UDC 004.432.2 **doi: 10.26906/SUNZ.2018.4.095**

S. Alyoshin, E. Borodina, A. Kikot, I. Zabran

Poltava National Technical Yuriy Kondratyuk University, Poltava, Ukraine

PYTHON PROGRAMMING FEATURES AND NEW POSSIBILITIES

Purpose. The purpose of the article is to demonstrate the profitability of new versions of the Python programming language, and also to demonstrate the volumes of this language together with new software. The content of the article is an overview that allows you to understand the features and new features of the Python language. **Results.** It was found that to create programs for different purposes, use a powerful tool Python. **Originality.** Programs written in Python work in exactly the same way, regardless of which operating system they are running on, and also Python adds features that make it widely used. **Practical value.** The cost-effectiveness of new versions of Python and the volumes of this language together with the new provision provides a detailed analysis of the areas where Python is used. The conclusion is that Python has some noteworthy features that make it widely used. **Conclusions.** The arguments we have presented prove that the program written in Python will function exactly the same regardless of which operating system it is running in. Differences arise only in rare cases, and they are easy to anticipate due to the availability of detailed documentation.

Keywords: Python, professional language, features, possibilities of Python.

Introduction

YouTube, DropBox, Google, Quora, Instagram, BitTorrent, Spotify, Reddit, Yahoo Maps, Hipmunk. The thing that unites them is Python. [1] "General purposeness" of the language resulted into extraordinary large field of its application.

A relevance of this article is that for the time being Python is the fifth most used as a professional language in Ukraine and the eighth in the list of languages a person wants to learn next according to Dou [2].

A new element is that there is no comprehensive and complex article that would cover all below-mentioned questions from different views and both for prepared and non-prepared readers.

The purpose of article is to demonstrate profitability of new versions of Python and show the scope of this language along with new soft.

The tasks are:

1. To consider new versions of Python and their features.
2. To describe a scope of Python and new soft for its usage.

Main part

Let us start with first item and examine advantages and specificities of 3.5 version of Python that was released in 2008. Despite the mentioned, most people still prefer to use v.2.7 [3].

Below is the list of minor and not so minor features supposed to change their preferences [3]:

1. Advanced unpacking

You can already do this:

```
>>> a, b = range(2)
>>> a
0
>>> b
1
```

Now you can do this:

```
>>> a, b, *rest = range (10)
>>> a
0
>>> b
1
>>> rest
[2, 3, 4, 5, 6, 7, 8, 9]
```

2. Keyword only arguments

```
def f(a, b, *args, option=True):
                ...
```

- option comes after *args.
- The only way to access it is to explicitly call f(a, b, option=True)
- You can write just a * if you do not want to collect *args.

```
def f(a, b, *, option=True):
```

3. Chained exceptions

Python 3 shows you the whole chain of exceptions.

4. Everything is an iterator

In Python 3, *range, zip, map, dict.values*, etc. all return memory-efficient iterables. If you want a list, just wrap the result with list.

This way it is harder to write code that accidentally uses too much memory, because the input was bigger than you expected.

5. No more comparison of everything to everything

In Python 2, you can do:

```
>>> max(['one', 2])  # One "is" the loneliest number
'one'
```

It is because in Python 2, you can < compare anything to anything. In Python 3, you cannot do this:

```
>>> 'one' > 2
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: unorderable types: str() > int()
```

6. Yield from

Instead of accumulating a list, just yield or yield from.

1) Bad

```
def dup(n):
    A = []
    for i in range(n):
        A.extend([i, i])
    return A
```

2) Good

```
def dup(n):
    for i in range(n):
        yield i
        yield i
```

3) Better

```
def dup(n):
    for i in range(n):
        yield from [i, i]
```

7. Asyncio

Uses new coroutine features and saved state of generators to do asynchronous IO.

8. Faulthandler

Shows traceback even if Python has crashed in a bad way.

9. Unicode variable names

```
>>> résumé = "knows Python"
>>> π = math.pi
```

10. Pathlib

In Python 2, path handling is verbose.

```
import os
directory = "/etc"
filepath = os.path.join(directory, "test_file.txt")
if os.path.exists(filepath):
    stuff
```

In Python 3, it is much simpler.

```
from pathlib import Path
directory = Path("/etc")
filepath = directory / "test_file.txt"
if filepath.exists():
stuff
```

In the second part of this article, we will take a closer look at both evident and little-known or specific fields of Python application.

Testing automation. As a programmer, you need to write tests all the time and Python code fits all requirements for test code: it is plain, simple and needs not so much time to be written. In addition, it has a lot of tooling which comes in handy as well.

Web sites. One can use Python to create dynamic web sites, a program that uses CGI (still it is not recommended), a test for such a program, let alone usage of Django, Flask, Tornado and other frameworks.

