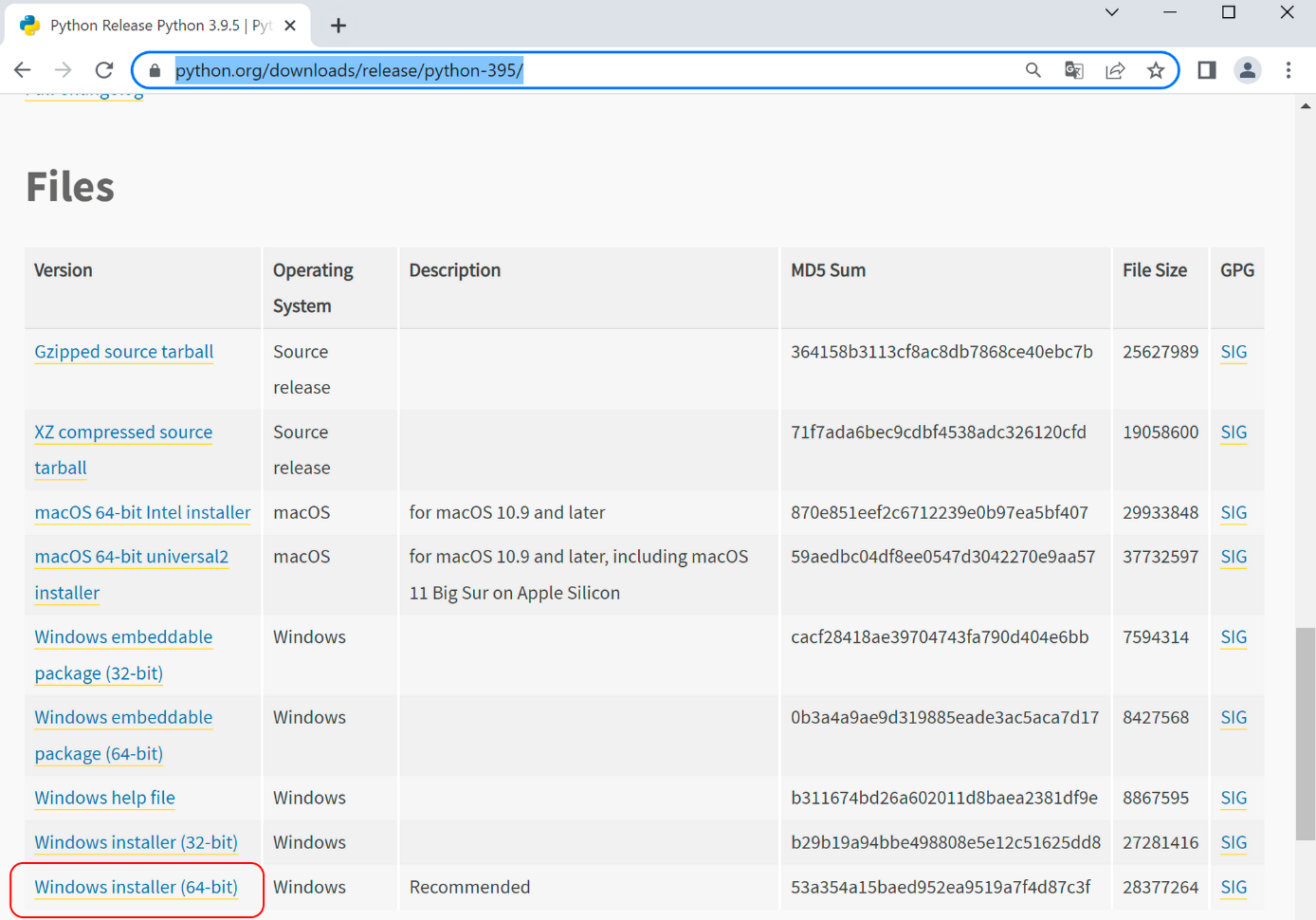
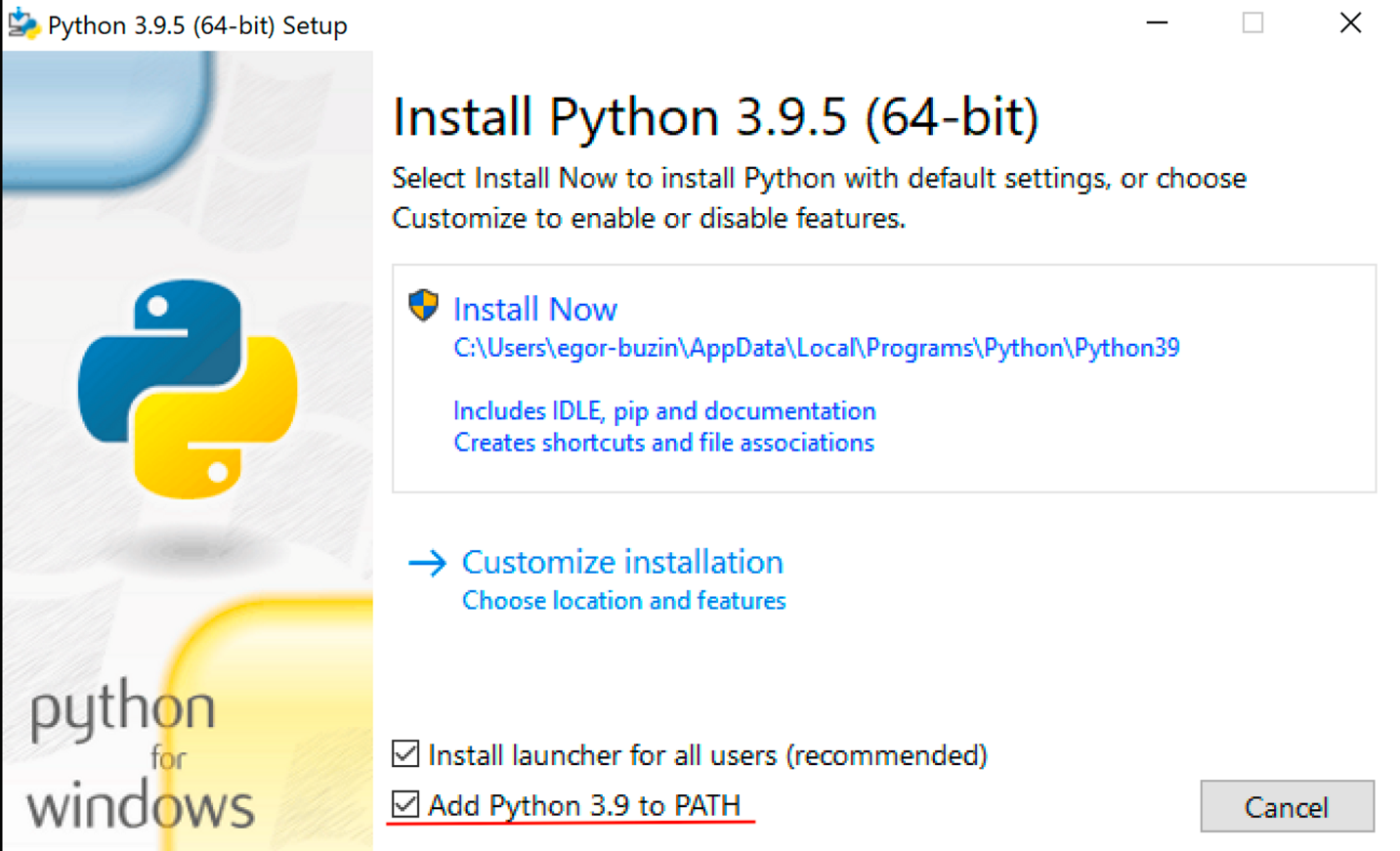
**Настройка окружения для Windows**

