

# PYTHON

## для начинающих

Паскаль  
Лафуркад

Малика  
Мор

**Лучшие задачи**  
для изучения языка  
программирования

Originally published in France as: 15 énigmes ludiques pour s'initier à la programmation Python By Pascal LAFOURCADE and Malika MORE © Dunod 2022, Malakoff

Во внутреннем оформлении использованы фотографии и иллюстрации:  
Foto-Ruhrgebiet, Ricardo Javier, Renata Sedmakova, Woodpond, Prokradyha,  
Studio Mazeve, ZKH, Babich Alexander / Shutterstock / FOTODOM  
Используется по лицензии от Shutterstock / FOTODOM;  
© Boston Globe, NurPhoto / GettyImages.ru;  
© Universal Images Group / Universal History Archive / DIOMEDIA;  
© Photos 12 Cinema / DR / DIOMEDIA;  
© VMichael Debets/Pacific Press / Alamy / Legion-media;  
© Dieter E. Hoppe / akg-images / Alamy / Legion-media;  
© SVEN SIMON / Alamy / Legion-media

**Лафуркад, Паскаль.**

Л29 Python для начинающих. Лучшие задачи для изучения языка программирования / Паскаль Лафуркад, Малика Мор; [перевод с французского А. Золотова]. — Москва : Эксмо, 2025. — 192 с. — (Мировой компьютерный бестселлер).

Это практическое руководство для тех, кто хочет освоить программирование на языке Python. Состоит из 15 увлекательных задач, которые помогут новичкам освоить ключевые концепции программирования, решая интересные кейсы и головоломки. Издание ориентировано на читателей с любым уровнем подготовки.

Книга «Python для начинающих. Лучшие задачи для изучения языка программирования» станет источником вдохновения для начинающих программистов, заинтересованных не только в овладении техническими навыками, но и в поиске оригинальных решений.

УДК 004.43

ББК 32.973.26-018.2

Все права защищены. Книга или любая ее часть не может быть скопирована, воспроизведена в электронной или механической форме, в виде фотокопии, записи в память ЭВМ, репродукции или каким-либо иным способом, а также использована в любой информационной системе без получения разрешения от издателя. Копирование, воспроизведение и иное использование книги или ее части без согласия издателя является незаконным и влечет уголовную, административную и гражданскую ответственность.

12+

Производственно-практическое издание  
МИРОВОЙ КОМПЬЮТЕРНЫЙ БЕСТСЕЛЛЕР

**Лафуркад Паскаль, Мор Малика**

**PYTHON ДЛЯ НАЧИНАЮЩИХ**

**ЛУЧШИЕ ЗАДАЧИ ДЛЯ ИЗУЧЕНИЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ**

Главный редактор *Р. Фасхутдинов*  
Руководитель направления *В. Обручев*  
Ответственный редактор *Л. Салихова*  
Продюсер *В. Палитко*  
Научный редактор *Н. Белявская*  
Литературный редактор *М. Молчанов*  
Младший редактор *П. Смирнов*  
Художественный редактор *Е. Анисина*  
Компьютерная верстка *Е. Матусовская*  
Корректоры *А. Ярлыкова, Ю. Киреева*

Страна происхождения: Российская Федерация  
Шығарушы ел: Ресей Федерациясы

ООО «Издательство «Эксмо»  
123308, Россия, г. Москва, ул. Зорге, д. 1, стр. 1, эт. 20, каб. 2013. Тел.: 8 (495) 411-68-86.  
Home page: [www.eksmo.ru](http://www.eksmo.ru) E-mail: [info@eksmo.ru](mailto:info@eksmo.ru)  
Өндүрүш: «Издательство «Эксмо» ЖШК  
123308, Ресей, Мәскеу қаласы, Зорге көшесі, 1-үй, 1-құрылыс, 20 қабат, 2013-қаб.  
Тел.: 8 (495) 411-68-86. Home page: [www.eksmo.ru](http://www.eksmo.ru) E-mail: [info@eksmo.ru](mailto:info@eksmo.ru).  
Тауар белгісі: «Эксмо»

Интернет-магазин : [www.book24.ru](http://www.book24.ru)  
Интернет-магазин : [www.book24.kz](http://www.book24.kz)  
Интернет-дуken : [www.book24.kz](http://www.book24.kz)

Импортер в Республику Казахстан ТОО «РДЦ-Алматы».  
Қазақстан Республикасына импорттаушы «РДЦ-Алматы» ЖШС.

