

Язык программирования

Python



История создания и современное положение
языка программирования Python

Python (в русском языке распространено название **ПИТОН**) — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Официальный сайт -

<https://www.python.org/> Официальный сайт -

<https://www.python.org/> Официальный сайт -

<https://www.python.org/> Справочник(Документация) -

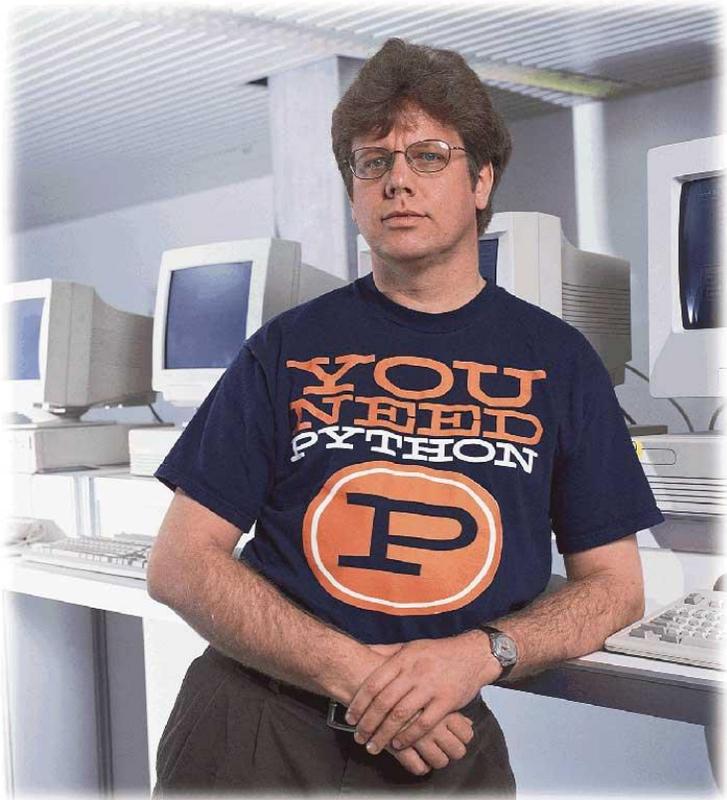
<https://docs.python.org/3/>

Python поддерживает несколько парадигм программирования:

- структурное;
- объектно-ориентированное(ООП);
- функциональное;
- императивное;
- аспектно-ориентированное.

Основные архитектурные черты:

- динамическая сильная неявная типизация;
- автоматическое управление памятью;
- полная интроспекция;
- механизм обработки исключений;
- поддержка многопоточных вычислений;
- и удобные высокоуровневые структуры данных.



Guido van Rossum

Гвидо ван Россум родился 31 января 1956 г. в Нидерландах, где вырос и закончил Амстердамский университет по специальности «математика и информатика» в 1982 году. Работал в исследовательских институтах и лабораториях Европы и США