**Шаг 1**

Пройдите по ссылке <https://www.python.org/downloads/release/python-395/> и скачайте Python версии 3.9.5.

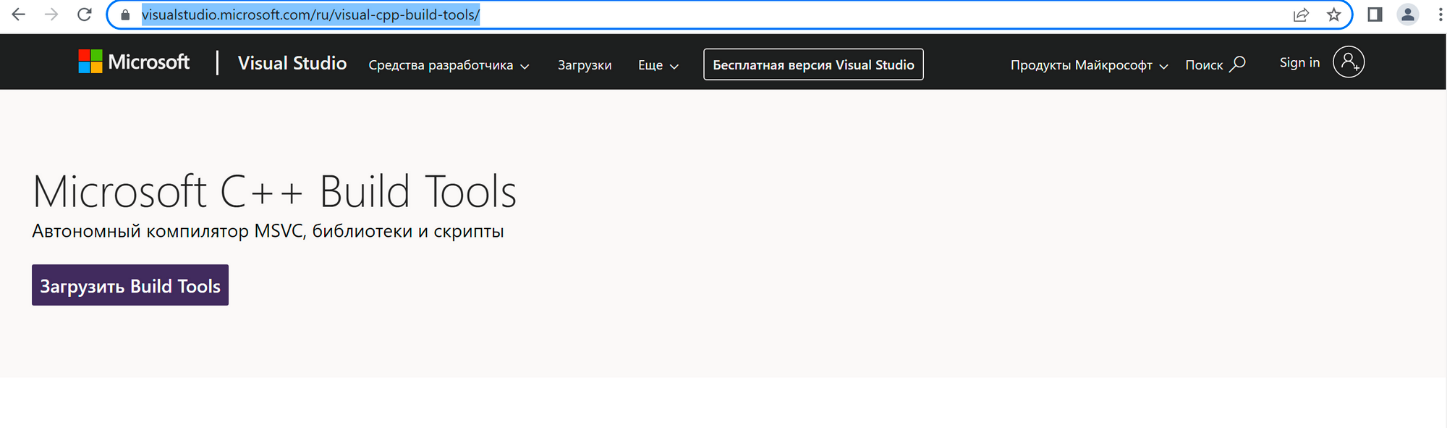
Запустите скачанный файл. Обязательно поставьте галочки напротив опций Add Python 3.9 to Path и Install launcher for all users (recommended), если они не выставлены по умолчанию. После чего нажмите кнопку Install now.



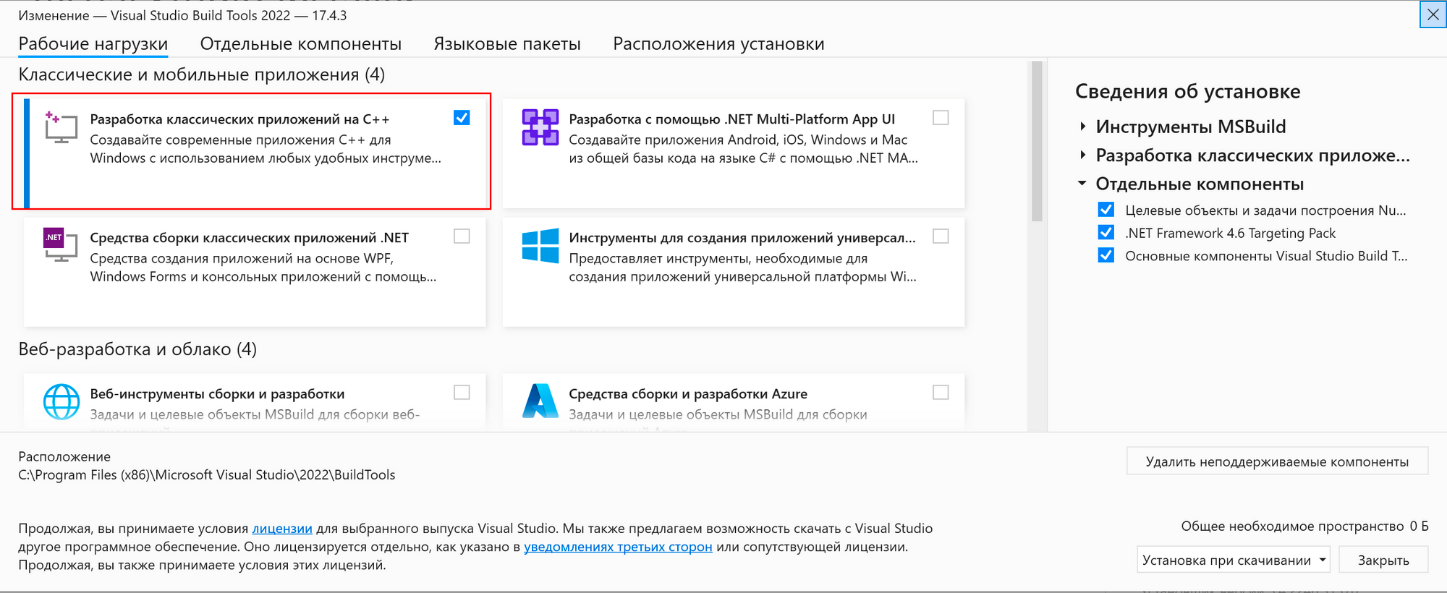
В следующих открывающихся окнах нажимайте кнопку OK.

