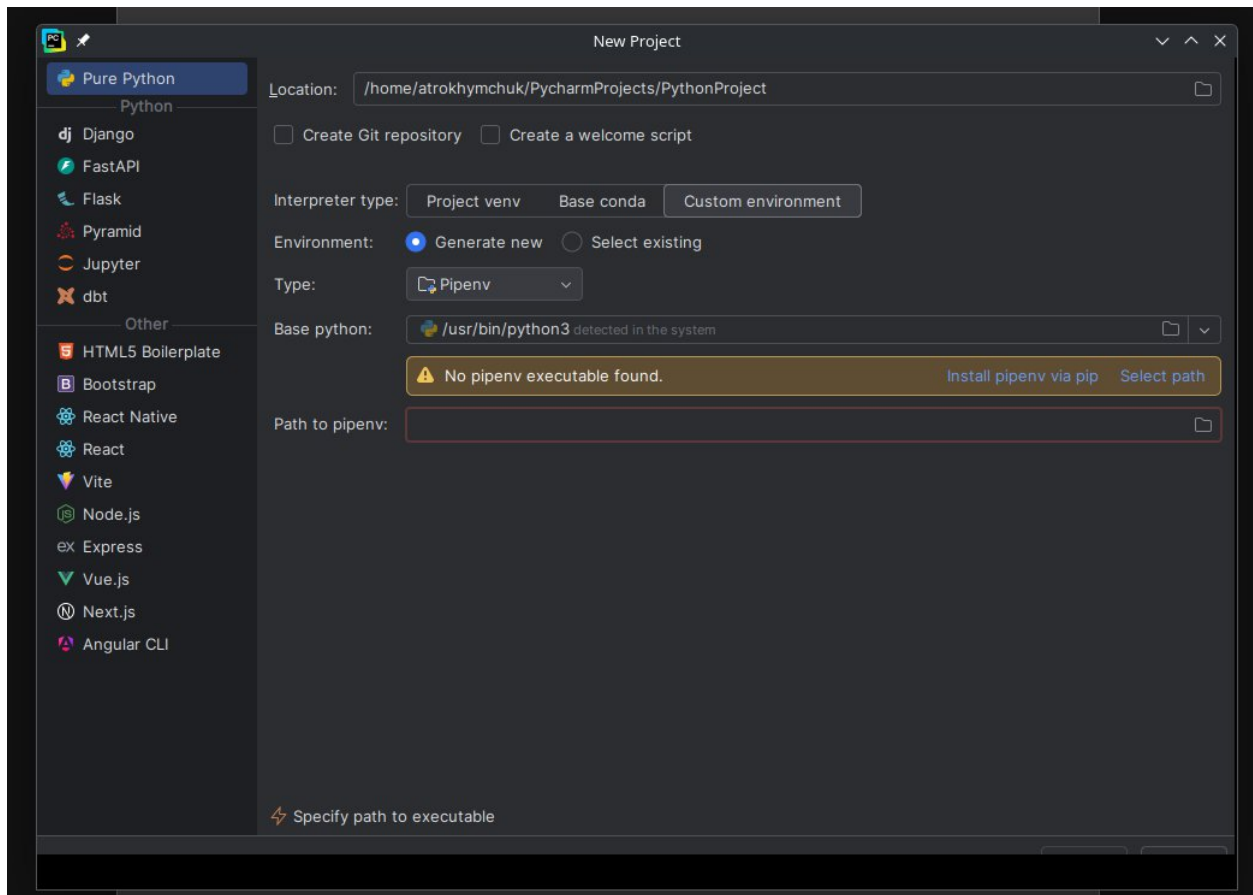


Трохимчук Артем, 3 група

Посилання: <https://github.com/trokhymchuk/pipenv>

Створення нового проєкту

Пайчарм при створенні свариться на відсутність pipenv, добре, шукаємо в репозиторіях



