

**РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное**  
**образовательное учреждение высшего образования**  
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**  
**Кафедра инфокоммуникаций**

«Работа с переменными окружения в Python3»

Отчет по лабораторной работе № 2.18

по дисциплине «Программирование на Python»

Выполнил студент группы ИВТ-б-о-21-1

Толубаев Рамиль Ахметович

Проверил Воронкин Р.А.

(подпись)

Ставрополь 2022

**Цель работы:** приобретение навыков по работе с переменными окружения с помощью языка программирования Python версии 3.x.

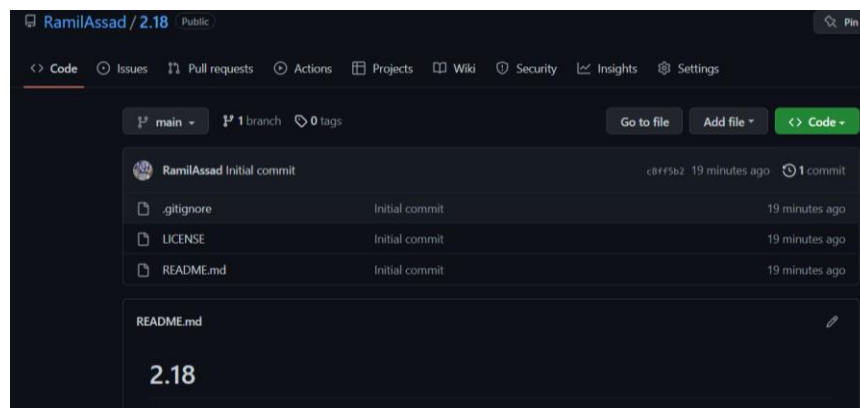


Рисунок 1 - Создание репозитория

```
C:\Users\User>cd C:\Users\User\Desktop\2 кypc Python\lab 21
C:\Users\User\Desktop\2 кypc Python\lab 21>git clone https://github.com/aikanyshkaukanbekova/2.18.git
Cloning into '2.18'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), 4.51 KiB | 660.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
C:\Users\User\Desktop\2 кypc Python\lab 21>
```

Рисунок 2 - Клонирование репозитория

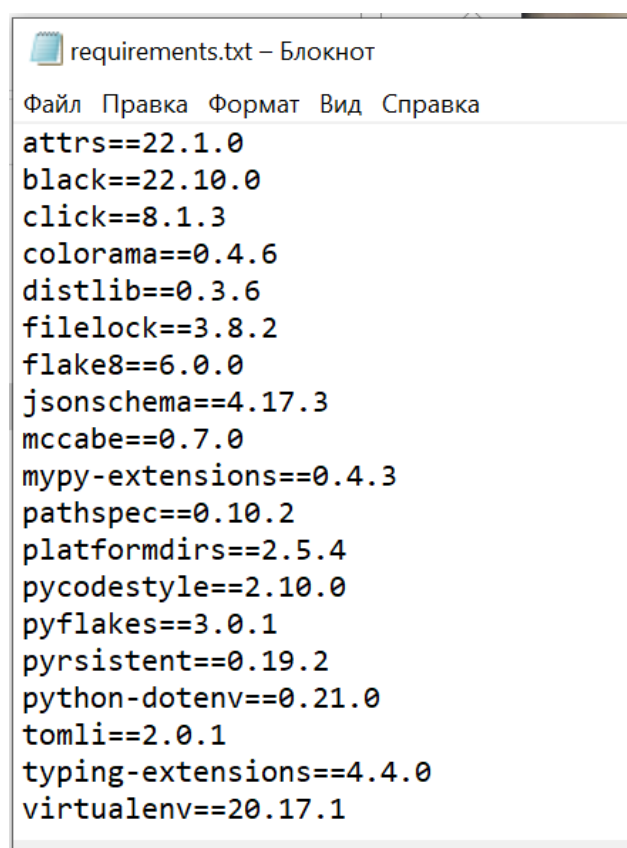


Рисунок 3 - Файл requirements.txt

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import argparse
5  import json
6  import os
7  import sys
8  from datetime import date
9
10
11 def add_worker(staff, name, post, year):
12     """
13     Добавить данные о работнике.
14     """
15     staff.append(
16         {
17             "name": name,
18             "post": post,
19             "year": year
20         }
21     )
22     return staff
23
24
25 def display_workers(staff):
26     """
27     Отобразить список работников.
28     """
29     # Проверить, что список работников не пуст.
30     if staff:
```

Рисунок 4 - Результат выполнения примера

## Индивидуальное задание 1