Затем нужно установить Microsoft C++ Build Tools. Эти компоненты нужны для корректной работы некоторых библиотек.

Пройдите по ссылке <https://visualstudio.microsoft.com/ru/visual-cpp-build-tools/> и скачайте файл с дистрибутивом vs\_BuildTools.exe. Откройте файл и отметьте галочкой указанный на скриншоте компонент. Чтобы его установить, вам понадобится доступ к интернету.

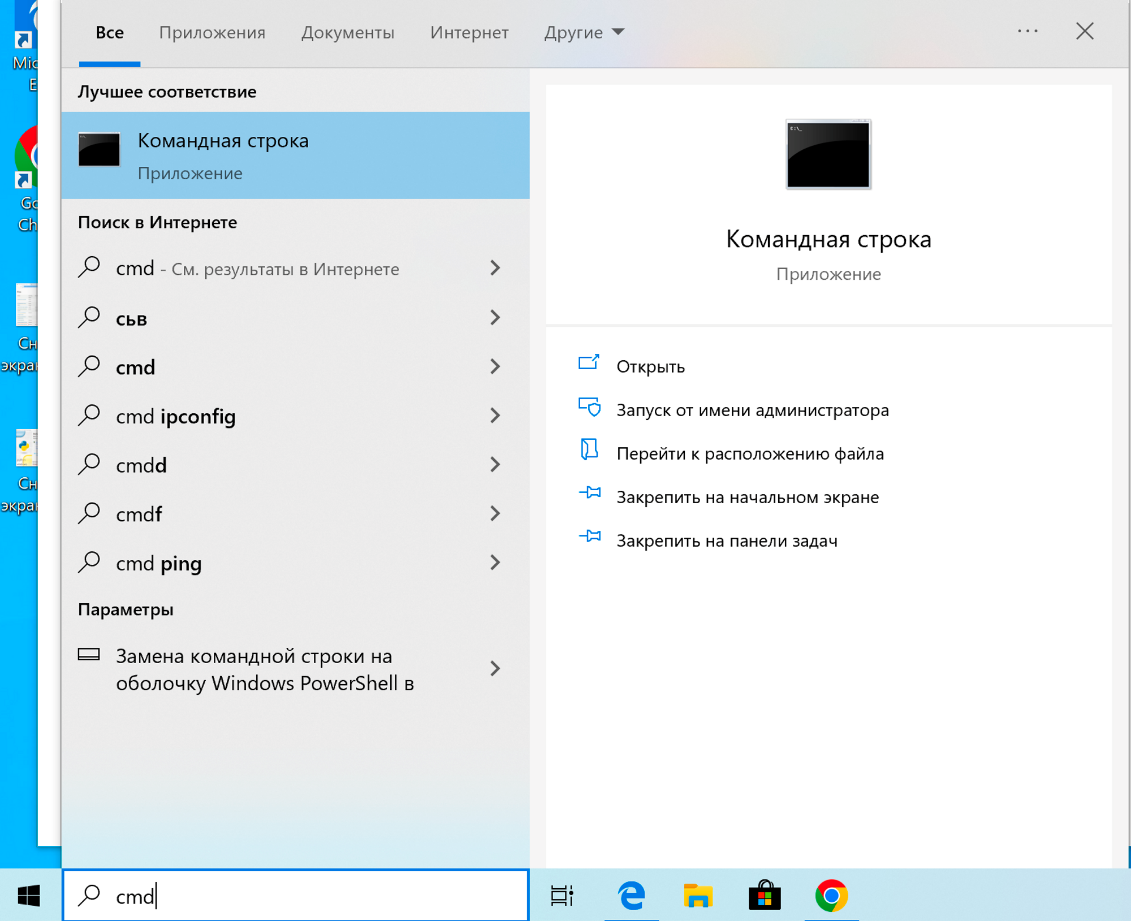


Установить нужно только выделенный компонент:

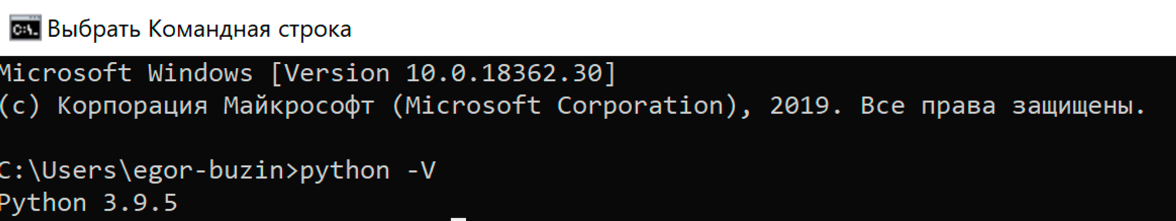


**Шаг 2**

Запустите командную строку.