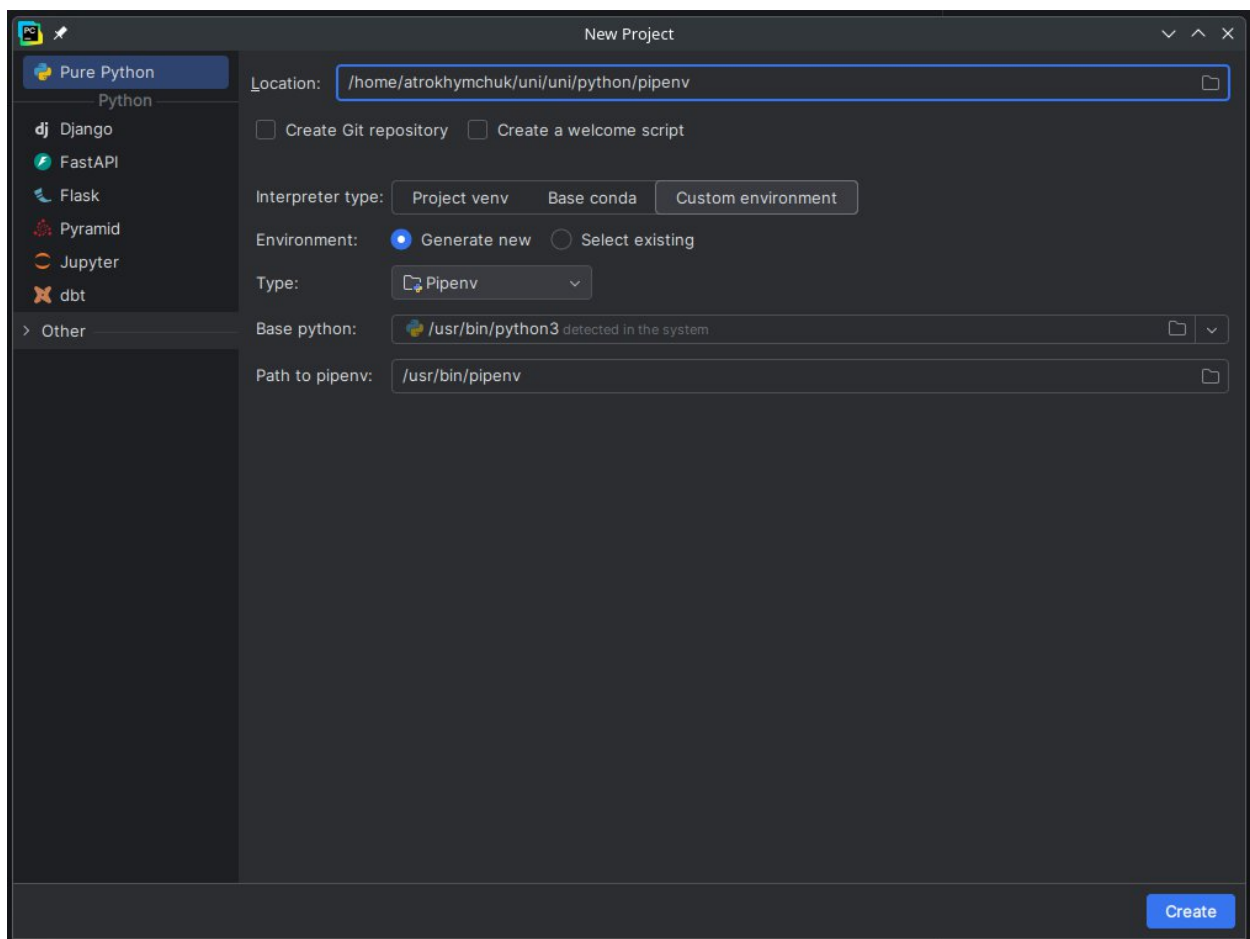
Удача, є, встановлюємо, пробуємо створити проєкт ще раз

```

Type help for instructions on how to use fish
atrokhymchuk@atrokhymchuk-laptop ~> pacman -Ss pipenv
extra/python-pipenv 2024.4.1-1
  Sacred Marriage of Pipfile, Pip, & Virtualenv.
extra/python-pipenv-to-requirements 0.9.0-10
  Generate requirements[-dev].txt from Pipfile using pipenv
extra/rye 0.44.0-1
  An experimental alternative to poetry, pip, pipenv, venv, virtualenv, pdm, hatch
atrokhymchuk@atrokhymchuk-laptop ~>