В 1991 г. – состоялся первый релиз языка Python

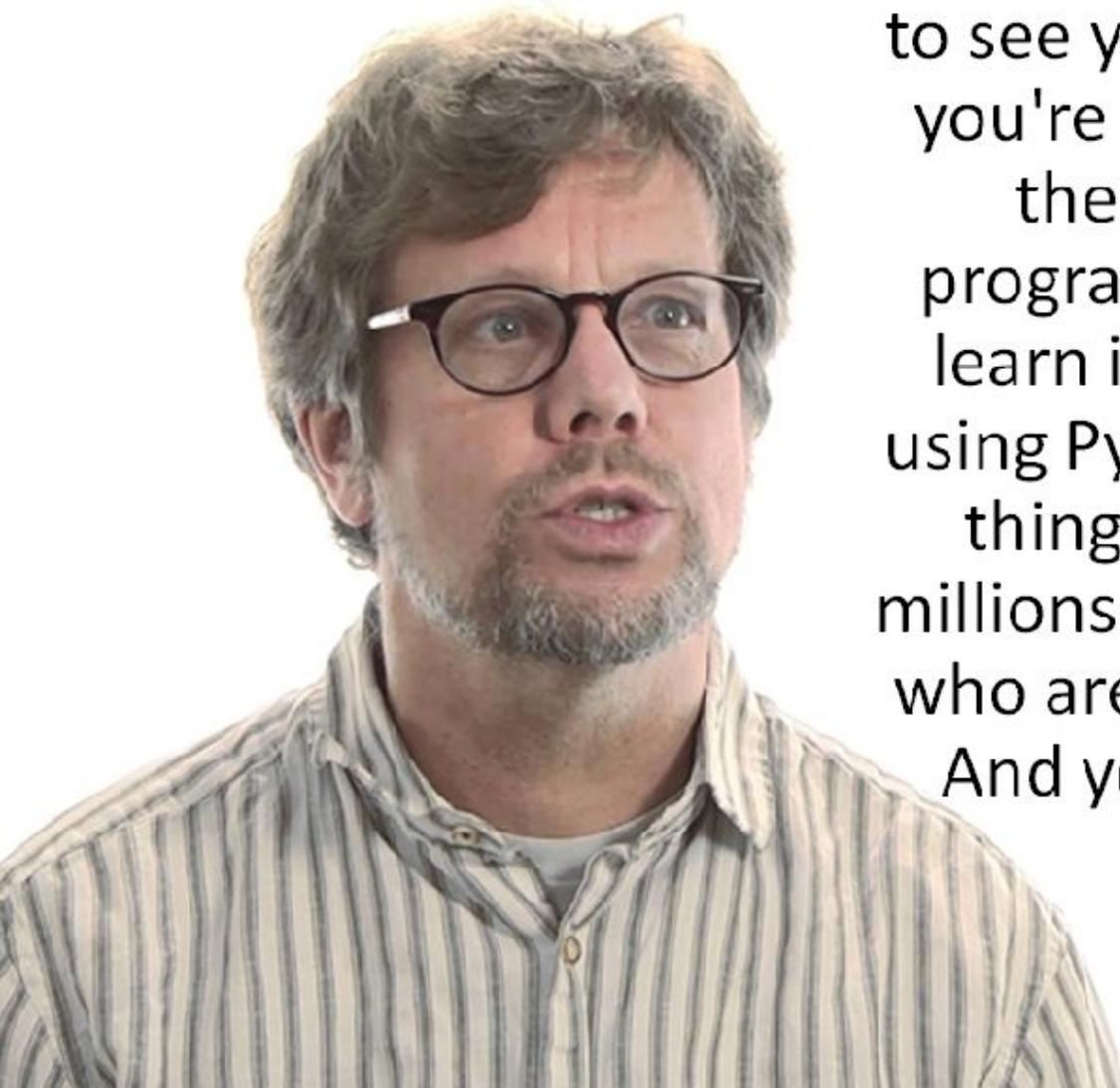
С 2005 – 2012 гг. – работал в Google.

С 2013 – инженер (Python BDFL) в Dropbox.

В настоящее время живёт в Сан-Франциско, в Калифорнии



Welcome to Python Programming



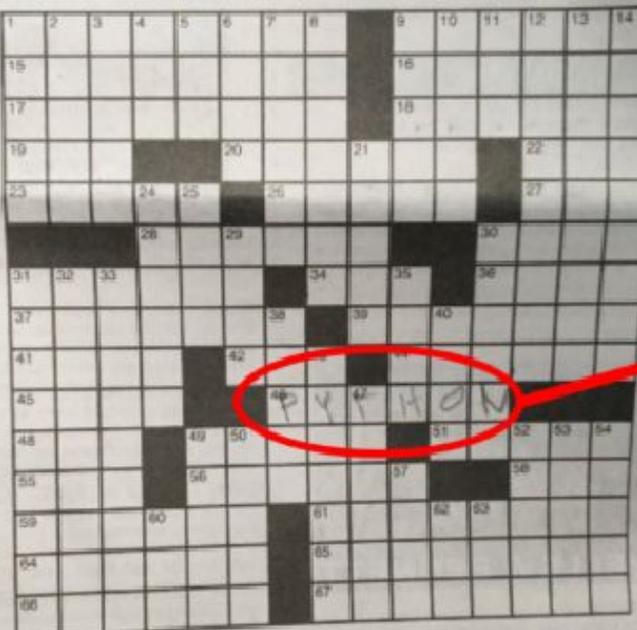
I'm Guido van Rossum. I created the Python language. I have worked on Python for 25 years and I have worked with a large group of others of course in the Python community. I'm very excited to see you all in this class. And I'm very proud that you're using my language to learn. Python is just the first step you're setting on the path to programming. There are many exciting things to learn in programming. Many that you can learn using Python, many that you can learn using other things. And you won't be alone. There will be millions of people who have gone before you or, or who are learning Python at the same time as you. And you, you can help each other, and you can learn together.

Меня зовут Гвидо ван Россум. Я создал язык программирования «Питон». Я работал над Питоном в течение 25 лет, и я работал, конечно же, с большой группой других людей. Я очень рад видеть вас всех в этом классе. И я очень горд, что вы используете мой язык для обучения. «Питон» - это всего лишь первый шаг, который вы делаете на своём пути к программированию. Есть много интересных вещей, которые вы узнаете в программировании. Многие вы узнаете, используя «Питон», многие вы узнаете, используя другие языки. И вы такие не одни. Миллионы людей изучали «Питон» до вас и миллионы изучают его одновременно с вами. И вы можете помогать друг другу и учиться вместе.

