

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»
Вариант 5.

Выполнила:
Михеева Елена Александровна
2 курс, группа ИВТ-б-з-20-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: 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. Было создано окружение Anaconda с именем репозитория.

```
(base) MacBook-Pro-Elena:Python_2.14 mikheeva$ conda create -n Python17
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 23.11.0

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=23.11.0

## Package Plan ##

  environment location: /Users/mikheeva/anaconda3/envs/Python17

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate Python17
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) MacBook-Pro-Elena:Python_2.14 mikheeva$
```

Рисунок 1. Установка окружения

2. Установили в виртуальное окружение пакеты `pip`, `NumPy`, `Pandas`, `SciPy`.

```
(Python17) MacBook-Pro-Elena:Python_2.14 mikheeva$ conda install numpy pandas scipy
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 23.11.0

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=23.11.0

## Package Plan ##

environment location: /Users/mikheeva/anaconda3/envs/Python17

added / updated specs:
- numpy
- pandas
- scipy

The following packages will be downloaded:
```

package	build	
blas-1.0	mk1	6 KB
bottleneck-1.3.5	py312h32608ca_0	136 KB
ca-certificates-2023.12.12	hecd8cb5_0	127 KB
expat-2.5.0	hcec6c5f_0	140 KB
intel-openmp-2023.1.0	ha357a0b_43548	638 KB
mk1-2023.1.0	h8e150cf_43560	181.0 MB
mk1-service-2.4.0	py312h6c40b1e_1	53 KB
mk1_fft-1.3.8	py312h6c40b1e_0	182 KB
mk1_random-1.2.4	py312ha357a0b_0	230 KB
numexpr-2.8.7	py312hac873b0_0	143 KB
numpy-1.26.3	py312hac873b0_0	11 KB
numpy-base-1.26.3	py312h6f81483_0	7.3 MB
pandas-2.1.4	py312hc282a81_0	14.0 MB

Рисунок 2. Загрузка дополнительных пакетов

3. Установили менеджером пакетов conda пакет TensorFlow

```
(Python17) MacBook-Pro-Elena:Python_2.14 mikheeva$ 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='2.7.*|3.10.*|3.9.*|3.8.*|3.11.*|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.

(Python17) MacBook-Pro-Elena:Python_2.14 mikheeva$
```

