

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

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

Выполнил:  
Шальнев Владимир Сергеевич,  
2 курс, группа ПИЖ-б-о-20-1,

Проверил:  
Доцент кафедры  
прикладной математики и  
компьютерной безопасности,  
Воронкин Р.А.

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

## Выполнение:

```
(base) F:\pythonProject>cd lab_2_2

(base) F:\pythonProject\lab_2_2>conda create -n lab_2_2
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 4.11.0

Please update conda by running

    $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: F:\conda\envs\lab_2_2

Proceed ([y]/n)? conda update conda
Invalid choice: conda update conda
Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate lab_2_2
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) F:\pythonProject\lab_2_2>conda activate lab_2_2
```

## Создание виртуального окружения Anaconda

```
(lab_2_2) F:\pythonProject\lab_2_2>conda update -n base -c defaults conda
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: F:\conda
added / updated specs:
- conda

The following packages will be downloaded:

package-----build-----
conda-4.11.0-----py38ha05532_0-----14.4 MB
xnlmodict-0.12.0-----pyhd3eb1b0_0-----13 KB
-----Total:-----14.5 MB

The following packages will be UPDATED:
conda 4.10.1-py38ha05532_1 --> 4.11.0-py38ha05532_0

The following packages will be DOWNGRADED:
xnlmodict 0.12.0-py_0 --> 0.12.0-pyhd3eb1b0_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
xnlmodict-0.12.0 | 13 KB | ##### 100%
conda 4.11.0 | 14.4 MB | ##### 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

## Обновление conda

```
(lab_2_2) F:\pythonProject\lab_2_2>python -m pip install -U pip
Requirement already satisfied: pip in c:\pycharm\lib\site-packages (21.3)
Collecting pip
  Downloading pip-22.0.3-py3-none-any.whl (2.1 MB)
  | 2.1 MB 1.1 MB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 21.3
    Uninstalling pip-21.3:
      Successfully uninstalled pip-21.3
Successfully installed pip-22.0.3
```

### Установка pip

```
(lab_2_2) F:\pythonProject\lab_2_2>pip install numpy
Requirement already satisfied: numpy in c:\pycharm\lib\site-packages (1.20.2)

(lab_2_2) F:\pythonProject\lab_2_2>pip install pandas
Requirement already satisfied: pandas in c:\pycharm\lib\site-packages (1.2.4)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\pycharm\lib\site-packages (from pandas) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in c:\pycharm\lib\site-packages (from pandas) (2021.1)
Requirement already satisfied: numpy>=1.16.5 in c:\pycharm\lib\site-packages (from pandas) (1.20.2)
Requirement already satisfied: six>=1.5 in c:\pycharm\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.15.0)

(lab_2_2) F:\pythonProject\lab_2_2>pip install scipy
Collecting scipy
  Downloading scipy-1.8.0-cp39-cp39-win_amd64.whl (36.9 MB)
  ----- 36.9/36.9 MB 3.6 MB/s eta 0:00:00
Requirement already satisfied: numpy<1.25.0,>=1.17.3 in c:\pycharm\lib\site-packages (from scipy) (1.20.2)
Installing collected packages: scipy
Successfully installed scipy-1.8.0