CROSSWORD PUZZLE By Kyle Mahowald

ACROSS

- 1 "You're kidding, right?"
- 9 Will matter
- 15 "I give up"
- 16 Onomatopoeic dance
- 17 Criminology concerns
- 18 Weapons used on mounts
- 19 "Out of the Blue" band
- 20 Beyond harmful
- 22 Pabst brand originally brewed in Washington, familiarly
- 23 They're slow to pick things up
- 26 Weaken
- 27 Runner's distance
- 28 Fall deliveries?
- 30 Play or school follower
- 31 Alouette I satellite launcher
- 34 Catch
- 36 "Good..."
- 37 Oakland's county
- 39 Like some beaches
- 41 Gimlet flavor
- 42 2011 World Golf Hall of Fame inductee
- 44 By the fire, say
- 46 Programming language named for a comedy group
- 49 Dryden
- 49 Mexican painter Frida
- 51 Teamsters president James
- 55 Go with carriers
- 56 Like old timers?
- 56 Word in many Mexican place names
- 59 Egg, perhaps
- 61 More than one can handle
- 64 Stuck at a chalet, maybe
- 65 French for "stick"



- 66 Fit one within another
 - 67 "There was a point here somewhere"
- DOWN
- 1 Babies-"R"-Us
 - 2 Even a little
 - 3 Half a playground exchange
 - 4 Stuff that goes kablooney
 - 5 Trouble
 - 6 Give a good toss
 - 7 Equally close
 - 8 1996 treaty subject
 - 9 Panache
 - 10 Fissile rock
 - 11 Light shade... or avoid the shade?
 - 12 Praise
 - 13 Gossip's stock-in-trade
 - 14 "Piece of cake"
 - 21 Goes after
 - 24 Holds accountable
 - 25 Perspective
 - 29 Ordered
 - 30 Pepperidge Farm treat
 - 31 Joining the radio show
 - 32 The Na'vi in "Avatar," for one
 - 33 Rat
 - 35 "Ice cream or cake?" response, perhaps
 - 38 Highest-ranking
 - 40 Bear with a "choful spot"
 - 43 Educational outlines
 - 47 "Poor baby!"
 - 49 Holmes of "Touched With Fire"
 - 50 Words spoken after coming to?
 - 52 High wind?
 - 53 They can be hard to face
 - 54 Grill leftovers
 - 57 1951 title role for Audrey
 - 60 Abbr. on a Miami itinerary
 - 62 Pull
 - 63 Marseille sight

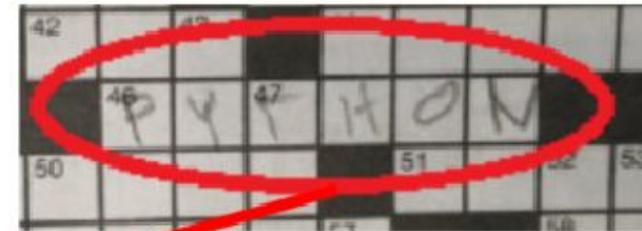
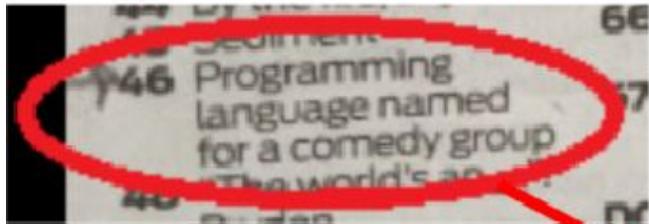


Фото из
Твиттера
Гвидо ван
Россума



Guido van Rossum @guvanrossum · 7 мая

46 across: Programming language named for a comedy group.

1,1 тыс. 1,5 тыс. ***

PYTHON и ПИТОН

Назван язык не в честь змеи, а в честь английской комик-группы «Монти Пайтон», снявшей в телешоу «Летающий цирк Монти Пайтона» в 1969-1974 годах в жанре абсурдистского юмора. (Пример: <https://www.youtube.com/watch?v=g9LEvEFGzlw>)

Влияние, оказанное коллективом на жанр комедии, сравнивают с влиянием, оказанным группой The Beatles на поп и рок-музыку.



Реализации языка PYTHON

- **CPython** – это и есть сам PYTHON, т.к. он написан на языке C.
- **Jython** – это реализация языка Python на языке Java.
- **IronPython** – реализация языка Python, предназначенная для платформы Microsoft .NET или Mono. Написан на C#.
- **ChinesePython** – язык, использующий лексику китайского языка. зарезервированные слова, имена переменных, и большинство операторов также на китайском.
- **PyPy** – в начале своего существования был интерпретатором Python, написанным на Python. Текущие версии PyPy транслируются из RPython в Си и компилируются. Ему пророчат будущее.
- **и многие другие**

Версии Python

PYTHON v. 2.0

16.10.2000

...