```

Не зовсім допомогло, але вручну вставив шлях до pipenv і пройшло



Створив .gitignore

```

atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv> echo ".idea/" >> .gitignore
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv> cat .gitignore
.idea/
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv>

```

Створив репозиторій, пушнув

```

atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (128)> git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/atrokhymchuk/uni/uni/python/pipenv/.git/
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> git status
On branch master

No commits yet

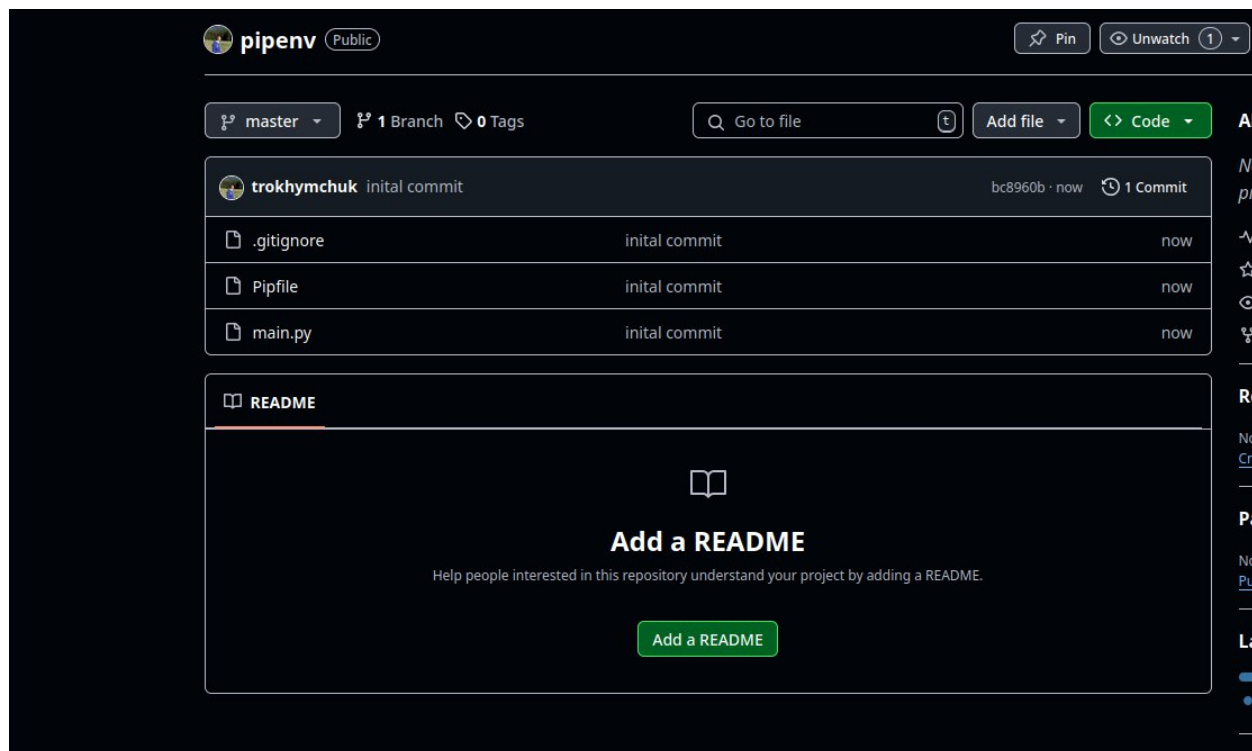
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        Pipfile
        main.py

nothing added to commit but untracked files present (use "git add" to track)
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> ls -al
total 12
drwxr-xr-x 1 atrokhymchuk atrokhymchuk  66 Apr  6 13:23 ./
drwxr-xr-x 1 atrokhymchuk atrokhymchuk  16 Apr  6 13:21 ../
drwxr-xr-x 1 atrokhymchuk atrokhymchuk  82 Apr  6 13:23 .git/
-rw-r--r-- 1 atrokhymchuk atrokhymchuk   7 Apr  6 13:22 .gitignore
drwxr-xr-x 1 atrokhymchuk atrokhymchuk 114 Apr  6 13:23 .idea/
-rw-r--r-- 1 atrokhymchuk atrokhymchuk 170 Apr  6 13:22 Pipfile
-rw-r--r-- 1 atrokhymchuk atrokhymchuk  65 Apr  6 13:23 main.py
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> git add .
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> git commit -m "inital commit"
[master (root-commit) bc8960b] inital commit
 3 files changed, 21 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 Pipfile
 create mode 100644 main.py
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> git branch -M master
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> git remote add origin git@github.com:trokhymchuk/pipenv.git
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> git push -u origin master

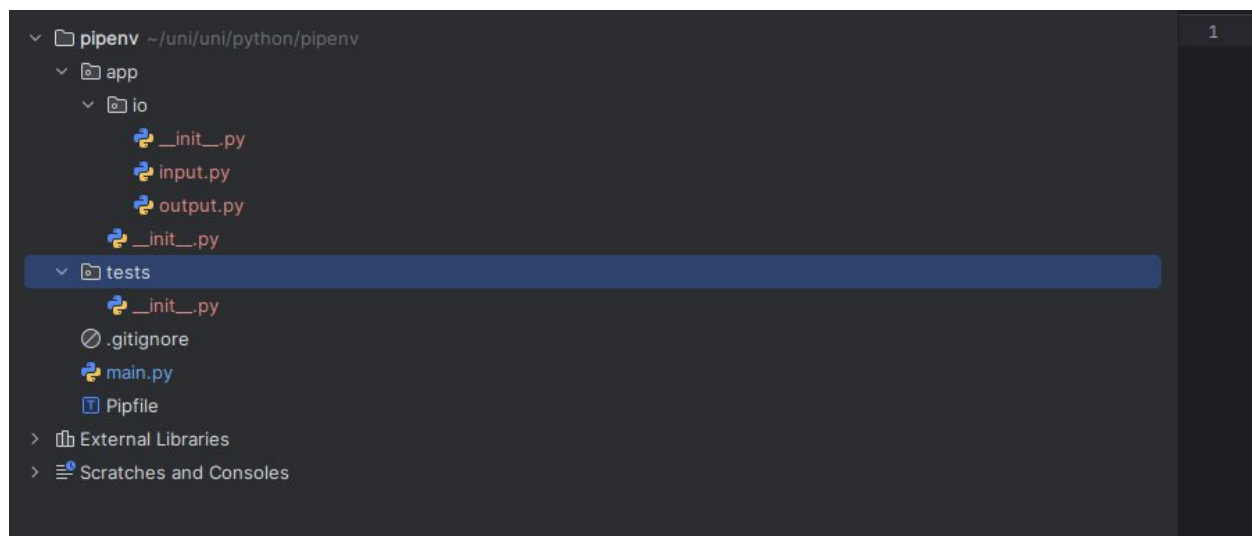
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 20 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 482 bytes | 482.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:trokhymchuk/pipenv.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)>

