

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

Стригалов Дмитрий.

«20» декабря 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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

Порядок выполнения работы:

Создание виртуального окружения.

```
d:\Git\Lab2.18>python -m venv .venv

d:\Git\Lab2.18>.venv\Scripts\activate

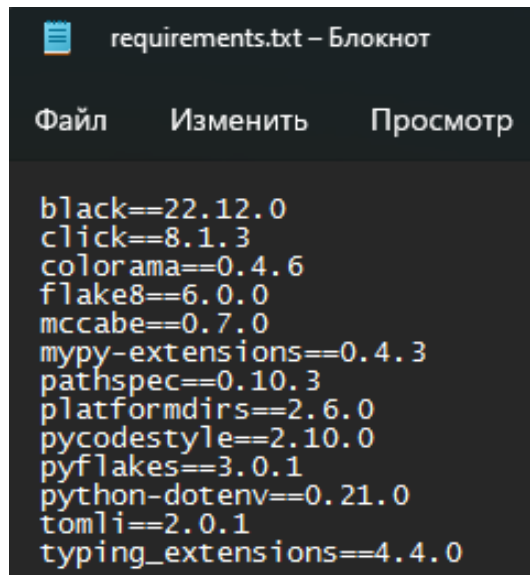
(.venv) d:\Git\Lab2.18>pip install black, flake8
ERROR: Invalid requirement: 'black,'

[notice] A new release of pip available: 22.3 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(.venv) d:\Git\Lab2.18>pip install black flake8
Collecting black
  Using cached black-22.12.0-cp311-cp311-win_amd64.whl (1.2 MB)
Collecting flake8
  Using cached flake8-6.0.0-py2.py3-none-any.whl (57 kB)
Collecting click>=8.0.0
```

Рисунок 1 - Виртуальное окружение.

Формирование файла requirements.txt.



```
requirements.txt - Блокнот

Файл  Изменить  Просмотр

black==22.12.0
click==8.1.3
colorama==0.4.6
flake8==6.0.0
mccabe==0.7.0
mypy_extensions==0.4.3
pathspec==0.10.3
platformdirs==2.6.0
pycodestyle==2.10.0
pyflakes==3.0.1
python-dotenv==0.21.0
tomli==2.0.1
typing_extensions==4.4.0
```

Рисунок 2 - Файл requirements.txt

Проработать примеры лабораторной работы.

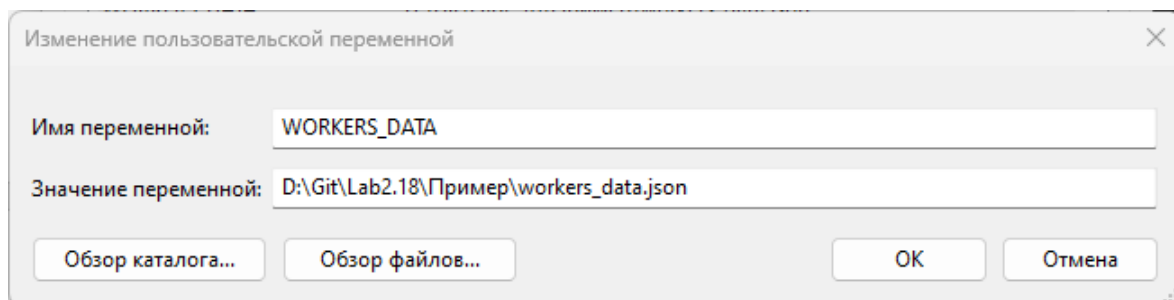


Рисунок 3 - Создание системной переменной WORKERS_DATA

```
d:\Git\Lab2.18\Пример>python primer.py add --name="Сидоров Сидор" --post="Главный инженер" --year=2012
d:\Git\Lab2.18\Пример>python primer.py display
```

| No | Ф.И.О. | Должность | Год |
|----|---------------|-----------------|------|
| 1 | Сидоров Сидор | Главный инженер | 2012 |

Рисунок 4 - Результат выполнения примера

8. Выполнить индивидуальные задания.

Задание 1

Для своего варианта лабораторной работы 2.17 добавьте возможность получения имени файла данных, используя соответствующую переменную окружения.

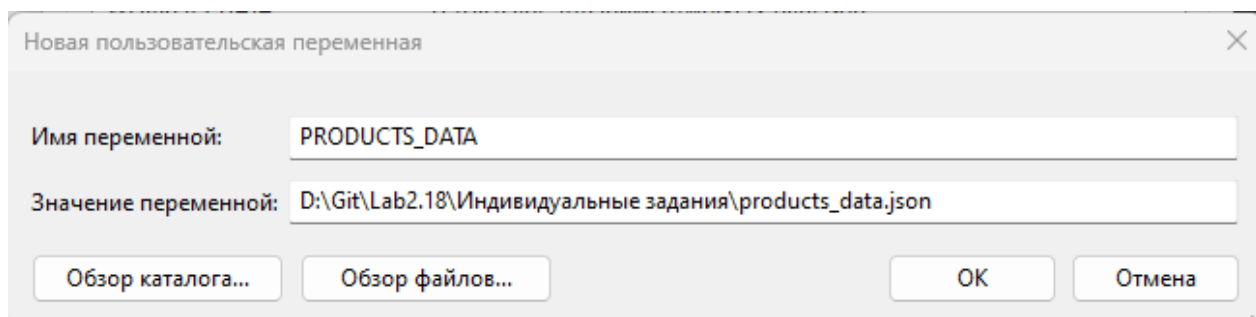


Рисунок 5 - Создание переменной окружения PRODUCTS_DATA

```
:\Git\Lab2.18\Индивидуальные задания>python ind1.py select -s="sapog"
```

| % | Название товара | Цена | Название магазина |
|---|-----------------|----------|-------------------|
| 1 | sapog | 53425.00 | dior |

```
:\Git\Lab2.18\Индивидуальные задания>python ind1.py display
```

| % | Название товара | Цена | Название магазина |
|---|-----------------|----------|-------------------|
| 1 | kurtka | 3213.00 | armani |
| 2 | sapog | 53425.00 | dior |

Рисунок 6 - Результат выполнения задания 1

Задание 2

Самостоятельно изучите работу с пакетом `python-dotenv`. Модифицируйте программу задания 1 таким образом, чтобы значения необходимых переменных окружения считывались из файла `.env`.

```
:\Git\Lab2.18\Индивидуальные задания>python ind2.py display
```

| % | Название товара | Цена | Название магазина |
|---|-----------------|----------|-------------------|
| 1 | kurtka | 3213.00 | armani |
| 2 | sapog | 53425.00 | dior |
| 3 | шапка | 500.00 | klips |
| 4 | носки | 300.00 | klips |

```
:\Git\Lab2.18\Индивидуальные задания>python ind2.py select -s="шапка"
```

| % | Название товара | Цена | Название магазина |
|---|-----------------|--------|-------------------|
| 1 | шапка | 500.00 | klips |

Рисунок 7 - 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.