

Основы языка программирования Python

Основные сведения о Python

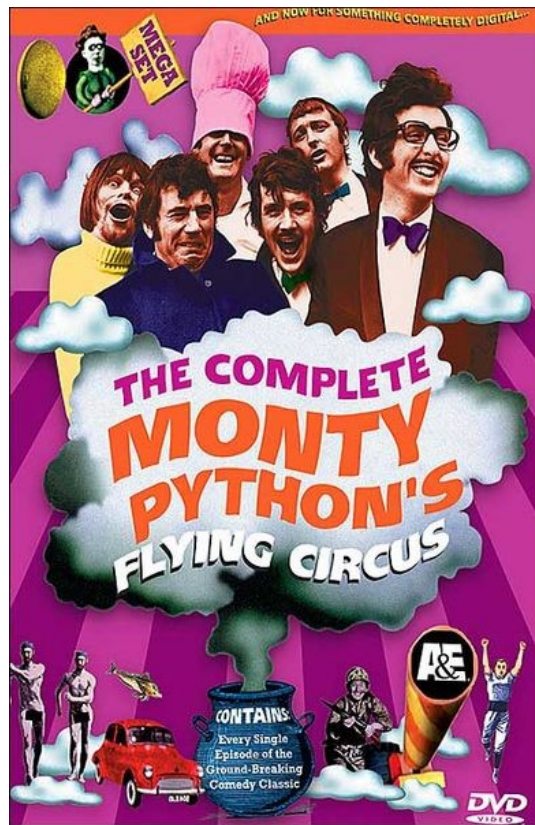
Немного истории



Python 1.0 — 1991 год.

Guido van Rossum (род. в 1956 г.)

Почему «Python»?



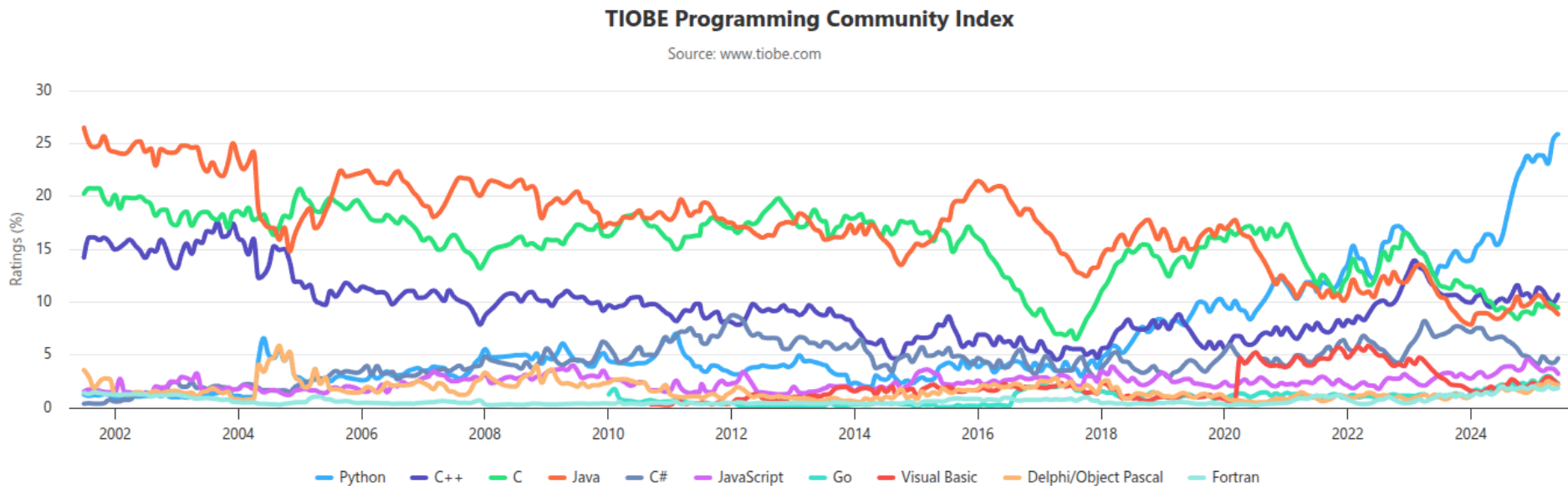
<https://python.org>



Текущая стабильная версия: 3.13

Рейтинг языков программирования. TIOBE Index. июнь 2025 г.