Рисунок 3. Неудачная установка TensorFlow

Из-за несовместимости версий решение было заморожено.

```

tf_low-select                pkgs/main/osx-64:tf_low-select-2.0.0-eigen
libssl-cpp                  pkgs/main/osx-64:libssl-cpp-28211182.0-h9ed5dce_0
absl-py                      pkgs/main/osx-64:absl-py-1.4.0-py39hcd8cb5_0
aiohttp                      pkgs/main/osx-64:aiohttp-3.9.0-py39hc6a0b1e_0
aiosignal                   pkgs/main/noarch:aiosignal-1.2.0-pyhd3eb1b0_0
astunparse                  pkgs/main/noarch:astunparse-1.6.3-py_0
async-timeout               pkgs/main/osx-64:async-timeout-4.0.3-py39hcd8cb5_0
attrs                        pkgs/main/osx-64:attrs-23.0.0-py39hcd8cb5_0
blinker                      pkgs/main/osx-64:blinker-1.6.2-py39hcd8cb5_0
brotli-python               pkgs/main/osx-64:brotli-python-1.0.9-py39h9d5dce_7
c-ares                       pkgs/main/osx-64:c-ares-1.19.1-h6c40b1e_0
cachetools                  pkgs/main/noarch:cachetools-4.2.2-pyhd3eb1b0_0
certifi                      pkgs/main/osx-64:certifi-2023.11.17-py39hcd8cb5_0
cffi                         pkgs/main/osx-64:cffi-1.16.0-py39hcd8cb5_0
charset-normalizer           pkgs/main/noarch:charset-normalizer-2.0.4-pyhd3eb1b0_0
click                        pkgs/main/osx-64:click-8.1.7-py39hcd8cb5_0
cryptography                 pkgs/main/osx-64:cryptography-41.0.3-py39h72381d6_0
flatbuffers                  pkgs/main/osx-64:flatbuffers-2.0.0-h23ba428_0
frozenlist                   pkgs/main/osx-64:frozenlist-1.4.0-py39hc6a0b1e_0
gast                         pkgs/main/noarch:gast-0.4.0-pyhd3eb1b0_0
gitdb                        pkgs/main/osx-64:gitdb-4.0.10-py39hcd8cb5_0
google-auth                  pkgs/main/osx-64:google-auth-2.22.0-py39hcd8cb5_0
google-auth-oauth2           pkgs/main/noarch:google-auth-oauth2-lib-0.4.4-pyhd3eb1b0_0
google-pasta                 pkgs/main/noarch:google-pasta-0.2.0-pyhd3eb1b0_0
grpc-cpp                     pkgs/main/osx-64:grpc-cpp-1.46.1-h64d96ca_1
grpcio                       pkgs/main/osx-64:grpcio-1.42.0-py39h29bfda_0
h5py                          pkgs/main/osx-64:h5py-3.9.0-py39hf9d4033_0
hubs                          pkgs/main/osx-64:hubs-0.12.0-h29ad972_0
icu                           pkgs/main/osx-64:icu-68.1-h23ba428_0
idna                          pkgs/main/osx-64:idna-3.4-py39hcd8cb5_0
importlib-metadata           pkgs/main/osx-64:importlib-metadata-7.0.1-py39hcd8cb5_0
jpeg                          pkgs/main/osx-64:jpeg-9e-h6c40b1e_1
keras                         pkgs/main/osx-64:keras-2.11.0-py39_0
keras-preprocessing          pkgs/main/noarch:keras-preprocessing-1.1.2-pyhd3eb1b0_0
krb5                          pkgs/main/osx-64:krb5-1.20.1-hbda5381_0
libcurl                       pkgs/main/osx-64:libcurl-8.1.4-h855b31_0
libid3                        pkgs/main/osx-64:libid3-3.1.2023828-h6c40b1e_0
libev                         pkgs/main/osx-64:libev-4.33-h9fed24_1
libnghttp2                   pkgs/main/osx-64:libnghttp2-1.52.0-h8c88b7d_1
libpng                        pkgs/main/osx-64:libpng-1.6.39-h6c40b1e_0
libprotobuf                  pkgs/main/osx-64:libprotobuf-3.20.3-hfff2838_0
libzlib                       pkgs/main/osx-64:libzlib-1.2.13-hd39d1d2_0
markdown                     pkgs/main/osx-64:markdown-3.4.1-py39hcd8cb5_0
markupsafe                   pkgs/main/osx-64:markupsafe-2.1.3-py39hc6a0b1e_0
multidict                    pkgs/main/osx-64:multidict-6.0.4-py39hc6a0b1e_0
oauthlib                     pkgs/main/osx-64:oauthlib-3.2.2-py39hcd8cb5_0
opt_einsum                   pkgs/main/noarch:opt_einsum-3.3.0-pyhd3eb1b0_1
packaging                     pkgs/main/osx-64:packaging-23.1-py39hcd8cb5_0
protobuf                     pkgs/main/osx-64:protobuf-3.9.0-py39hcd8cb5_0
psutil                       pkgs/main/noarch:psutil-0.8.2-py39hcd8cb5_0
pyasn1-modules               pkgs/main/noarch:pyasn1-modules-0.2.8-py_0

