

Основы языка программирования Python

Основные сведения о Python

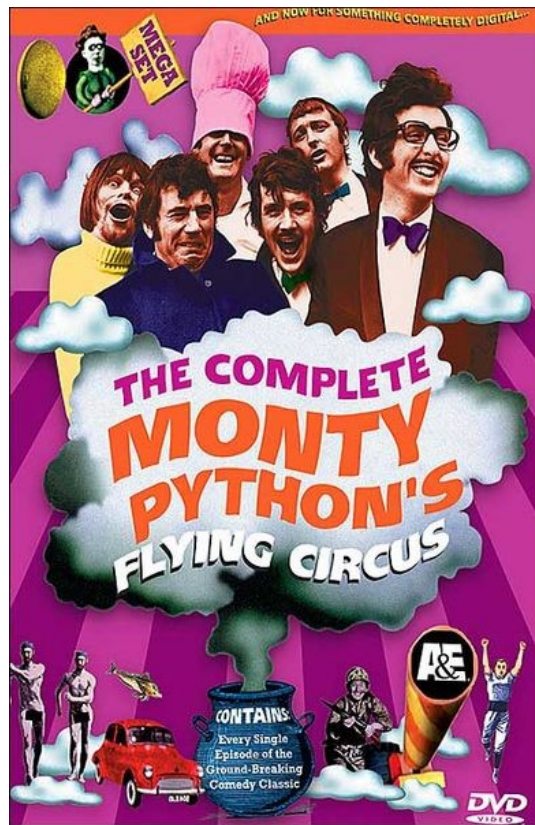
Немного истории



Python 1.0 — 1991 год.

Guido van Rossum (род. в 1956 г.)

Почему «Python»?



Официальный сайт

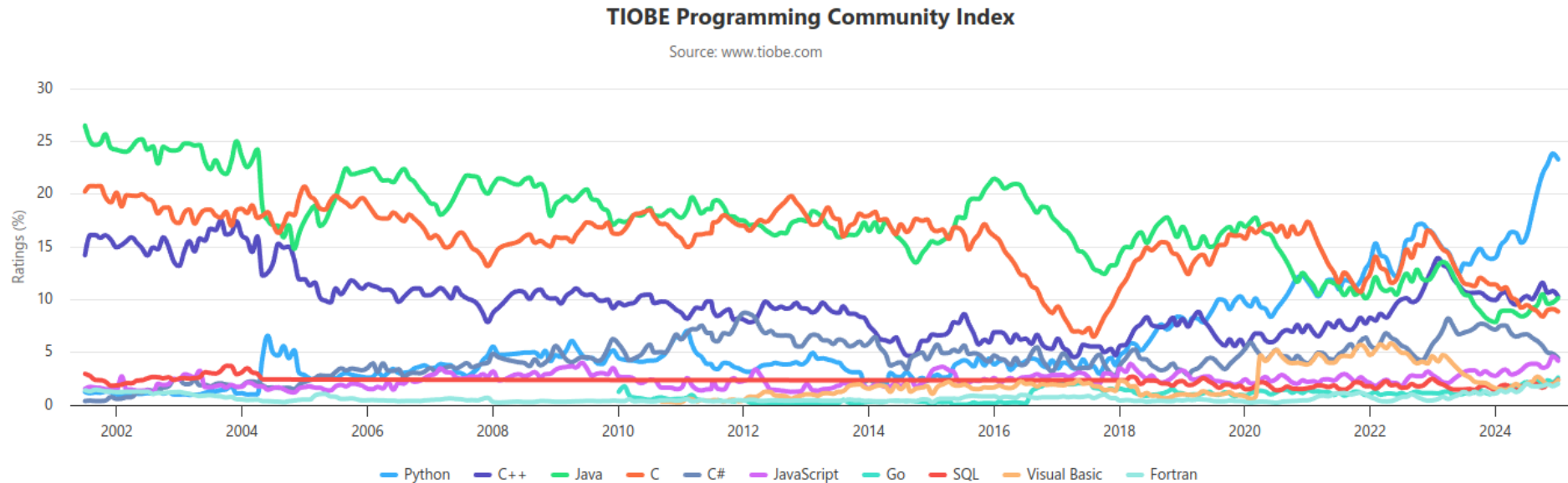
<https://python.org>



Текущая стабильная версия: 3.13

Рейтинг языков программирования. TIOBE Index. Январь 2025 г.