Mixing of Python and HTML is possible and provides with libraries, templates and template engines (Mako, Genshi, Jinja) that help a developer to maintain code (fig. 1) [5].

```
>>> template = "<html><body><h1>Hello %s!</h1></body></html>"
>>> print template % "Reader"
<html><body><h1>Hello Reader!</h1></body></html>

>>> from string import Template
>>> template = Template("<html><body><h1>Hello ${name}</h1></body></html>")
>>> print template.substitute(dict(name='Dinsdale'))
<html><body><h1>Hello Dinsdale!</h1></body></html>
```

Fig. 1. Mixed Python and HTML code template

As a Data analyst, you would use Python for getting data from databases, from the web, text files etc., for analysis itself and especially visualization for Python is accompanies by a vast number of great visualization libraries and tools. Here are some examples (fig. 2, 3).

The most common tools for scientific and numeric computing are SciPy (collection of packages for mathematics, science, and engineering), Pandas (data analyzing and modeling library), iPython (interactive shell supporting visualizations and parallel computing) [7].

Another chance to apply this language is teaching or learning programming at any level up to advanced. There is a lot books and online courses, both free and paid services.

Now about future scope of Python. For a Linux admin, Python is crucial, nowadays almost 1/5 of Linux job require Python code to be hired [6]. Python programming skills come in handy for Database developers and boost their productivity.

In view of the above and given fact of Python is used in such an amount of fields, the fact that this language in addition has its own GUIs sounds unbelievably.

However, here they are:

Widgets

Kivy, for writing multitouch applications.

Qt via pyqt or pyside [7]

Specialists say that Python has a bright future because it has been voted as most favorite language beating C, C++ and Java; it is robust, comprehensive and versatile.