```

З'явилось на гітхабі



Створив директорії, трохи не зрозуміло, чи в тому місці створив tests, далі побачимо



Закомітив і пушнув

```

atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> git add .
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> git commit -m "add project structure"
[master 332b9bf] add project structure
6 files changed, 1 insertion(+), 1 deletion(-)
create mode 100644 app/__init__.py
create mode 100644 app/io/__init__.py
create mode 100644 app/io/input.py
create mode 100644 app/io/output.py
create mode 100644 tests/__init__.py
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 20 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 634 bytes | 634.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:trokhymchuk/pipenv.git
   bc8960b..332b9bf  master -> master
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)>

```

Підготувати рір не вийшло — цей модуль приїжджає із репозиторіїв дистрибутиву, має бути і так свіжий

```

atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> python3 -m pip install --upgrade pip
error: externally-managed-environment

× This environment is externally managed
  ╰─> To install Python packages system-wide, try 'pacman -S
python-xyz', where xyz is the package you are trying to
install.

If you wish to install a non-Arch-packaged Python package,
create a virtual environment using 'python -m venv path/to/venv'.
Then use path/to/venv/bin/python and path/to/venv/bin/pip.

If you wish to install a non-Arch packaged Python application,
it may be easiest to use 'pipx install xyz', which will manage a
virtual environment for you. Make sure you have python-pipx
installed via pacman.

Note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can override this, at the risk of breaking your Python installation or OS, by passing --break-system-packages.
Hint: See PEP 668 for the detailed specification.
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master) [1]:> python3 -m pip --version
pip 25.0.1 from /usr/lib/python3.13/site-packages/pip (python 3.13)
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)>

```

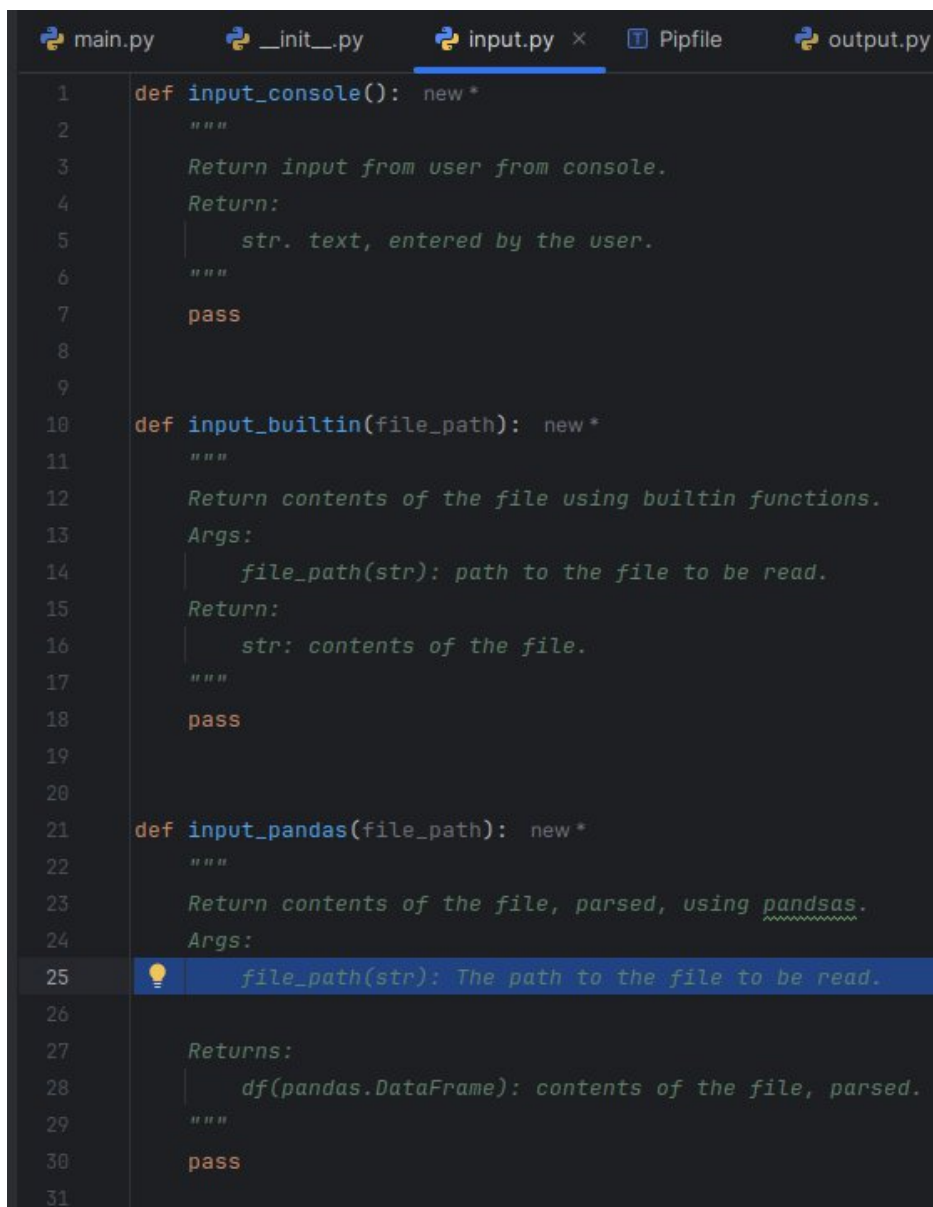
Встановимо залежності