(lab_2_2) F:\pythonProject\lab_2_2>
```

### Установка NumPy, Pandas, SciPy

```
(lab_2_2) F:\pythonProject\lab_2_2>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: done
```

### Запуск установки TensorFlow

# ## Package Plan ##

environment location: F:\conda\envs\lab\_2\_2

added / updated specs:

- tensorflow

The following packages will be downloaded:

package	build	
-----	-----	
_tflow_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
aiosignal-1.2.0	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
bratliipy-0.7.0	py39h2bbff1b_1003	411 KB
ca-certificates-2021.10.26	haa95532_4	116 KB
cachetools-4.2.2	pyhd3eb1b0_0	13 KB
certifi-2021.10.8	py39haa95532_2	152 KB
cffi-1.15.0	py39h2bbff1b_1	224 KB
charset-normalizer-2.0.4	pyhd3eb1b0_0	35 KB
click-8.0.3	pyhd3eb1b0_0	79 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
frozenlist-1.2.0	py39h2bbff1b_0	77 KB
gast-0.4.0	pyhd3eb1b0_0	13 KB
google-auth-1.33.0	pyhd3eb1b0_0	80 KB
google-auth-oauthlib-0.4.1	py_2	20 KB

Сообщение о том, что будут скачаны доп. пакеты

The following NEW packages will be INSTALLED:

_tflow_select	pkgs/main/win-64::_tflow_select-2.3.0-mk1
abseil-cpp	pkgs/main/win-64::abseil-cpp-20210324.2-hd77b12b_0
absl-py	pkgs/main/noarch::absl-py-0.15.0-pyhd3eb1b0_0
aiohttp	pkgs/main/win-64::aiohttp-3.8.1-py39h2bbff1b_0
aiosignal	pkgs/main/noarch::aiosignal-1.2.0-pyhd3eb1b0_0
astor	pkgs/main/win-64::astor-0.8.1-py39haa95532_0
astunparse	pkgs/main/noarch::astunparse-1.6.3-py_0
async-timeout	pkgs/main/noarch::async-timeout-4.0.1-pyhd3eb1b0_0
attrs	pkgs/main/noarch::attrs-21.4.0-pyhd3eb1b0_0
blac	pkgs/main/win-64::blas-1.0-mkl

Сообщение о том, что будут установлены доп. пакеты



```

werkzeug          pkgs/main/noarch::werkzeug-2.0.2-pyhd3eb1b0_0
wheel             pkgs/main/noarch::wheel-0.35.1-pyhd3eb1b0_0
win_inet_pton     pkgs/main/win-64::win_inet_pton-1.1.0-py39haa95532_0
wincertstore      pkgs/main/win-64::wincertstore-0.2-py39haa95532_2
wrapit            pkgs/main/win-64::wrapit-1.13.3-py39h2bbbff1b_2
yarl              pkgs/main/win-64::yarl-1.6.3-py39h2bbbff1b_0
zipp              pkgs/main/noarch::zipp-3.7.0-pyhd3eb1b0_0
zlib              pkgs/main/win-64::zlib-1.2.11-h8cc25b3_4

Proceed ([y]/n)? y

```

## Сообщение о подтверждении установки

```

keras-preprocessing 35 KB ##### 100%
certifi-2021.10.8    152 KB ##### 100%
wheel-0.35.1         139 KB ##### 100%
win_inet_pton-1.1.0  1.6 MB ##### 100%
google-auth-1.31.0   98 KB ##### 100%
http-1.8.1           487 KB ##### 100%
icu-60.1             11.0 MB ##### 100%
google-auth-oauthlib 20 KB ##### 100%
jpeg-9d              283 KB ##### 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

```

## Скачивание и установка пакетов завершена