```
File Edit View Navigate Code Refactor Run Tools Git Window Help
OneDrive Рабочий стол Python 2.18 ind1.py
Project primer1.py ind1.py Сложная задача.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  import argparse
6  import json
7  import os.path
8  import sys
9
10
11 def add_route(routes, start, finish, number):
12     """
13     Добавить данные о маршруте
14     """
15
16     routes.append(
17         {
18             'start': start,
19             'finish': finish,
20             'number': number
21         }
22     )
23     return routes
24
25
26 def display_route(routes):
27     """
28     Отобразить список маршрутов
29     """
30     if routes:
```

Рисунок 5 - Результат выполнения задания 1

## Индивидуальное задание 2

```
15         "И",
16         "Место прибытия",
17         "Номер самолёта",
18         "Тип"))
19     print(line)
20     for i, num in enumerate(flights, 1):
21         if typing == num.get('value', ''):
22             count += 1
23             print(
24                 '| {:<4} | {:<20} | {:<15} | {:<16} |'.format(
25                     count,
26                     num.get('stay', ''),
27                     num.get('number', ''),
28                     num.get('value', 0)))
29     print(line)
30
31
32 def opening(filename):
33     with open(filename, "r", encoding="utf-8") as f_in:
34         return json.load(f_in)
35
36
37 def main():
38     cli()
39
40
41 if __name__ == '__main__':
42     main()
```

Рисунок 6 - Результат выполнения задания 2

Контрольные вопросы:

### 1. Каково назначение переменных окружения?

Переменная среды (переменная окружения) – это короткая ссылка на какой-либо объект в системе. С помощью таких сокращений, например, можно создавать универсальные пути для приложений, которые будут работать на любых ПК, независимо от имен пользователей и других параметров.

### 2. Какая информация может храниться в переменных окружения?

Переменная окружения может хранить информацию о путях к исполняемым файлам, заданном по умолчанию текстовом редакторе, браузере, языковых параметрах (локали) системы или настройках раскладки клавиатуры.

### 3. Как получить доступ к переменным окружения в ОС Windows?

Получить информацию о существующих переменных можно в

свойствах системы. Для этого кликаем по ярлыку Компьютера на рабочем столе правой кнопкой мыши и выбираем соответствующий пункт.

Переходим в «Дополнительные параметры».

В открывшемся окне с вкладкой «Дополнительно» нажимаем кнопку «Переменные среды».

#### **4. Каково назначение переменных PATH и PATHEXT?**

«PATH» позволяет запускать исполняемые файлы и скрипты,

«лежащие» в определенных каталогах, без указания их точного местоположения.

PATHEXT, в свою очередь, дает возможность не указывать даже расширение файла, если оно прописано в ее значениях.

#### **5. Как создать или изменить переменную окружения в Windows?**

Нажимаем кнопку Создать. Сделать это можно как в пользовательском разделе, так и в системном.

Вводим имя, например, desktop. Обратите внимание на то, чтобы такое название еще не было использовано (просмотрите списки).

В поле Значение указываем путь.

#### **6. Что представляют собой переменные окружения в ОС Linux?**

Переменные окружения в Linux представляют собой набор именованных значений, используемых другими приложениями.

#### **7. В чем отличие переменных окружения от переменных оболочки?**

Переменные окружения (или «переменные среды») — это переменные, доступные в масштабах всей системы и наследуемые всеми дочерними процессами и оболочками.

Переменные оболочки — это переменные, которые применяются только к текущему экземпляру оболочки. Каждая оболочка, например, bash или zsh, имеет свой собственный набор внутренних переменных.

#### **8. Как вывести значение переменной окружения в Linux?**

Наиболее часто используемая команда для вывода переменных окружения — `printenv`.

#### **9. Какие переменные окружения Linux Вам известны?**

USER — текущий пользователь. PWD — текущая директория.

OLDPWD — предыдущая рабочая директория. Используется оболочкой для того, чтобы вернуться в предыдущий каталог при выполнении команды `cd -`.

HOME — домашняя директория текущего пользователя.

SHELL — путь к оболочке текущего пользователя (например, `bash` или `zsh`).

EDITOR — заданный по умолчанию редактор. Этот редактор будет вызываться в ответ на команду `edit`.

LOGNAME — имя пользователя, используемое для входа в систему.

PATH — пути к каталогам, в которых будет производиться поиск вызываемых команд. При выполнении команды система будет проходить по данным каталогам в указанном порядке и выберет первый из них, в котором будет находиться исполняемый файл искомой команды.

LANG — текущие настройки языка и кодировки. TERM — тип текущего эмулятора терминала.

MAIL — место хранения почты текущего пользователя.

LS\_COLORS — задает цвета, используемые для выделения объектов (например, различные типы файлов в выводе команды `ls` будут выделены разными цветами).

## **10. Какие переменные оболочки Linux Вам известны?**

BASHOPTS — список задействованных параметров оболочки, разделенных двоеточием.

BASH\_VERSION — версия запущенной оболочки `bash`.

COLUMNS — количество столбцов, которые используются для отображения выходных данных.

DIRSTACK — стек директорий, к которому можно применять команды `pushd` и `popd`.

HISTFILESIZE — максимальное количество строк для файла истории команд.

HISTSIZE — количество строк из файла истории команд, которые можно хранить в памяти.

HOSTNAME — имя текущего хоста.

IFS — внутренний разделитель поля в командной строке (по умолчанию используется пробел).

PS1 — определяет внешний вид строки приглашения ввода новых команд.

PS2 — вторичная строка приглашения.

SHELLOPTS — параметры оболочки, которые можно устанавливать с помощью команды `set`.

UID — идентификатор текущего пользователя.

## **11. Как установить переменные оболочки в Linux?**

Чтобы создать новую переменную оболочки с именем, например, `NEW_VAR` и значением `Ravesli.com`, просто введите:

```
$ NEW_VAR='Ravesli.com'
```

## **12. Как установить переменные окружения в Linux?**

Команда `export` используется для задания переменных окружения.

## **13. Для чего необходимо делать переменные окружения Linux постоянными?**

Чтобы переменная сохранялась после закрытия сеанса оболочки.

## **14. Для чего используется переменная окружения PYTHONHOME?**

Переменная среды `PYTHONHOME` изменяет расположение стандартных библиотек Python.

## **15. Для чего используется переменная окружения PYTHONPATH?**

Переменная среды `PYTHONPATH` изменяет путь поиска по умолчанию для файлов модуля.

## **16. Какие еще переменные окружения используются для управления работой интерпретатора Python?**

`PYTHONSTARTUP`, `PYTHONOPTIMIZE`, `PYTHONBREAKPOINT`,  
`PYTHONDEBUG`, `PYTHONINSPECT`, `PYTHONUNBUFFERED`,

`PYTHONVERBOSE`, `PYTHONCASEOK`,  
`PYTHONDONTWRITEBYTECODE`, `PYTHONPYCACHEPREFIX`,  
`PYTHONHASHSEED`, `PYTHONIOENCODING`, `PYTHONNOUSERSITE`,  
`PYTHONUSERBASE`, `PYTHONWARNINGS`, `PYTHONFAULTHANDLER`,

PYTHONTRACEMALLOC, PYTHONPROFILEIMPORTTIME,  
PYTHONASYNCIODEBUG, PYTHONMALLOC,  
PYTHONMALLOCSTATS,

PYTHONLEGACYWINDOWSFSENCODING,  
PYTHONLEGACYWINDOWSSTDIO, PYTHONCOERCECLOCALE,  
PYTHONDEVMODE, PYTHONUTF8, PYTHONWARNDEFAULTENCODING,  
PYTHONTHREADDEBUG, PYTHONDUMPPREFS.

**17. Как осуществляется чтение переменных окружения в программах на языке программирования Python?**

Для доступа к переменным среды в Python используется объект `os.environ`.

**18. Как проверить, установлено или нет значение переменной окружения в программах на языке программирования Python?**

Для чтения значений переменных мы используем модуль `os`, а модуль `sys` — для прекращения работы приложения.

**19. Как присвоить значение переменной окружения в программах на языке программирования Python?**

Для присвоения значения любой переменной среды используется функция `setdefault()`.

**Вывод:** были приобретены навыки по работе с переменными окружения с помощью языка программирования Python версии 3.x.