```

Просмотр

Рисунок 3. Удачная установка TensorFlow

4. Создали два файла: requirements.txt – хранит все пакеты, environment.yml – параметры окружения.

```

1 requirements.txt
2 1 whn1-py @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_8By0u639n/croot/abs1-py_1688852452492/work
3 asintox @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_b5c1b1b/croot/asintox_1711112611818/work
4 asynsploit @ file:///tmp/build/88754f9/asynsploit_16378463377/work
5 astunemem-0.3
6 asynce @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_9f5d5wdw/croot/asynce_1748390733766/work
7 attr @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_8d7c1t9j/croot/attr_1695717834289/work
8 blinker @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_80du6dajag/croot/blinker_1605307675331/work
9 bottleneck @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_9c51am6d/croot/bottleneck_16371566485/work
10 boto3 @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_81d5u8dc/croot/boto3_169516663272/work
11 Brotli @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_449c4myt/croot/cffi_1780254431144/work
12 cachetools @ file:///tmp/build/88754f9/cachetools_1610597188817/work
13 certifi @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_3c0t9u3w/croot/certifi_170851615943/work/certifi
14 cffi @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_449c4myt/croot/cffi_1780254431144/work
15 charset-normalizer @ file:///tmp/build/88754f9/charset-normalizer_16388327954/work
16 click @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_2763b3/croot/click_168812816333/work
17 cryptography @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_19uc_33nq/croot/cryptography_166421354768/work
18 cycler @ file:///tmp/build/88754f9/cycler_16348912618/work
19 cvxopt @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_6e532xx3/croot/frozennet_164785761576/work
20 cymem @ file:///Users/alexiz/dmcc/conda-bld/cymem_16285888823/work
21 google-auth @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_gq3/croot/google-auth_1094512712814/work
22 google-auth-authlib @ file:///tmp/build/88754f9/google-auth-authlib_161726569481/work
23 google-pasta @ file:///Users/alexiz/dmcc/conda-bld/google-pasta_16357799134/work
24 grpcio @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_3d6p3e/croot/grpcio_16372312888/work
25 http @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_45ad6p/croot/http_1691589712322/work
26 idna @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_40f0mtr/croot/idna_1666125173144/work
27 jenkins-client @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_13c0ed/croot/importlib_metadata-suite_17048133105/work
28 keras @ file:///tmp/local/conda-bld/keras_16794518104/keras-2.1.10-py3-none-any.whl
29 Keras-Preprocessing @ file:///tmp/build/88754f9/keras-preprocessing_16122363496/work
30 libcmap-4.4
31 Markdown @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_906_p1p/croot/markdown_1671541313695/work
32 MarkupSafe @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_3a_c4d/croot/MarkupSafe_17424952951/work
33 mfr @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_8f643n/croot/mfr_161958181833/work
34 ml-random @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_dq2848d/croot/ml-random_1695858852526/work
35 ml-service-2.4.0
36 multiselect @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_b20uvc/croot/multiselect_1781878656495/work
37 numpy @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_3c2oer/croot/numpy_16951530457/work
38 opencv-python @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_8f643n/croot/opencv-python_1695858852526/work
39 oauthlib @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_81q63w/croot/oauthlib_16749304531/work
40 opt-einsum @ file:///tmp/build/88754f9/opt-einsum_16215882389/work
41 optuna @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_2u3g9p/croot/packaging_16915717883/work
42 pandas @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_8Bpc3ic/croot/pandas_178231794061/work/dist/pandas-2.1.4-cp39-cp39-macosx_10_9_arm_64.whl#sha256:64618db27267c787858660228894928869432ca6c18049449492a62
43 parso @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_81q63w/croot/parso_16915717883/work
44 pyasn1 @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_81q63w/croot/pyasn1_16915717883/work
45 pyasn1-modules-0.2.0
46 pyinstaller @ file:///tmp/build/88754f9/pyinstaller_16954132334/work
47 PyM @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_4e748d-fc8-4a86-1bca-1316111bdc7dca/croot/recipe.pyv_16574458958/work
48 pyopenssl @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_7u6dJ84B1/croot/pyopenssl_16892348883/work
49 Pysock @ file:///opt/concourse/workers/uv/L109112288-8C8-4473-786c-f4d41a/volume/psycos_16026341084/work
50 python-dateutil @ file:///tmp/build/88754f9/python-dateutil_16531744494/work
51 pytz @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_4y3l2hd/croot/pytz_16915717883/work
52 requests @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_81q63w/croot/requests_169484872395/work
53 requests-auth-1.0.0
54 rsa @ file:///tmp/build/88754f9/rsa_16454326049/work
55 scikit-learn @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_T614t4T2/croot/scipy_1781280491576/work/dist/scipy-1.11.4-cp39-cp39-macosx_10_9_arm_64.whl#sha256:5c149a3b3a187867a5156b6d44c4c4ec90732a4772a5f7889337481
56 tensorflow @ file:///tmp/local/conda-bld/tensorflow_17479511979/work/tensorflow-2.11.0-py3-none-any.whl
57 tensorflow-data-server @ file:///tmp/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_83n1l/croot/tensorflow-data-server_16785359314/work/tensorflow_data_server-0.6.1-py3-none-macosx_10_9_arm_64.whl
58 tensorflow-plugin-utils-0.6.0
59 tensorflow @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_7u6dJ84B1/croot/tensorflow-plugin-utils-2.11.0-cp39-cp39-macosx_10_9_arm_64.whl
60 tensorflow-tensorflow @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_7u6dJ84B1/croot/tensorflow-tensorflow-2.11.0-cp39-cp39-macosx_10_9_arm_64.whl
61 tensorflowio @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_7u6dJ84B1/croot/tensorflowio-2.11.0-cp39-cp39-macosx_10_9_arm_64.whl
62 termcolor @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_7u6dJ84B1/croot/termcolor_168944847901/work
63 testing_extensions @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_81q63w/croot/testing_extensions_17885637480/work
64 tzdata @ file:///croot/pythn-tzdata_1696112552/work
65 urllib @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_82wv72/croot/urllib_169025794533/work
66 Werkzeug @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_81q63w/croot/Werkzeug_1678499762631/work
67 wrapt @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_1d61168-354-408-83b-37817281577/work/croot/recipeurl_1657814447132/work
68 yarl @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_81q63w/croot/yarl_16915717883/work
69 zip @ file:///private/var/folders/sy/f16z6s0x2311mrb0vq00000g/7/abs_d32923/croot/zip_178628091176/work

