

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**Дисциплина «Языки программирования»**

**Отчет по практической работе № 2.14**

**Установка пакетов в Python. Виртуальные окружения**

Выполнил: студент группы ИТС-б-о-21-1  
Крамаренко Илья Витальевич

---

(подпись)

Проверил: к.т.н., доцент  
Кафедры инфокоммуникаций  
Воронкин Р.А.

---

(подпись)

Ставрополь, 2022

## Установка пакетов в Python. Виртуальные окружения

**Цель работы:** приобретение навыков по работе с менеджером пакетов pip и виртуальными окружениями с помощью языка программирования Python версии 3.x.

### Ход работы:

1. Создали новое виртуальное окружения в анаконде

```
(prog) C:\Users\Admin\Desktop\4_lr\prog> python3 -m venv env  
(prog) C:\Users\Admin\Desktop\4_lr\prog> env\Scripts\activate  
(env) (prog) C:\Users\Admin\Desktop\4_lr\prog> python3 -m venv /home/user/envs/project1_env
```

Рисунок 1. Процесс создания виртуального окружения

2. Активировали свое виртуальное окружение

```
(base) C:\Users\Admin\Desktop\4_lr\prog>conda activate prog  
(prog) C:\Users\Admin\Desktop\4_lr\prog>
```

Рисунок 2. Активация виртуального окружения

```
(prog) C:\Users\Admin\Desktop\4_lr\prog>conda install pip, NumPy, Pandas, SciPy.  
Collecting package metadata (current_repodata.json): done  
Solving environment: failed with initial frozen solve. Retrying with flexible solve.  
Collecting package metadata (repodata.json): done  
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
```

Рисунок 3. Установка в виртуальное окружение следующие пакеты:

pip, NumPy, Pandas, SciPy.

Процесс скачивания TensorFlow.

```
(prog) C:\Users\Admin\Desktop\4_lr\prog> pip install TensorFlow  
Collecting TensorFlow  
  Downloading tensorflow-2.10.0-cp37-cp37m-win_amd64.whl (455.9 MB)  
10.5/455.9 MB 444.2 kB/s eta 0:16:43
```

Рисунок 4. Процесс скачивание TensorFlow.

```
(prog) C:\Users\Admin\Desktop\4_lr\prog>pip freeze > requirements.txt  
(prog) C:\Users\Admin\Desktop\4_lr\prog>conda env export > environment.yml
```

Рисунок 4. Сформированы файлы requirements.txt и environment.yml .

### Вопросы для защиты работы

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

Существует так называемый Python Package Index (PyPI) – это репозиторий, открытый для всех Python разработчиков, в нем вы можете найти пакеты для решения практически любых задач. Там также есть возможность выкладывать свои пакеты. Для скачивания и установки используется специальная утилита, которая называется `pip`.

2. Как осуществить установку менеджера пакетов `pip`?

При развертывании современной версии Python (начиная с Python 2.7.9 и Python 3.4), `pip` устанавливается автоматически. Но если, по какой-то причине, `pip` не установлен на вашем ПК, то сделать это можно вручную. Существует несколько способов. Для того, чтобы это сделать, скачайте скрипт `get-pip.py`

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

`Pip` позволяет установить самую последнюю версию пакета, конкретную версию или воспользоваться логическим выражением, через которое можно определить, что вам, например, нужна версия не ниже указанной. Также есть поддержка установки пакетов из репозитория.

4. Как установить последнюю версию пакета с помощью `pip`?

Установка последней версии пакета

```
$ pip install ProjectName
```

5. Как установить заданную версию пакета с помощью `pip`?

Установка определенной версии

```
$ 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?

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?

Основные этапы работы с виртуальным окружением:

1. Создаём через утилиту новое виртуальное окружение в отдельной папке для выбранной версии интерпретатора Python.

2. Активируем ранее созданное виртуального окружения для работы.

3. Работаем в виртуальном окружении, а именно управляем пакетами используя pip и запускаем выполнение кода.

4. Деактивируем после окончания работы виртуальное окружение.

5. Удаляем папку с виртуальным окружением, если оно нам больше не нужно.

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>
```

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

Создание виртуального окружения с утилитой `virtualenv` отличается от стандартного. Например, создание в текущей папке виртуального окружения для интерпретатора доступного через команду `python3` с названием папки окружения `env`:

```
virtualenv -p python3 env
```

14. Изучите работу с виртуальными окружениями `pipenv`. Как осуществляется работа с виртуальными окружениями `pipenv`?

Благодаря этому инструменту работа с Python и Виртуальными Окружениями выходит на новый уровень, поскольку он сочетает в себе поддержку управления пакетами и окружением.

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

**requirements.txt** является стандартной договоренностью и используется некоторыми утилитами автоматически. Установка пакетов из файла зависимостей в новом виртуальном окружении так же выполняется одной командой:

```
pip install -r requirements.txt
```

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.

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?

Для Windows, если используется дистрибутив Anaconda, то необходимо вначале запустить консоль Anaconda Powershell Prompt. Делается это из системного меню, посредством выбора следующих пунктов: Пуск Anaconda3 (64-bit) Anaconda Powershell Prompt (Anaconda3). В результате будет отображено окно консоли,

19. Как создать виртуальное окружение conda?

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

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

```
conda activate %PROJ_NAME%
```

21. Как деактивировать и удалить виртуальное окружение conda?

```
conda deactivate
```

Если вы хотите удалить только что созданное окружение, выполните:

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

22. Каково назначение файла environment.yml ?

Файл environment.yml позволит воссоздать окружение в любой нужный момент.

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

```
conda env create -f environment.yml
```