



Настройка окружения

План занятия

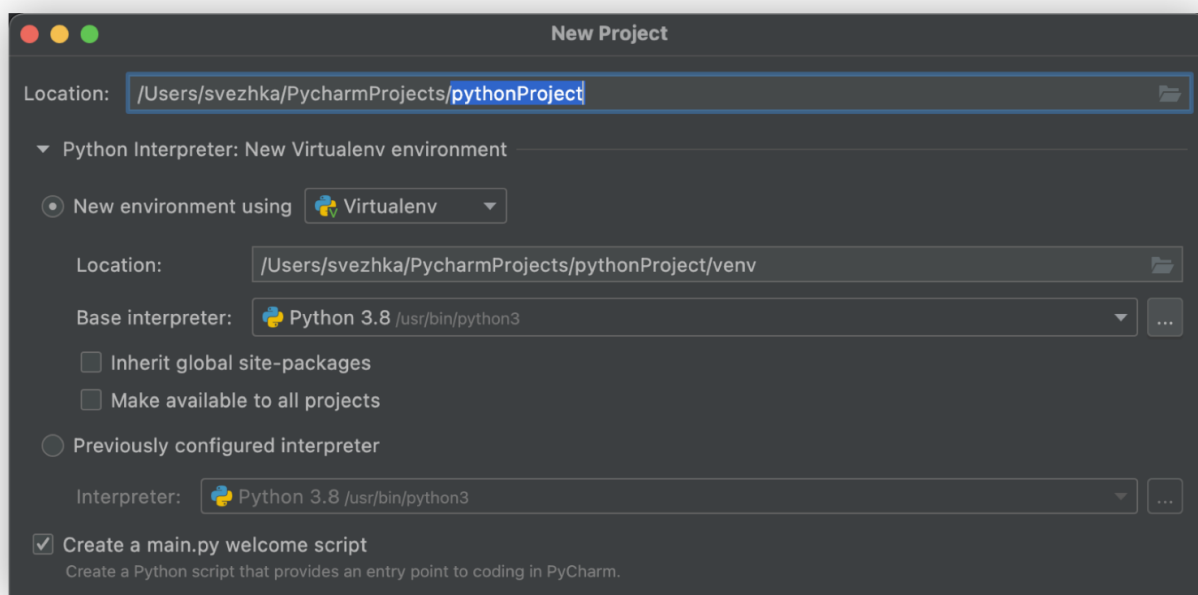
Работа с PyCharm	3
Начало работы	3
Примеры6	
Виртуальное окружение	8
Установка	8
Работа с пакетами.....	11