5



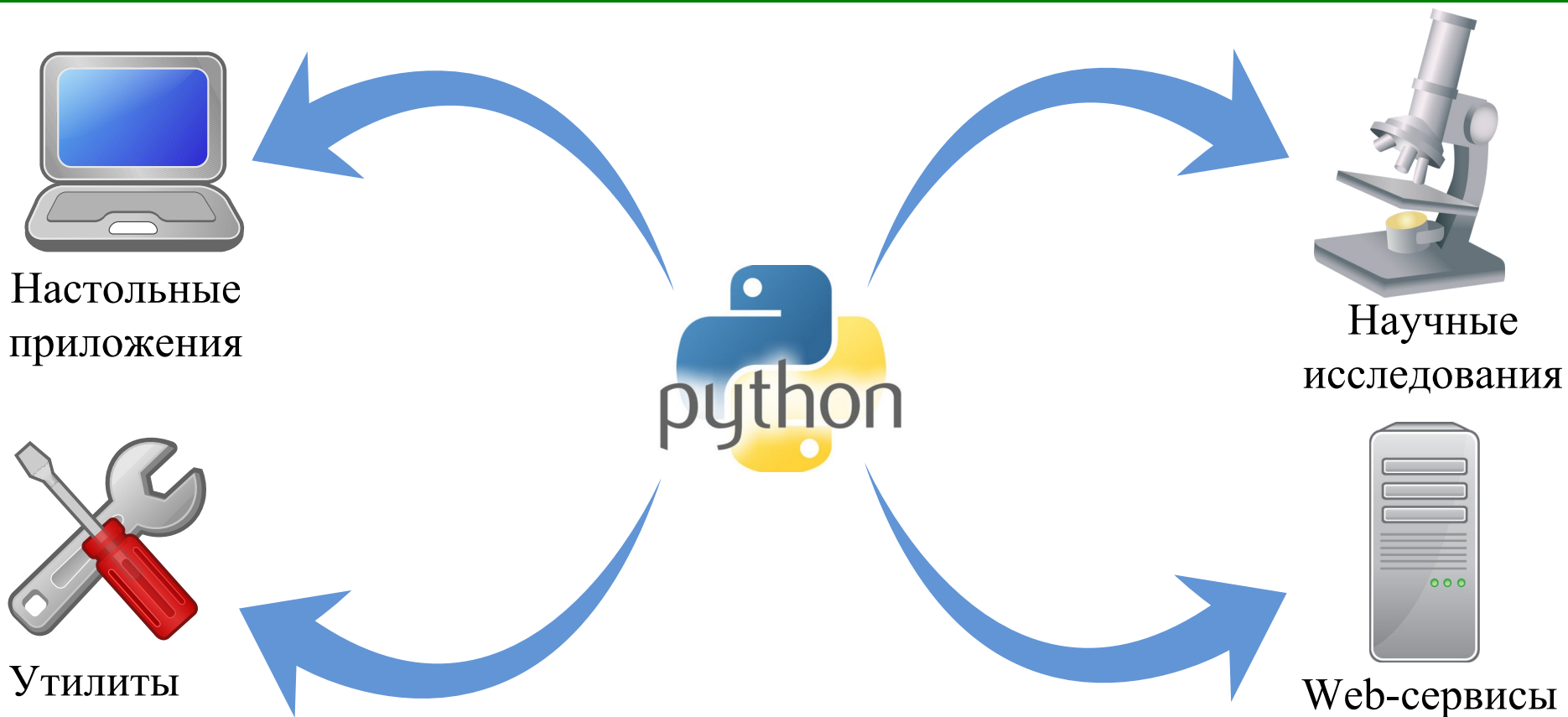
<https://www.tiobe.com/tiobe-index/>

Рейтинг языков программирования. TIOBE Index. Январь 2025 г.

6

Jan 2025	Jan 2024	Change	Programming Language		Ratings	Change
1	1			Python	23.28%	+9.32%
2	3	^		C++	10.29%	+0.33%
3	4	^		Java	10.15%	+2.28%
4	2	v		C	8.86%	-2.59%
5	5			C#	4.45%	-2.71%
6	6			JavaScript	4.20%	+1.43%
7	11	^^		Go	2.61%	+1.24%
8	9	^		SQL	2.41%	+0.95%
9	8	v		Visual Basic	2.37%	+0.77%
10	12	^		Fortran	2.04%	+0.94%
11	13	^		Delphi/Object Pascal	1.79%	+0.70%

Области применения языка Python



Преимущества Python

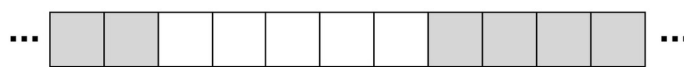
- Низкий порог вхождения.
- Высокая скорость написания кода.
- Лаконичность написанного кода.
- Постоянное развитие языка.
- Большое количество библиотек (<https://pypi.org/>).
- Множество инструментов на основе Python.
- Язык используют в крупных компаниях.