2.6
2008

2.7
2010

END

PYTHON v. 1.0
1994

...

1.6
2000

PYTHON v. 3.0

03.12.2008

...

3.4
2014

3.5
2015

3.6
2016

3.7
2018

3.8 ..20??

Применение Python

ИНДУСТРИЯ

- Google
 - Apple
 - IBM
 - Facebook
 - NASA
 - Яндекс и Mail.Ru
 - Dropbox
 - Instagram
- и многие другие.

ОБРАЗОВАНИЕ

- Наибольшее распространение получил в США
- Это первый язык программирования в MIT – флагмане обучения computer science в мире
- В США рекомендован как первый язык программирования для школ и университетов
- Изучается в школах и ВУЗАХ России

С 2015 года задания на Python входят в ЕГЭ(8 заданий):

часть 1 – 5 заданий, часть 2 – 3 задания

11 Ниже на пяти языках программирования записан рекурсивный алгоритм F .

Бейсик	Python
<pre> DIM N, S AS INTEGER N = 1 S = 0 WHILE N <= 150 S = S + 30 N = N * 5 WEND PRINT S </pre>	<pre> def F(n): if n > 2: print(n) F(n - 3) F(n - 4) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел n, s n := 1 s := 0 нц пока n <= 150 s := s + 30 n := n * 5 кц вывод s кон </pre>	<pre> var n, s: integer; begin n := 1; s := 0; while n <= 150 do begin s := s + 30; n := n * 5; end; write(s) end. </pre>
Си	
<pre> #include<stdio.h> int main() { int n, s; n = 1; </pre>	

8 Определите, какое число будет напечатано в результате выполнения программы, записанной ниже на пяти языках программирования.

Бейсик	Python
<pre> DIM N, S AS INTEGER N = 1 S = 0 WHILE N <= 150 S = S + 30 N = N * 5 WEND PRINT S </pre>	<pre> n = 1 s = 0 while n <= 150: s = s + 30 n = n * 5 print(s) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел n, s n := 1 s := 0 нц пока n <= 150 s := s + 30 n := n * 5 кц вывод s кон </pre>	<pre> var n, s: integer; begin n := 1; s := 0; while n <= 150 do begin s := s + 30; n := n * 5; end; write(s) end. </pre>
Си	
<pre> #include<stdio.h> int main() { int n, s; n = 1; </pre>	

19 В программе используется одномерный целочисленный массив A с индексами от 0 до 9. Значения элементов равны 1, 2, 5, 8, 9, 3, 4, 0, 7, 6 соответственно, т.е. $A[0] = 1, A[1] = 2$ и т.д. Определите значение переменной j после выполнения следующего фрагмента программы (записанного ниже на пяти языках программирования).

Бейсик	Python
<pre> j = 5 WHILE A(j) < A(j-1) t = A(j) A(j) = A(j-1) A(j-1) = t j = j - 1 WEND </pre>	<pre> j = 5 while A[j] < A[j-1]: A[j], A[j-1] = A[j-1], A[j] j -= 1 </pre>
Алгоритмический язык	Паскаль
<pre> j := 5 нц пока A[j] < A[j-1] t := A[j] A[j] := A[j-1] A[j-1] := t j := j - 1 кц </pre>	<pre> j := 5; while A[j] < A[j-1] do begin t := A[j]; A[j] := A[j-1]; A[j-1] := t; j := j - 1; end; </pre>
Си	

20 Ниже на пяти языках программирования записан алгоритм. Получив на вход натуральное число x , этот алгоритм печатает число R . Укажите такое число x , при вводе которого алгоритм печатает двузначное число, сумма цифр которого равна 16. Если таких чисел x несколько, укажите наименьшее из них.

Бейсик	Python
<pre>DIM X,D,R AS LONG INPUT X R = 0 WHILE X>0 D = X MOD 10 R = 10*R + D X = X \ 10 WEND PRINT R</pre>	<pre>x = int(input()) R = 0 while x>0: d = x % 10 R = 10*R + d x = x // 10 print(R)</pre>

Алгоритм
 алг
 нач
 цел x ,
 ввод x
 $R := 0$
 нц пока
 $d :=$
 $R :=$
 $x :=$
 кц
 вывод
 кон

Си
 #include
 int main
 {

25 Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых десятичная запись хотя бы одного числа оканчивается на 2. В данной задаче под парой подразумевается два подряд идущих элемента массива. Например, для массива из пяти элементов: 16 3 142 55 22 – ответ: 3.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Бейсик	Python
<pre>CONST N = 40 DIM A (1 TO N) AS INTEGER DIM I, J, K, AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>//допускается также использовать //две целочисленные переменные j и k a = [] n = 40 for i in range(0, n): a.append(int(input())) ... </pre>

Алгоритмический язык	Паскаль
<pre>алг нач цел N = 40 целтаб a[1:N] цел i, j, k нц для i от 1 до N ввод a[i] кц</pre>	<pre>const N = 40; var a: array [1..N] of integer; i, j, k: integer; begin for i := 1 to N do readln(a[i]);</pre>

21 Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на пяти языках программирования).