5



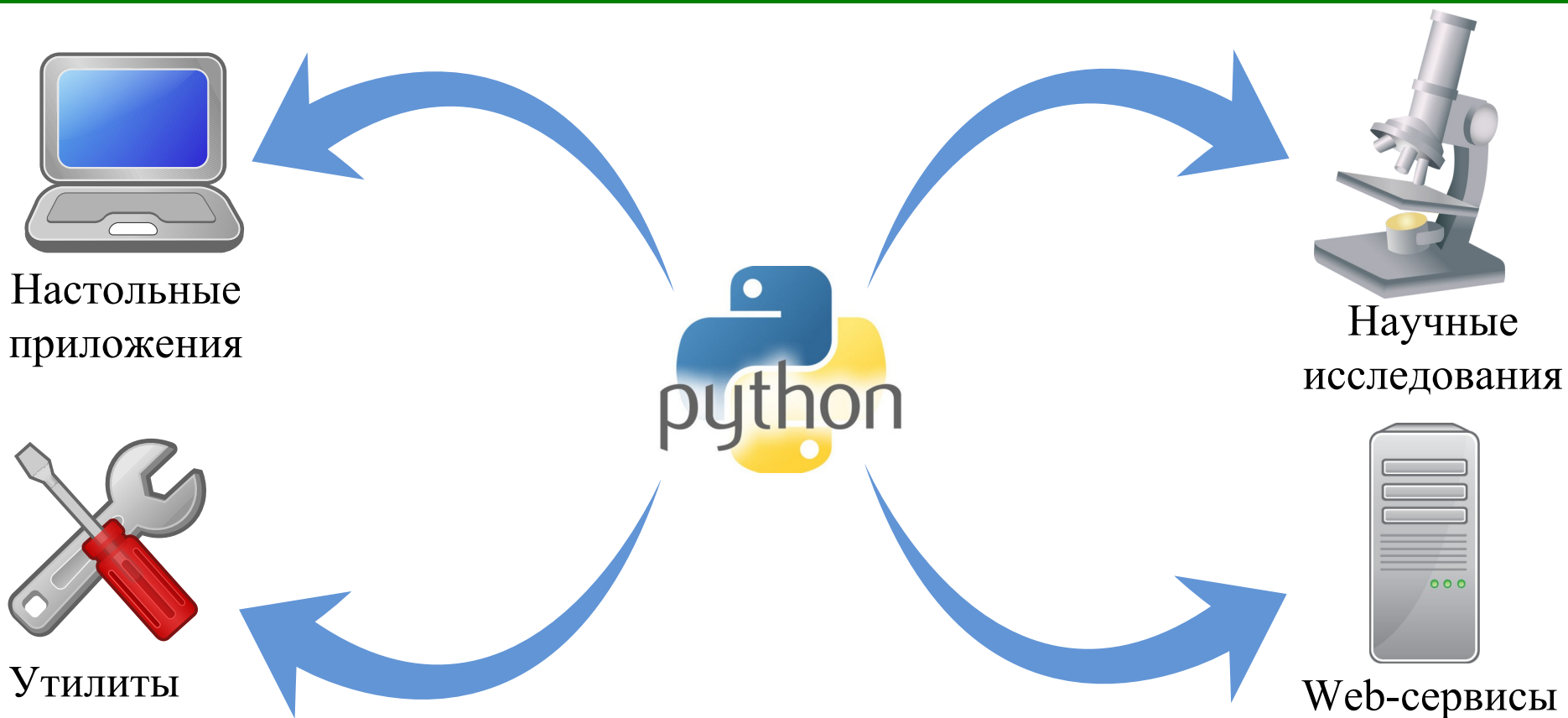
<https://www.tiobe.com/tiobe-index/>

Рейтинг языков программирования. TIOBE Index. Январь 2025 г.