```
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> pipenv install numpy
Pipfile.lock not found, creating...
Locking [packages] dependencies...
Locking [dev-packages] dependencies...
Updated Pipfile.lock (c48f752b1faa88b8518fb30e2dd92d35b53d7f495af82ecd10aa1cbc7e7aab87)!
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing numpy...
✓ Installation Succeeded
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (7aab87)...
All dependencies are now up-to-date!
Upgrading numpy in dependencies.
Building requirements...
Resolving dependencies...
✓ Success!
Building requirements...
Resolving dependencies...
✓ Success!
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (cc23f8)...
All dependencies are now up-to-date!
Installing dependencies from Pipfile.lock (cc23f8)...
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> pipenv install seaborn==0.13.1
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing seaborn==0.13.1...
✓ Installation Succeeded
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (cc23f8)...
All dependencies are now up-to-date!
Upgrading seaborn==0.13.1 in dependencies.
Building requirements...
Resolving dependencies...
✓ Success!
Building requirements...
Resolving dependencies...
✓ Success!
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (d35e9a)...
All dependencies are now up-to-date!
Installing dependencies from Pipfile.lock (d35e9a)...
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> pipenv install matplotlib pylint black
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing matplotlib...
✓ Installation Succeeded
Installing pylint...
✓ Installation Succeeded
Installing black...
✓ Installation Succeeded
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (d35e9a)...
All dependencies are now up-to-date!
Upgrading matplotlib, pylint, black in dependencies.
Building requirements...
Resolving dependencies...
✓ Success!
Building requirements...
Resolving dependencies...
✓ Success!
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (78a495)...
All dependencies are now up-to-date!
Installing dependencies from Pipfile.lock (78a495)...
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)>
```