Работа с PyCharm

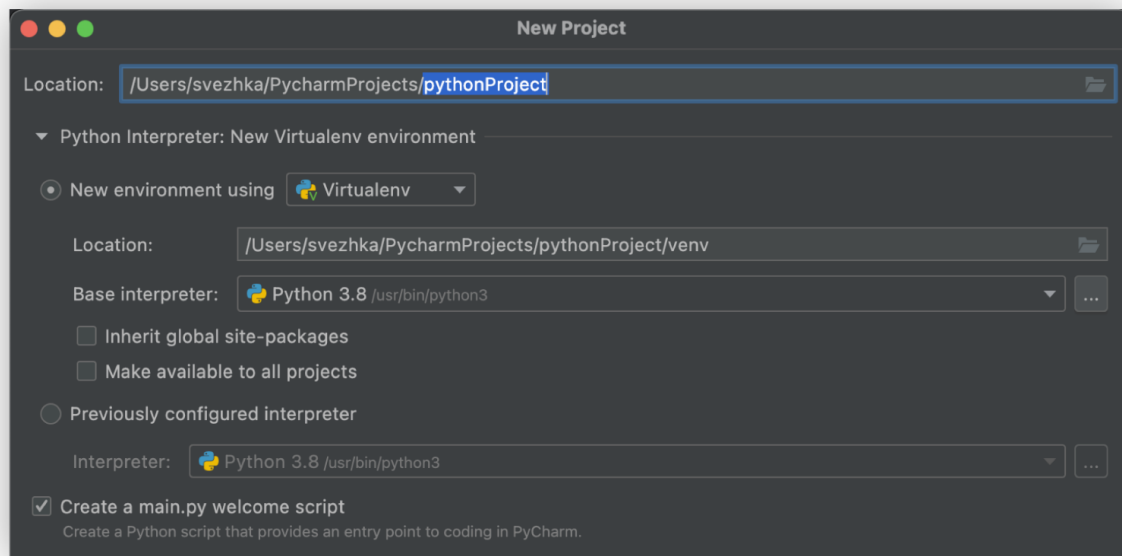
Начало работы

Скачать PyCharm: jetbrains.com/ru-ru/pycharm/download

Создать новый проект:

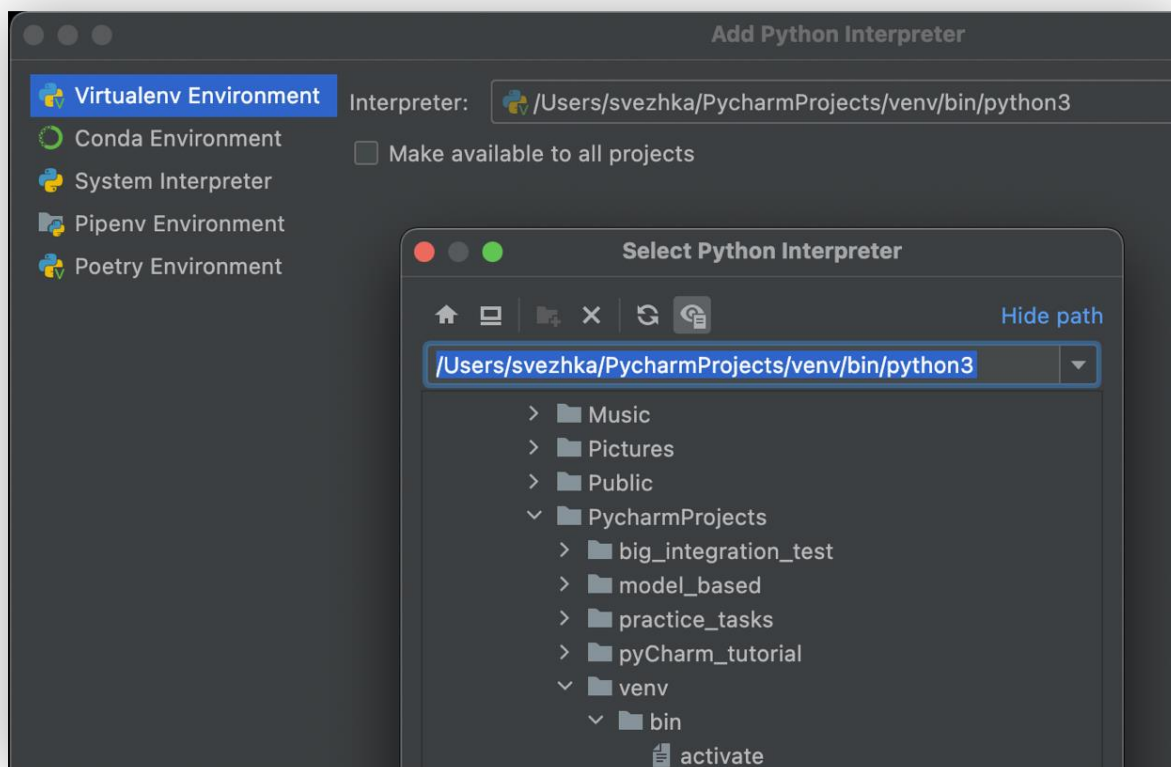


При создании проекта существует две опции: создать новый интерпретатор или выбрать из списка существующих:



Примеры популярных окружений:

- Virtualenv
- Conda
- System
- Pipenv
- Portry



Особенности некоторых из них:



Если Pip устанавливает пакеты Python6, то Conda устанавливает пакеты, которые могут содержать программное обеспечение, написанное на любом языке.