Бейсик	Python
<pre>DIM A, B, N, t AS INTEGER A = -100: B = 100</pre>	<pre>def f(x): return (x - 16)*(x + 25)</pre>

```
a = -100
b = 100
n = 0

for t in range(a, b + 1):
  if f(t) <= 0:
    n = n + 1
print(n)
```

Паскаль

```
var a, b, N, t: integer;
Function F(x: integer):integer;
begin
  F := (x - 16)*(x + 25)
end;
begin
  a := -100; b := 100;
  N := 0;
  for t := a to b do begin
    if (F(t) <= 0) then
      N := N + 1
    end;
  end;
  write(N)
end.
```

25)

24 Дано целое положительное число N , не превосходящее 1000. Необходимо определить, является ли это число степенью числа 3. То есть требуется определить, существует ли такое целое число K , что $3^K = N$, и вывести это число либо сообщение, что такого числа не существует.

Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre>DIM N, K AS INTEGER INPUT N K = 0</pre>	<pre>n = int(input()) k = 0 while k%3 == 0: k = k + 1 n = n // 3 if n > 0: print(k) else: print("Не существует")</pre>

24)

«не существует»

Паскаль

```
var n, k: integer;
begin
  read(n);
  k := 0;
  while k mod 3 = 0 do begin
    k := k + 1;
    n := n div 3;
  end;
  if n > 0 then
    writeln(k)
  else
    writeln('Не существует')
end.
```

РЕЙТИНГИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

TIOBE Programming Community Index (Индекс TIOBE)

Ежемесячно рассчитывается на основе количества ссылок, найденных по запросам вида: +"**<language> programming**" в поисковых системах и Google, YouTube, Baidu, Yahoo!, Amazon, Wikipedia, Bing и другие. Более высокая позиция языка в рейтинге свидетельствует, что о нём найдено больше страниц по сравнению с другими языками. При этом предполагается, что существует корреляция между количеством найденных страниц и числом специалистов, использующих этот язык.

Каждый год, начиная с 2003, авторами TIOBE выбирается язык года

(Programming Language of the Year):

2017 C
2016 Go
2015 Java
2014 Javascript
2013 Transact-SQL
2012 Objective-C
2011 Objective-C
2010 Python
2009 Go
2008 C
2007 Python
2006 Ruby
2005 Java
2004 PHP
2003 C++

TIOBE Programming Community Index (Индекс TIOBE)

Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5	▲	Python	7.653%	+4.67%
4	3	▼	C++	7.394%	+1.83%
5	8	▲	Visual Basic .NET	5.308%	+3.33%
6	4	▼	C#	3.295%	-1.48%
7	6	▼	PHP	2.775%	+0.57%
8	7	▼	JavaScript	2.131%	+0.11%
9	-	▲▲	SQL	2.062%	+2.06%
10	18	▲▲	Objective-C	1.509%	+0.00%

РЕЙТИНГИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

IEEE Spectrum's Top Programming Languages

Рейтинг синтезирует результаты, полученные по 12 метрикам из 10 источников: поиск Google, Google Trends, Twitter, GitHub, Stack Overflow, Reddit, Hacker News, Career Builder, Dice, IEEE Xplore Digital Library.

Рейтинг обновляется ежегодно.

10 самых популярных языков
программирования 2018 года
по данным IEEE Spectrum

IEEE Spectrum's Top Programming Languages

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	98.4
3. C	  	98.2
4. Java	  	97.5
5. C#	  	89.8
6. PHP		85.4
7. R		83.3
8. JavaScript	 	82.8
9. Go	 	76.7
10. Assembly		74.5

РЕЙТИНГИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

PYPL Popularity of Programming Language Index

Этот рейтинг основан на том, как часто разработчики ищут руководства по тому или иному языку программирования. Данные собираются в Google Trends по запросам "<programming language> tutorial".

Существуют также и др. рейтинги

Worldwide, Sept 2018 compared to a year ago:

Rank	Change	Language	Share	Trend
1	↑	Python	24.58 %	+5.7 %
2	↓	Java	22.14 %	-0.6 %
3	↑	Javascript	8.41 %	+0.0 %
4	↓	PHP	7.77 %	-1.4 %
5		C#	7.74 %	-0.4 %
6		C/C++	6.22 %	-0.8 %
7		R	4.04 %	-0.2 %
8		Objective-C	3.33 %	-0.9 %
9		Swift	2.65 %	-0.9 %
10		Matlab	2.1 %	-0.3 %

Дальнейшая разработка

PEP – Python Enhancement Proposal (Предложения по развитию Python) — это документ со стандартизированным дизайном, предоставляющий общую информацию о языке Python, включая новые предложения, описания и разъяснения возможностей языка.