6

Jun 2025	Jun 2024	Change	Programming Language		Ratings	Change
1	1			Python	25.87%	+10.48%
2	2			C++	10.68%	+0.65%
3	3			C	9.47%	+0.24%
4	4			Java	8.84%	+0.44%
5	5			C#	4.69%	-1.96%
6	6			JavaScript	3.21%	-0.11%
7	7			Go	2.28%	+0.35%
8	9	⬆		Visual Basic	2.20%	+0.54%
9	11	⬆		Delphi/Object Pascal	2.15%	+0.62%
10	10			Fortran	1.86%	+0.33%
11	25	⬆		Ada	1.70%	+0.91%

Области применения языка Python



Преимущества Python

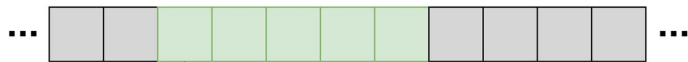
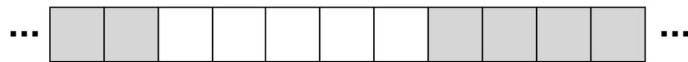
- Низкий порог вхождения.
- Высокая скорость написания кода.
- Лаконичность написанного кода.
- Постоянное развитие языка.
- Большое количество библиотек (<https://pypi.org/>).
- Множество инструментов на основе Python.
- Язык используют в крупных компаниях.

Архитектурные особенности Python

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

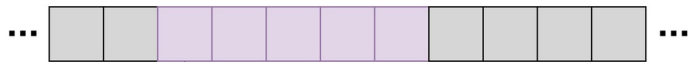
Ошибки работы с памятью

Object
foo()



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

`*o`



2) `o->foo();`

`*o`



3) `delete o;`

`*o`

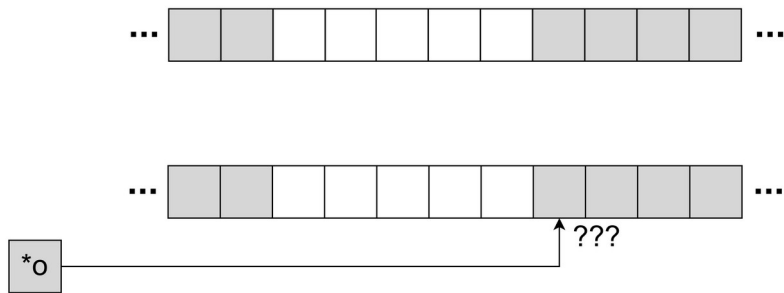


3)

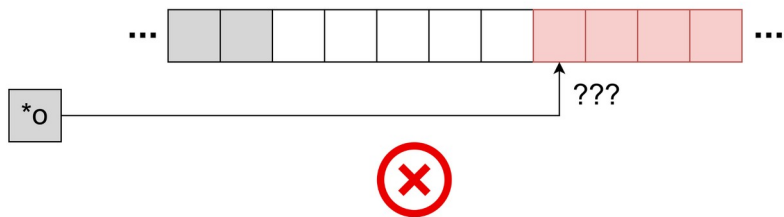


Object
foo()

1) Object *o;



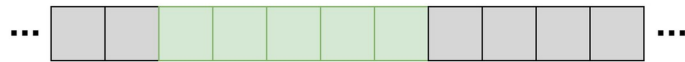
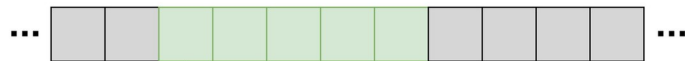
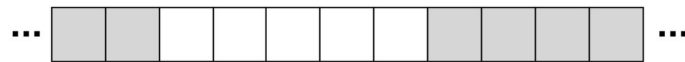
2) `o->foo();`



Object
foo()