Дистрибутор и представитель по приему претензий на продукцию в Республике Казахстан: ТОО «РДЦ-Алматы»  
Дистрибутор және Қазақстан Республикасында өнімге шағымдар қабылдау жөніндегі өкіл: «РДЦ-Алматы» ЖШС.  
Алматы қ., Дембровский қыс., 3-қ., литер Б, офис 1.  
Тел.: 8 (727) 251-59-90/91/92. E-mail: [RDC-Almaty@eksmo.kz](mailto:RDC-Almaty@eksmo.kz)

Сведения о подтверждении соответствия издания согласно законодательству РФ о техническом регулировании можно получить на сайте Издательства «Эксмо»:  
[www.eksmo.ru/certification](http://www.eksmo.ru/certification)

Техникалық реттеу туралы РФ заңнамасына сай басылмыштың сайкестігін растау туралы мәліметтерді мына адрес бойынша алуға болады: <http://eksmo.ru/certification/>

Произведено в Российской Федерации  
Ресей Федерациясында өндiрiлген  
Сертификаттауға жатпайды

Дата изготовления / Подписано в печать 01.10.2024.  
Формат 70x100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 15,56.  
Тираж экз. Заказ

ЛИТРЕС  
Я ТАК ЧИТАЮ

**БОМБОРА**  
ИЗДАТЕЛЬСТВО

БОМБОРА – лидер на рынке полезных и вдохновляющих книг.  
Мы любим книги и создаем их, чтобы вы могли творить, открывать мир, пробовать новое, расти. Быть счастливыми. Быть на волне.

[bomбора.ru](http://bomбора.ru) [bomбораbooks](https://bomбораbooks) [bomбора](https://bomбора)

ЧИТАЙ · ГОРОД



ТЕРИТОРИЯ  
КНИЖНЫЙ МАГАЗИН  
Официальная франшиза  
издательства «Эксмо»



**eksmo.ru**

Официальный  
интернет-магазин  
издательства «Эксмо»



Хочешь стать  
автором «Эксмо»?



9 785041 926465 >

ISBN 978-5-04-192646-5

© Антон Золотов, перевод на русский язык, 2025

© Оформление. ООО «Издательство «Эксмо», 2025



# Содержание

Предисловие .....	9
<b>1. РYТНОН 3 .....</b>	<b>11</b>
Краткое введение .....	12
<b>2. ГОЛОВОЛОМКИ.....</b>	<b>19</b>
1. Аскет ★ .....	21
2. Криптарифм ★ .....	23
3. Кулинарная лопатка ★ .....	25
4. Шадоки считали ★ .....	27
5. Стеганография ★ .....	29
6. Скрытые каналы ★ .....	32
7. Раскраска в два цвета ★★ .....	34
8. Электронное голосование ★★ .....	39
9. Биткоин ★★ .....	43
10. Сиракузы ★★ .....	45
11. Кривые Безье ★★ .....	47
12. Во времена древних греков ★★ .....	51
13. Четверичное дерево ★★ .....	54
14. Семь мостов Кёнигсберга ★★★ .....	56
15. Точный счет ★★★ .....	59
<b>3. ПОДСКАЗКИ... НА ВСЯКИЙ СЛУЧАЙ .....</b>	<b>61</b>
1. Подсказки первого уровня .....	63
2. Подсказки второго уровня .....	65
3. Подсказки третьего уровня .....	69

<b>4. РЕШЕНИЯ</b> .....	73
1. Аскет ★ .....	75
2. Криптарифм ★ .....	83
3. Кулинарная лопатка ★ .....	93
4. Шадоки считали ★ .....	102
5. Стеганография ★ .....	109
6. Скрытые каналы ★ .....	117
7. Раскраска в два цвета ★★ .....	121
8. Электронное голосование ★★ .....	127
9. Биткоин ★★ .....	132
10. Сиракузы ★★ .....	137
11. Кривые Безье ★★ .....	150
12. Во времена древних греков ★★ .....	159
13. Четверичное дерево ★★ .....	165
14. Семь мостов Кёнигсберга ★★★ .....	172
15. Точный счет ★★★ .....	181
Список иллюстраций .....	187
Список аббревиатур .....	188
Алфавитный указатель .....	189

*Всем, кто готов программировать и зарабатывать этим деньги*



# Предисловие

Пятнадцать головоломок, собранных в этой книге, позволят вам приобрести необходимые навыки программирования и вместе с тем неплохо провести время. Их решение требует логики, размышлений, а иногда — хитрости, однако для понимания вполне достаточно знаний на уровне средней школы.