І заморозимо їх

```
Installing dependencies from Pipfile.lock (78a495)...
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> pipenv lock
Locking [packages] dependencies...
Building requirements...
Resolving dependencies...
✓ Success!
Locking [dev-packages] dependencies...
Updated Pipfile.lock (25f31096c6b2b2579f69bc5646af2c5b0d327bc2b05ba888c1f5c6670a78a495)!
atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)> █
```

Додав заглушки до функцій



```
main.py  __init__.py  input.py  Pipfile  output.py

1  def input_console(): new *
2      """
3      Return input from user from console.
4      Return:
5      |     str. text, entered by the user.
6      """
7      pass
8
9
10 def input_builtin(file_path): new *
11     """
12     Return contents of the file using builtin functions.
13     Args:
14     |     file_path(str): path to the file to be read.
15     Return:
16     |     str: contents of the file.
17     """
18     pass
19
20
21 def input_pandas(file_path): new *
22     """
23     Return contents of the file, parsed, using pandas.
24     Args:
25     |     file_path(str): The path to the file to be read.
26
27     Returns:
28     |     df(pandas.DataFrame): contents of the file, parsed.
29     """
30     pass
31
```

```
main.py  __init__.py  input.py  Pipeline  output.py x
1  def output_console(text):  Artem Trokhymchuk
2      """
3      Print text to the console.
4      Args:
5          text (str): text to display to print in console.
6      """
7      pass
8
9
10 def output_builtin(text, file_path):  Artem Trokhymchuk
11     """
12     Print text to file using built-in functions.
13     Args:
14         text (str): text to display to print in console.
15         file_path(str): The path to the file to output to.
16     """
17     pass
18
```

