

# Python для школьников

Занятие 1. «Знакомство с Python:  
ввод/вывод, целочисленная арифметика»

Паньгина Н.Н. 2018



# История языка Python

Первая версия языка – 1994 год

Вторая версия языка – 2000 год

Третья версия языка – 2008 год



Гвидо ван Россум

## Историческая справка

Гвидо ван Россум известен среди разработчиков как «Великодушный пожизненный диктатор Python».

Это означает, что он продолжает наблюдать за разработкой языка и принимает окончательные решения по его развитию, когда это необходимо.

# Возможности языка Python
















Язык программирования **Python 3** — это мощный инструмент для создания программ самого разнообразного назначения, доступный даже для новичков.

- **Python** - интерпретируемый язык программирования
- Кроссплатформенность и бесплатность
- Понятный и лаконичный синтаксис
- Динамическая типизация
- Хорошая поддержка модульности
- Поддержка объектно-ориентированного программирования
- И др.

# Установка языка Python и среды разработки

- Скачать дистрибутив с официального сайта <http://www.python.org>
- Рекомендуется использование версии 3..., как более современные
- Прямая ссылка на дистрибутив 3.6.0 для Windows: [32 бита](#), [64 бита](#). При установке рекомендуется выбирать вариант "Install for all users"
- Можно использовать IDLE – стандартную среду разработки для Python
- Также есть среда [JetBrains PyCharm](#), более богатая возможностями, но и более сложная в использовании для начинающих
- Простая кроссплатформенная бесплатная среда для обучения – Wing IDE 101 (<http://wingware.com/downloads/wingide-101>)

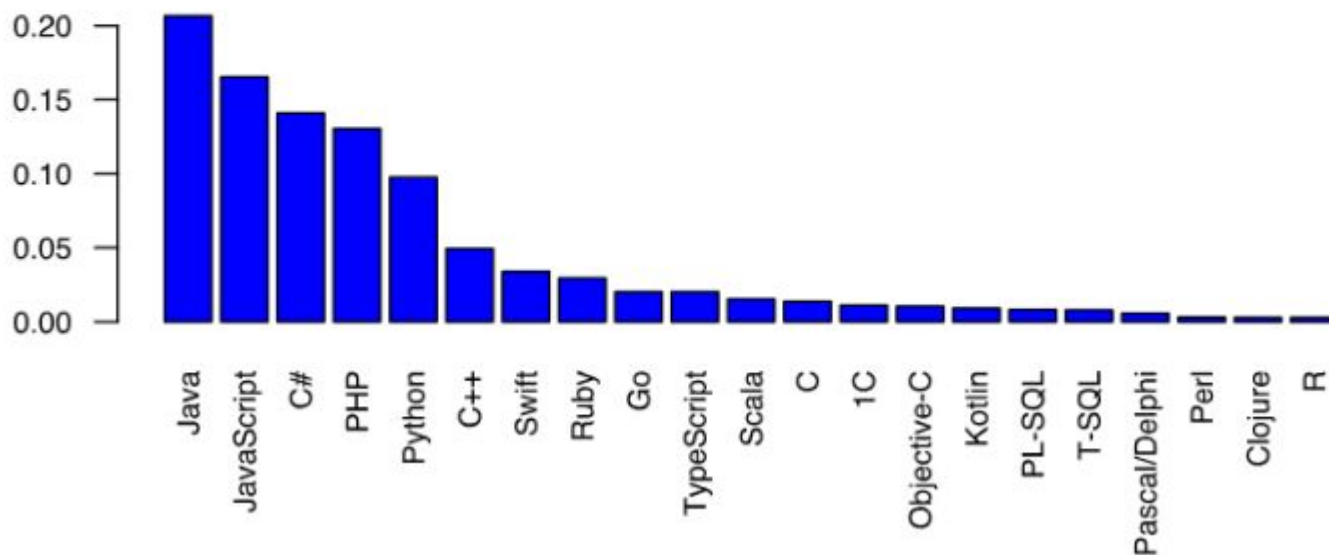
# 2017г. Популярность Python

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

# 2018г. Популярность Python

Коммерческое использование

На каком языке вы пишете для работы сейчас



# 2019г. Популярность Python

Language Types (click to hide)