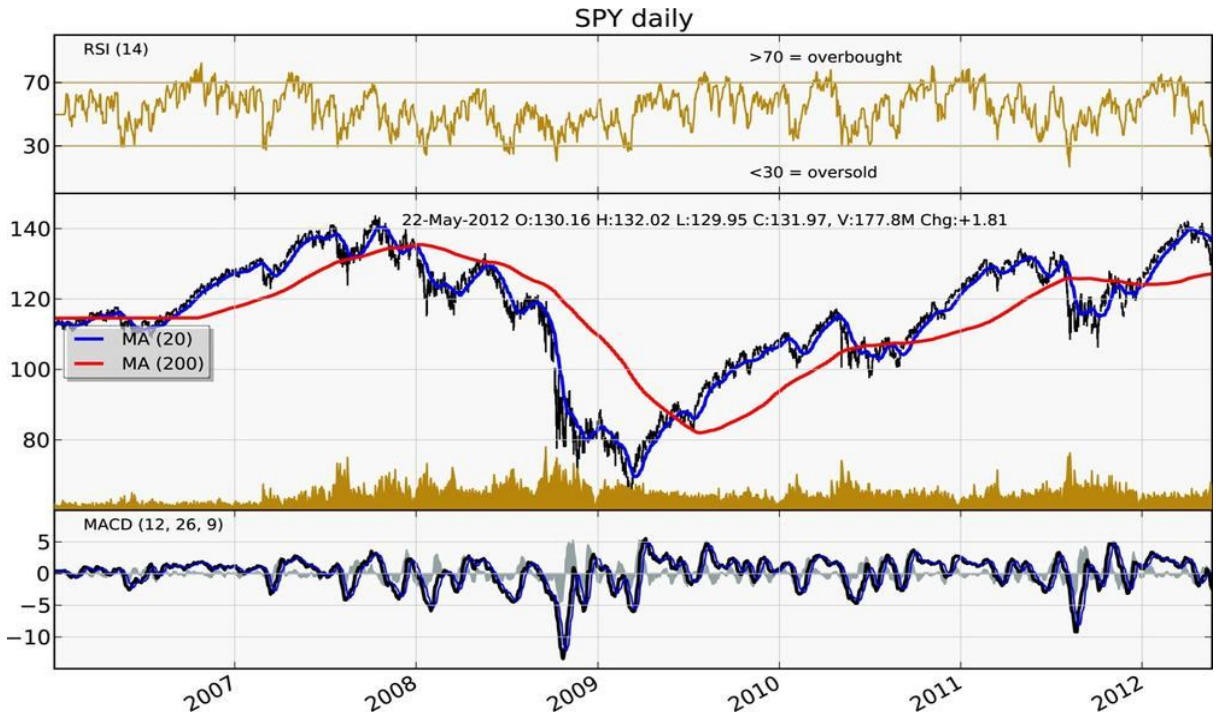

Fig. 2. Matplolib financial plot

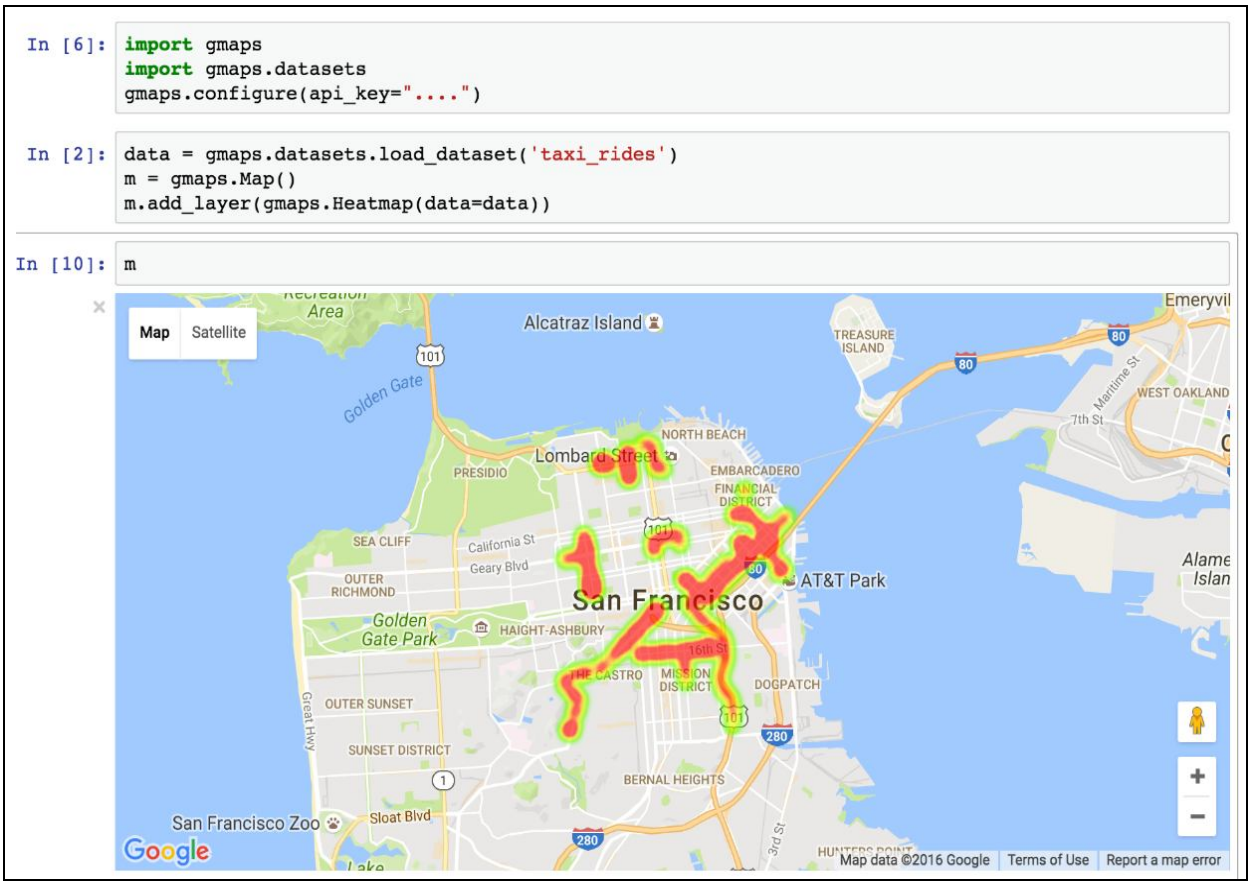

Fig. 3. Heatmap in Python

Conclusions

The conclusion is that Python has some noteworthy features that make it widely used. The arguments we have presented prove that the program written in Python will function exactly the same regardless of which operating system it is running in. Differences arise only in rare cases, and they are easy to anticipate due to the availability of detailed documentation.

REFERENCES

1. WHAT ARE THE 10 MOST FAMOUS SOFTWARE PROGRAMS WRITTEN IN PYTHON? Available at: http://www.hartmannsoft ware.com/Blog/Articles_from_Software_Fans/Most-Famous-Software-Programs-Written-in-Python (accessed 13 May 2018).
2. Рейтинг языков программирования 2018: Go и TypeScript вошли в высшую лигу, Kotlin стоит воспринимать серьезно Available at: https://dou.ua/lenta/articles/language-rating-jan-2018/ (accessed 13 May 2018).
3. 10 awesome features of Python that you can't use because you refuse to upgrade to Python 3 Available at: https://www.asmeurer.com/python3-presentation/slides.html#1 (accessed 13 May 2018).
4. Testing with Python: Just Makes Sense Available at: https://blog.smartbear.com/ testing/testing-with-python-and-why-it-makes-sense/ (accessed 13 May 2018).
5. HOWTO Use Python in the web Available at: https://docs.python.org/2/howto/webservers.html (accessed 13 May 2018).
6. Future Scope of Python Programming Language Available at: https://legacy.gitbook .com/book/sarahjohan/what-is-the-scope-of-python-programming-language/details (accessed 13 May 2018).
7. Applications for Python Available at: https://www.python.org/about/apps/ (accessed 13 May 2018).

