

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №2.14 по дисциплине:
основы программной инженерии**

Выполнила:

студент группы ПИЖ-б-о-20-1

Лазарева Дарья Олеговна

Проверил:

доцент кафедры инфокоммуникаций

Романкин Р.А.

Ставрополь, 2022 г.

1. Установка пакетов в Python

```
(base) C:\Users\79616>conda create -n laba2
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.3
  latest version: 4.11.0

Please update conda by running

    $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\79616\anaconda3\envs\laba2

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate laba2
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:\Users\79616>
```

2. Активация виртуального окружения

```
(base) C:\Users\79616>activate laba2

(laba2) C:\Users\79616>
```

3. Установка пакетов pip

```
(lab2a) C:\Users\79616>conda install -n lab2a pip
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.3
  latest version: 4.11.0

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\79616\anaconda3\envs\lab2a

  added / updated specs:
    - pip

The following packages will be downloaded:

  package | build | size
  -----|-----|-----
  ca-certificates-2022.2.1 | haa95532_0 | 123 KB
  certifi-2021.10.8 | py39haa95532_2 | 152 KB
  openssl-1.1.1m | h2bbff1b_0 | 4.8 MB
  pip-21.2.4 | py39haa95532_0 | 1.8 MB
  python-3.9.7 | h6244533_1 | 16.5 MB
  setuptools-58.0.4 | py39haa95532_0 | 778 KB
  sqlite-3.37.2 | h2bbff1b_0 | 799 KB
  tzdata-2021e | hda174b7_0 | 112 KB
  wheel-0.37.1 | pyhd3eb1b0_0 | 33 KB
  setuptools-58.0.4 | py39haa95532_0 | 15 KB
  -----|-----|-----
                                         Total: 25.1 MB

The following NEW packages will be INSTALLED:

  ca-certificates pkgs/main/win-64::ca-certificates-2022.2.1-haa95532_0
  certifi pkgs/main/win-64::certifi-2021.10.8-py39haa95532_2
  openssl pkgs/main/win-64::openssl-1.1.1m-h2bbff1b_0
  pip pkgs/main/win-64::pip-21.2.4-py39haa95532_0
  python pkgs/main/win-64::python-3.9.7-h6244533_1
  setuptools pkgs/main/win-64::setuptools-58.0.4-py39haa95532_0
```

4. Список пакетов после установки

```
(laba2) C:\Users\79616>conda list
# packages in environment at C:\Users\79616\anaconda3\envs\laba2:
#
# Name                      Version                      Build      Channel
ca-certificates             2022.2.1                    haa95532_0
certifi                     2021.10.8                   py39haa95532_2
openssl                     1.1.1m                      h2bbff1b_0
pip                         21.2.4                      py39haa95532_0
python                     3.9.7                       h6244533_1
setuptools                  58.0.4                      py39haa95532_0
sqlite                     3.37.2                      h2bbff1b_0
tzdata                     2021e                       hda174b7_0
vc                          14.2                       h21ff451_1
vs2015_runtime              14.27.29016                 h5e58377_2
wheel                      0.37.1                     pyhd3eb1b0_0
wincertstore                0.2                        py39haa95532_2
```

5. Установка пакета TensorFlow. Ошибки отсутствуют

```
(laba2) C:\Users\79616>conda install -n laba2 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: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.3
  latest version: 4.11.0

Please update conda by running

    $ conda update -n base -c defaults conda

## Package Plan ##

environment location: C:\Users\79616\anaconda3\envs\laba2

added / updated specs:
- tensorflow

The following packages will be downloaded:
```

package	build	
tensorflow-select-2.3.0	mk1	3 KB
abseil-cpp-20210324.2	hd77b12b_0	1.6 MB
absl-py-0.15.0	pyhd3eb1b0_0	103 KB
aiohttp-3.8.1	py39h2bbff1b_0	487 KB
aiohttp-3.8.1	pyhd3eb1b0_0	12 KB
astor-0.8.1	py39haa95532_0	47 KB
astunparse-1.6.3	py_0	17 KB
async-timeout-4.0.1	pyhd3eb1b0_0	10 KB
attrs-21.4.0	pyhd3eb1b0_0	51 KB
blinker-1.4	py39haa95532_0	23 KB
brotlipy-0.7.0	py39h2bbff1b_1003	411 KB
cachetools-4.2.2	pyhd3eb1b0_0	13 KB
cffi-1.15.0	py39h2bbff1b_1	224 KB
charset-normalizer-2.0.4	pyhd3eb1b0_0	35 KB
click-8.0.4	py39haa95532_0	155 KB
cryptography-3.4.8	py39h71e12ea_0	638 KB
dataclasses-0.8	pyh6d0b6a4_7	8 KB
flatbuffers-2.0.0	h6c2663c_0	1.4 MB