```

(lab_2_2) F:\pythonProject\lab_2_2>pip install tensorflow
Requirement already satisfied: tensorflow in f:\conda\envs\lab_2_2\lib\site-packages (2.6.0)
Requirement already satisfied: opt-einsum==3.3.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: wheel==0.35 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (0.35.1)
Requirement already satisfied: h5py>3.1.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (3.6.0)
Requirement already satisfied: wrapt==1.11.2 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.13.3)
Requirement already satisfied: six>1.15.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: typing-extensions>3.7.4 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (3.10.0.2)
Requirement already satisfied: protobuf>3.9.2 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (3.14.0)
Requirement already satisfied: gast==0.4.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: astunparse==1.6.3 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.6.3)
Collecting flatbuffers==1.12
  Downloading flatbuffers-1.12-py2.py3-none-any.whl (15 kB)
Collecting keras==2.8.0
  Downloading keras-2.8.0-py2.py3-none-any.whl (1.4 MB)
  1.4 MB 1.1 MB/s
Requirement already satisfied: tensorflow-estimator==2.6 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (2.6.0)
Requirement already satisfied: keras-preprocessing==1.1.2 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.1.2)
Collecting clang==5.0
  Downloading clang-5.0.tar.gz (38 kB)
Requirement already satisfied: numpy>1.19 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.21.5)
Requirement already satisfied: google-pasta==0.2 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: tensorboard==2.6 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (2.6.0)
Requirement already satisfied: termcolor==1.1.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: absl-py==0.10 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (0.15.0)
Requirement already satisfied: grpcio<2.0, >=1.37.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.42.0)
Requirement already satisfied: google-auth-oauthlib<0.5, >=0.4.1 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (0.4.1)
Requirement already satisfied: setuptools==41.0.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (58.0.4)
Requirement already satisfied: tensorboard-data-server<0.7.0, >=0.6.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (0.6.0)
Requirement already satisfied: tensorflow-plugin-wit>1.6.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.6.0)
Requirement already satisfied: requests<3, >=2.21.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (2.27.1)
Requirement already satisfied: markdown>2.6.8 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (3.3.4)
Requirement already satisfied: werkzeug==0.11.15 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (2.0.2)
Requirement already satisfied: google-auth2>1.6.3 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.33.0)
Requirement already satisfied: pyasn1-modules>0.2.1 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (0.2.8)
Requirement already satisfied: cachetools<5.0, >=2.0.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (4.2.2)
Requirement already satisfied: rsa<5, >=3.1.4 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (4.7.2)
Requirement already satisfied: requests-oauthlib<0.7.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.3.0)
Requirement already satisfied: pyasn1<0.5.0, >=0.4.6 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (0.4.8)
Requirement already satisfied: charset-normalizer==2.0.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (2.0.4)
Requirement already satisfied: idna<4, >=2.5 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (3.3)
Requirement already satisfied: urllib3<1.27, >=1.21.1 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (1.26.8)
Requirement already satisfied: certifi>2017.4.17 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (2021.10.8)
Requirement already satisfied: oauthlib<3.0.0 in f:\conda\envs\lab_2_2\lib\site-packages (from tensorflow) (3.1.1)
Building wheels for collected packages: clang
  Building wheel for clang (setup.py) ... done
  Created wheel for clang: filename=clang-5.0-py3-none-any.whl size=30701 sha256=07a9def3c8009a9d707b515cdd7033b6daaf9626fbd424f0ad6e3248865e21
  Stored in directory: c:\users\W041K-nk\appdata\local\pip\cache\wheels\3a\ce\7a\27094f689461801c934296d070773603663dfcac63bb064
Successfully built clang
Installing collected packages: keras, flatbuffers, clang
  Attempting uninstall: flatbuffers
    Found existing installation: flatbuffers 20210226132247
    Uninstalling flatbuffers-20210226132247:
      Successfully uninstalled flatbuffers-20210226132247
Successfully installed clang-5.0 flatbuffers-1.12 keras-2.8.0

```

## Установка TensorFlow с помощью pip

```

(lab_2_2) F:\pythonProject\lab_2_2>conda env export > environment.yml
(lab_2_2) F:\pythonProject\lab_2_2>pip freeze > requirements.txt
(lab_2_2) F:\pythonProject\lab_2_2>

```

## Создания файла requirements.txt и environment.yml

```

1 abs1-py @ file:///opt/conda/conda-bld/abs1-py_1639803114343/work
2 aiohttp @ file:///C:/ci/aiohttp_1637857274009/work
3 aiosignal @ file:///tmp/build/80754af9/aiosignal_1637843061372/work
4 astor==0.8.1
5 astunparse==1.6.3
6 async-timeout @ file:///tmp/build/80754af9/async-timeout_1637851218186/work
7 attrs @ file:///opt/conda/conda-bld/attrs_1642510447205/work
8 blinker==1.4
9 brotli==0.7.0
10 cachetools @ file:///tmp/build/80754af9/cachetools_1619597386817/work
11 certifi==2021.10.8
12 cffi @ file:///C:/ci_310/cffi_1642682485096/work
13 charset-normalizer @ file:///tmp/build/80754af9/charset-normalizer_1630003229654/work

```

Часть содержимого файла requirements.txt

```

1 name: lab_2_2
2 channels:
3   - defaults
4 dependencies:
5   - _tflow_select=2.3.0=mkl
6   - abseil-cpp=20210324.2=hd77b12b_0
7   - absl-py=0.15.0=pyhd3eb1b0_0
8   - aiohttp=3.8.1=py39h2bbff1b_0

```

Часть содержимого файла environment.yml