PEP предлагаются как основной источник для предложения новых возможностей и для разъяснения выбора того или иного дизайна для всех основных элементов языка.

Выдающиеся PEP рецензируются, комментируются и утверждаются Гвидо ван Россумом, имеющим в проекте статус

«великодушного пожизненного диктатора».

Плюсы и минусы Python

+

- открытый код;
- простота синтаксиса;
- наличие мощных библиотек;
- многоплатформенность;
- бесплатность;
- прост в изучении;
- сообщество поддержки.

-

- невысокая скорость выполнения программ;
- эволюция языка.

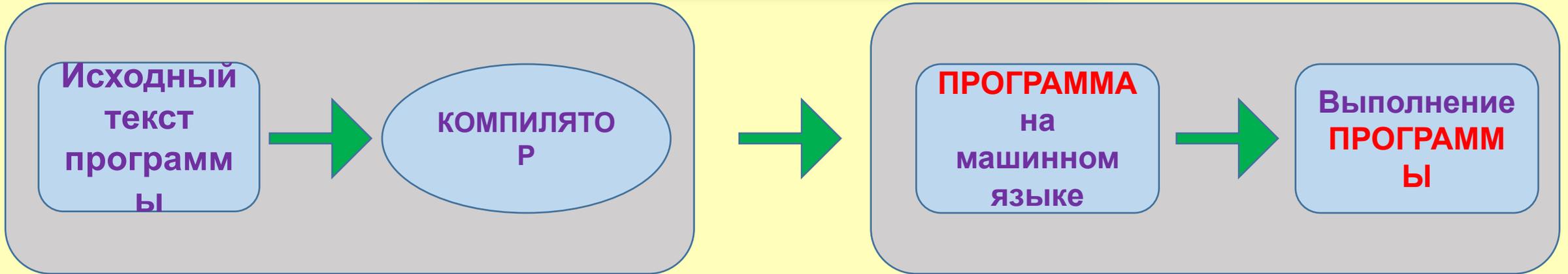
Среды разработки (IDE) для Python

IDE – Интегрированная среда разработки (англ. Integrated Development Environment)

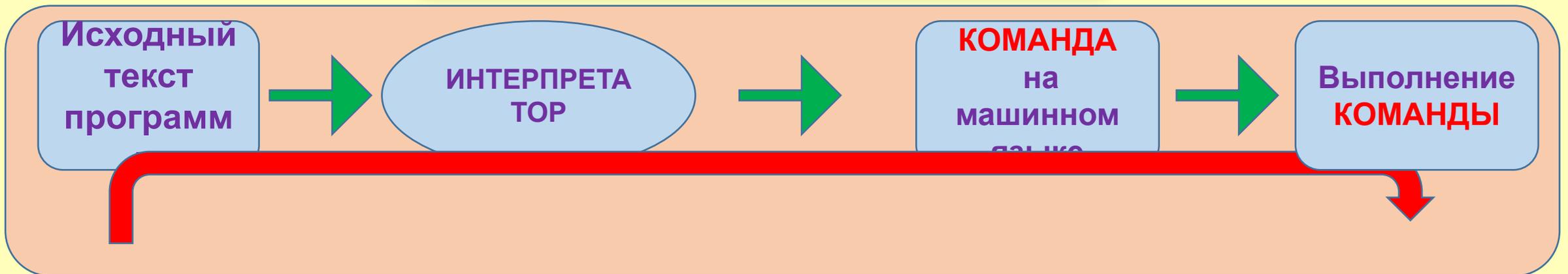
- **IDLE**
- **PyDev**
- **Aptana**
- **Wing**
- **PyCharm**
- **Eclipse**
- **Emacs**
- **Vim**
- **NotePad++**

Полный список редакторов можно найти на странице
<http://wiki.python.org/moin/PythonEditors>.

КОМПИЛЯЦИЯ



ИНТЕРПРЕТАЦИЯ



ПЛЮСЫ И МИНУСЫ КОМПИЛЯЦИИ И ИНТЕРПРЕТАЦИИ

КОМПИЛЯТОРЫ

+

- Быстрота работы программ
- Отсутствие необходимости компилятора на компьютере пользователя

-

- Программа имеет зависимость от ОС, под которую была скомпилирована
- Сложность отладки кода программы

ИНТЕРПРЕТАТОРЫ

+

- Работают на любой платформе
- Более совершенные и наглядные средства диагностики ошибок
- Упрощение отладки исходных кодов программ
- Меньшие размеры машинных кодов

-

- Не может выполняться отдельно без программы - интерпретатора
- Программа выполняется медленнее

Виртуальная машина PYTHON



Питон компилирует программу в байт-код, который передается механизму под названием виртуальная машина Python(PVM).