6. Наличие пакета TensorFlow

requests-oauthlib-1.3.0	py_0	23 KB
rsa-4.7.2	pyhd3eb1b0_1	28 KB
scipy-1.7.3	py39h0a974cb_0	13.9 MB
six-1.16.0	pyhd3eb1b0_1	18 KB
tensorboard-2.6.0	py_1	4.9 MB
tensorboard-data-server-0.6.0	py39haa95532_0	17 KB
tensorboard-plugin-wit-1.6.0	py_0	630 KB
tensorflow-2.6.0	mk1_py39h31650da_0	4 KB
tensorflow-base-2.6.0	mk1_py39h9201259_0	65.1 MB
tensorflow-estimator-2.6.0	pyh7b7c402_0	267 KB
termcolor-1.1.0	py39haa95532_1	9 KB
typing-extensions-3.10.0.2	hd3eb1b0_0	12 KB
typing_extensions-3.10.0.2	pyh06a4308_0	31 KB
urllib3-1.26.8	pyhd3eb1b0_0	106 KB
werkzeug-2.0.3	pyhd3eb1b0_0	221 KB
wheel-0.35.1	pyhd3eb1b0_0	38 KB
win_inet_pton-1.1.0	py39haa95532_0	35 KB
wrapt-1.13.3	py39h2bbff1b_2	50 KB
yaml-1.6.3	py39h2bbff1b_0	153 KB
zipp-3.7.0	pyhd3eb1b0_0	12 KB
zlib-1.2.11	h8cc25b3_4	112 KB
Total:		238.3 MB

7. Запуск Python

```
(laba2) C:\Users\79616>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
Failed calling sys.__interactivehook__
Traceback (most recent call last):
  File "C:\Users\79616\anaconda3\envs\laba2\lib\site.py", line 449, in register_readline
    readline.read_history_file(history)
  File "C:\Users\79616\anaconda3\envs\laba2\lib\site-packages\pyreadline\rlmain.py", line 165, in read_history_file
    self.mode._history.read_history_file(filename)
  File "C:\Users\79616\anaconda3\envs\laba2\lib\site-packages\pyreadline\lineeditor\history.py", line 82, in read_history_file
    for line in open(filename, 'r'):
  File "C:\Users\79616\anaconda3\envs\laba2\lib\encodings\cp1251.py", line 23, in decode
    return codecs.charmap_decode(input,self.errors,decoding_table)[0]
UnicodeDecodeError: 'charmap' codec can't decode byte 0x98 in position 120: character maps to <undefined>
```

8. Импорт пакета Tensorflow

```
>>> import tensorflow
>>> print(tensorflow.__version__)
2.6.0
```

9. Содержимое файла requirements.txt

```
requirements.txt - Блокнот
Файл Правка Формат Вид Справка
alabaster @ file:///home/ktietz/src/ci/alabaster_1611921544520/work
anaconda-client==1.7.2
anaconda-navigator==2.0.4
anaconda-project @ file:///tmp/build/80754af9/anaconda-project_1610472525955/work
anyio @ file:///C:/ci/anyio_1620153418380/work/dist
appdirs==1.4.4
argh==0.26.2
argon2-cffi @ file:///C:/ci/argon2-cffi_1613037959010/work
asn1crypto @ file:///tmp/build/80754af9/asn1crypto_1596577642040/work
astroid @ file:///C:/ci/astroid_1613501047216/work
astropy @ file:///C:/ci/astropy_1617745647203/work
async-generator @ file:///home/ktietz/src/ci/async_generator_1611927993394/work
atomicwrites==1.4.0
attrs @ file:///tmp/build/80754af9/attrs_1604765588209/work
autopep8 @ file:///tmp/build/80754af9/autopep8_1615918855173/work
Babel @ file:///tmp/build/80754af9/babel_1607110387436/work
backcall @ file:///home/ktietz/src/ci/backcall_1611930011877/work
backports.functools_lru_cache @ file:///tmp/build/80754af9/backports.functools_lru_cache_
backports.shutil_get_terminal_size @ file:///tmp/build/80754af9/backports.shutil_get_term
backports.tempfile @ file:///home/linux1/recipes/ci/backports.tempfile_1610991236607/work
backports.weakref==1.0.post1
bcrypt @ file:///C:/ci/bcrypt_1597936263757/work
beautifulsoup4 @ file:///home/linux1/recipes/ci/beautifulsoup4_1610988766420/work
bitarray @ file:///C:/ci/bitarray_1618435038389/work
bkcharts==0.2
black==19.10b0
bleach @ file:///tmp/build/80754af9/bleach_1612211392645/work
bokeh @ file:///C:/ci/bokeh_1620784067744/work
boto==2.49.0
Bottleneck==1.3.2
brotlipy==0.7.0
certifi==2020.12.5
cffi @ file:///C:/ci/cffi_1613247279197/work
chardet @ file:///C:/ci/chardet_1607690654534/work
click @ file:///home/linux1/recipes/ci/click_1610990599742/work
cloudpickle @ file:///tmp/build/80754af9/cloudpickle_1598884132938/work
clyent==1.2.2
colorama @ file:///tmp/build/80754af9/colorama_1607707115595/work
```

