



Python programming - introduction

Pusztai Pál

pusztai@sze.hu

Takács Gábor

gtakacs@math.sze.hu

Outline

- Basic programming concepts
- Programming languages
- Integrated Development Environments
- Platform independency
- The Python programming language
 - Characteristics of Python
 - History
 - Evolution of the language, PEPs
 - Prominent applications
 - Implementations
- Python Development Environments
 - Installation
 - Windows
 - Linux

Basic programming concepts

- **Algorithm:** Finite sequence of steps for solving a specific problem.
 - The concept can be applied for everyday tasks too (e.g. making a Sacher cake, cleaning the bookshelf).
- **Data structure:** A scheme for the storage and efficient use of data elements (example: list).
- **Programming language:** A language defined by rigorous rules, that can be used to communicate instructions to the computer.
- **Programming:** Designing algorithms and data structures, and implementing them in a programming language (coding).

Programming languages

Sep 2023	Sep 2022	Change	Programming Language		Ratings
1	1			Python	14.16%
2	2			C	11.27%
3	4	▲		C++	10.65%
4	3	▼		Java	9.49%
5	5			C#	7.31%
6	7	▲		JavaScript	3.30%
7	6	▼		Visual Basic	2.22%
8	10	▲		PHP	1.55%
9	8	▼		Assembly language	1.53%
10	9	▼		SQL	1.44%
11	15	▲▲		Fortran	1.28%

Source: <https://www.tiobe.com/tiobe-index/>

Integrated Development Environments



■ Input data

- Source codes
- Compiled elements
 - Components, libraries (DLL), ...
- ...

■ IDE

- MS Visual Studio
 - Visual Basic, C#, C++, J#, Python, ...
- RAD Studio
 - Delphi, C/C++
- NetBeans, Eclipse
 - Java, C/C++, PHP, Python, ...
- ...

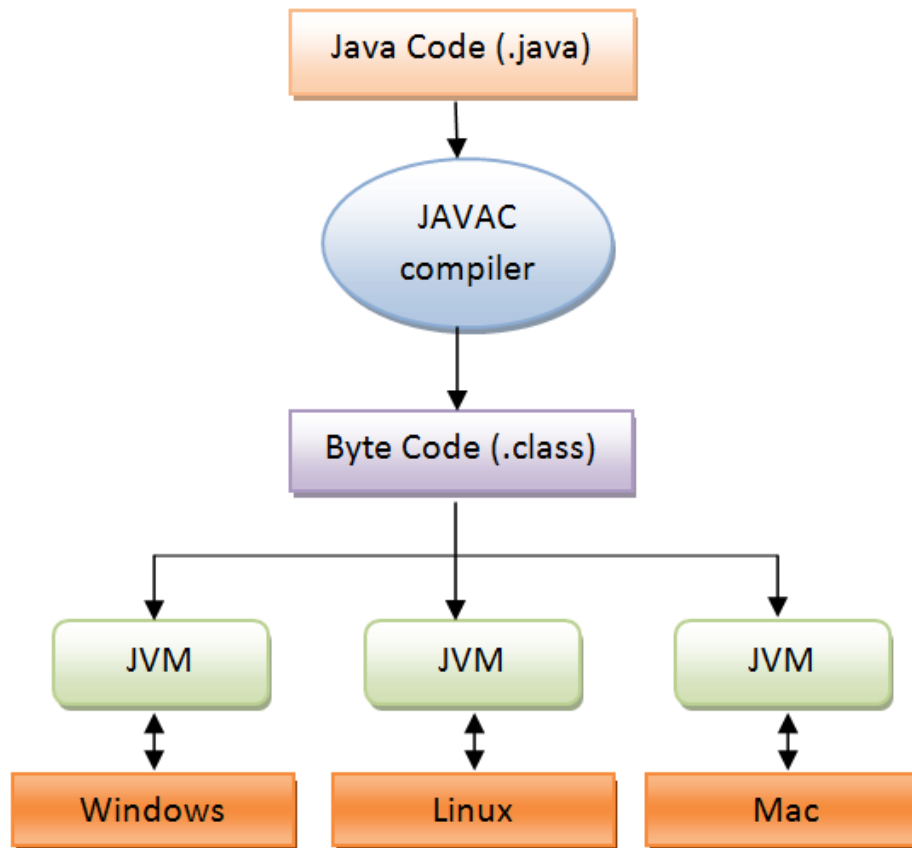
■ Output data

- Compiled elements
 - *.obj, byte code, ...
- Runnable software
 - *.exe, ...
- ...