```

95 - pip:
96   - clang==5.0
97   - flatbuffers==1.12
98   - keras==2.8.0
99 prefix: F:\conda\envs\lab_2_2
100

```

Часть содержимого файла environment.yml

В результате анализа стало ясно, что в файле environment.yml содержатся более подробные сведения о зависимостях, содержится список модулей для команды pip, имя виртуального окружения. В файле requirements.txt содержатся только данные о пакетах.

Ответы на вопросы:

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

С помощью команды `pip install <Название пакета>` или `conda install <Название пакета>`

**2. Как осуществить установку менеджера пакетов pip?**

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

`python get-pip.py`

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

Python Package Index (PyPI) – это репозиторий, открытый для всех Python разработчиков,

**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?**

`pip list`

**11. Каковы причины появления виртуальных окружений в языке Python?**

В системе для интерпретатора Python может быть установлена глобально только одна версия пакета. Это порождает ряд проблем.

**1. Проблема обратной совместимости**

Чем опасно обновление пакетов или версий интерпретатора? В новой версии пакета могут измениться названия функций или методов объектов и число и/или порядок передаваемых в них параметров. В следующей версии интерпретатора могут появиться новые ключевые слова, которые совпадают с именами переменных уже существующих приложений.

**2. Проблема коллективной разработки**

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 нам нужен механизм, который вместе с обменом проектами быстро устанавливал бы локально и все необходимые для них пакеты, при этом не мешая работе других проектов.

**12. Каковы основные этапы работы с виртуальными окружениями?**

Создание виртуального окружения

Активация

Деактивация

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

Для создания виртуального окружения достаточно дать команду в формате: `python3 -m venv <путь к папке виртуального окружения>`

Чтобы активировать виртуальное окружение под Windows команда выглядит иначе: `env\\Scripts\\activate`

Чтобы переключиться с одного окружения на другое нам нужно выполнить команду деактивации и команду активации другого виртуального окружения, например, так:

```
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: `python3 -m pip install virtualenv`

Например, 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.

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

Установка – `pip install pipenv`

Создание оболочки виртуальной среды – `pipenv shell`

Установка нового пакета – `pipenv install ProjectName`

Установка пакета с репозитория – `pipenv install -e git+https://github.com/requests/requests.git#egg=requests`

Если необходимо установить зависимости, которые будут нужны только во время процесса разработки можно воспользоваться командой: `pipenv install pytest --dev`

Для переноса в рабочую среду необходимо заблокировать свою локальную среду командой – `pipenv lock`

Эта команда создаст/обновит файл **Pipfile.lock**, который вам никогда не придется (и никогда не нужно) редактировать вручную.

Теперь, нам нужно перенести свой код проекта в рабочую среду включая файлы **Pipfile** и **Pipfile.lock**. Далее создать там собственную среду окружения командой `pipenv shell`. И далее установить все зависимости командой: `pipenv install --ignore-pipfile`

—`ignore-pipfile` говорит Pipenv игнорировать **Pipfile** для установки и использовать то, что находится в **Pipfile.lock**. Учитывая **Pipfile.lock**, Pipenv создаст ту же среду, которая была у нас, когда мы запустили блокировку зависимостей в `pipenv`.

Теперь допустим, что другой разработчик хочет внести некоторые дополнения в наш код. В этой ситуации он склонирует весь код себе на компьютер, включая **Pipfile**, и воспользуются этой командой для установки всех зависимостей у себя локально:

```
pipenv install --dev
```

Эта команда установит все зависимости, необходимые для разработки, которые включают в себя как обычные зависимости, так и те, которые вы указали в аргументе —`dev` во время установки.



Мы также можем отобразить граф зависимостей, воспользовавшись командой `pipenv graph`

Проверить наличие уязвимостей безопасности (и требований PEP 508) в вашей среде: `pipenv check`

Полностью стереть все установленные пакеты из вашей виртуальной среды: `pipenv uninstall --all`

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

Все пакеты, которые вы установили перед выполнением команды и предположительно использовали в каком-либо проекте, будут перечислены в файле с именем «`requirements.txt`». Формат файла – текст

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

Основная проблема заключается в том, что `pip`, `easy_install` и `virtualenv` ориентированы на Python. Эти инструменты игнорируют библиотеки зависимостей, реализованные с использованием других языков. Например, XSLT, HDF5, MKL и другие, которые не имеют `setup.py` в исходном коде и не устанавливают файлы в директорию `site-packages`.