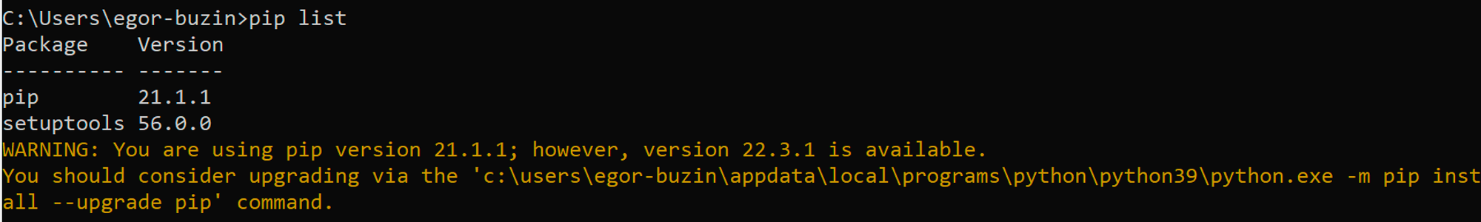


Чтобы убедиться, что всё установилось правильно, пропишите в терминале команду python -V. Должна отобразиться установленная версия Python.



Если выводится другая версия — ничего страшного! В шаге 3 мы расскажем, как активировать окружение именно на версии 3.9.5.

Последнее на этом этапе — нужно убедиться, что у вас установлен pip. Это менеджер пакетов в репозитории, с помощью которого можно устанавливать библиотеки в Python. Пропишите в терминале pip list и вы увидите, какие библиотеки сейчас установлены.

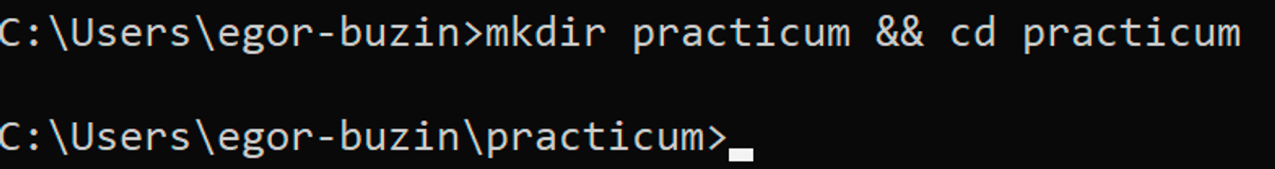


Если отображается дисклеймер WARNING, необходимо обновить pip командой pip install --upgrade pip --user.

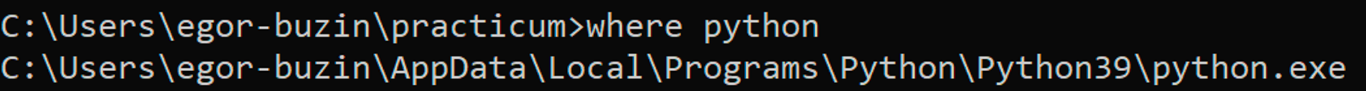
**Шаг 3**

Установите виртуальное окружение командой pip install virtualenv. Команда mkdir practicum && cd practicum создаст папку practicum для окружения и сразу же её откроет. В этом окружении вы будете работать, туда и надо установить все необходимые библиотеки.

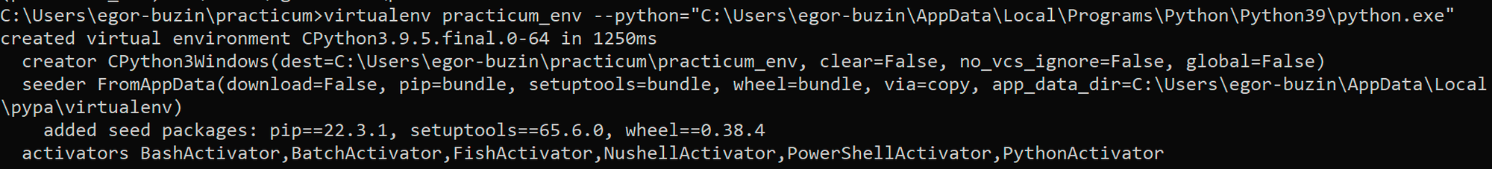
Если по умолчанию командная строка открывается в директории (папке)C:\Windows\System32 или какой-то другой, необходимо сначала открыть нужную директорию командой cd C:\Users\имя\_пользователя. Затем используйте команду mkdir practicum && cd practicum, чтобы создать папку для окружения и перейти в неё.



Команда where python покажет путь, куда устанавливается Python.

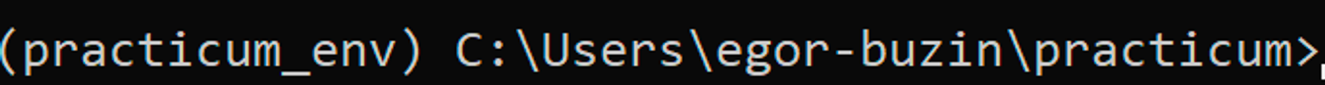


Этот путь нужно вставить после команды virtualenv practicum\_env --, чтобы активировать нужную версию Python в окружении. Целиком команда будет выглядеть примерно так: virtualenv practicum\_env --python="C:\Users\ваше\_имя\_пользователя\AppData\Local\Programs\Python\Python39\python.exe". Так вы создадите новое виртуальное окружение с версией Python 3.9.5. Часть practicum\_env обозначает название нового окружения. Не забудьте указать ваше имя пользователя и Python39.



Чтобы активировать окружение, пропишите команду practicum\_env\Scripts\activate.bat (либо practicum\_env\Scripts\activate.ps1 для PowerShell версии меньше 7).