Цепочка операций выглядит следующим образом:

1. CPython компилирует ваш исходный код на Питоне в байткод.
2. Этот байткод запускается(интерпретируется) на виртуальной машине CPython.

Байт-код – это внутреннее представление программы на языке Python.

Сильно упрощая: машинный код намного быстрее, но байткод



Разработчики языка
Python
придерживаются
определённой
философии
программирования,
называемой
«The Zen of Python»
(«Дзен Питона»)

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

«Дзен Питона»

1. Красивое лучше, чем уродливое.
2. Явное лучше, чем неявное.
3. Простое лучше, чем сложное.
4. Сложное лучше, чем запутанное.
5. Плоское лучше, чем вложенное.
6. Разреженное лучше, чем плотное.
7. Читаемость имеет значение.
8. Особые случаи не настолько особые, чтобы нарушать правила.
9. При этом практичность важнее безупречности.
10. Ошибки никогда не должны замалчиваться.
11. Если не замалчиваются явно.
12. Встретив двусмысленность, отбрось искушение угадать.
13. Должен существовать один — и, желательно, только один — очевидный способ сделать это.
14. Хотя он поначалу может быть и не очевиден, если вы не голландец
15. Сейчас лучше, чем никогда.
16. Хотя никогда зачастую лучше, чем прямо сейчас.
17. Если реализацию сложно объяснить — идея плоха.
18. Если реализацию легко объяснить — идея, возможно, хороша.
19. Пространства имён — отличная штука! Будем делать их побольше!

Python - самый стильный язык программирования в мире. Он не допускает двойного написания кода.

Так, языку Perl присущи зависимость от контекста и множественность синтаксиса, и часто два программиста пишущих на Perl, просто не понимают код друг друга.

В Python же код можно написать только одним способом.

В нем отсутствуют лишние конструкции.

Все программисты должны придерживаться стандарта PEP-8.

Более читаемого кода нет ни в одном другом языке программирования.

На английском: <https://www.python.org/dev/peps/pep-0008/>

На русском: <http://pep8.ru/doc/pep8/>

PEP 8 – руководство по написанию кода на

Python

Per (англ.) — бодрость духа, энергия

Этот документ описывает соглашение(стандарт) о том, как писать код для языка python, включая стандартную библиотеку, входящую в состав python.

PEP 8 создан на основе рекомендаций ван Россума с добавлениями от Барри Уорсоуа.

