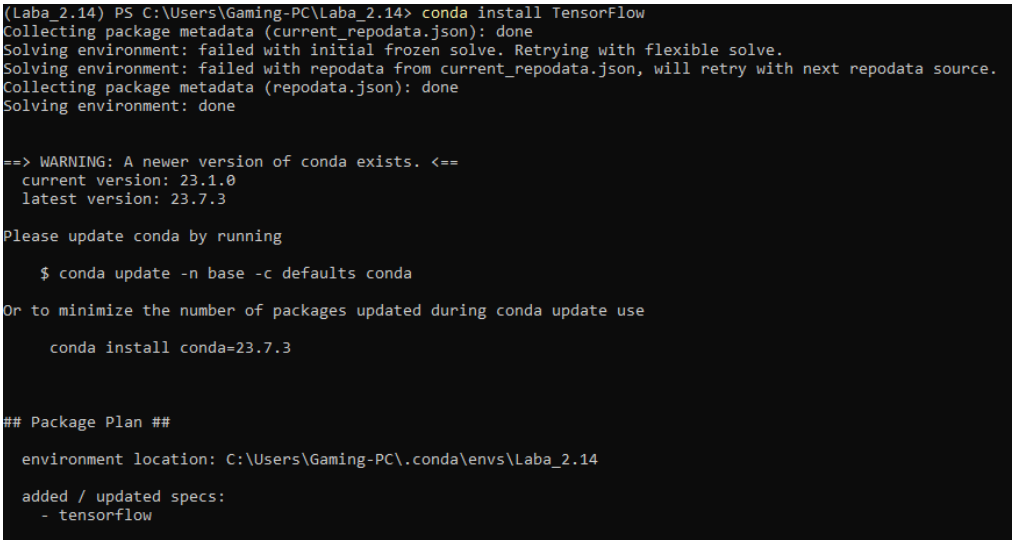


Рисунок 6. Установка TensorFlow через conda

После попытки установки пакета TensorFlow через conda, воспользовалась менеджером пакетов pip. После чего выполнила установку данного пакета.

Рисунок 7. Установка TensorFlow через pip

Задание 6. Сформировали файл environment.yml, определяющий окружение conda, включая название, каналы (источники пакетов) и список зависимостей. Он позволяет указать не только Python-пакеты, но и другие зависимости (например, библиотеки, необходимые для компиляции), а также версию Python.

C: > Users > HP > Documents > ИТС-б-о-22-1 > Программирование Pyth

```yaml
name: Lab_2.14
channels:
  - defaults
dependencies:
  - blas=1.0=mkl
  - bottleneck=1.3.5=py311h5bb9823_0
  - bzip2=1.0.8=he774522_0
  - ca-certificates=2023.08.22=haa95532_0
  - icc_rt=2022.1.0=h6049295_2
  - intel-openmp=2023.1.0=h59b6b97_46319
  - libffi=3.4.4=hd77b12b_0
  - mkl=2023.1.0=h6b88ed4_46357
  - mkl-service=2.4.0=py311h2bbff1b_1
  - mkl_fft=1.3.6=py311hf62ec03_1
  - mkl_random=1.2.2=py311hf62ec03_1
  - numexpr=2.8.4=py311h1fcbade_1
  - numpy=1.25.2=py311hdab7c0b_0
  - numpy-base=1.25.2=py311hd01c5d8_0
  - openssl=3.0.10=h2bbff1b_2
  - pandas=2.0.3=py311hf62ec03_0
  - pip=23.2.1=py311haa95532_0
  - python=3.11.4=he1021f5_0
  - python-dateutil=2.8.2=pyhd3eb1b0_0
  - python-tzdata=2023.3=pyhd3eb1b0_0
  - pytz=2022.7=py311haa95532_0
  - scipy=1.11.1=py311hc1ccb85_0
  - setuptools=68.0.0=py311haa95532_0
  - six=1.16.0=pyhd3eb1b0_1
  - sqlite=3.41.2=h2bbff1b_0
  - tbb=2021.8.0=h59b6b97_0
  - tk=8.6.12=h2bbff1b_0
  - tzdata=2023c=h04d1e81_0
  - vc=14.2=h21ff451_1
  - vs2015_runtime=14.27.29016=h5e58377_2
  - wheel=0.38.4=py311haa95532_0
```

Рисунок 8. Содержимое файла

Далее сформировала файл requirements.txt он содержит список зависимостей и их версий в текстовом формате. Каждая строка представляет собой один пакет с указанием версии или диапазона версий.