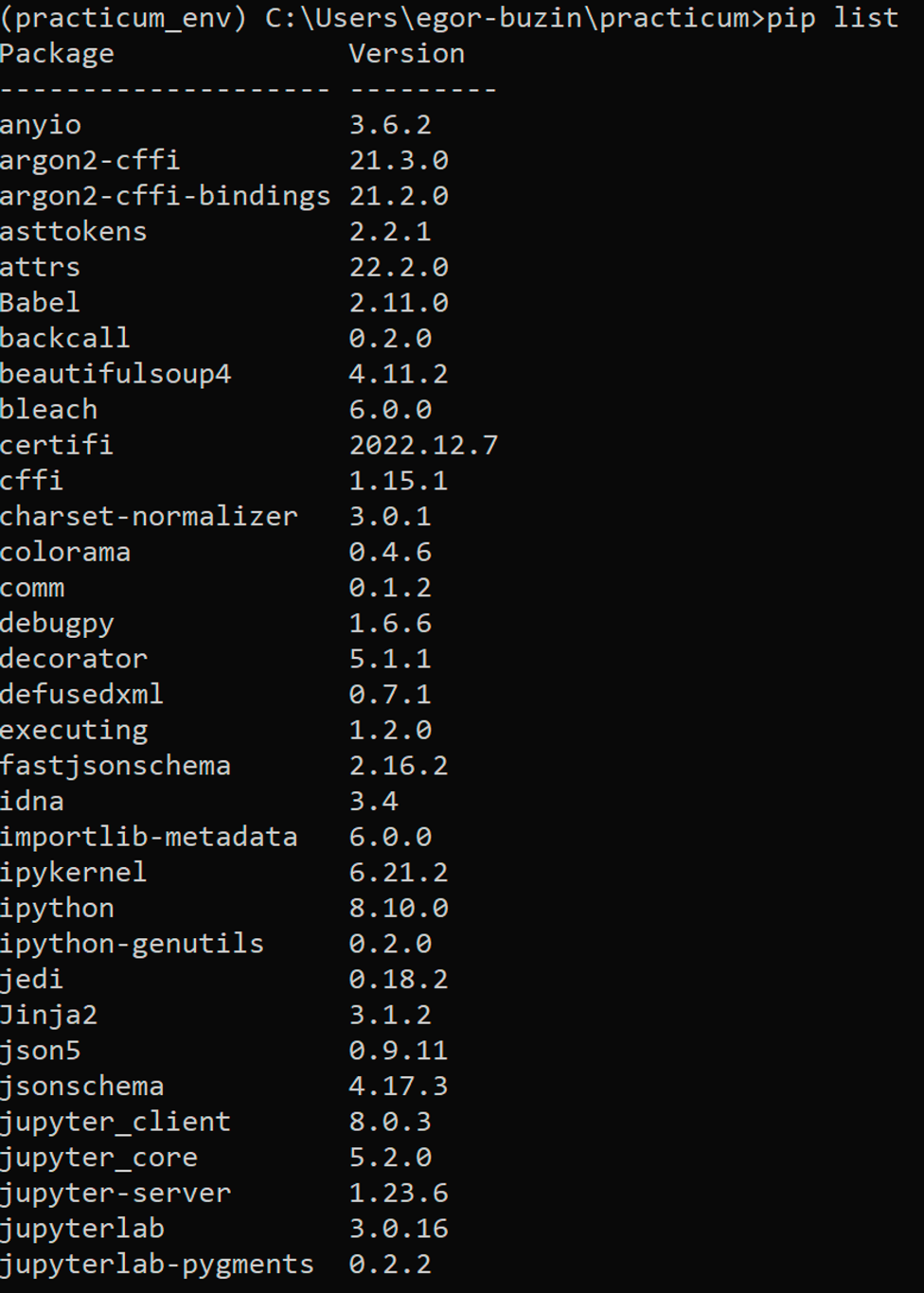
Активированное окружение выглядит так:



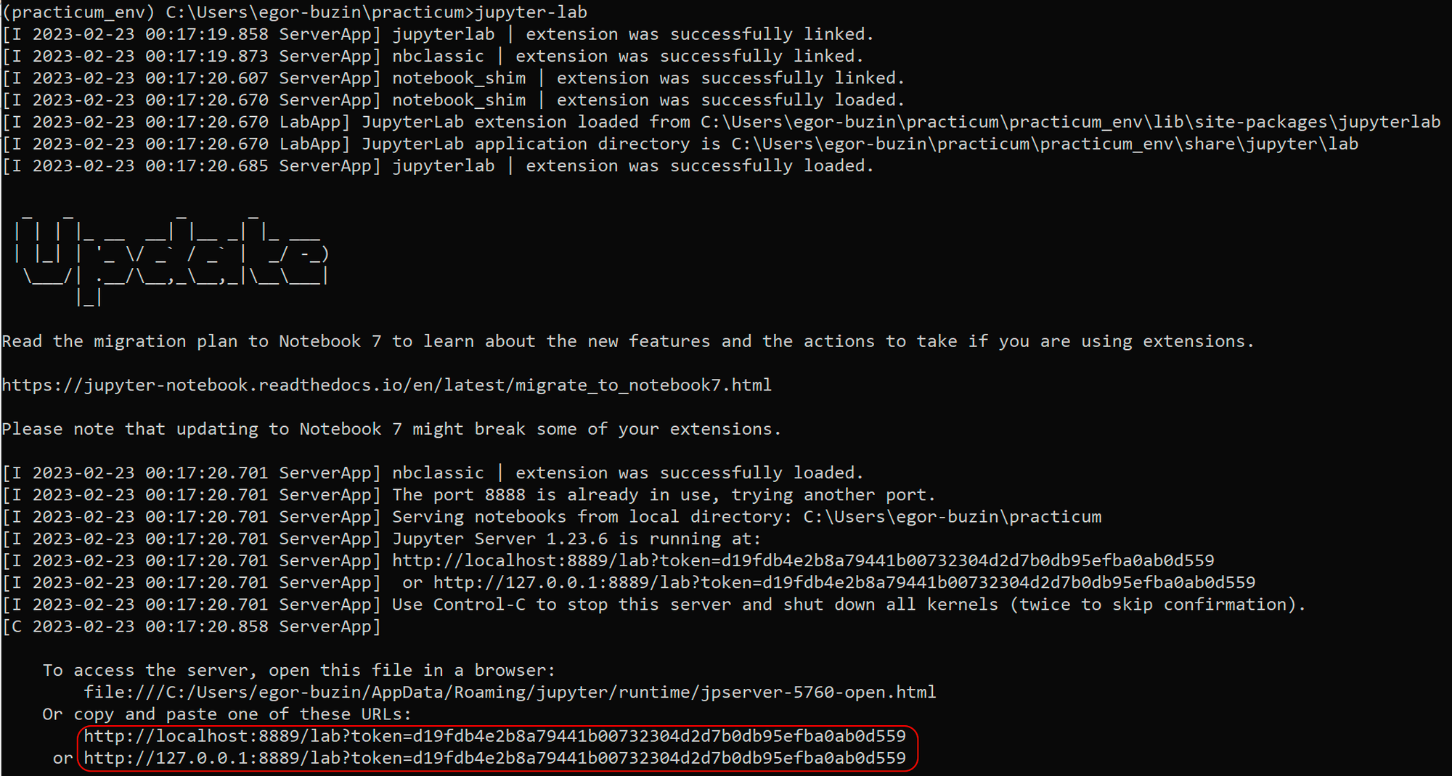
Чтобы деактивировать окружение, пропишите команду practicum\_env\Scripts\deactivate.bat (либо practicum\_env\Scripts\deactivate.ps1 для PowerShell версии меньше 7).