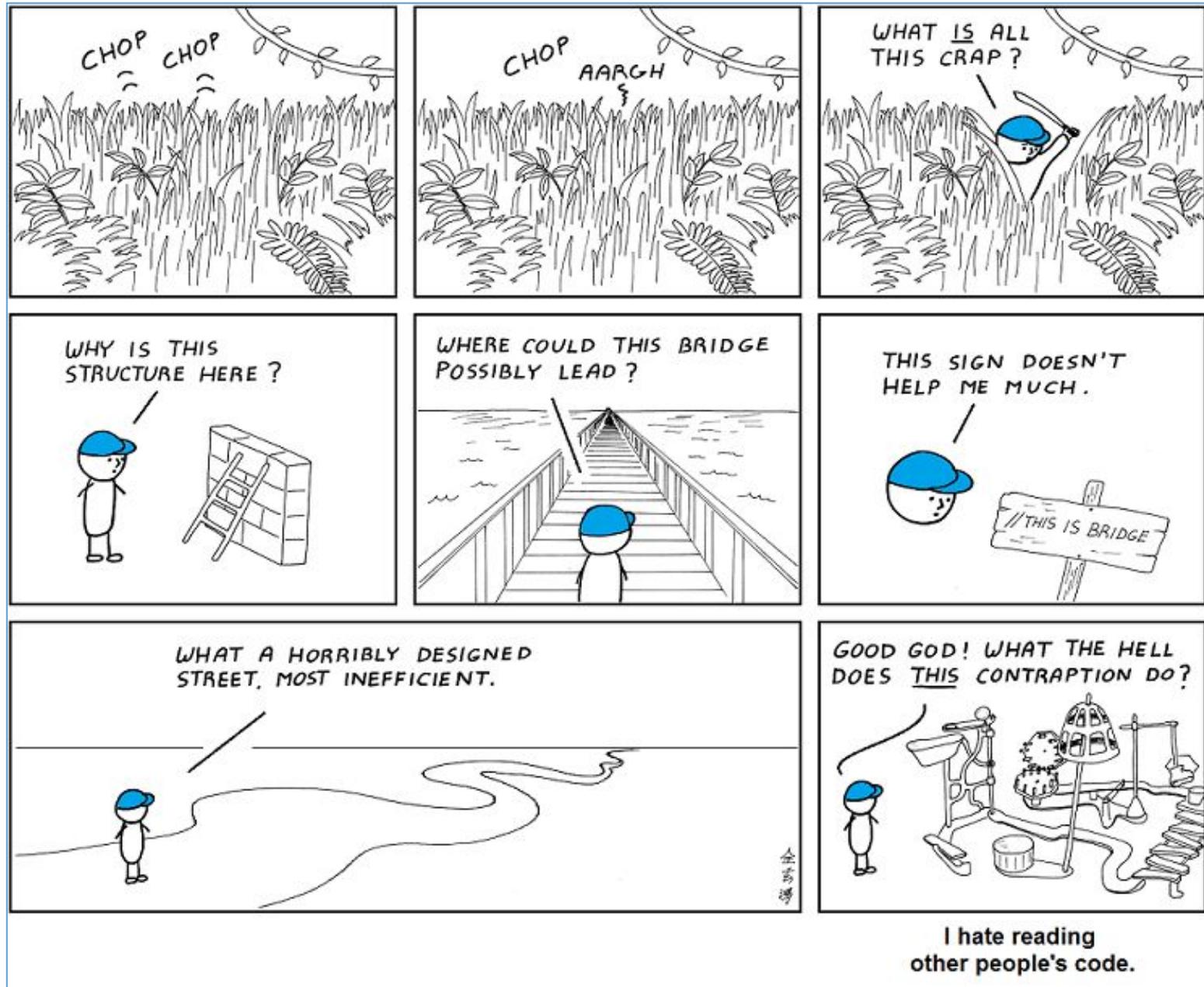
Ключевая идея такова: код читается намного больше раз, чем пишется. Собственно, рекомендации о стиле написания кода направлены на то, чтобы улучшить читаемость кода и сделать его согласованным между большим числом проектов.

В идеале, весь код будет написан в едином стиле, и любой сможет легко его прочесть.

Две причины для того, чтобы нарушить данные правила:

– Когда применение правила сделает код менее читаемым даже для того, кто привык читать код, который следует правилам.

– Чтобы писать в едином стиле с кодом, который уже есть в проекте и который нарушает правила (возможно, в



Начинающим разработчикам на Python рекомендуется распечатать и повесить на стену

- используйте 4 пробела для отступа и не используйте табы, не смешивайте их
- максимальная длина строки 79 символов; для разрыва строки используйте отступы или бэкслеш
- функции верхнего уровня и определения классов отделяйте двумя пустыми строками
- определения методов внутри класса отделяйте одной пустой строкой
- дополнительные пустые строки используйте для логической группировки методов
- кодировка файлов должна быть Latin-1, а в Питоне 3 и выше — utf-8
- каждый импортируемый модуль с новой строки
- порядок импортов: стандартные модули, third-party модули, локальные модули приложения
- пустая строка между каждой группой импортов
- избегайте лишних пробелов внутри скобок; перед запятыми, точкой с запятой и двоеточиями
- избегайте лишних пробелов перед скобками с аргументами функций и скобками с индексами
- избегайте больше чем одного пробела между операторами (= — + итд)
- используйте одиночный пробел для выделения операторов
- избегайте пробелов вокруг '=', который используется для параметров по-умолчанию и keyword аргументов
- обновляйте комментарии вместе с кодом
- пишите комментарии по-английски
- избегайте имён из одиночных I, O, l
- для модулей и пакетов используйте короткие

Среды разработки (IDE) для Python

IDE – Интегрированная среда разработки (англ. Integrated Development Environment)

- **IDLE**
- **PyDev**
- **Aptana**
- **Wing**
- **PyCharm**
- **Eclipse**
- **Emacs**
- **Vim**
- **NotePad++**

Shebang

Основная идея - улучшение переносимости между платформами и компьютерами. Не гарантируется, что на различных системах исполняемый файл будет лежать по пути, который указан в shebang. Использование `env` позволяет снизить этот риск за счет запуска команды на основе данных из переменной среды `PATH`

```
#!/usr/bin/env python3
```

Если собираемся внутри скрипта на питоне писать текст на русском или ещё каком языке кроме английского, то в начало файла нужно вставить строку:

```
# -*- coding: utf-8 -*-
```

Эти две строки кода желательно писать в самом начале, если пишете сценарий в обычных текстовых редакторах



ИТАК!!!

...ТЫ СДЕЛАЛ ВЫБОР?



ВЫБОР ОЦЕРИДЕНИИ