В середине книги есть специальный раздел, где для каждой головоломки предложены подсказки трех последовательных уровней. Если вы вдруг забуксовали и не знаете, что делать дальше, эти подсказки помогут вам найти путь к решению.

Уровень сложности головоломки обозначен звездочками. Есть головоломки легкого уровня ★. Их под силу решить каждому, достаточно лишь некоторой усидчивости. Требуемый код, как правило, довольно простой.

Решение головоломок промежуточного уровня ★★ потребует более сложных размышлений или вычислений. Иногда нужно будет применять приемы не столь очевидные, как на первом уровне. Для написания программ, несомненно, понадобится приложить больше усилий.

И наконец, есть две сложные головоломки ★★★, решение которых потребует большего напряжения, а также продвинутых математических и компьютерных навыков. Для составления необходимых программ придется проявить определенное искусство.

Эта книга предназначена в первую очередь для учащихся средних специальных учебных заведений соответствующего профиля. Цель книги — представить задачи, которые можно решить с помощью программ на языке Python 3, хотя на самом деле программы можно писать и на любом другом языке программирования, который вам хорошо знаком. В начале книги мы посвящаем несколько страниц описанию основных команд Python 3, которые пригодятся для решения головоломок.

Темы головоломок дают возможность познакомиться со многими важными понятиями информатики. Основную часть книги составляют детальные

## Предисловие

решения всех головоломок. Каждое решение содержит не только программу на языке Python 3, с помощью которой решается головоломка, но и подробное объяснение того, как алгоритм был составлен. Каждое решение сопровождается научной или литературной цитатой, связанной с темой головоломки или ее решением.

Игровая составляющая данной книги послужит дополнительным стимулом для многих читателей и подтолкнет их к творческому подходу при составлении программ для решения головоломок.

Эта книга ориентирована на тех, кто любит программировать. В ней представлены задачи, доступные для программистов и помогающие узнать важные элементы информатики. Косвенно она также адресована всем преподавателям информатики, которые хотят обучить своих учеников программированию на Python 3. Настоящий сборник головоломок представляет собой банк упражнений, пригодных для включения в учебную программу.

Головоломки, подсказки и решения содержат многочисленные врезки: биографические, исторические, культурные, технические, математические, связанные с языком Python 3, решения на языке Python 3, — а также полезные советы для педагогов. Каждая врезка обозначена соответствующим значком:



**Благодарности:** Мы благодарим Седрика Лораду за то, что подсказал нам путь к созданию этих головоломок. Мы также выражаем нашу благодарность Флавьену Бине, Генаэль де Жюли, Эммануэлю Диле, Кристин Сольнон и Тибо Рали за их вклад в подготовку этой книги. Кроме того, мы адресуем нашу сердечную признательность Жан-Люку Блану, Матье Даниэлю и Анн Ле Дюк за их замечания и конструктивные предложения, сделанные после тщательного прочтения этой книги.

Малика Мор и Паскаль Лафуркад  
Клермон-Ферран, 13 сентября 2022 год

**1**

# Python 3

# Краткое введение

Все решения головоломок для наглядности приводятся на языке Python 3, бесплатном и свободном для скачивания\*. Разумеется, решения можно писать на любом другом языке программирования по вашему выбору, имеющем функции для арифметических вычислений, присваивания переменных, выполнения условных операторов и циклов (`while`), что необходимо для написания требуемых алгоритмов.

В данной главе описаны команды, наиболее часто используемые при решении представленных головоломок. Иногда в тексте книги или подсказках приводятся дополнительные сведения. Для получения более подробной информации о Python 3, а также для систематизированного изучения программирования на этом языке существует множество пособий\*\* и специализированных сайтов с курсами. Рекомендуем начать с официального руководства, доступного в интернете по адресу <https://docs.python.org/3/>, хоть оно и не самое простое. Если вы уже хорошо знакомы с Python 3, то, скорее всего, вам не придется обращаться к этой главе; в противном случае будет нелишним заглянуть сюда за помощью, если для программирования правильного решения какой-либо из головоломок вам не хватает уже известных инструментов.