**Шаг 4**

Теперь можно установить сам JupyterLab. Для этого пропишите команду pip install jupyterlab==3.0.16. Выполнив команду pip list, можно увидеть, что вместе с JupyterLab установилось много новых библиотек.



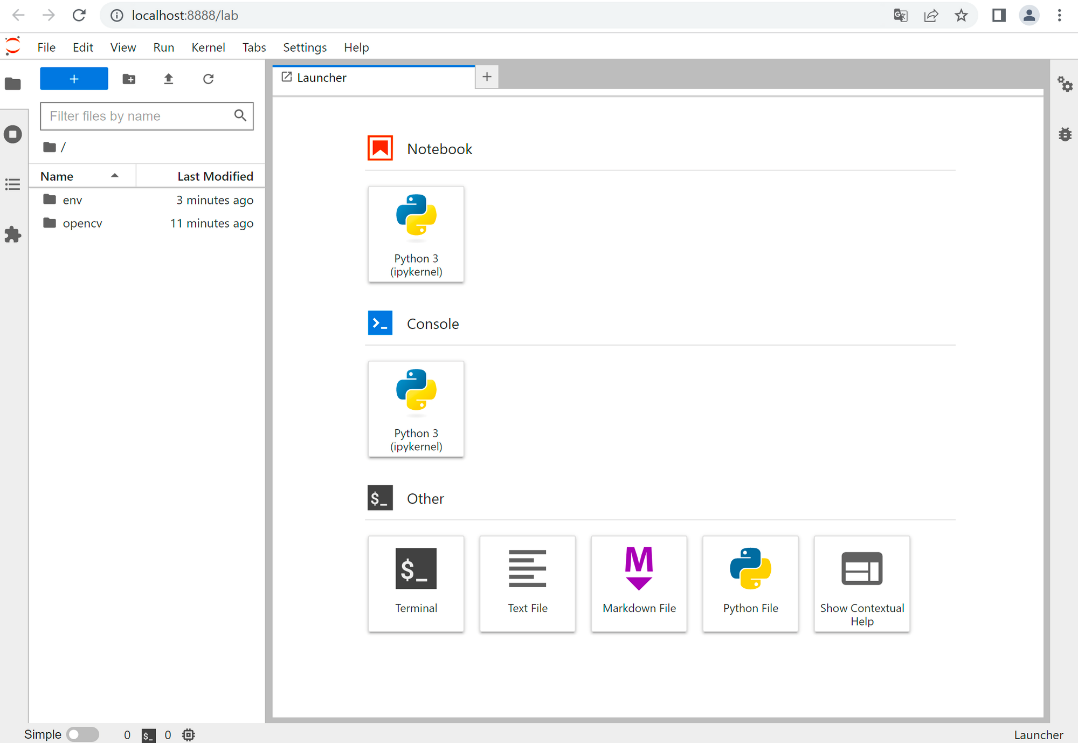
Теперь запустите JupyterLab командой jupyter-lab.



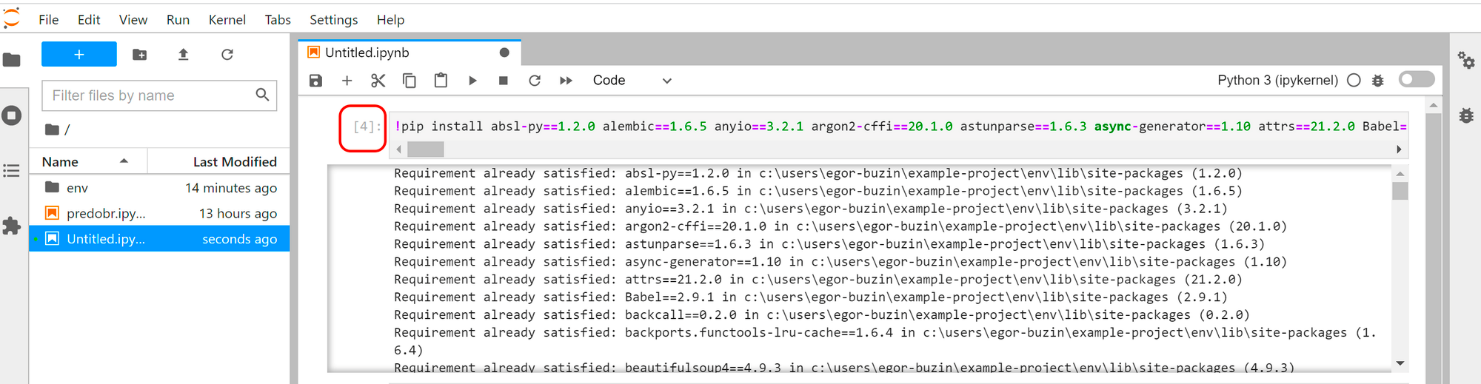
JupyterLab должен автоматически открыться в браузере. Если этого не произошло, нужно скопировать любую из выделенных на изображении ссылок и вставить её в строку браузера. Для удобства ссылку можно добавить в закладки или закрепить.

**Шаг 5**

В JupyterLab выберите Python 3 (ipykernel) в разделе Notebook, чтобы открыть тетрадь.