Web



Mobile



Enterprise



Embedded

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C++		99.7
3. Java		97.5
4. C		96.7
5. C#		89.4
6. PHP		84.9
7. R		82.9
8. JavaScript		82.6
9. Go		76.4
10. Assembly		74.1



# Оболочка IDLE для Python

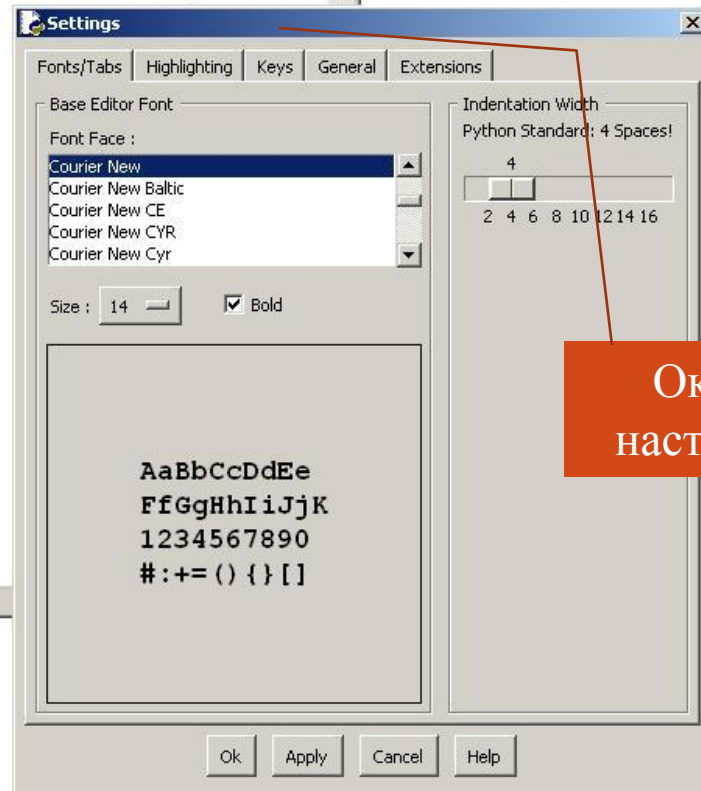
Главное меню

Пункт меню «Настройки»

Встроенный калькулятор

Ввод/вывод данных

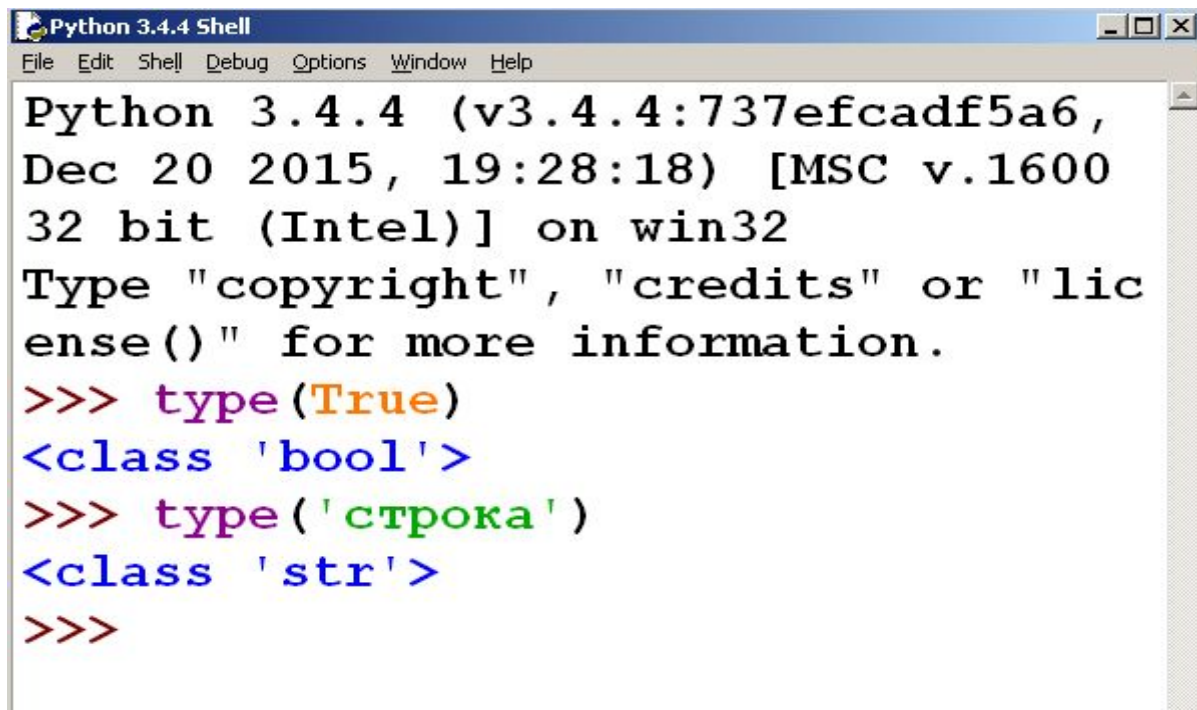
```
Python 3.4.4 Shell
File Edit Shell Debug Options Window Help
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015,
19:28:18) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for
more information.
>>> (2+8)*(54-14)/40
10.0
>>> ((2+8)*(54-14)/40)**3
SyntaxError: invalid syntax
>>> ((2+8)*(54-14)/40)**3
1000.0
>>>
```