Conda же способна управлять пакетами как для Python, так и для C/ C++, R, Ruby, Lua, Scala и других. Conda устанавливает двоичные файлы, поэтому работу по компиляции пакета самостоятельно выполнять не требуется (по сравнению с `pip`).

**18. В какие дистрибутивы Python входит пакетный менеджер `conda`?**  
Anaconda и Miniconda.

**19. Как создать виртуальное окружение `conda`?**

```
mkdir %PROJ_NAME%
```

```
cd %PROJ_NAME%
```

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

Создаем директорию проекта с файлом `main.py` внутри.

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

Создаем виртуальное окружение

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

```
conda activate %PROJ_NAME%
```

Активация

```
conda install <имя пакета>
```

**21. Как деактивировать и удалить виртуальное окружение `conda`?**

```
conda deactivate
```

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

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

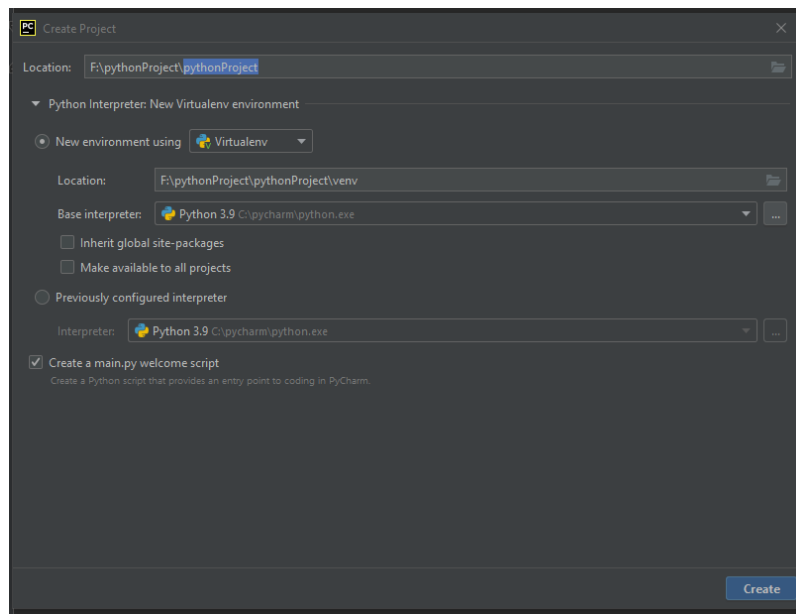
```
conda env export > environment.yml
```

В нем хранятся все зависимости из проектов

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

conda env create -f environment.yml

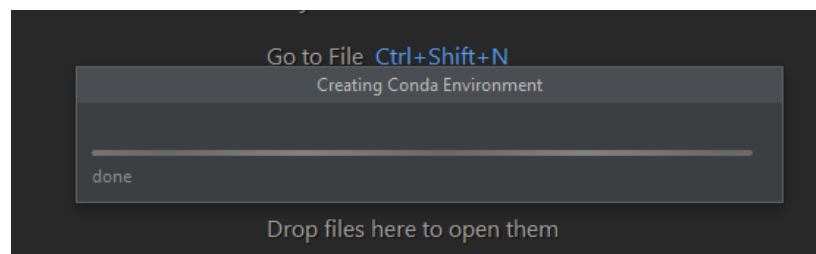
**24. Самостоятельно изучите средства IDE PyCharm для работы с виртуальными окружениями conda. Опишите порядок работы с виртуальными окружениями conda в IDE PyCharm.**



Окно создания нового проекта



Выбираем окружение conda



Создание окружение conda

```
(pythonProject) F:\pythonProject\pythonProject>conda list
# packages in environment at F:\conda\envs\pythonProject:
#
# Name                        Version      Build    Channel
ca-certificates              2021.10.26   haa95532_4
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
```

Создано виртуальное окружение

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

Потому что при разработке проекта, должны быть установлены одинаковые версии `python`, все установленные пакеты должны также быть одинаковы, именно поэтому необходимо хранить эти файлы в репозитории.