Прежде всего отметим, что в Python 3 символ «#» используется для комментирования остальной части строки, а значит, все написанное после этого символа в той же строке не будет учтено при выполнении программы. Другое необходимое знание — важную роль в Python 3 играют *табуляции* (отступы), и их наличие или отсутствие в нужных местах может изменить поведение программы и даже привести к возникновению ошибок. И третье — чтобы разбивать длинный текст, в Python 3 существует оператор «\», известный

---

\* <https://www.python.org>.

\*\* Например: Julien Guillod, *Programmation Python par la pratique Problèmes et exercices corrigés*, Dunod, 2021.

также как принудительный перенос строки. Он разбивает одну длинную непрерывную строку на несколько коротких, более удобных для чтения.

```
1 string = «Этот» + « текст» + « просто» + « чересчур» + « длинный,»\  
2 + « чтобы» + « отображать» + « его» + « на» + « одной» + « строке.»
```

## Присвоение и простые математические операции

Символ «=» используется для того, чтобы присвоить значение выражения переменной. Например, две команды  $x = 3$  и  $x = x + 1$  последовательно присваивают переменной  $x$  значение 3, а затем значение  $3 + 1$ , т. е. 4.

Числовые операции сложения, вычитания, умножения, деления и возведения в степень, а также логические (булевы) операции *отрицания*, *и*, или обозначаются соответственно  $a+b$ ,  $a-b$ ,  $a*b$ ,  $a/b$ ,  $a**b$ ,  $\text{not } a$ ,  $a \text{ and } b$ ,  $a \text{ or } b$ . Для деления с остатком, если  $a = b.q + r$  при  $0 \leq r < b$ , то  $q = a//b$ ,  $a \cdot r = a\%b$  или  $r = \text{divmod}(a,b)$ .

В случае, когда эти операции возможны, для преобразования переменной  $y$  в целое число пишут  $x = \text{int}(y)$ , а для преобразования переменной  $x$  в строку символов пишут  $y = \text{str}(x)$ .

## Условный оператор if

Для выражения условия используются ключевые слова **if**, **else** и **elif**. В приведенном ниже коде переменной  $res$  присваивают значение 3, если значение переменной  $x$  строго больше 3; иначе, если  $x$  равно нулю, то  $res$  принимает значение 0; в противном случае  $res$  принимает значение 1.

```
1 if x > 3:  
2     res = 3  
3 elif x == 0:  
4     res = 0  
5 else:  
6     res = 1
```

Обратите внимание, что запись «==», используемая при проверке условий, отличается от символа присваивания. Кроме того, обязательными являются как символ «:» после сравнения, так и табуляция в строках после условных операторов. Части программы, соответствующие **elif** и **else**, используются

## 1. Python 3

только в том случае, когда это необходимо по алгоритму; в противном случае они могут быть опущены.

Обычными операторами сравнения являются:  $a < b$ ,  $a \leq b$ ,  $a > b$ ,  $a \geq b$ ,  $a == b$  и  $a != b$  (для  $a$ , отличного от  $b$ ).

## Циклы `while` и `for`

Следующая программа выводит на экран целые числа от 0 до 9 с помощью цикла `while`:

```
1 i=0
2 while (i<10):
3     print(i)      #0 1 2 3 4 5 6 7 8 9
4     i=i+1
```

Следующая программа делает то же самое с помощью цикла `for`. В Python 3 функция `range(i)` возвращает набор целых значений от 0 до  $i-1$ :

```
1 for i in range(10):
2     print(i)      #0 1 2 3 4 5 6 7 8 9
```

## Строки символов

- Пустая символьная строка обозначается «`«»`» (пара кавычек без пробела между ними).
- Для объединения двух строк, например, "фото" и "выставка", достаточно использовать символ сложения «`+`». Таким образом, объединение строк «фото» и «выставка» будет равно «фотовыставка».
- Функция `chr` позволяет преобразовать целое число в символ, для которого оно является кодом ASCII\*, если такой символ существует. Например, код `chr(97)` возвращает букву 'a', а `chr(122)` возвращает букву 'z', однако `chr(123456789)` вызовет ошибку, поскольку в таблице кодов ASCII столь больших значений просто нет.
- В общем случае разницы между одинарными «'» и двойными «"» кавычками для обозначения символьной строки (см. примеры выше) нет.