Окно настроек

# Типы данных в Python

- Числовые (целые – **int** (43), вещественные – **float** (3.1415))
- Строковые – **str** ('Строка', "Текст")
- Булевские – **bool** (True, False)



```
Python 3.4.4 Shell
File Edit Shell Debug Options Window Help
Python 3.4.4 (v3.4.4:737efcadf5a6,
Dec 20 2015, 19:28:18) [MSC v.1600
32 bit (Intel)] on win32
Type "copyright", "credits" or "lic
ense()" for more information.
>>> type(True)
<class 'bool'>
>>> type('строка')
<class 'str'>
>>>
```

# Основные операции в Python

**Для чисел:**

$A + B$  — сумма;

$A - B$  — разность;

$A * B$  — произведение;

$A / B$  — частное;

$A // B$  — деление нацело;

$A \% B$  — остаток от деления;

$A ** B$  — возведение в степень.

**Для строк:**

$A + B$  — конкатенация;

$A * n$  — повторение  $n$  раз,  
значение  $n$  должно быть целого  
типа.

# Приоритеты операций в Python

1. Выполняются операции возведения в степень **справа налево**.
2. Выполняются унарные минусы (отрицания).
3. Выполняются умножения и деления слева направо. Операции умножения и деления имеют одинаковый приоритет.
4. Выполняются сложения и вычитания слева направо. Операции сложения и вычитания имеют одинаковый приоритет.

**Пример 1: чему будет равно  $1 + 2 * 3 ** 1 + 1$ ?**

**8**

**Пример 2: чему будет равно  $3 ** 3 ** 3$  ?**

**762559748498**

**7**

# Преобразование типов в Python

Для этого используются функции, одноименные с именем типа, то есть **int**, **float**, **str**.

Например, `int('123')` вернет целое число `123`, а `str(123)` вернет строку `'123'`.

Пример:

Что будет в результате `str(2 + 2) * int('2' + '2')`?

