Fundamentals of Python programming

Presentation · April 2018

DOI: 10.13140/RG.2.2.13071.20642

CITATION
1

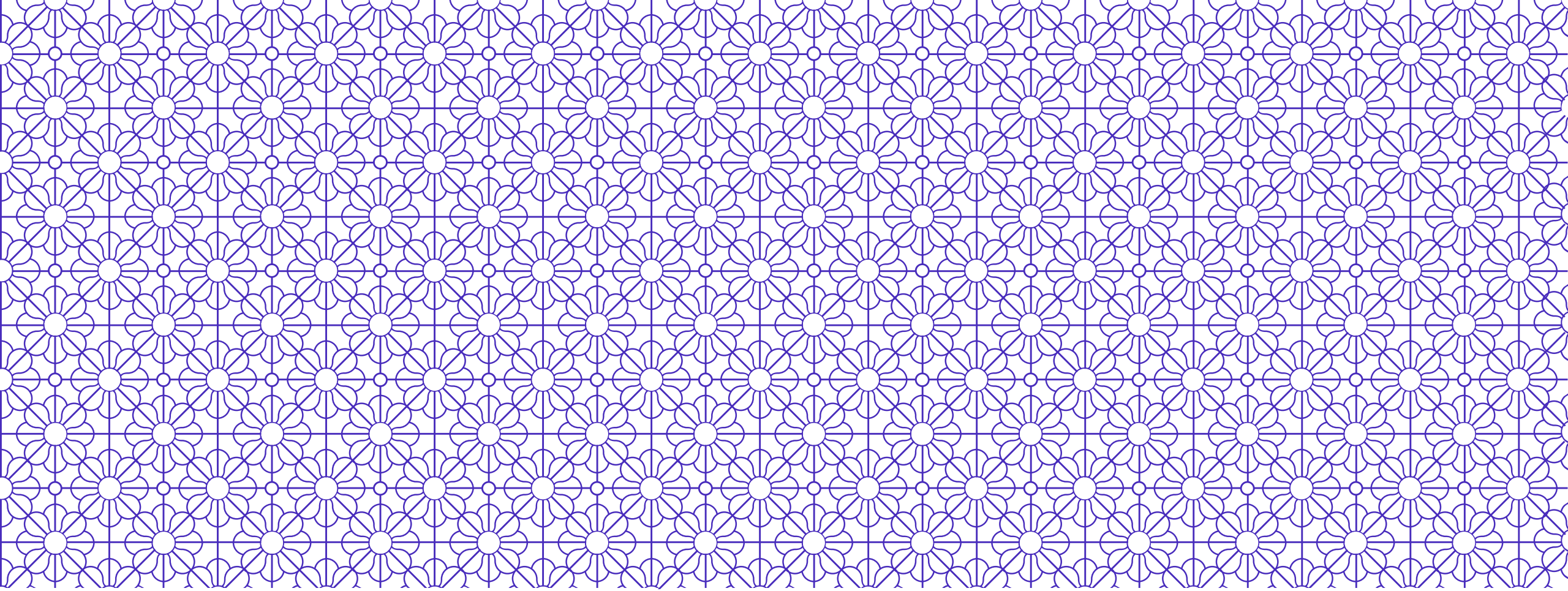
READS
11,595

1 author:

Saman Siadati

29 PUBLICATIONS **133** CITATIONS

SEE PROFILE



FUNDAMENTALS OF PYTHON PROGRAMMING

Saman Siadati


python

INTRODUCTION

- Python is an interpreted, high-level, general-purpose programming language.
- Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.
- Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

INTRODUCTION

- Python is dynamically typed and garbage-collected.
- It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming.
- Python is often described as a "batteries included" language due to its comprehensive standard library.
- Python was conceived in the late 1980s as a successor to the ABC language.
- Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

INTRODUCTION

- Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.
- No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported.
- Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation.
- A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

