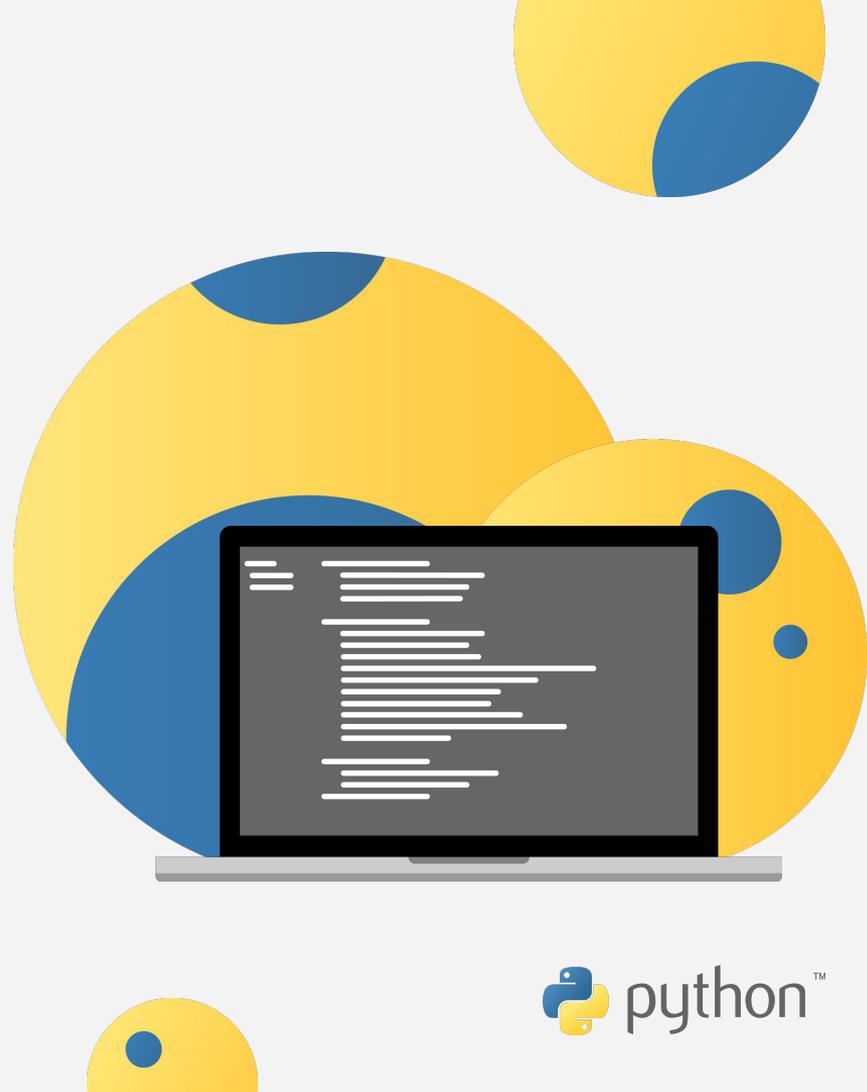


## Урок 3

# Работа с целыми числами

Поколение Python:  
курс для начинающих





# Вспоминаем прошлый урок

2

1. Для чего используется параметр `sep`?
2. Для чего используется параметр `end`?
3. Что такое переменная?
4. Для чего нужны комментарии в коде?
5. Что такое PEP 8?



# Целые числа



## Целые числа

Все предыдущие программы, работали с **текстовыми данными**  
Во многих случаях нам нужно работать именно с **числами**

Для того, чтобы в языке Python создать переменную целого типа, нужно опустить кавычки при объявлении переменной:

```
num1 = 7          # num1 - это число
num2 = 10         # num2 - это число
num3 = num1 + num2 # num3 - это число
print(num3)
```

Будет выведено  
число 17



числа обозначаются без кавычек, а строки с кавычками

# Арифметические операции

В Python над числами можно совершать **4 основные операции**:

Операция	Описание
+	сложение
-	вычитание
*	умножение
/	деление

```
a = 3
b = 2
print(a + b)
print(a - b)
print(a * b)
print(a / b)
```



```
5
1
6
1.5
```

# Приоритет арифметических операций

Порядок выполнения арифметических операций в Python аналогичен порядку выполнения операций в математике:

Операция	Приоритет
*	1
/	1
+	2
-	2

```
num1 = 2 + 3 * 4
num2 = (2 + 3) * 4
print(num1)
print(num2)
```



```
14
20
```



для изменения приоритета понадобятся скобки

# Преобразование строки к целому числу

7

Чтобы преобразовать строку к целому числу, мы используем команду `int()`:

```
age = '1992'  
year = int(age)  
grade = int(input())
```

Переменная `age` имеет строковый тип

Переменная `year` имеет целочисленный тип

Переменная `grade` имеет целочисленный тип



`int()`: возьми то, что указано в скобках и преврати это в целое число

# Преобразование строки к целому числу

8

Программа, которая считывает два целых числа и выводит на экран их сумму:

```
num1 = int(input())  
num2 = int(input())  
print(num1 + num2)
```



чтобы считать одно целое число, мы пишем код:

```
num = int(input())
```



# Задача



Указать значение переменной **s** после выполнения следующего кода.

```
s = 0
k = 30
d = k - 5
k = 2 * d
s = k - 100
```