`'444444444444444444444444444444'`

# Работа со встроенным калькулятором

Упражнения (для самостоятельного выполнения):

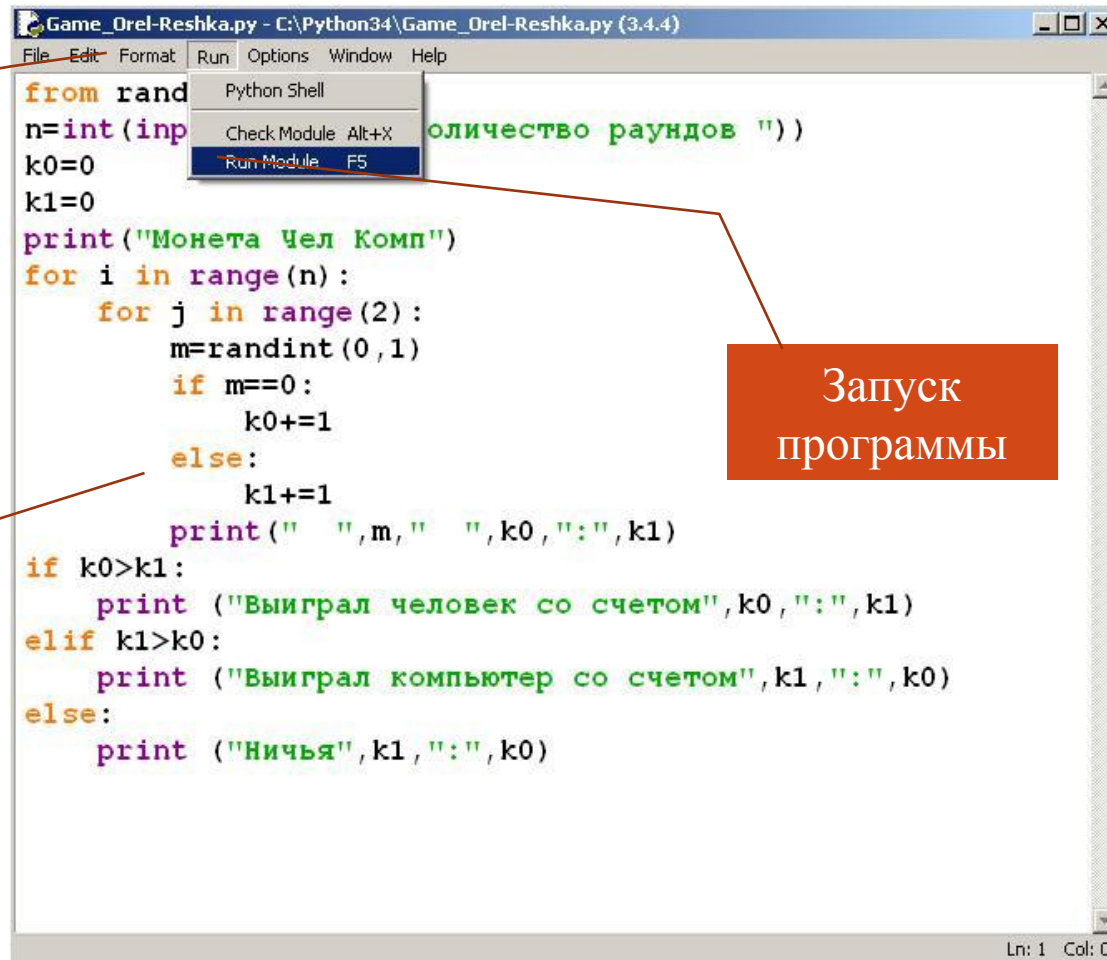
- Вычислить выражение  $(16386 - 396) : 78 + 402 \times 306$  123217
  
- Вычислить факториал числа  $17!$  355687428096000
  
- Какой цифрой оканчивается сумма  $29^{2019} + 5$  4
  
- Найти частное и остаток от деления числа  $23^{13}$  на целую часть от числа  $[2019:67]$  1680121206454891  
2  
23

# Окно написания программы на Python

Главное меню

Код программы

Запуск программы



The screenshot shows a Python IDE window titled "Game\_Orel-Reshka.py - C:\Python34\Game\_Orel-Reshka.py (3.4.4)". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". A dropdown menu is open under "Run", showing "Python Shell", "Check Module Alt+X", and "Run Module F5". The code in the editor is as follows:

```
from random import randint
n=int(input("Введите количество раундов "))
k0=0
k1=0
print("Монета Челл Комп")
for i in range(n):
    for j in range(2):
        m=randint(0,1)
        if m==0:
            k0+=1
        else:
            k1+=1
        print(" ", m, " ", k0, ":", k1)
if k0>k1:
    print ("Выиграл человек со счетом", k0, ":", k1)
elif k1>k0:
    print ("Выиграл компьютер со счетом", k1, ":", k0)
else:
    print ("Ничья", k1, ":", k0)
```

At the bottom right of the window, the status bar shows "Ln: 1 Col: 0".