Написав main


```
main.py ×  output.txt  input.csv  input.py  Pipfile  output.py  .gitignore
1  from app.io.input import input_builtin, input_console, input_pandas
2  from app.io.output import output_console, output_builtin
3
4
5  def main(): 1 usage  Artem Trokhymchuk *
6      output_console("Please, enter path to the input file: ")
7      input_file_path = input_console()
8      string_file_contents = input_builtin(input_file_path)
9      pandas_file_contents = input_pandas(input_file_path)
10
11      output_console(input_file_path)
12      output_console(string_file_contents)
13      output_console(pandas_file_contents.to_string())
14
15      output_console("Please, enter path to the output file: ")
16      output_file_path = input_console()
17      output_builtin(input_file_path, output_file_path)
18      output_builtin(string_file_contents, output_file_path)
19      output_builtin(pandas_file_contents.to_string(), output_file_path)
20      pass
21
22
23  if __name__ == '__main__':
24      main()
25
```

Реалізував функції

main.py output.txt input.csv input.py × Pipfile output.py .gitignore

```
1 import pandas
2
3 def input_console(): 3 usages  ⚠ Artem Trokhymchuk *
4     """
5     Return input from user from console.
6     Return:
7     |     str. text, entered by the user.
8     """
9     return input()
10
11
12 def input_builtin(file_path): 2 usages  ⚠ Artem Trokhymchuk *
13     """
14     Return contents of the file using builtin functions.
15     Args:
16     |     file_path(str): path to the file to be read.
17     Return:
18     |     str: contents of the file.
19     """
20     return open(file_path, encoding='utf-8').read()
21
22
23 def input_pandas(file_path, format = 'csv'): 2 usages  ⚠ Artem Trokhymchuk *
24     """
25     Return contents of the file, parsed, using pandas.
26     Args:
27     |     file_path(str): The path to the file to be read.
28     |     format(str): format of the input file, csv and json are supported.
29     Returns:
30     |     df(pandas.DataFrame): contents of the file, parsed.
31     Throws:
32     |     ValueError: if format is unsupported
33     """
34     if format == 'csv':
35         return pandas.read_csv(file_path)
36     elif format == 'json':
37         return pandas.read_json(file_path)
38     else:
39     💡 raise ValueError(f"Format '{format}' is not supported")
```

```
main.py  output.txt  input.csv  input.py  Pipfile  output.py  .gitign
1  def output_console(text): 6 usages  Artem Trokhymchuk *
2      """
3      Print text to the console.
4      Args:
5          text (str): text to display to print in console.
6      """
7      print(text)
8
9
10 def output_builtin(text, file_path): 4 usages  Artem Trokhymchuk *
11     """
12     Print text to file using built-in functions.
13     Args:
14         text (str): text to display to print in console.
15         file_path(str): The path to the file to output to.
16     """
17     open(file_path, 'w', encoding='utf-8').write(text)
18
```

Закомітили, ідемо далі, написання тестів, отака структура пакетів

```
1 2 3 4 5 6
pipenv ~/uni/uni/python/pipenv
├── app
│   ├── io
│   │   ├── __init__.py
│   │   ├── input.py
│   │   ├── output.py
│   │   └── __init__.py
│   ├── data
│   │   ├── input.csv
│   │   └── output.txt
│   ├── tests
│   │   ├── test_io
│   │   │   ├── __init__.py
│   │   │   └── test_input.py
│   │   ├── __init__.py
│   │   ├── .gitignore
│   │   ├── main.py
│   │   ├── Pipfile
│   │   └── Pipfile.lock
├── External Libraries
└── Scratches and Consoles
```

Стикнувся із помилкою:

```
main.py  output.txt  input.csv  input.py  Pipfile  output.py  .gitignore  test_input.py x
1      from input import input_console
2
3
4  def test_input_console():
5      assert True
6
```