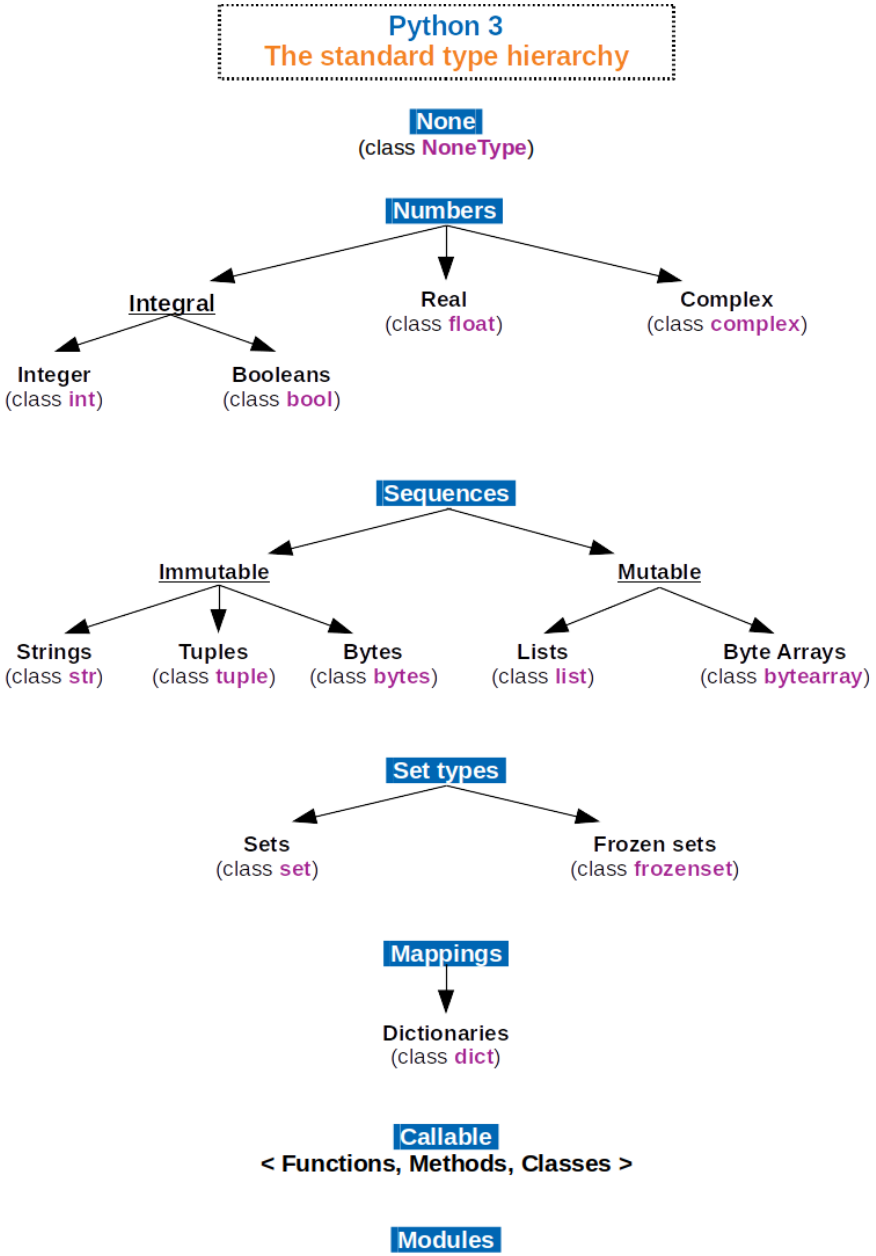
INTRODUCTION

- Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.
- Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6. and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

FEATURES AND PHILOSOPHY

- Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)).
- Many other paradigms are supported via 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. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.
- Python's design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

THE STANDARD TYPE HIERARCHY IN PYTHON 3



LIBRARIES

- Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported.
- It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.
- Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333), but most modules are not.
- They are specified by their code, internal documentation, and test suites. However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

D

- Python Package Index (PyPI), the official repository for third-party Python software, contains over 200,000 packages with a wide range of functionality, including:

- Automation
- Data analytics
- Databases
- Documentation
- Graphical user interfaces
- Image processing
- Machine learning
- Mobile App
- Multimedia
- Networking
- Scientific computing
- System administration
- Test frameworks
- Text processing
- Web frameworks
- Web scraping[105]

REFERENCES

Piotrowski, Przemyslaw (July 2006). "Build a Rapid Web Development Environment for Python Server Pages and Oracle". Oracle Technology Network. Oracle.

Batista, Facundo (17 October 2003). "PEP 327 – Decimal Data Type". Python Enhancement Proposals. Python Software Foundation.

Eby, Phillip J. (7 December 2003). "PEP 333 – Python Web Server Gateway Interface v1.0". Python Enhancement Proposals. Python Software Foundation.

van Rossum, Guido (29 August 2000). "SETL (was: Lukewarm about range literals)". Python-Dev (Mailing list).

van Rossum, Guido (20 January 2009). "A Brief Timeline of Python". The History of Python.

"Guido van Rossum Stepping Down from Role as Python's Benevolent Dictator For Life | Linux Journal". www.linuxjournal.com.

Kuchling, A. M.; Zadka, Moshe (16 October 2000). "What's New in Python 2.0". Python Software Foundation.

Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.

"About Python". Python Software Foundation. Retrieved 24 April 2012., second section "Fans of Python use the phrase "batteries included" to describe the standard library, which covers everything from asynchronous processing to zip files."

"History and License". Retrieved 5 December 2016. "All Python releases are Open Source"

Venners, Bill (13 January 2003). "The Making of Python". Artima Developer. Artima.

van Rossum, Guido (5 April 2006). "PEP 3000 – Python 3000". Python Enhancement Proposals. Python Software Foundation. Archived from the original on 3 March 2016.

REFERENCES

The Cain Gang Ltd. "Python Metaclasses: Who? Why? When?" (PDF). Archived from the original (PDF) on 30 May 2009.

Hettinger, Raymond (30 January 2002). "PEP 289 – Generator Expressions". Python Enhancement Proposals. Python Software Foundation.

Siadati, Saman. (2013). Evolutionary Computation. 10.13140/RG.2.2.36143.15526.

Siadati, Saman. (2017). Just a moment with Machine Learning. 10.13140/RG.2.2. 31765.35042.

Siadati, Saman. (2010). Fundamentals of Mathematics. 10.13140/RG.2.2.31142.75841/1.

Siadati, Saman. (2010). A short review of advanced Statistics. 10.13140/RG.2.2.35310.84804/1.

Siadati, Saman. (2012). Introduction to Data Mining. 10.13140/RG.2.2.21050.21447.

Siadati, Saman. (2014). The first course in Algebra. 10.13140/RG.2.2.16017.04962.

Siadati, Saman. (2012). First lessons of Algorithms, Languages and Logic. 10.13140/RG.2.2.16954.21440