# Оператор вывода на Python

```
*Untitled*  
File Edit Format Run Options Window Help  
print (  
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

```
print_1.py - C:/Python34/print_1.py (3.4.4)  
File Edit Format Run Options Window Help  
print (10, 20, 50.5, 3141592e-6)  
print (10//2 + 20/5 - 5**2 + 3 * 6)  
print (10,20,30,sep=":")  
print (10,20,30,sep=" ")  
print (100)  
print (200)  
print (400,end = ' ')  
print (500,end = ' ')  
print (600)  
print ("Я сразу смазал карту будня,")  
print ('Плеснувши краску из стакана.')
```

```
Python 3.4.4 Shell  
File Edit Shell Debug Options Window Help  
C:/Python34/print_1.py  
=====  
10 20 50.5 3.141592  
2.0  
10:20:30  
10 20 30  
100  
200  
400 500 600  
Я сразу смазал карту будня,  
Плеснувши краску из стакана.  
>>> |
```

Конец строки

Разделитель

Результат работы программы

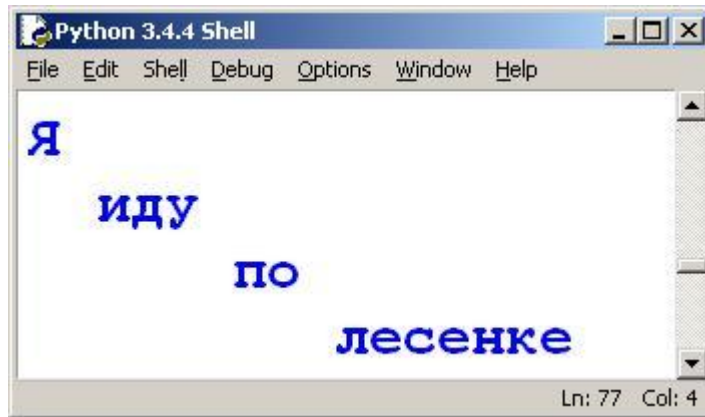


# Закрепление пройденного

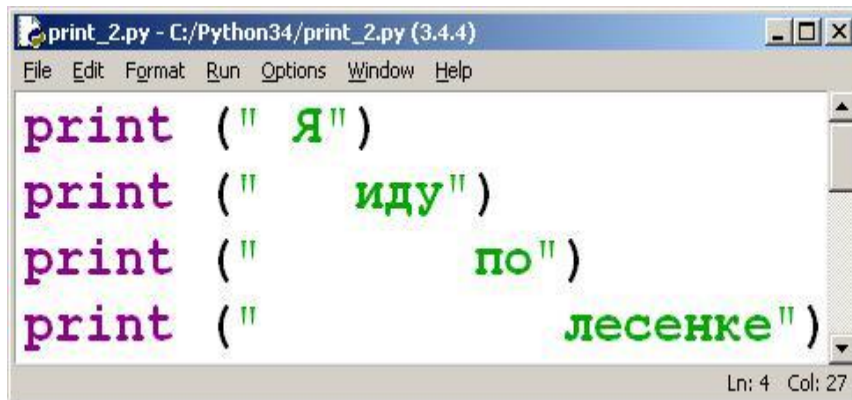
Упражнения (для самостоятельного выполнения):

Напечатать текст «лесенкой»

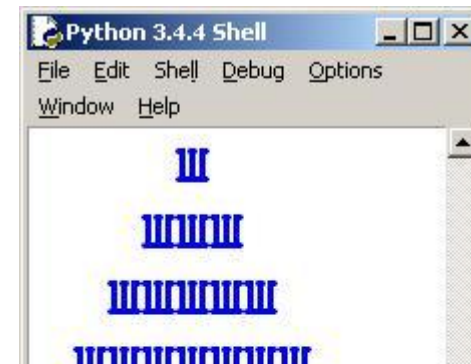
Нарисовать «Елочку»



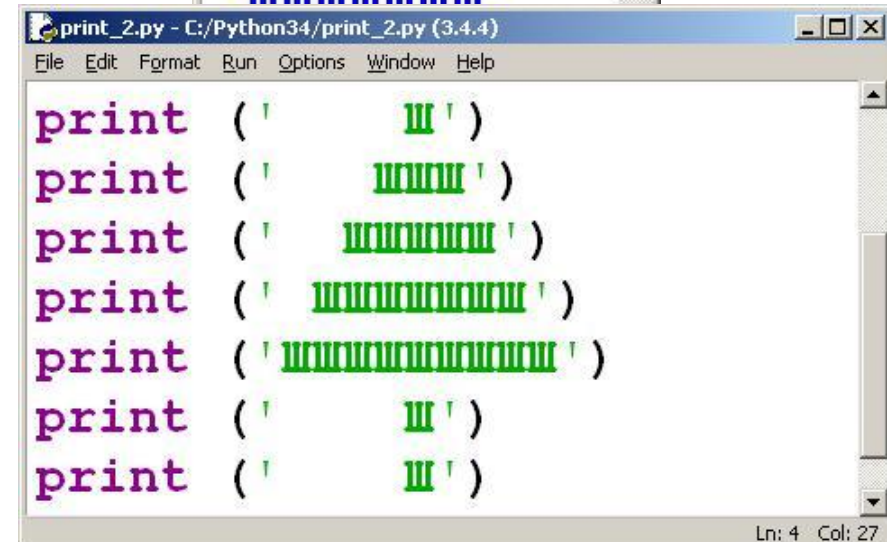
```
Python 3.4.4 Shell
File Edit Shell Debug Options Window Help
Я
    иду
        по
            лесенке
Ln: 77 Col: 4
```



