УДК 621.396.49

L. V. Tulchak
A. O. Marchuk

History of Python

Вінницький національний технічний університет

Анотація

Метою статті є дослідження розвитку мови програмування Python, її особливості та можливості. Також у статті приділяється увага першим засновникам даної мови програмування, перевагам та недолікам Python, її функціональності, типам й структурам даних, бібліотекам, впливу інших мов на Python.

Ключові слова

Історія виникнення Python, Гвідо ван Россум, філософія Python, перша публікація, переваги та недоліки Python, розвиток Python, можливості та реалізація.

Abstract

The development of the Python programming language, its features and capabilities are considered in the article. The first founders of the programming language, advantages and disadvantages of Python, its functionality, types and structures of data, librarie, and the influence of other languages to Python are also focused here.

Keyword

History of Python, Guido van Rossum, the philosophy of Python, the first publication, the advantages and disadvantages of Python, the development of Python, opportunities and implementation.

The history of the Python programming language dates back to the late 1980s [1, 27]. Python was conceived in the late 1980s and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to the ABC programming language capable of exception handling and interfacing with the Amoeba operating system. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community. Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library. Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Python is a multi-paradigm programming language: object-oriented programming and structured programming are fully supported, and there are a number of language features which support functional programming and aspect-oriented programming. Many other paradigms are supported using extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution. The design of Python offers some support for functional programming in the Lisp tradition. The language has map, reduce and filter functions; comprehensions for lists, dictionaries, and sets; and generator expressions [2, 91]. The standard library has two modules itertools and functools. Also, Python has significant advantages over the various programming languages, for example:

- clean syntax (for allocation of blocks to use derogations);
- tolerance programs (that is characteristic of most interpreted languages);
- normal distribution has a lot of useful modules (including the module for developing GUI);
- the use Python in interactive mode (very useful for experimentation and solve simple problems);
- normal distribution is simple but at the same time quite powerful development environment, which is called IDLE and what is written in Python;

- suitable for solving mathematical problems (a means of working with complex numbers, can operate with integers of arbitrary size, a dialog can be used as a powerful calculator).

However, Python still has some disadvantages. Python like many other interpreted languages that do not apply, for example, JIT-compilers have a common drawback - the relatively low rate of program implementation. In addition, the lack of static typing and some other reasons do not allow you to implement a Python function overloading mechanism at compile time.

Python has a large standard library, commonly cited as one of Python's greatest strengths [3, 178], providing tools suited to many tasks. This is deliberate and has been described as a "batteries included" Python philosophy. For Internet-facing applications, a large number of standard formats and protocols (such as MIME and HTTP) are supported. Modules for creating graphical user interfaces, connecting to relational databases, pseudorandom number generators, arithmetic with arbitrary precision decimals, manipulating regular expressions and doing unit testing are also included. Some parts of the standard library are covered by specifications, but the majority of the modules are not. They are specified by their code, internal documentation, and test suite (if supplied). However, because most of the standard library is cross-platform Python code, there are only a few modules that must be altered or completely rewritten by alternative implementations. The primary mechanism for proposing major new features, for collecting community input on an issue and for documenting the design decisions that have gone into Python.

Enhancement of the language goes along with development of the CPython reference implementation. The mailing list python-dev is the primary forum for discussion about the language's development; specific issues are discussed in the Roundup bug tracker maintained at python.org. Development takes place on a self-hosted source code repository running Mercurial. A number of alpha, beta, and release-candidates are also released as previews and for testing before the final release is made. Although there is a rough schedule for each release, this is often pushed back if the code is not ready. The development team monitor the state of the code by running the large unit test suite during development, and using the BuildBot continuous integration system. The community of Python developers has also contributed over 72,000 software modules (as of January 2016) to the Python Package Index (called PyPI), the official repository of third-party libraries for Python. The major academic conference on Python is named PyCon. There are special mentoring programs like the Pyladies.

Python's name is derived from the television series Monty Python's Flying Circus, and it is common to use Monty Python references in example code [4, 161]. For example, the metasyntactic variables often used in Python literature are spam and eggs, instead of the traditional foo and bar. As well as this, the official Python documentation often contains various obscure Monty Python references. Since 2003, Python has consistently ranked in the top ten most popular programming languages as measured by the TIOBE Programming Community Index. As of January 2016, it is in the fifth position. It was ranked as Programming Language of the Year for the year 2007 and 2010. It is the third most popular language whose grammatical syntax is not predominantly based on C.

Python - stable and common language. It is used in many projects and in various ways: as a basic programming language or to create extensions and application integration. In Python implemented a large number of projects, it is also widely used for prototyping of future programs. Python is widely used in many large companies.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Making of Python [Artima Developer] – 2007.

2. Hettinger Raymond PEP 289 – Generator Expressions. Python Enhancement Proposals. / Raymond Hettinger // Python Software Foundation. – 2012.

3. Piotrowski Przemyslaw Build a Rapid Web Development Environment for Python Server Pages and Oracle / Przemyslaw Piotrowski // Oracle Technology Network. – Oracle. –2012.

4. Whetting Your Appetite. The Python Tutorial. // Python Software Foundation. –2012.

Науковий керівник: Тульчак Людмила Володимірівна – старший викладач кафедри іноземних мов, Вінницький національний технічний університет

Марчук Анастасія Олександрівна – студента групи І-15б, факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет, м. Вінниця, zanzarah98@mail.ru

Supervisor: Liudmyla Tulchak – senior teacher of English, the Foreign Languages Department, Vinnytsia National Technical University, Vinnytsia
Anastasiia Marchuk – student, group I-15b, Faculty of Computer Systems and Automatics, Vinnytsia National Technical University, Vinnytsia*,*