Виявилось, треба було імпортувати через `app.io.input`

```
1      from app.io.input import input_console
2
3
4  def test_input_console():
5      assert True
6
```

```
pipenv atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master) [2]> pytest
===== test session starts =====
platform linux -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0
rootdir: /home/atrokhymchuk/uni/uni/python/pipenv
collected 1 item

tests/test_io/test_input.py .
===== 1 passed in 0.16s =====
pipenv atrokhymchuk@atrokhymchuk-laptop ~/u/u/p/pipenv (master)>
```

А щоб отримати `pytest` в PATH достатньо виконати `pipenv shell`. Але тепер я дочитав і зрозумів, що тестувати інпут з консолі не треба, змінив на тестування `input_builtin`. Використав тимчасові файли для тестування

```
6  def test_input_builtin():
7      test_content = "Temporary file content for testing."
8
9      with tempfile.NamedTemporaryFile(mode='w+', encoding='utf-8', delete=True) as tmp_file:
10         tmp_file.write(test_content)
11         tmp_file.flush()
12         result = input_builtin(tmp_file.name)
13
14         assert result == test_content
15
```

З `Pandas` прийшлося поводитись, але вийшло.

```

17 ▶ def test_input_pandas_no_extension():
18     expected_pandas_dataframe = pandas.DataFrame({
19         "a": ["1", "artem"],
20         "b": ["2", "trokhymchuk"],
21         "c": ["3", "artem"],
22     })
23
24     ⚡ test_content = expected_pandas_dataframe.to_csv(index=False)
25
26     with tempfile.NamedTemporaryFile(mode='w+', encoding='utf-8', delete=True) as tmp_file:
27         tmp_file.write(test_content)
28         tmp_file.flush()
29         result = input_pandas(tmp_file.name)
30
31     # print(expected_pandas_dataframe)
32     # print("===")
33     # print(result)
34     pandas.testing.assert_frame_equal(result, expected_pandas_dataframe)
35
36
37 ▶ def test_input_pandas_csv_extension():
38     expected_pandas_dataframe = pandas.DataFrame({
39         "a": ["1", "artem"],
40         "b": ["2", "trokhymchuk"],
41         "c": ["3", "artem"],
42     })
43
44     test_content = expected_pandas_dataframe.to_csv(index=False)
45
46     with tempfile.NamedTemporaryFile(mode='w+', encoding='utf-8', delete=True) as tmp_file:
47         tmp_file.write(test_content)
48         tmp_file.flush()
49         result = input_pandas(tmp_file.name, format='csv')
50     pandas.testing.assert_frame_equal(result, expected_pandas_dataframe)
51
52
53 ▶ def test_input_pandas_json_extension():
54     expected_pandas_dataframe = pandas.DataFrame({
55         "a": ["1", "artem"],
56         "b": ["2", "trokhymchuk"],
57         "c": ["3", "artem"],
58     })
59
60     test_content = expected_pandas_dataframe.to_json()
61
62     with tempfile.NamedTemporaryFile(mode='w+', encoding='utf-8', delete=True) as tmp_file:
63         tmp_file.write(test_content)
64         tmp_file.flush()
65         result = input_pandas(tmp_file.name, format='json')
66
67     pandas.testing.assert_frame_equal(result, expected_pandas_dataframe)
68

```

Висновки

- a. Створили проект, в ньому використали `pipenv` для створення віртуального середовища для `pip`, щоб встановлені пакети не конфліктували із ситемними пакетами, в проекті створили декілька пакетів, `app.io` відповідає за ввід-вивід інформації користувачем, а `test.test_io` за тестування вводу.
- b. Дізнався декілька моментів
 - a. По-перше, я раніше думав що `pipenv` це скоріше як `version manager` для `python + pip`, виявилось, це не так.
 - b. В пітоні дуже зручно створювати тимчасові файли, я чогось думав, що буде складніше і мені не можна буде ім'я тимчасового файлу взяти.
 - c. Існує бібліотека для читання файлів різних форматів і вона називається на честь панд.
- c. Корисно дізнатися про організацію пакетів в пітон, про віртуальні середовища для `pip`.
- d. Може трошки детальніше про структуру директорій, а то я десь до останнього пункту не був впевнений, що правильно все створив, а так все прекрансо.