```
print_2.py - C:/Python34/print_2.py (3.4.4)
File Edit Format Run Options Window Help
print (" Я")
print ("    иду")
print ("        по")
print ("            лесенке")
Ln: 4 Col: 27
```



```
Python 3.4.4 Shell
File Edit Shell Debug Options
Window Help
      III
     IIIIII
    IIIIIIII
   IIIIIIIIII
  IIIIIIIIII
 IIIIIIIIII
IIIIIIIII
IIIIIIIII
Ln: 77 Col: 4
```



```
print_2.py - C:/Python34/print_2.py (3.4.4)
File Edit Format Run Options Window Help
print ('      III ')
print ('     IIIIII ')
print ('    IIIIIIII ')
print ('   IIIIIIIIII ')
print ('  IIIIIIIIII ')
print (' IIIIIIIIII ')
print ('IIIIIIIII ')
print ('IIIIIIIII ')
Ln: 4 Col: 27
```

# Операторы ввода и присваивания в Pythonе

```
*inp_1.py - C:/Python34/inp_1.py (3.4.4)*
File Edit Format Run Options Window Help
a = input(
input([prompt]) -> string
```

Подсказка

Вводятся понятия переменных и констант

Написать программу «Диалог с компьютером»

```
*inp_1.py - C:/Python34/inp_1.py (3.4.4)*
File Edit Format Run Options Window Help
# Диалог с компьютером
a = input("Как тебя зовут? ")
print ("Привет, ", a, "! А я ПК!", sep = " ")
n = input("Сколько тебе лет? ")
print ("Тебе ", n, ", а мне всего 5!", sep = " ")
h = input("Назови свое хобби ")
print ("Твое ", h, ", а мое - решать задачи!", sep = " ")
Ln: 7 Col: 13
```

Компьютер

Результат работы программы

# Задачи на линейные алгоритмы

- Вводится целое число. Вывести предыдущее и последующее числа.

## Пример

*Входные данные:*

8

*Выходные данные:*

Перед 8 число 7

За 8 число 9

- Вводятся длины двух сторон прямоугольника (в метрах). Вывести его площадь и периметр.

## Пример

*Входные данные:*

1-я сторона 8

2-я сторона 5

*Выходные данные:*

Площадь 40 кв.м

Периметр 26 м