СПИСОК ЛІТЕРАТУРИ

1. ЧТО ТАКОЕ 10 САМЫХ ИЗВЕСТНЫХ ПРОГРАММНЫХ ПРОГРАММ, НАПИСАННЫХ В ПИТОНЕ? [Электронный ресурс] - Режим доступа: http://www.hartmannsoftware.com/Blog/Articles_from_Software_Fans/Most-Famous-Software-Programs-Written-in-Python.
2. Рейтинг языков программирования 2018: Go и TypeScript вошли в высшую лигу, Котлин следует воспринимать всерьез. [Электронный ресурс] - Режим доступа: https://dou.ua/lenta/articles/language-rating-jan-2018/
3. 10 удивительных особенностей Python. Python 3 [Электронный ресурс] - Режим доступа: https://www.asmeurer.com/python3-presentation/slides.html#1.
4. Тестирование с помощью Python: просто имеет смысл. [Электронный ресурс] - Режим доступа: https://blog.smartbear.com/ testing / testing-with-python-and-why-it-make-sense /
5. Как использование Python в Интернете [Электронный ресурс] - Режим доступа: https://docs.python.org/2/howto/webservers.html
6. Будущий масштаб языка программирования Python [Электронный ресурс] - Режим доступа: https: //legacy.gitbook .com / book / sarahjohan / what-is-the-scope-of-python-programming-language / details (доступно 13 мая 2018 года).
7. Приложения для Python [Электронный ресурс] - Режим доступа: https://www.python.org /about/apps/

Рецензент: д-р техн. наук, проф. І. В. Шостак,
Національний аерокосмічний університет імені М. Є. Жуковського «ХАІ», Київ
Received (Надійшла) 7.06.2018
Accepted for publication (Прийнята до друку) 22.08.2018

Особливости програмування та нові можливості мови програмування Python

С. П. Альошин, О. О. Бородіна, А. С. Кікоть, І. Б. Жабран

Мета. Метою статті є необхідність продемонструвати рентабельність нових версій мови програмування Python, а також продемонструвати обсяги цієї мови разом з новим програмним забезпеченням. Зміст статті представляє оглядовий характер, що дозволяє зрозуміти особливості та нові можливості мови Python. **Результати.** Було встановлено, що для створення програм для різних цілей застосовують потужної інструмент програмування Python. **Наукова новизна.** Програми, написані на мові програмування Python працюють так само та незалежать від того, на якій операційній системі працює, а також в Python додані особливості, які роблять його широко застосовуваним. **Практична значимість.** Рентабельність нових версій Python та обсяги цієї мови разом з новим забезпеченням дає детальний аналіз областей, де застосовується Python. **Висновки.** Висновок полягає в тому, що у Python є деякі функції, які роблять його широко використовуваними. Наведені аргументи доводять, що програма, написана на Python, буде працювати однаково, незалежно від того, в якій операційній системі вона працює. Різниця виникає лише в рідкісних випадках, і їх легко передбачити через доступність докладної документації.

Ключові слова: Python, професійна мова, особливості, можливості Python.

Особенности программирования и новые возможности языка программирования python

С. П. Алёшин, Е. А. Бородина, А. С. Кикоть, И. Б. Жабран

Цель. Целью статьи является необходимость продемонстрировать рентабельность новых версий языка программирования Python, а также продемонстрировать объемы этого языка вместе с новым программным обеспечением. Содержание статьи представляет собой обзорный характер, которая позволяет понять особенности и новые возможности языка Python. **Результаты.** Было установлено, что для создания программ для разных целей применяют мощной инструмент Python. **Научная новизна.** Программы, написанные на Python работают точно так же, независимости от того, на какой операционной системе работает, а также в Python добавлены особенности, которые делают его широко применяемым. **Практическая значимость.** Рентабельность новых версий Python и объемы этого языка вместе с новым обеспечением дает детальный анализ областей, где применяется Python. **Выводы.** Вывод заключается в том, что у Python есть некоторые функции, которые делают его широко используемым. Представленные аргументы доказывают, что программа, написанная на Python, будет работать одинаково, независимо от того, в какой операционной системе она работает. Различия возникают только в редких случаях, и их легко предвидеть из-за доступности подробной документации.

Ключевые слова: Python, профессиональный язык, особенности, возможности Python.