

УПРАВЛЕНИЕ ЗАВИСИМОСТЯМИ. ИНСТРУМЕНТЫ PIP, PIPENV, PYENV, POETRY, UV

Выполнили студенты ИВТ 2.2
Киняев И.К., Магер Е.В.,
Нахатакян А.Р.

УПРАВЛЕНИЕ ЗАВИСИМОСТЯМИ

Это процесс контроля внешних библиотек (пакетов), которые использует проект, включая:

- **Установку нужных версий.**
- **Изоляцию окружений для разных проектов.**
- **Гарантию воспроизводимости сборки.**

PIP

PIP

PIP - это система управления пакетами в Python, которая упрощает установку, обновление и удаление программных пакетов, написанных на Python. PIP, начиная с версии Python 2.7.9 и Python 3.4, поставляется по умолчанию вместе с интерпретатором Python. Устанавливает пакеты из PyPI.

Основные команды

- `pip install package_name`
- `pip install --upgrade package_name`
- `pip install -r requirements.txt`
- `pip freeze > requirements.txt`

PIPVENV

В чем отличие от pip

Pipenv — инструмент, объединяющий pip и virtualenv. Автоматически:

- Создаёт виртуальные окружения.
- Управляет зависимостями через Pipfile и Pipfile.lock.

Основные команды

- `pipenv shell`
- `pipenv install scrapy`
- `pipenv install`
- `pipenv sync`

```
[[source]]
name = "pypi"
url = "https://pypi.org/simple"
verify_ssl = true

[dev-packages]

[packages]
questionary = "*"

[requires]
python_version = "3.7"
```

PYENV

Управление версиями python

Pyenv — это мощный инструмент для управления версиями Python, который позволяет легко устанавливать и переключаться между различными версиями Python

Основные команды

- `pyenv install <версия>`
- `pyenv local <версия>`
- `pyenv global 3.9.1`
- `pyenv versions`

POETRY

Poetry — современный инструмент для управления зависимостями и сборки Python-пакетов.

Poetry предоставляет следующие возможности:

- управление зависимостями с воспроизводимыми установками и резолвером конфликтов;
- автоматическое управление виртуальными окружениями;
- сборка и публикация.

Основные команды

- `poetry init`
- `poetry add package_name`
- `poetry install`
- `poetry shell`

UV

```
.gitignore  
.python-version  
hello.py  
pyproject.toml  
README.md
```

```
[project]  
name = "venv"  
version = "0.1.0"  
description = "Add your description here"  
readme = "README.md"  
requires-python = ">=3.13"  
dependencies = [  
    "requests>=2.32.3",  
]
```

Управление версиями python

Uv - довольно быстрый пакетный менеджер и менеджер виртуальных окружений для Python, написанный на Rust. По заявлению разработчиков, uv способен заменить такие Python инструменты, как pip, pip-tools, pipx, poetry, pyenv, twine, virtualenv и другие.

Основные команды

- uv init
- uv add requests
- uv sync
- pyenv versions

СПАСИБО
ЗА ВНИМАНИЕ!