```
Python 3.4.4 Shell
File Edit Shell Debug Options Window Help
```

```
1_2_0.py - F:/1_Python/Python/1_2_0.py (3.4.4)
File Edit Format Run Options Window Help
a=int(input())
print('Перед',a,'число',
print('За',a,'число',a+1
```

```
Python 3.4.4 Shell
File Edit Shell Debug Options Window Help
```

```
3_1.py - F:/1_Python/Python/3_1.py (3.4.4)
File Edit Format Run Options Window Help
a=int(input("1-я сторона "))
b=int(input("2-я сторона "))
print('Площадь',a*b,'кв.м')
print('Периметр',(a+b)*2,'м')
```

Ln: 4 Col: 29

Ln: 9 Col: 4

# Задачи на линейные алгоритмы

(для самостоятельного выполнения)

□ Вводится трехзначное число.

Вывести сумму его цифр.

Пример

*Входные данные:*

**876**

*Выходные данные:*

**21**

□ Вводится целое число. Вывести следующее за ним четное число.

Пример

*Входные данные:*

**7            8**

*Выходные данные:*

**8            10**

□ Дано число  $n$ . С начала суток прошло  $n$  минут. Определите, сколько часов и минут будут показывать электронные часы в этот момент. Программа должна вывести два числа: количество часов (от 0 до 23) и количество минут (от 0 до 59). Учтите, что число  $n$  может быть больше, чем количество минут в сутках.

Пример

*Входные данные:*

**150            1441**

*Выходные данные:*

**2:30            0:1**

# Дистанционная подготовка

- <https://informatics.mccme.ru/>
- <https://informatics.msk.ru/>

The screenshot shows a web browser window with the address bar displaying <https://informatics.mccme.ru/>. The page header features a graduation cap icon and the text "Дистанционная подготовка по информатике". A notification in the top right corner states "Вы не прошли идентификацию (Вход)".

**Контакты**  
По всем текущим вопросам, а также в случае, если у вас что-то не работает, пишите на [форум](#) или в [группу telegram](#).  
По вопросам авторизации учителей пишите на почту [informatics.msk@gmail.com](mailto:informatics.msk@gmail.com)

**Основное меню**

- Каталог по темам и источникам
- Мои посылки
- Мои группы
- Мои настройки
- Сообщения об ошибках
- Топ идеальных решений
- Рейтинг пользователей

**О сайте**

- Команда проекта
- Новые материалы
- FAQ
- Виртуальные турниры и командные логины

**Организации-разработчики**

- Центр педагогического

**Изучение языка программирования** [Развернуть всё](#)

- Ввод-вывод, оператор присваивания, арифметические операции
- Условный оператор
- Операторы цикла
- Типы данных
- Массивы
- Функции и процедуры. Рекурсия
- Особенности языков программирования
- Составление тестов к задачам
- Учим python (в стадии разработки)

**Авторские курсы (5)**

- Ф. Меньшиков. Олимпиадные задачи по программированию
- Е. В. Андреева. Программирование — это так просто, программирование — это так сложно.
- Д. П. Кириенко. Программирование на языке Python (школа 179 г. Москвы)
- фирма "1С". "Алгоритмы. Олимпиадное программирование на языке Java для школьников"
- Задачи из учебника К.Ю. Полякова и Е.А. Еремина

**Группа техподдержки в telegram**

**Вход**

Логин:   
Пароль:

[Регистрация](#)  
[Регистрация учителя](#)  
[Регистрация команды](#)  
[Забыли пароль?](#)

**К задаче №**

**Поиск**

**Новостной форум**

- 16 окт 22:35: Онлайн-курсы олимпиадной информатики
- 17 авг 17:26: Начинается отбор на курсы алгоритмов и

СПАСИБО ЗА ВНИМАНИЕ!

Паньгина Нина Николаевна,  
E-mail: [nina\\_pangina@mail.ru](mailto:nina_pangina@mail.ru)