```

Рисунок 4. Файл requirements.txt

```
(Python_2.14) MacBook-Pro-Elena:Python_2.14 mikheeva$ conda env export > environment.yml
(Python_2.14) MacBook-Pro-Elena:Python_2.14 mikheeva$ pip freeze > requirements.txt
(Python_2.14) MacBook-Pro-Elena:Python_2.14 mikheeva$ conda deactivate
(base) MacBook-Pro-Elena:Python_2.14 mikheeva$ mkdir doc
```

Рисунок 4. Создание файлов

```

1 name: Python_2.14
2 channels:
3   - defaults
4 dependencies:
5   - _tflow_select=2.2.0=eigen
6   - absl-cpp=20211102.0=he9d5cce_0
7   - absl-py=1.4.0=py39hcd8cb5_0
8   - aiohttp=3.9.0=py39h6c40b1e_0
9   - aiosignal=1.2.0=pyhd3eb1b0_0
10  - astunparse=1.6.3=py_0
11  - async-timeout=4.0.3=py39hcd8cb5_0
12  - attrs=23.1.0=py39hcd8cb5_0
13  - blas=1.0=mkl
14  - blinker=1.6.2=py39hcd8cb5_0
15  - bottleneck=1.3.5=py39h67323c0_0
16  - brotli-python=1.0.9=py39he9d5cce_7
17  - c-ares=1.19.1=h6c40b1e_0
18  - ca-certificates=2.023.12.12=hecd8cb5_0
19  - cachetools=4.2.2=pyhd3eb1b0_0
20  - certifi=2023.11.17=py39hcd8cb5_0
21  - cffi=1.16.0=py39h6c40b1e_0
22  - charset-normalizer=2.0.4=pyhd3eb1b0_0
23  - click=8.1.7=py39hcd8cb5_0
24  - cryptography=41.0.3=py39ha2381d6_0
25  - flatbuffers=2.0.0=h23ab428_0
26  - frozenlist=1.4.0=py39h6c40b1e_0
27  - gast=0.4.0=pyhd3eb1b0_0
28  - giflib=5.2.1=h6c40b1e_3
29  - google-auth=2.22.0=py39hcd8cb5_0
30  - google-auth-oauthlib=0.4.4=pyhd3eb1b0_0
31  - google-pasta=0.2.0=pyhd3eb1b0_0
32  - grpc-cpp=1.46.1=h64d96ca_1
33  - grpcio=1.42.0=py39ha29bfda_0
34  - h5py=3.9.0=py39hf9d4033_0
35  - hdf5=1.12.1=h2b2ad87_2
36  - icu=68.1=h23ab428_0
37  - idna=3.4=py39hcd8cb5_0
38  - importlib-metadata=7.0.1=py39hcd8cb5_0
39  - intel-openmp=2023.1.0=ha357a0b_43548
40  - jpeg=9e=h6c40b1e_1
41  - keras=2.11.0=py39_0
42  - keras-preprocessing=1.1.2=pyhd3eb1b0_0
43  - krb5=1.20.1=hdba6334_1
44  - libcurl=8.2.1=ha585b31_0
45  - libcxx=14.0.6=h9765a3e_0
46  - libedit=3.1.20230828=h6c40b1e_0
47  - libev=4.33=h9ed2024_1
48  - libffi=3.4.4=hecd8cb5_0
49  - libgfortran=5.0.0=11_3_0_hcd8cb5_28
50  - libgfortran5=11.3.0=h9dfd629_28
51  - libnghttp2=1.52.0=h1c88b7d_1
52  - libpng=1.6.39=h6c40b1e_0
53  - libprotobuf=3.20.3=hfff2838_0
54  - libssh2=1.10.0=hdb2fb19_2
55  - llvm-openmp=14.0.6=h0dcd299_0
56  - markdown=3.4.1=py39hcd8cb5_0
57  - markupsafe=2.1.3=py39h6c40b1e_0
58  - mkl=2023.1.0=h8e150cf_43560
59  - mkl-service=2.4.0=py39h6c40b1e_1
60  - mkl_fft=1.3.8=py39h6c40b1e_0