1) Object *o = new Object();

*o



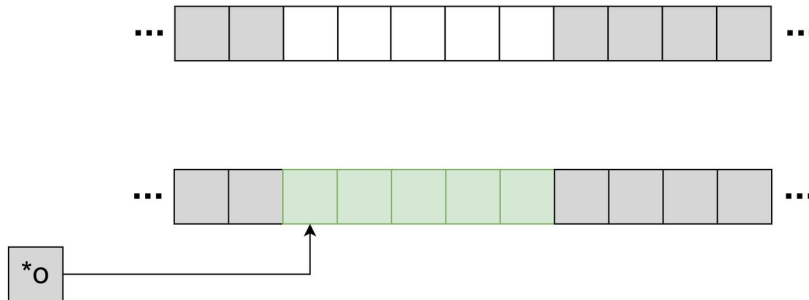
2)



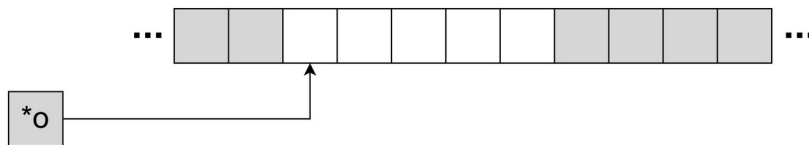
Утечка памяти

Object
foo()

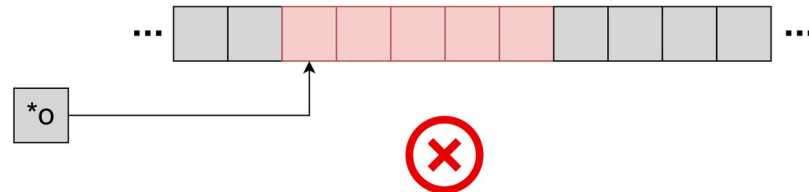
1) Object *o = new Object();



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

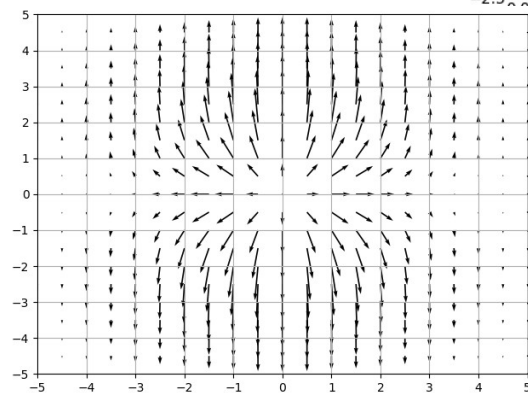
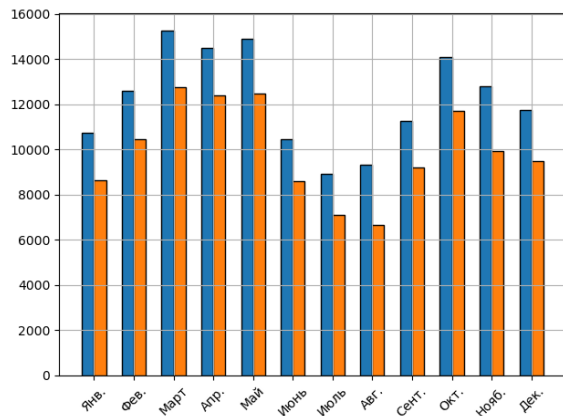
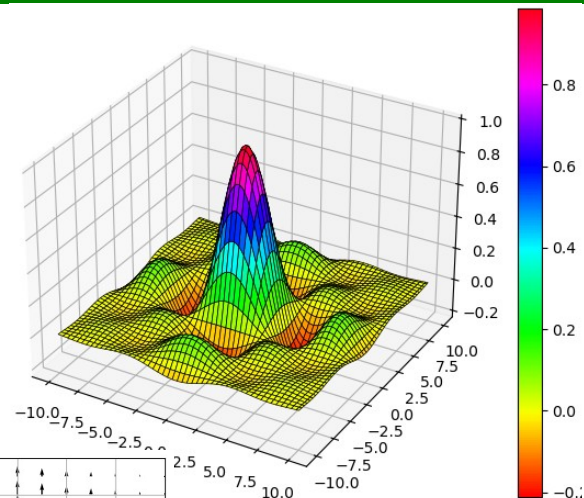
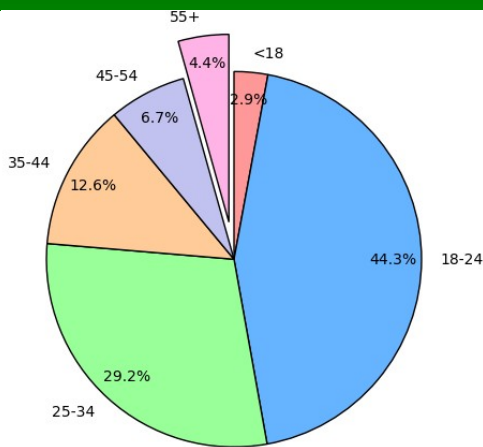
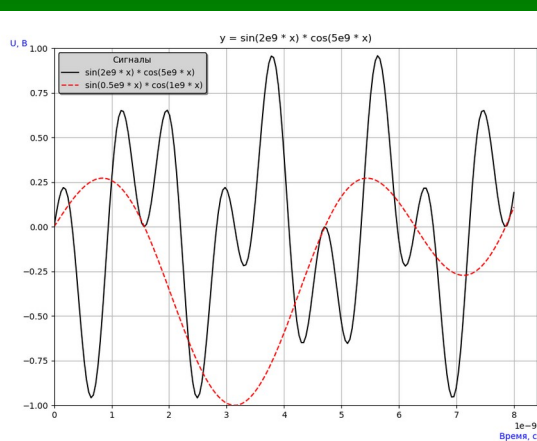
18