Если Pip хранит данные о зависимостях в файле requirements.txt, то Poetry хранит информацию в файле pyproject.toml .

Если Pip хранит данные о зависимостях в файле requirements.txt, то Poetry хранит информацию в файле pyproject.toml .



```
[tool.poetry]
name = "pycharm_tutorial"
version = "0.1.0"
description = ""
authors = ["asvezh"]

[tool.poetry.dependencies]
python = "^3.8"
pytest = "^7.1.2"
requests = "^2.28.1"

[tool.poetry.dev-dependencies]

[build-system]
requires = ["poetry-core>=1.0.0"]
build-backend = "poetry.core.masonry.api"
```

Примеры

Пример просто теста:

```
# test_example

def addition(x, y):
    return x + y

def test_sum(expect=8):
    data = 3, 5
    assert addition(*data) == expect
```

Запуск теста:

- Для запуска используется библиотека pytest.
- Флаг -v даст подробный вывод.

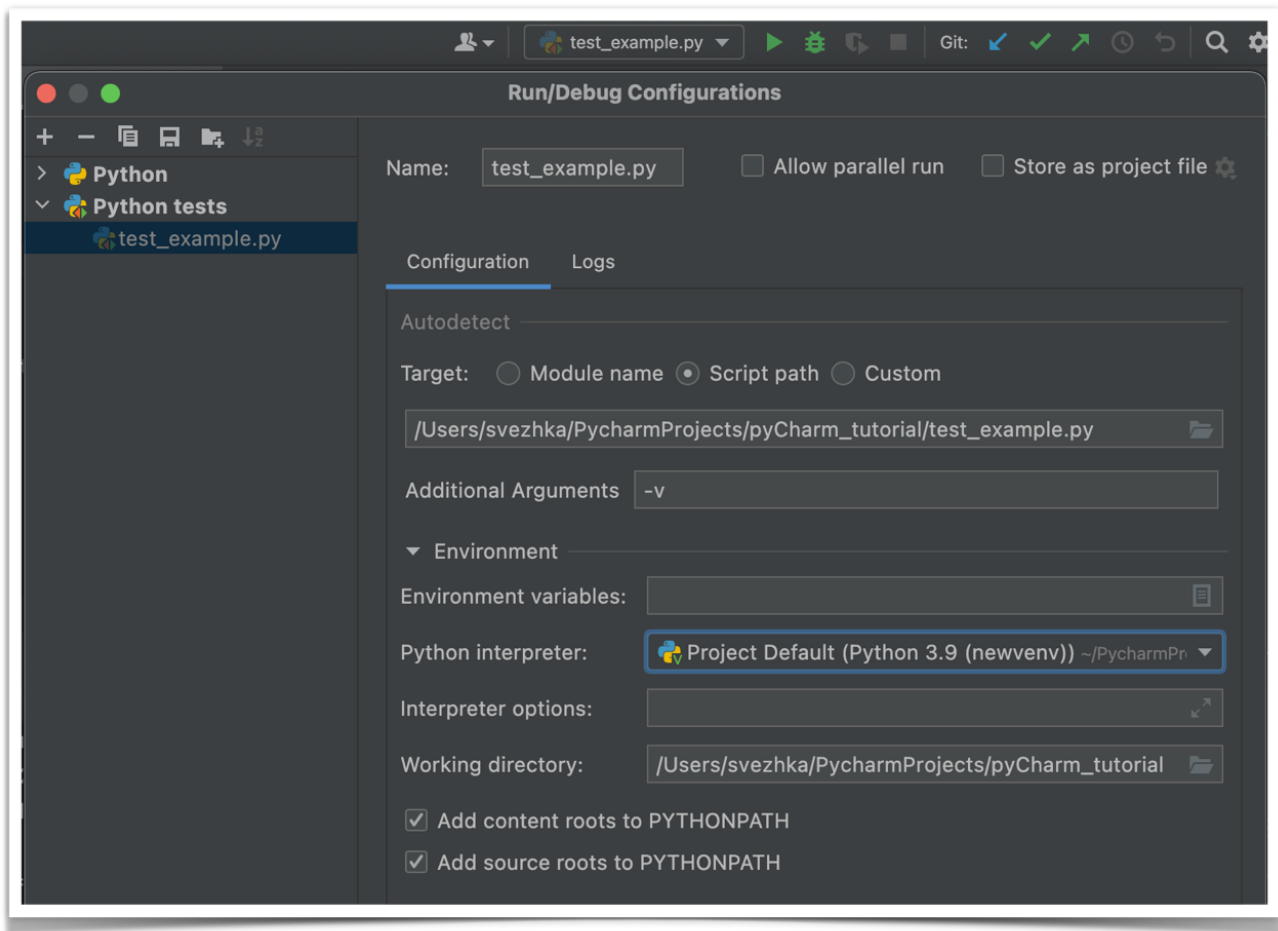
```
Terminal: Local x + v
(venv) svezhka@MacBook-Air-Alexander pyCharm_tutorial % pytest -v test_example.py
===== test session starts =====
platform darwin -- Python 3.8.2, pytest-7.1.2, pluggy-1.0.0 -- /Users/svezhka/PycharmProjects/pyCharm_tutorial/venv/bin/python
cachedir: .pytest_cache
rootdir: /Users/svezhka/PycharmProjects/pyCharm_tutorial
collected 1 item

test_example.py::test_sum PASSED

===== 1 passed in 0.00s =====
(venv) svezhka@MacBook-Air-Alexander pyCharm_tutorial %
```