10. Содержимое файла environment.yml

```
1  name: base
2  channels:
3    - defaults
4  dependencies:
5    - _ipyw_jlab_nb_ext_conf=0.1.0=py38_0
6    - alabaster=0.7.12=pyhd3eb1b0_0
7    - anaconda=2021.05=py38_0
8    - anaconda-client=1.7.2=py38_0
9    - anaconda-navigator=2.0.4=py38_0
10   - anaconda-project=0.9.1=pyhd3eb1b0_1
11   - anyio=2.2.0=py38haa95532_2
12   - appdirs=1.4.4=py_0
13   - argh=0.26.2=py38_0
14   - argon2-cffi=20.1.0=py38h2bbff1b_1
15   - asn1crypto=1.4.0=py_0
16   - astroid=2.5=py38haa95532_1
17   - astropy=4.2.1=py38h2bbff1b_1
18   - async_generator=1.10=pyhd3eb1b0_0
19   - atomicwrites=1.4.0=py_0
20   - attrs=20.3.0=pyhd3eb1b0_0
21   - autopep8=1.5.6=pyhd3eb1b0_0
22   - babel=2.9.0=pyhd3eb1b0_0
23   - backcall=0.2.0=pyhd3eb1b0_0
24   - backports=1.0=pyhd3eb1b0_2
25   - backports.functools_lru_cache=1.6.4=pyhd3eb1b0_0
26   - backports.shutil_get_terminal_size=1.0.0=pyhd3eb1b0_3
27   - backports.tempfile=1.0=pyhd3eb1b0_1
```

Вопросы:

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

Существует так называемый Python Package Index (PyPI) – это репозиторий, открытый для всех Python разработчиков, в нем вы можете найти пакеты для решения практически любых задач. Для скачивания и установки используется специальная утилита, которая называется `pip`.

2. Как осуществить установку менеджера пакетов `pip`?

Универсальный способ: скачать скрипт `get-pip.py`

```
$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

```
$ python get-pip.py
```

При этом, вместе с `pip` будут установлены `setuptools` и `wheels`. `Setuptools` – это набор инструментов для построения пакетов Python. `Wheels` – это формат дистрибутива для пакета Python.

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

По умолчанию в Linux `Pip` устанавливает пакеты в `/usr/local/lib/python2.7/dist-packages`. Использование `virtualenv` или `—user` во время установки изменит это местоположение по умолчанию. Важный момент: по умолчанию `pip` устанавливает пакеты глобально. Это может привести к конфликтам между версиями пакетов.

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

```
$ pip install ProjectName
```

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

```
$ pip install ProjectName==3.2
```

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

```
$ pip install -e git+https://gitrepo.com/ProjectName.git
```

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

```
$ pip install ./dist/ProjectName.tar.gz
```

8. Как удалить установленный пакет с помощью pip?

```
$ pip uninstall ProjectName
```

9. Как обновить установленный пакет с помощью pip?