---

\* American Standard Code for Information Interchange.

## Считывание данных из файла

Чтобы не перепечатывать текст вручную, данные, необходимые для решения тех или иных головоломок, содержатся в файлах, которые можно скачать по адресу <https://addons.eksmo.ru/it/Enigmes.zip>.

Поэтому важно знать, как считывать эти данные, чтобы работать с ними в Python 3.

Следующая программа считывает слова из файла `f.txt` построчно, одно за другим:

```
1 file = open('f.txt', 'r')
2 for line in file:
3     for word in line.split():
4         print(word)
5 file.close()
```

Следующая программа позволяет считывать содержимое файла `f.txt` по-символьно:

```
1 file = open('f.txt', 'r')
2 character = file.read(1)
3 while character:
4     print(character)
5     character = file.read(1)
6 file.close()
```

## Списки

Пустой список обозначается пустыми квадратными скобками: «`[]`». Для добавления элемента `x` в конец списка `L` используется оператор `L.append(x)`. Для доступа к первому элементу списка `L` следует написать `L[0]`, поскольку в Python 3 индексы списков начинаются с 0. Чтобы получить последний элемент списка, можно написать `L[-1]`.

```
1 L=[]
2 L.append(4)
3 L.append(5)
```

## 1. Python 3

```
4 L.append(6)
5 print(L)           #[4, 5, 6]
6 print(L [1])      #5
7 print(len(L))     # Выводит 3, количество элементов в списке L
8 print(L[-1])      # Выводит последний элемент списка L
9 print(L[len(L) - 1]) # Выводит последний элемент списка L
```

## Функции и передача параметров

Чтобы определить функцию в Python 3, мы должны использовать ключевое слово **def**, за которым следует имя функции, например, `exchange`, как в приведенном ниже примере, затем задать в скобках параметры функции — в нашем случае `x` и `y`.

```
1 def exchange(x, y):
2     x=x+y
3     y=x-y
4     x=x-y
5     return (x, y)
6
7 a=3
8 b=4
9 print(exchange(3, b)) # Выводит (4, 3)
10 print(a, b)         # Выводит 3 4
```

Переменные, определенные внутри функции, называются *локальными* и не имеют никакой связи с переменными, определенными вне функции, хотя могут иметь такие же имена.

Локальные переменные существуют только внутри функции и получают значения при ее запуске. В примере выше `x` принимает значение 3, а `y` — значение переменной `b`, т. е. 4.

В Python 3 так называемые *неизменяемые* объекты, то есть целые числа и символьные строки, могут быть переданы в функции как значения. Это означает, что их значение не может быть изменено функцией.

С другой стороны, *изменяемые* объекты, такие как списки, передаются в функции как ссылки (см. пример ниже). Это означает, что их значение может быть изменено функцией:

```

1 def addList(my_list, x):
2     my_list.append(x)
3
4 L=[1,2,3]
5 print(L)          # [1, 2, 3]
6 addList(L,4)
7 print(L)          # [1, 2, 3, 4]

```

Изменения в списке, переданном в качестве параметра функции, действуют на список, определенный вне функции `addList`.

## Библиотеки

В Python 3 есть возможность подключения библиотек, в основном тематических, состоящих из уже запрограммированных функций. Для этого в начале программы необходимо указать имя библиотеки, предварив его ключевым словом `import`. Например, `import math` позволяет использовать при написании программы математические функции `math.exp(x)`, `math.sqrt(x)`, `math.cos(x)` или `math.sin(x)`, которые, соответственно, вычисляют экспоненту, квадратный корень, косинус и синус числа  $x$ .

В Python 3 доступны и другие библиотеки, такие как, например, библиотеки `random` или `time`. Библиотека `random` позволяет генерировать случайные числа. С помощью команды `time()` из библиотеки `time` можно узнать текущее время и измерить таким образом, например, продолжительность выполнения программы.

```

1 import time
2 import random
3
4 start = time.time()          # Отображает текущее время
5 r = random.randint(10,100)  # Случайным образом выбирает целое число в диапазоне от 10 до 100
6 end = time()
7 duration = end - start

```

Также можно добавить библиотеки, которые по умолчанию в Python 3 не включены. Например, чтобы добавить библиотеку `turtle`, необходимую для решения головоломки 11, можно использовать следующую команду:

```
python3 -m pip install turtle
```