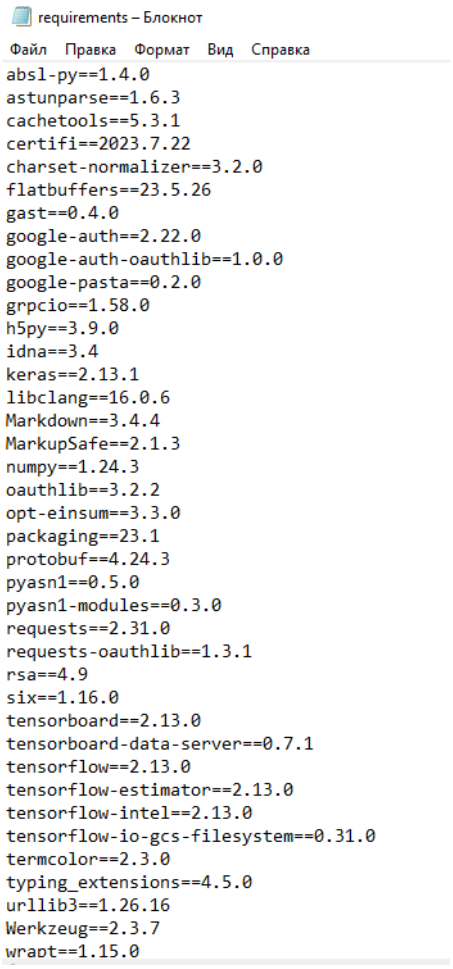
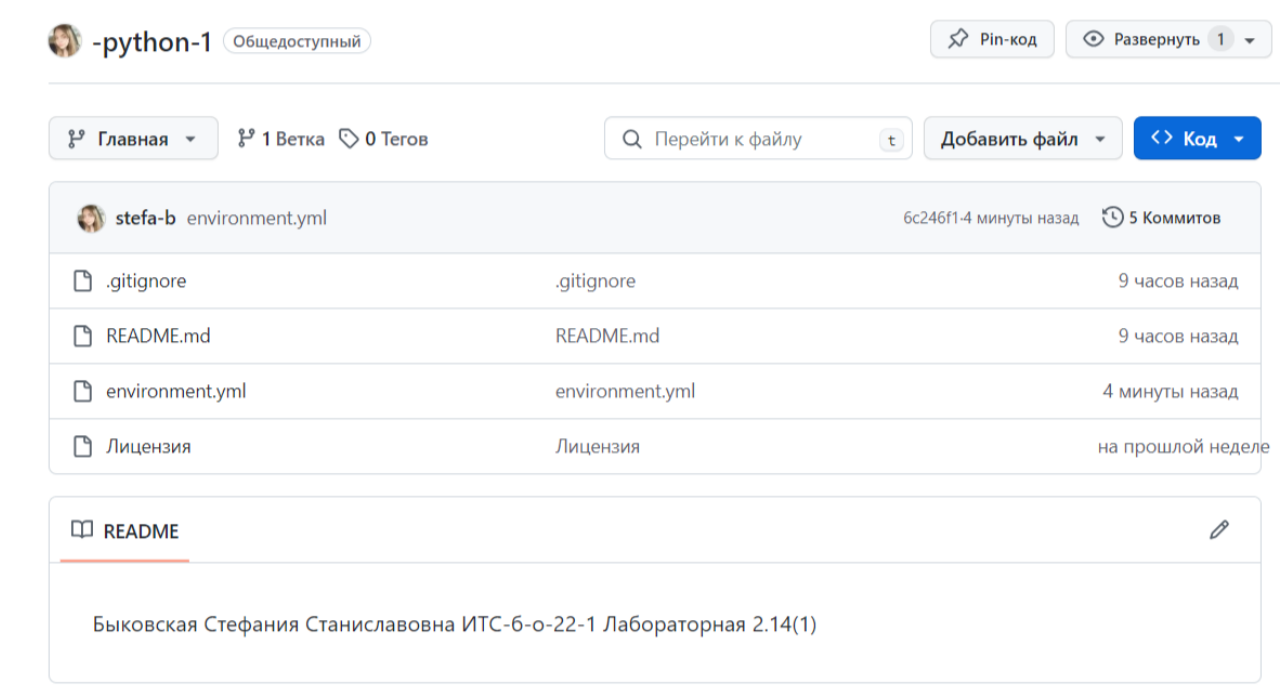


Рисунок 9. Содержимое файла requirements.txt

Задание 7. Закоммитила изменения и прикрепила на GitHub.