Так выглядит открытая тетрадь в JupyterLab:



Библиотеки, которые понадобятся для прохождения курса, вы сможете установить уже в тетради JupyterLab.

Вставьте следующую строку в любую ячейку и запустите её. Эту строку не нужно разбивать или делить на разные ячейки.

Скопировать кодPYTHON

!pip install absl-py==1.2.0 alembic==1.6.5 anyio==3.2.1 argon2-cffi==20.1.0 astunparse==1.6.3 async-generator==1.10 attrs==21.2.0 Babel==2.9.1 backcall==0.2.0 backports.functools-lru-cache==1.6.4 beautifulsoup4==4.9.3 bleach==3.3.0 blinker==1.4 blis==0.7.8 branca==0.5.0 brotlipy==0.7.0 cachetools==5.2.0 catalogue==2.0.8 catboost==1.0.3 certifi==2022.6.15 certipy==0.1.3 cffi==1.14.5 chardet==4.0.0 click==8.1.3 convertdate==2.4.0 cryptography==3.4.7 cycler==0.11.0 cymem==2.0.6 Cython DAWG-Python==0.7.2 debugpy decorator==5.0.9 defusedxml==0.7.1 docopt==0.6.2 entrypoints==0.3 et-xmlfile==1.1.0 filelock==3.8.0 findspark==1.4.2 flatbuffers==2.0.7 folium==0.12.1.post1 fuzzywuzzy==0.18.0 gast==0.4.0 gensim==4.1.2 google-auth==2.11.0 google-auth-oauthlib==0.4.6 google-pasta==0.2.0 graphviz==0.20.1 greenlet==1.1.0 grpcio==1.47.0 gspread==5.0.0 h5py==3.7.0 hijri-converter==2.2.4 holidays==0.11.3.1 huggingface-hub==0.9.1 idna==2.10 importlib-metadata==4.6.1 ipykernel==6.0.1 ipython==7.25.0 ipython-genutils==0.2.0 ipywidgets==7.6.3 jedi==0.18.0 Jinja2==3.0.1 joblib==1.1.0 json5==0.9.5 jsonschema==3.2.0 jupyter==1.0.0 jupyter-client==6.1.12 jupyter-console==6.4.2 jupyter-contrib-core==0.4.0 jupyter-contrib-nbextensions==0.5.1 jupyter-core==4.7.1 jupyter-dash jupyter-highlight-selected-word==0.2.0 jupyter-latex-envs==1.4.6 jupyter-nbextensions-configurator==0.5.0 jupyter-server==1.9.0 jupyter-telemetry==0.1.0 jupyterhub==1.4.2 jupyterlab-pygments==0.1.2 jupyterlab-server==2.6.0 jupyterlab-widgets==3.0.2 keras==2.7.0 Keras-Preprocessing==1.1.2 kiwisolver==1.4.4 korean-lunar-calendar==0.2.1 langcodes==3.3.0 libclang==14.0.6 lightgbm==3.3.1 llvmlite==0.39.0 lxml==4.6.3 Mako==1.1.4 Markdown==3.4.1 MarkupSafe==2.1.1 matplotlib==3.3.4 matplotlib-inline==0.1.2 murmurhash==1.0.8 nbclassic==0.3.1 nbclient==0.5.3 nbconvert==6.1.0 nbformat==5.1.3 nbgrader==0.6.1 nest-asyncio==1.5.1 nltk==3.6.1 notebook==6.4.0 numba==0.56.0 numpy==1.21.1 oauthlib==3.1.1 openpyxl==3.0.9 opt-einsum==3.3.0 packaging==21.3 pamela==1.0.0 pandas==1.2.4 pandocfilters==1.4.2 parso==0.8.2 pathy==0.6.2 patsy==0.5.2 pexpect==4.8.0 pickleshare==0.7.5 Pillow==8.4.0 plotly==5.4.0 preshed==3.0.7 prometheus-client==0.11.0 prompt-toolkit==3.0.19 protobuf==3.19.4 psycopg2-binary==2.9.2 ptyprocess==0.7.0 py4j==0.10.9 pyarrow==5.0.0 pyasn1==0.4.8 pyasn1-modules==0.2.8 pycosat==0.6.3 pycparser==2.20 pydantic==1.8.2 Pygments==2.9.0 PyJWT==2.1.0 PyMeeus==0.5.11 pymorphy2==0.9.1 pymorphy2-dicts-ru==2.4.417127.4579844 pymystem3==0.2.0 pyod==0.9.5 pyOpenSSL==20.0.1 pyparsing==2.4.7 PySocks==1.7.1 pyspark==3.0.2 python-editor==1.0.4 python-json-logger==2.0.1 pytz==2021.1 PyYAML==6.0 qtconsole==5.3.2 QtPy==2.2.0 regex==2022.8.17 requests==2.25.1 requests-oauthlib==1.3.1 requests-unixsocket==0.2.0 rsa==4.9 ruamel.yaml==0.17.10 ruamel.yaml.clib==0.2.2 sacremoses==0.0.53 scikit-learn==0.24.1 scipy==1.9.1 seaborn==0.11.1 setuptools==49.6.0 six==1.16.0 smart-open==5.2.1 sniffio==1.2.0 soupsieve==2.3.2.post1 spacy==3.2.0 spacy-legacy==3.0.10 spacy-loggers==1.0.3 SQLAlchemy==1.4.20 srsly==2.4.4 statsmodels==0.13.2 tenacity==8.0.1 tensorboard==2.10.0 tensorboard-data-server==0.6.1 tensorboard-plugin-wit==1.8.1 tensorflow-cpu==2.7.0 tensorflow-estimator==2.7.0 tensorflow-io-gcs-filesystem==0.26.0 termcolor==1.1.0 terminado==0.10.1 testpath==0.5.0 thinc==8.0.17 threadpoolctl==3.1.0 tokenizers==0.10.3 torch==1.10.0 tqdm==4.61.2 transformers==4.12.5 typer==0.4.2 typing-extensions==4.3.0 urllib3==1.26.6 wasabi==0.10.1 wcwidth==0.2.5 webencodings==0.5.1 websocket-client==0.57.0 Werkzeug==2.2.2 wheel==0.36.2 widgetsnbextension==3.5.2 wrapt==1.14.1 xgboost==1.5.1 xlrd==2.0.1 xmltodict==0.12.0 zipp==3.5.0

Процесс установки может занять время, и рядом с запущенной ячейкой будет отображаться знак [\*]. Когда ячейка кода выполнится, рядом с ней появится число ([1], [2], как на скриншоте выше).