Альтернативный вариант запуска через IDE:

Для запуска необходимо выбрать раннер и указать аргументы, например тот же флаг -v.



В поле environment variables можно передать переменные окружения, если они нужны.

В поле interpreter options вводятся опции самого интерпретатора.

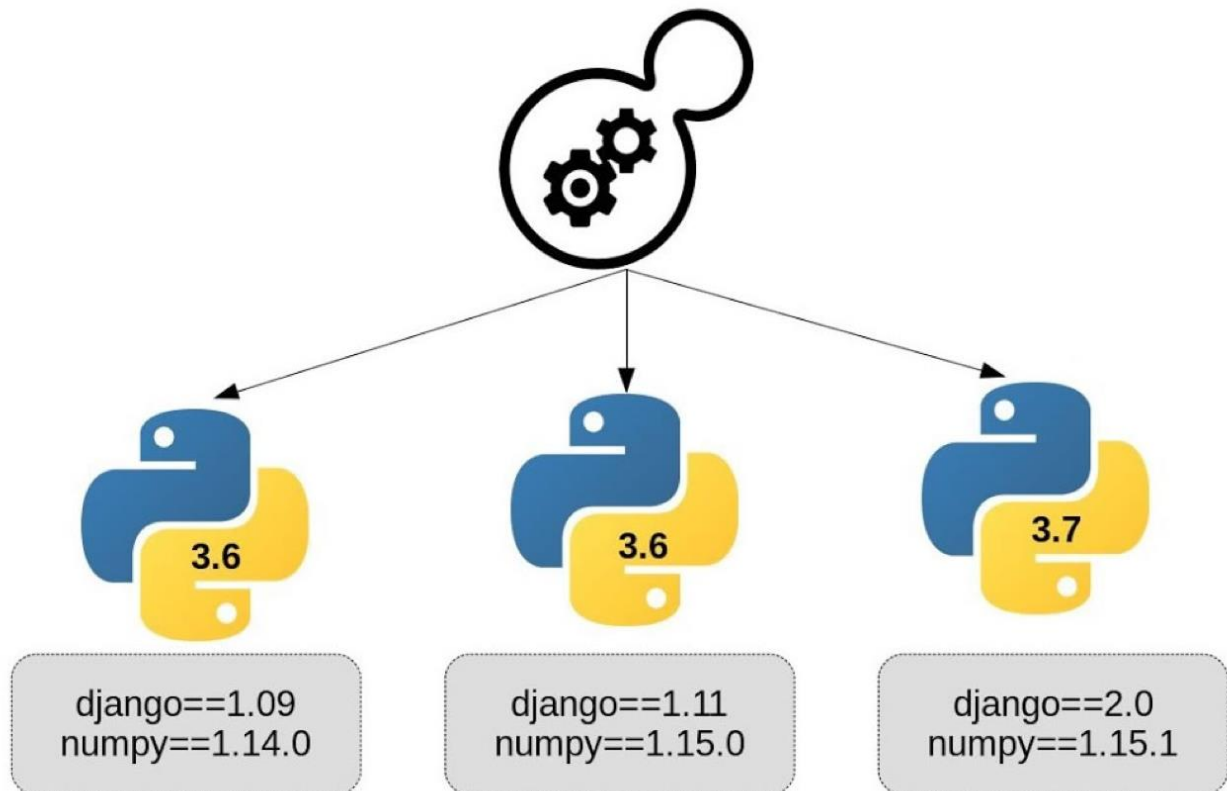
Виртуальное окружение

Установка

Для чего оно нужно:

- Установленные приложения/программы могут использовать разные версии одной и той же библиотеки.
- Версии библиотек могут быть несовместимы.
- Есть потребность в повторяемости окружения (для воспроизведения его на других машинах).

Пример использования:

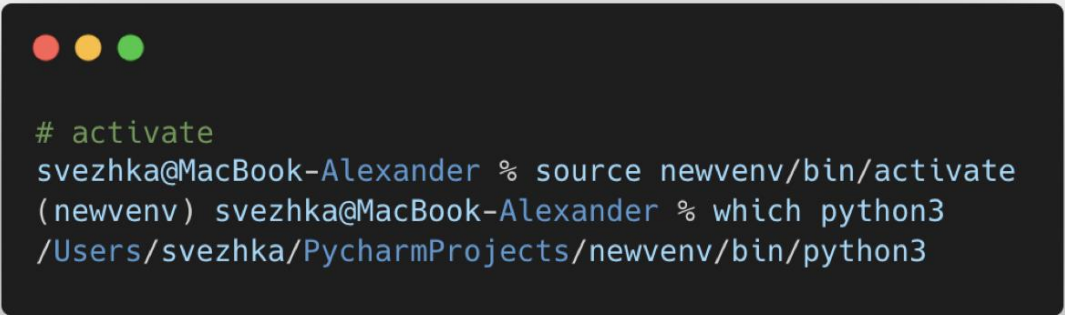


Основная цель – создание изолированной среды для python-проектов.

Создание окружения:

```
# create
svezhka@MacBook-Alexander % which python3
/opt/homebrew/bin/python3
svezhka@MacBook-Alexander % python3 -m venv newvenv
```

Активация окружения:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains the following text:

```
# activate
svezhka@MacBook-Alexander % source newenv/bin/activate
(newenv) svezhka@MacBook-Alexander % which python3
/Users/svezhka/PycharmProjects/newenv/bin/python3
```

```
# activate
svezhka@MacBook-Alexander % source newenv/bin/activate
(newenv) svezhka@MacBook-Alexander % which python3
/Users/svezhka/PycharmProjects/newenv/bin/python3
```

Деактивация окружения:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains the following text:

```
# deactivate
(newenv) svezhka@MacBook-Alexander % deactivate
svezhka@MacBook-Alexander %
```

```
# deactivate
(newenv) svezhka@MacBook-Alexander % deactivate
svezhka@MacBook-Alexander %
```

Содержание venv:

```
tree env

env
├── bin
│   ├── activate
│   ├── ...
│   ├── pip
│   ├── ...
│   └── python3 -> ../python3
├── lib
│   └── python3.6
│       ├── site-packages
│       │   ├── pip
│       │   │   └── ...
│       │   ├── ...
│       │   └── setuptools-40.6.2.dist-info
│       │       └── ...
│       └── ...
└── ...
```

Работа с пакетами

Руп — глобальный индекс для Python библиотек.