Архитектурные особенности Python

- Поддержка объектно-ориентированного программирования.
- Поддержка элементов функционального программирования.
- Динамическая типизация.
- Строгая типизация.
- Автоматическая сборка мусора.

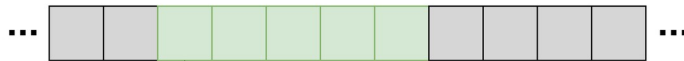
Ошибки работы с памятью

Object
foo()



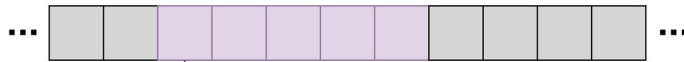
1) `Object *o = new Object();`

*o



2) `o->foo();`

*o

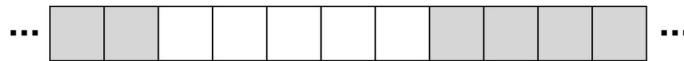


3) `delete o;`

*o

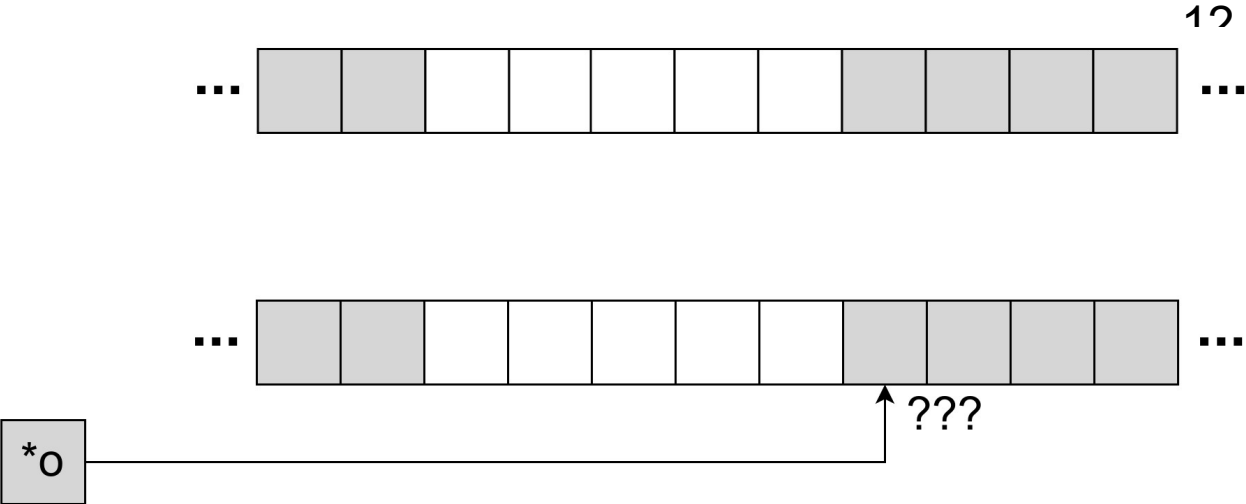


3)

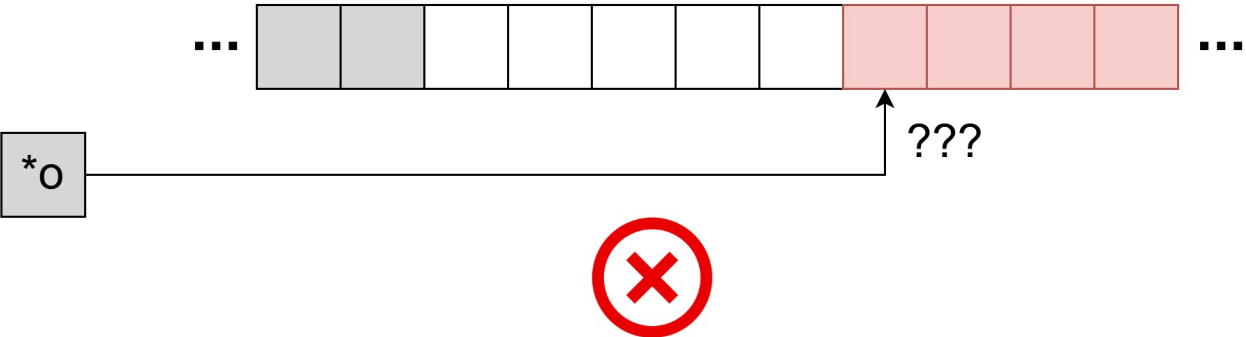


Object
foo()

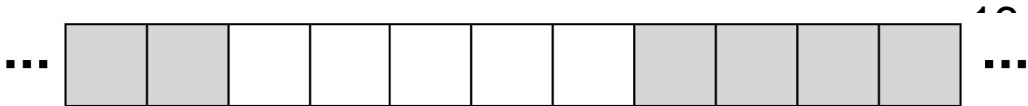
1) Object *o;