Ссылка: https://github.com/stefa-b/-python-1.git

Ответы на контрольные 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. Для этого нужно выполнить команду pip install package_name

2. Как осуществить установку менеджера пакетов pip?

Pip обычно устанавливается автоматически вместе с Python. Если он не установлен можно использовать команду python -m ensurepip --default-pip

3. Откуда менеджер пакетов pip по умолчанию устанавливает пакеты?

По умолчанию, pip устанавливает пакеты из Python Package Index (PyPI)

4. Как установить последнюю версию пакета с помощью pip?

Для установки последней версии пакета с помощью pip, нужно выполнить команду pip install --upgrade package_name

5. Как установить заданную версию пакета с помощью pip?

Для установки заданной версии пакета с помощью pip, нужно выполнить команду pip install package_name==desired_version

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

Для установки пакета из git репозитория, можно использовать команду pip install git+URL

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

Для установки пакета из локальной директории, нужно выполнить команду pip install /path/to/package_directory

8. Как удалить установленный пакет с помощью pip?

Чтобы удалить установленный пакет с помощью pip, нужно выполнить команду pip uninstall package_name

9. Как обновить установленный пакет с помощью pip?

Для обновления установленного пакета с помощью pip, нужно использовать команду pip install --upgrade package_name

10. Как отобразить список установленных пакетов с помощью pip?

Чтобы отобразить список установленных пакетов с помощью pip, нужно

выполнить команду pip list или pip freeze.

11. Каковы причины появления виртуальных окружений в языке Python?

Виртуальные окружения в Python используются для изоляции проектов и их зависимостей. Основные причины их использования включают изоляцию проектов друг от друга, управление версиями зависимостей и предотвращение конфликтов между пакетами.

12. Каковы основные этапы работы с виртуальными окружениями?

1. Создание виртуального окружения.
2. Активация виртуального окружения.
3. Установка необходимых пакетов в виртуальное окружение.
4. Работа с проектом в активированном виртуальном окружении.
5. Деактивация виртуального окружения (по завершении работы).

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

1) Создание виртуального окружения: python -m venv myenv
2) Активация виртуального окружения
3) Установка пакетов: pip install package_name
4) Деактивация: deactivate

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

Работа с виртуальными окружениями с помощью virtualenv аналогична venv. Создаем окружение с помощью virtualenv myenv, активируем его, устанавливаем пакеты и деактивируем.

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

1) Создание виртуального окружения: pipenv --python 3.8 (здесь указывается версия Python)
2) Установка пакетов: pipenv install package_name
3) Запуск оболочки в виртуальном окружении: pipenv shell

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

Файл requirements.txt используется для указания списка зависимостей

проекта. Создать его можно с помощью команды pip freeze > requirements.txt, и он будет содержать список установленных пакетов и их версий.

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

Преимущества conda по сравнению с pip включают в себя возможность установки бинарных зависимостей, управление средами и зависимостями, а также поддержку многих языков программирования, не только Python.

18. В какие дистрибутивы Python входит пакетный менеджер conda?

Пакетный менеджер conda входит в дистрибутивы Anaconda и Miniconda. Он также может быть установлен отдельно на другие дистрибутивы Python.

19. Как создать виртуальное окружение conda?

Для создания виртуального окружения conda нужно использовать команду conda create --name myenv

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

Для активации и установки пакетов в виртуальное окружение conda, нужно выполнить команду conda activate myenv, а затем команду conda install package_name для установки пакетов.

21. Как деактивировать и удалить виртуальное окружение conda?

Чтобы деактивировать и удалить виртуальное окружение conda, нужно выполнить команду conda deactivate для деактивации, а затем conda env remove --name myenv для удаления.

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

Файл environment.yml используется для определения окружения conda, включая зависимости. Его можно создать вручную, указав пакеты и их версии, или 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 помощью файла environment.yml, нужно использовать 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 среде PyCharm можно создавать и управлять виртуальными окружениями conda через интерфейс IDE. Это делается через "File" -> "Settings" -> "Project: <your project>" -> "Python Interpreter

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

Файлы requirements.txt и environment.yml содержат информацию о зависимостях проекта и их версиях. Хранение их в репозитории Git позволяет другим разработчикам воссоздать окружение проекта и установить необходимые зависимости для работы с кодом.

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