Find, install and publish Python packages
with the Python Package Index

Search projects

Or [browse projects](#)

350,291 projects 3,155,107 releases 5,435,745 files 563,902 users

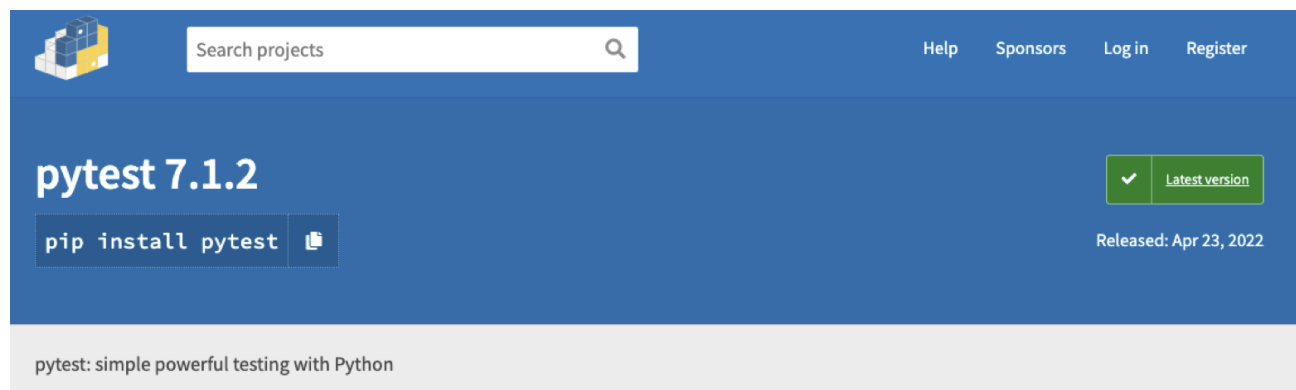
python™
Package
Index

The Python Package Index (PyPI) is a repository of software for the Python programming language.

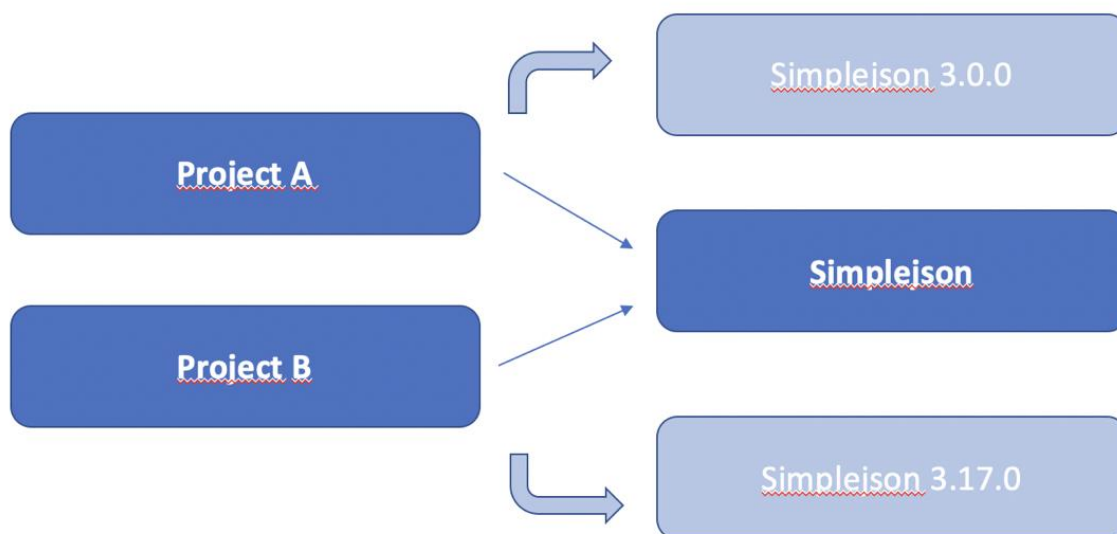
PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages >](#)

Через команду `pip install` можно установить пакет.