- 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.

Matplotlib



JupyterLab

File Edit View Run Kernel Tabs Settings Help

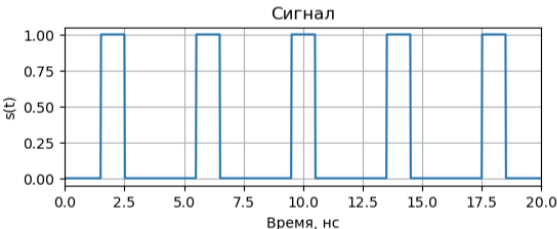
example_31.ipynb

Open in... Python 3 (ipykernel)

Последовательность видеоимпульсов

```
[25]: time = np.arange(signal_len) * dt
      signal = meander(time, duty, period)

[26]: fig = plt.figure(figsize=(6, 2))
      ax_signal = fig.add_subplot()
      ax_signal.plot(time / 1e-9, signal)
      ax_signal.set_title("Сигнал")
      ax_signal.set_xlabel("Время, нс")
      ax_signal.set_ylabel("s(t)")
      ax_signal.set_xlim(0, 20)
      ax_signal.grid()
```



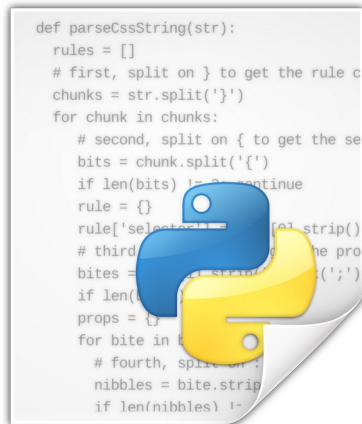
Спектр последовательности видеоимпульсов

$$\dot{S}(k) = \sum_{n=0}^{N-1} \dot{s}(n) e^{-i \frac{2\pi}{N} nk}, k = 0, 1, \dots, N-1$$

Simple Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 example_31.ipynb

Python для разработки настоольных приложений

pyInstaller / cx_Freeze



Библиотеки для создания графического интерфейса

24