61  - mkl_random=1.2.4=py39ha357a0b_0
62  - multidict=6.0.4=py39h6c40b1e_0
63  - ncurses=6.4=hcec6c5f_0
56  - markdown=3.4.1=py39hcd8cb5_0
57  - markupsafe=2.1.3=py39h6c40b1e_0
58  - mkl=2023.1.0=h8e150cf_43560
59  - mkl-service=2.4.0=py39h6c40b1e_1
60  - mkl_fft=1.3.8=py39h6c40b1e_0
61  - mkl_random=1.2.4=py39ha357a0b_0
62  - multidict=6.0.4=py39h6c40b1e_0
63  - ncurses=6.4=hcec6c5f_0
64  - numexpr=2.8.7=py39h827a554_0
65  - numpy=1.26.3=py39h827a554_0
66  - numpy-base=1.26.3=py39ha186be2_0
67  - oauthlib=3.2.2=py39hcd8cb5_0
68  - openssl=1.1.1w=hca72f7f_0
69  - opt_einsum=3.3.0=pyhd3eb1b0_1
70  - packaging=23.1=py39hcd8cb5_0
71  - pandas=2.1.4=py39h3ea8b11_0
72  - pip=23.3.1=py39hcd8cb5_0
73  - pyasn1=0.4.8=pyhd3eb1b0_0
74  - pyasn1-modules=0.2.8=py_0
75  - pycparser=2.21=pyhd3eb1b0_0
76  - pyjwt=2.4.0=py39hcd8cb5_0
77  - pyopenssl=23.2.0=py39hcd8cb5_0
78  - pysocks=1.7.1=py39hcd8cb5_0
79  - python=3.9.18=h218abb5_0
80  - python-dateutil=2.8.2=pyhd3eb1b0_0
81  - python-flatbuffers=2.0=pyhd3eb1b0_0
82  - python-tzdata=2023.3=pyhd3eb1b0_0
83  - pytz=2023.3.post1=py39hcd8cb5_0
84  - re2=2022.04.01=h9d5cce_0
85  - readline=8.2=hca72f7f_0
86  - requests=2.31.0=py39hcd8cb5_0
87  - requests-oauthlib=1.3.0=py_0
88  - rsa=4.7.2=pyhd3eb1b0_1
89  - scipy=1.11.4=py39hdb2ea58_0
90  - setuptools=68.2.2=py39hcd8cb5_0
91  - six=1.16.0=pyhd3eb1b0_1
92  - snappy=1.1.10=hcec6c5f_1
93  - sqlite=3.41.2=h6c40b1e_0
94  - tbb=2021.8.0=ha357a0b_0
95  - tensorboard=2.11.0=py39_0
96  - tensorboard-data-server=0.6.1=py39h7242b5c_0
97  - tensorboard-plugin-wit=1.6.0=py_0
98  - tensorflow=2.11.0=eigen_py39hcf9afec_0
99  - tensorflow-base=2.11.0=eigen_py39hbf87084_0
100 - tensorflow-estimator=2.11.0=py39_0
101 - termcolor=2.1.0=py39hcd8cb5_0
102 - tk=8.6.12=h5d9f67b_0
103 - typing_extensions=4.9.0=py39hcd8cb5_0
104 - tzdata=2023d=h04d1e81_0
105 - urllib3=1.26.18=py39hcd8cb5_0
106 - werkzeug=2.2.3=py39hcd8cb5_0
107 - wheel=0.35.1=pyhd3eb1b0_0
108 - wrapt=1.14.1=py39hca72f7f_0
109 - xz=5.4.5=h6c40b1e_0
110 - yarl=1.9.3=py39h6c40b1e_0
111 - zipp=3.17.0=py39hcd8cb5_0
112 - zlib=1.2.13=h4dc903c_0
113 - pip:
114   - libclang==16.0.6
115   - protobuf==3.19.6
116   - tensorflow-io-gcs-filesystem==0.35.0
117 prefix: /Users/mikheeva/anaconda3/envs/Python_2.14
118

```

Рисунок 5. Файл environment.yml

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. 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.