2) `o->foo();`



Object
foo()



1) Object *o = new Object();



2)

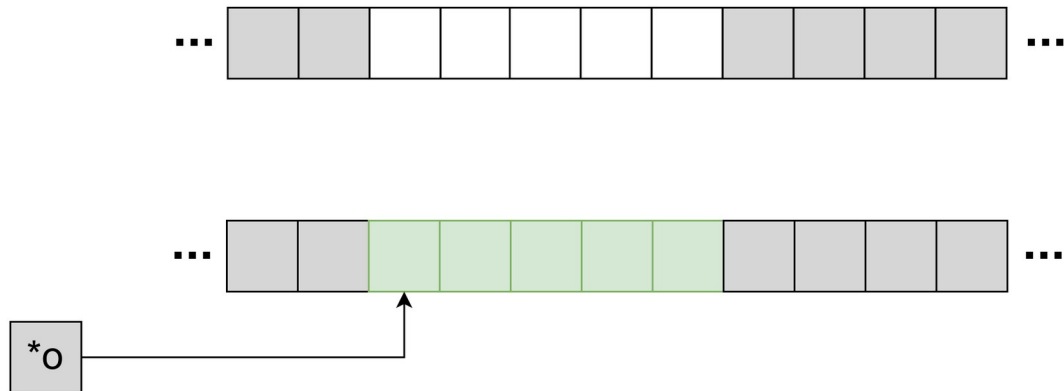


Утечка памяти

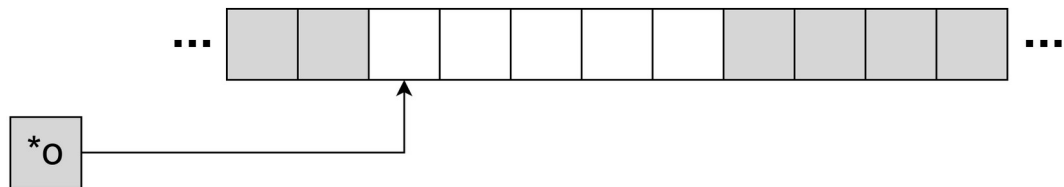
Object
foo()

14

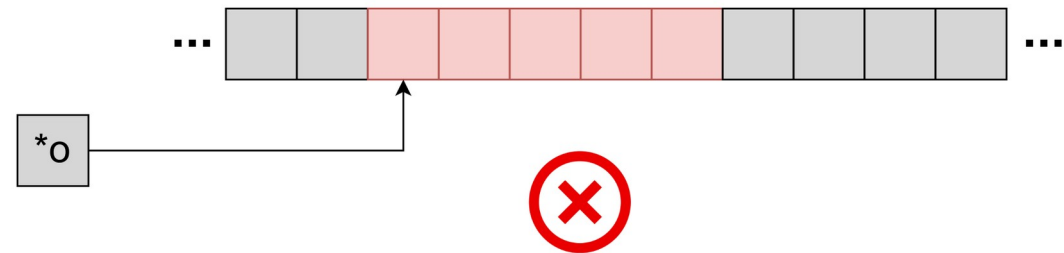
1) Object *o = new Object();



2) delete o;



3) o->foo();



Недостатки Python

- Скорость работы.
- Сложность реализации автоматического рефакторинга.
- Global Interpreter Lock (GIL).

GIL

#1

#2

#3



Время

Python в научных исследованиях



Наиболее известные библиотеки для научных вычислений

18

- NumPy / Scipy
- Matplotlib
- Pandas
- SymPy
- Scikit-Learn
- IPython / Jupyter / JupyterLab

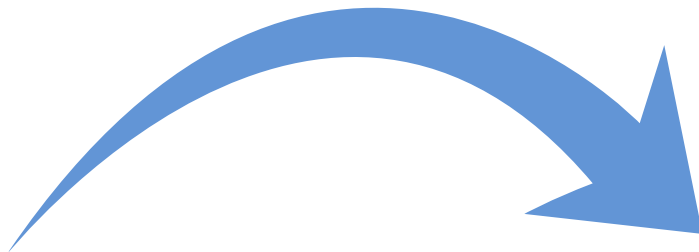
Возможности NumPy и SciPy

- Чтение и запись текстовых данных.
- Матрицы и матричные операции.
- Математические функции.
- Преобразование Фурье.
- Генераторы псевдослучайных чисел.
- Статистические расчеты.
- Алгоритмы оптимизации.
- Алгоритмы аппроксимации и интерполяции.
- И др.

Python для разработки настольных приложений

Создание desktop-приложений

pyInstaller / cx_Freeze



Библиотеки для создания графического интерфейса