■ Integrated Development Environment

- Source code editor
- Compiler, interpreter → executing
- Debugger, code completion
- ...

Platform independency - Java



■ Java Virtual Machine

- Executes the Java byte codes on the given operation system.

Characteristics of Python

- Positive
 - concise and elegant syntax
 - easy to learn ("brain-friendly")
 - tens of thousands of external packages (<https://pypi.org/>)
 - strong community, annual PyCon conferences
 - free and open source
 - platform independent
 - dynamically typed, interpreted language
 - multi-paradigm language
- Negative
 - can be slow for certain tasks
 - its multi-threaded capabilities are limited

History



- 1994 Python 1.0 was released.
- 2000 Python 2.0 was released.
- 2001 The Python Software Foundation was launched.
- 2001 The first PyCon conference was held.
- 2008 Python 3.0 was released. It was not compatible with version 2. The transition was slow, but finally happened.
- 2018 Guido van Rossum steps down as BDFL. The main authority behind the language will be a five-person steering committee (see: PEP 8016).

Guido van Rossum



Evolution of the language, PEPs



The screenshot shows a web browser window with the address bar displaying `legacy.python.org/dev/peps/pep-0238/`. The page features the Python logo and the title `» PEP Index > PEP 238 -- Changing the Division Operator`. On the left, a sidebar contains navigation links: ABOUT, NEWS, DOCUMENTATION, DOWNLOAD, a logo, COMMUNITY, FOUNDATION, and CORE DEVELOPMENT, each with a right-pointing arrow. Below these are links for Python Wiki, Python Insider Blog, Python 2 or 3?, and Help Fund Python. The Help Fund Python section includes a PayPal logo and a world map. The main content area displays the following information for PEP 238:

- PEP:** 238
- Title:** Changing the Division Operator
- Version:** e2b5d1a8a663
- Last-Modified:** 2009-01-18 09:50:42 +0000 (Sun, 18 Jan 2009)
- Author:** Moshe Zadka <moshez at zadka.site.co.il>, Guido van Rossum
- Status:** Final
- Type:** Standards Track
- Created:** 11-Mar-2001
- Python-Version:** 2.2
- Post-History:** 16-Mar-2001, 26-Jul-2001, 27-Jul-2001

Abstract

The current division (/) operator has an ambiguous meaning for numerical arguments: it returns the floor of the mathematical result of division if the arguments are ints or longs, but it returns a reasonable approximation of the division result if the arguments are floats or complex. This makes expressions expecting float or complex results error-prone when integers are not expected but possible as inputs.

We propose to fix this by introducing different operators for

Prominent applications

- Youtube

- *„Python has been an important part of Google since the beginning, and remains so as the system grows and evolve.” (Peter Norvig, Google)*

- Dropbox

- *„Python became my favorite programming language because it had a balance of simplicity, flexibility, and elegance.” (Drew Houston, CEO, Dropbox)*

Implementations

- CPython (<http://python.org/>)
- PyPy (<http://pypy.org/>)
- IronPython (<http://ironpython.net/>)
- Jython (<http://www.jython.org/>)
- MicroPython (<https://micropython.org/>)
- ...



Python development environments

■ Heavyweight

- PyCharm (<http://www.jetbrains.com/pycharm/>)
- Visual Studio Code (<https://code.visualstudio.com/>)
- PyScripter (<https://sourceforge.net/projects/pyscripter/>)
- Spyder (<https://code.google.com/p/spyderlib/>)
- ...



■ Lightweight

- Emacs / Vim / ...
- Sublime Text (<http://www.sublimetext.com/>), \$70
- IDLE (shipped with the interpreter)
- Jupyter Notebook
- ...



Installation

- Windows: the easiest way is to use a Python distribution
 - Anaconda (<https://www.continuum.io/downloads>)
 - Miniconda (<http://conda.pydata.org/miniconda.html>)
 - WinPython (<http://winpython.sourceforge.net/>)
- Linux: There are more viable options
 - Using the package manager of the operating system.
 - Installing the interpreter with the package manager, installing the external packages with pip.
 - Using a Python distribution.

Let's do programming, but how?



„keyboard monkey”

VS.



„real programmer”