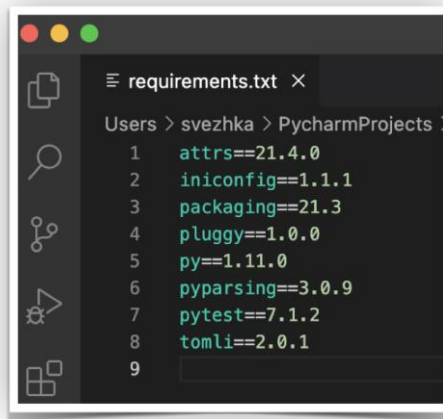
Например: **`pip install pytest==7.1.2`**



Все сторонние пакеты устанавливаются менеджером `pip` глобально. Чтобы установить пакет в нужное окружение, необходимо заранее не забыть активировать его.



Как перенести приложение на другую машину, сохранив все зависимости? Все зависимости проекта перечисляются в файле `requirements.txt`:

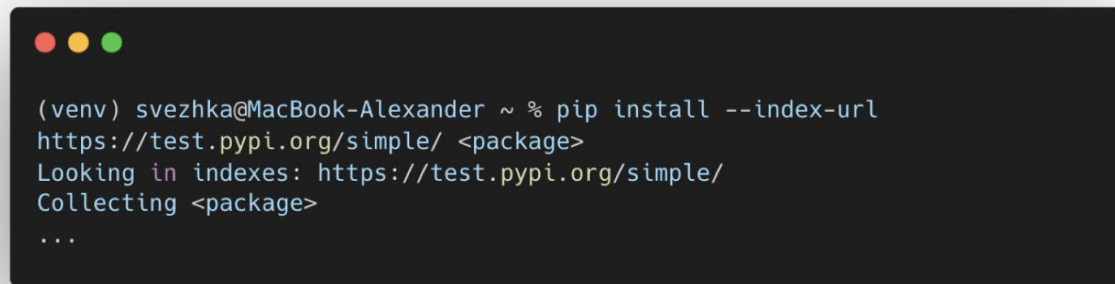


```
(newvenv) svezhka@MacBook-Alexander % pip list
Package      Version
-----
attrs        21.4.0
iniconfig    1.1.1
packaging    21.3
pip          22.1.2
pluggy       1.0.0
py           1.11.0
pyparsing    3.0.9
pytest       7.1.2
setuptools   60.10.0

(newvenv) svezhka@MacBook-Alexander % echo $(pip freeze)
attrs==21.4.0 iniconfig==1.1.1 packaging==21.3 pluggy==1.0.0 py==1.11.0 pyparsing==3.0.9 pytest==7.1.2
(newvenv) svezhka@MacBook-Alexander % pip freeze > requirements.txt
```

Установка зависимостей из файла: **pip install -r requirements.txt**

Если пакет доступен только из внутреннего репозитория, можно прописать доступ к пакетам, используя команду **pip install --index-url <url> .**

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The terminal displays the following text:

```
(venv) svezhka@MacBook-Alexander ~ % pip install --index-url  
https://test.pypi.org/simple/ <package>  
Looking in indexes: https://test.pypi.org/simple/  
Collecting <package>  
...
```

Для установки как внешних, так и внутренних зависимостей можно использовать опцию **pip install --extra-index-url <url> .**

Гайд с видеоуроками по IDE PyCharm jetbrains.com/pycharm/guide/playlists/42/.