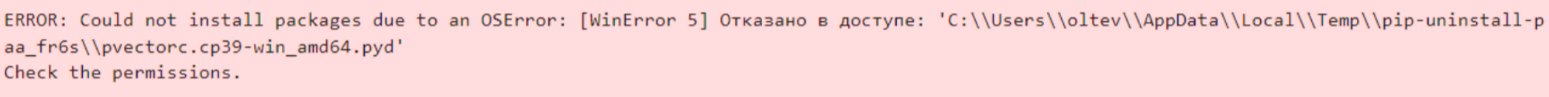
Этой же командой !pip install ... можно установить любую библиотеку, которая понадобится для выполнения проектов.

**Обратите внимание:** эта команда выполняется один раз. После успешной установки команду нужно удалить из ячейки, чтобы библиотеки не устанавливались каждый раз заново.

**Возможные ошибки**

При установке библиотек вы можете столкнуться с ошибками. Самые распространённые:

* Нет прав на установку библиотеки из определённой папки. Эта ошибка может выглядеть так:



* Неправильно установленная библиотека. Эта ошибка может выглядеть так:

https://pictures.s3.yandex.net/resources/image_1685966665.png

Решение:

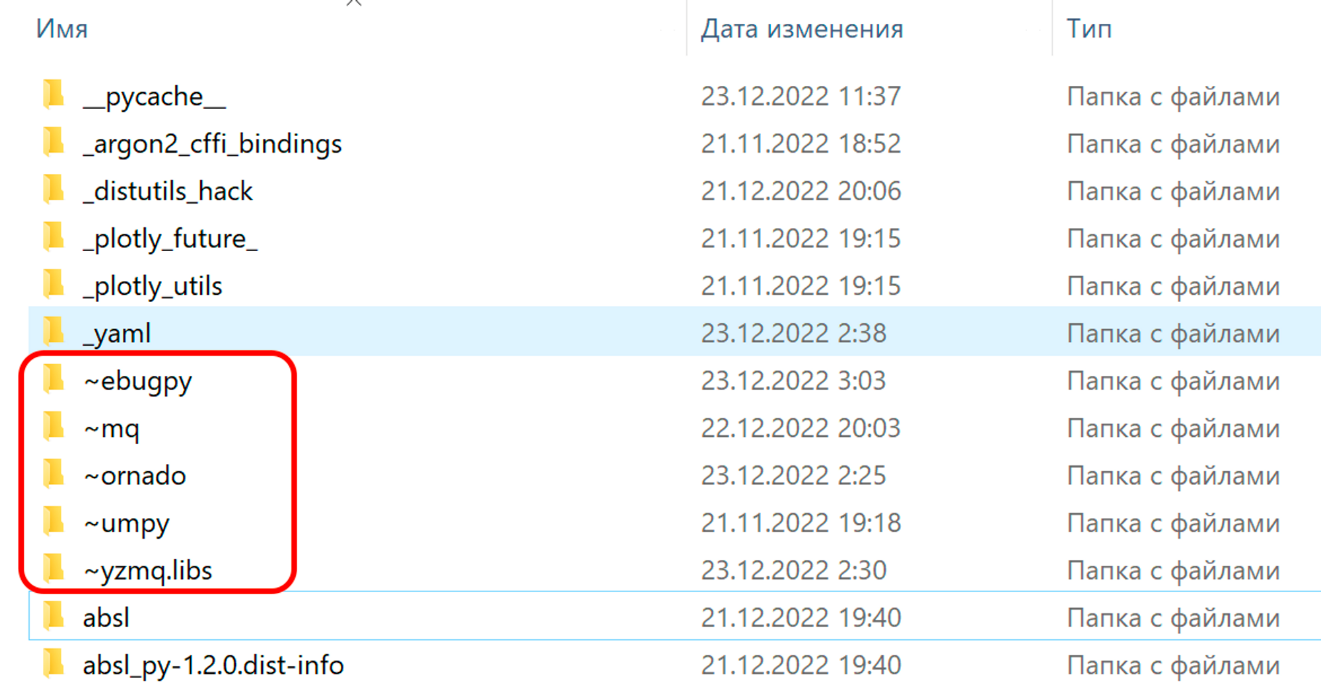
1) Удалите название библиотеки, которая вызвала ошибку, из строки с командой установки !pip install... в ячейке тетрадки.

2) Через Проводник зайдите в папку с библиотеками, она может находиться по подобному пути:

C:\Users\имя\_вашего\_пользователя\practicum\practicum\_env\Lib\site-packages.

Название диска может отличаться, например, оно может быть не C:\, а D:\.

3) Удалите все папки, которые начинаются с символа ~:



4) Снова запустите ячейку тетрадки с командой !pip install... в Jupyter.

При установке некоторых библиотек могут возникать ошибки: такие библиотеки нужно либо удалять, либо устанавливать другие версии, которые отличаются от окружения на платформе Практикума.

Чаще всего такие библиотеки не будут влиять на выполнение проектов. Вы можете работать над проектом локально, а потом запустить его на платформе. В редких случаях происходит конфликт версий библиотек: тогда код, который работает локально без проблем, может выдавать ошибки на платформе.

**После установки окружения**

Теперь, когда вы будете открывать проект в следующий раз, вам достаточно написать три команды в cmd.exe:

1. cd practicum — зайти в папку.
2. practicum\_env\Scripts\activate.bat — активировать окружение.
3. jupyter-lab — запустить JupyterLab.

Также можно запустить все команды сразу одной строкой в cmd.exe:

cd practicum & practicum\_env\Scripts\activate.bat & jupyter-lab.