```
$ pip install --upgrade ProjectName
```

10. Как отобразить список установленных пакетов с помощью pip?

Для вывода списка всех установленных пакетов применяется команда `pip list`. Если необходимо получить более подробную информацию о конкретном пакете `pip show ProjectName`

11. Каковы причины появления виртуальных окружений в языке Python?

Если разработчик работает над проектом не один, а с командой, ему нужно передавать и получать список зависимостей, а также обновлять их на своем компьютере таким образом, чтобы не нарушалась работа других его проектов. Значит нужен механизм, который вместе с обменом проектами быстро устанавливал бы локально и все необходимые для них пакеты, при этом не мешая работе других проектов. Попыток было несколько, но в основу PEP 405 легла утилита `virtualenv` Яна Бикинга. Были проанализированы возникающие при работе с ней проблемы. После этого в работу интерпретатора Python версии 3.3 добавили их решения. Так был создан встроенный в Python модуль `venv`, а утилита `virtualenv` теперь дополнительно использует в своей работе и его.

12. Каковы основные этапы работы с виртуальными окружениями?

- 1) Создаём через утилиту новое виртуальное окружение в отдельной папке для выбранной версии интерпретатора Python.
- 2) Активируем ранее созданное виртуальное окружения для работы.
- 3) Работаем в виртуальном окружении, а именно управляем пакетами используя `pip` и запускаем выполнение кода.

4) Деактивируем после окончания работы виртуальное окружение.

5) Удаляем папку с виртуальным окружением, если оно нам больше не нужно.

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

Создадим виртуальное окружение в папке проекта. Для этого перейдём в корень любого проекта на Python ≥ 3.3 и дадим команду:

```
$ python3 -m venv env
```

Чтобы активировать виртуальное окружение под Windows нужно дать команду:

```
> env\\Scripts\\activate
```

После активации приглашение консоли изменится. В его начале в круглых скобках будет отображаться имя папки с виртуальным окружением.

При размещении виртуального окружения в папке проекта стоит позаботиться об его исключении из репозитория системы управления версиями.

```
$ python3 -m venv /home/user/envs/project1_env
```

Чтобы переключиться с одного окружения на другое нам нужно выполнить команду деактивации и команду активации другого виртуального окружения.

```
$ deactivate  
$ source /home/user/envs/project1_env2/bin/activate
```

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

Для начала пакет нужно установить. Установку можно выполнить командой:

```
# Для python 3
python3 -m pip install virtualenv

# Для единственного python
python -m pip install virtualenv
```

Создание в текущей папке виртуального окружения для интерпретатора доступного через команду python3 с названием папки окружения env:

```
virtualenv -p python3 env
```

Активация и деактивация такая же, как у стандартной утилиты Python.

```
> env\\Scripts\\activate

(env) > deactivate
```

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

Грубо говоря, pipenv можно рассматривать как симбиоз утилит pip и venv (или virtualenv), которые работают вместе, пряча многие неудобные детали от конечного пользователя.

Pipenv умеет:

- автоматически находить интерпретатор Python нужной версии (находит даже интерпретаторы, установленные через pyenv и asdf!);
- запускать вспомогательные скрипты для разработки;
- загружать переменные окружения из файла .env;
- проверять зависимости на наличие известных уязвимостей.

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

Просмотреть список зависимостей можно командой: `pip freeze > requirements.txt`

Имя файла хранения зависимостей requirements.txt выбрано не зря. Оно является стандартной договоренностью и используется некоторыми утилитами автоматически.

Установка пакетов из файла зависимостей в новом виртуальном окружении так же выполняется одной командой: `pip install -r requirements.txt`

Расширение: `.txt`

18. В какие дистрибутивы Python входит пакетный менеджер conda?

Anaconda, miniconda и PyCharm.

19. Как создать виртуальное окружение conda?

Начиная проект, создайте чистую директорию и дайте ей понятное короткое имя.

Для Windows необходимо вначале запустить консоль Anaconda Powershell Prompt..

В этом окне необходимо ввести следующую последовательность команд:

```
mkdir %PROJ_NAME% cd %PROJ_NAME%
```

```
copy NUL > main.py
```

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

```
conda create -n %PROJ_NAME% python=3.7 conda activate %PROJ_NAME%
```

Установите пакеты, необходимые для реализации проекта. `conda install django, pandas.`

21. Как деактивировать и удалить виртуальное окружение conda?

Для Windows необходимо использовать следующую команду:

```
conda deactivate
```

Если вы хотите удалить только что созданное окружение, выполните:

```
conda remove -n $PROJ_NAME
```

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

6. Файл `environment.yml` позволит воссоздать окружение в любой нужный момент.

Достаточно набрать:

```
conda env create -f environment.yml
```

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

```
conda env export > enviromant.yml
```

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

Создавать отдельное окружение Conda и устанавливать только нужные библиотеки для каждого проекта. PyCharm позволяет легко создавать и выбирать правильное окружение.

25. Почему файлы `requirements.txt` и `environment.yml` должны храниться в репозитории git? Предоставляет доступ другим пользователям к файлам.