Варианты ответов:

A. -70

B. 50

C. -50

# Задача



Указать значение переменной **s** после выполнения следующего кода.

```
s = 0  
k = 30  
d = k - 5  
k = 2 * d  
s = k - 100
```

Ответ:

-50

# Задача



Указать значение переменной **x** после выполнения следующего кода.

```
x = 3
y = 4
z = x + y
z = z + 1
x = y
y = 5
x = z + y + 7
```

Варианты ответов:

A. 14

B. 15

C. 20

# Задача



Указать значение переменной **x** после выполнения следующего кода.

```
x = 3
y = 4
z = x + y
z = z + 1
x = y
y = 5
x = z + y + 7
```

Ответ:

20

# Задача



Что появится на экране после выполнения следующей программы?

```
a = 4  
print(a, 'a')
```

Варианты ответов:

- A. a, 'a'
- B. 4 4
- C. 4 a

# Задача



Что появится на экране после выполнения следующей программы?

```
a = 4  
print(a, 'a')
```

Ответ:

```
4 a
```

# Задача «Три последовательных числа»

16



Напишите программу вывода на экран трех последовательно идущих чисел, каждое на отдельной строке. Первое число вводит пользователь, остальные числа вычисляются в программе.

Ввод	Вывод
7	7 8 9

В курсе: урок 2.4, шаг 5

# Задача «Сумма трёх чисел»

17



Напишите программу, которая считывает три целых числа и выводит на экран их сумму. Каждое число записано в отдельной строке.

Ввод	Вывод
9 11 2	22

В курсе: урок 2.4, стэп 6

# Задача «Следующее и предыдущее»

18



Напишите программу, которая считывает целое число, после чего на экран выводится следующее и предыдущее целое число с пояснительным текстом.

Ввод	Вывод
20	Следующее за числом 20 число: 21 Для числа 20 предыдущее число: 19

В курсе: урок 2.4, стэп 9



# Дополнительные операции



# Дополнительные операции

20

В Python над числами можно совершать **3 дополнительные операции:**

1. возведение в степень \*\*
2. целочисленное деление //
3. нахождение остатка %

# Возведение в степень

Операция возведения в степень `a**n` возводит число `a` в степень `n`:

```
print(2 ** 0)
print(2 ** 1)
print(2 ** 2)
print(2 ** 3)
print(2 ** (-1))
```



```
1
2
4
8
0.5
```



В отличие от многих языков программирования, в Python для возведения в степень используется символ `**`, а не `^`

# Целочисленное деление

Операция целочисленного деления `//` отбрасывает десятичную часть результата:

```
print(10 // 3)
print(10 // 4)
print(10 // 5)
print(10 // 6)
print(10 // 12)
print(-10 // 12)
```



```
3
2
2
1
0
-1
```



при целочисленном делении округление берётся в меньшую сторону

# Нахождение остатка от деления

23

Операция нахождения остатка `%` возвращает остаток от деления двух целых чисел:

```
print(10 % 3)
print(10 % 4)
print(10 % 5)
print(10 % 6)
print(10 % 12)
print(3 % 7)
```



```
1
2
0
4
10
3
```



при нахождении остатка от деления на целое число **n**,  
получаются числа **0, 1, 2, ..., n-1**

# Примечания

Операция **нахождения остатка** очень полезна при решении задач:



число делится на **n** нацело, если остаток от деления на **n** равен **0**

Приоритет дополнительных операций:

Операция	Приоритет
**	0
//	1
%	1

операции **//** и **%** имеют такой же приоритет как и операции умножения и обычного деления

# Задача



Вычислите результат возведения в степень

1.  $1 ** 10$
2.  $2 ** 3$
3.  $3 ** 2$
4.  $5 ** 4$

# Задача



Вычислите результат возведения в степень

1.  $1 ** 10 = 1$
2.  $2 ** 3 = 8$
3.  $3 ** 2 = 9$
4.  $5 ** 4 = 625$

# Задача



Вычислите результат деления

1.  $24 // 5$
2.  $34 // 3$
3.  $100 // 12$
4.  $120 // 90$

# Задача



Вычислите результат деления

1.  $24 // 5 = 4$
2.  $34 // 3 = 11$
3.  $100 // 12 = 8$
4.  $120 // 90 = 1$

# Задача



Вычислите результат деления

1.  $24 \div 5$
2.  $34 \div 3$
3.  $100 \div 15$
4.  $120 \div 90$

# Задача



Вычислите результат деления

1.  $24 \div 5 = 4$

2.  $34 \div 3 = 1$

3.  $100 \div 15 = 10$

4.  $120 \div 90 = 30$

# Задача



Что будет выведено на экран в результате выполнения следующей программы?

```
a = 82 // 3 ** 2 % 5  
print(a)
```

# Задача



Что будет выведено на экран в результате выполнения следующей программы?

```
a = 82 // 3 ** 2 % 5  
print(a)
```

Ответ:

4

# Задача «Расстояние в метрах»

33



Напишите программу, которая находит полное число метров по заданному числу сантиметров.

Ввод	Вывод
345	3
200	2

В курсе: урок 2.5, стэп 7

# Задача «Мандарины»



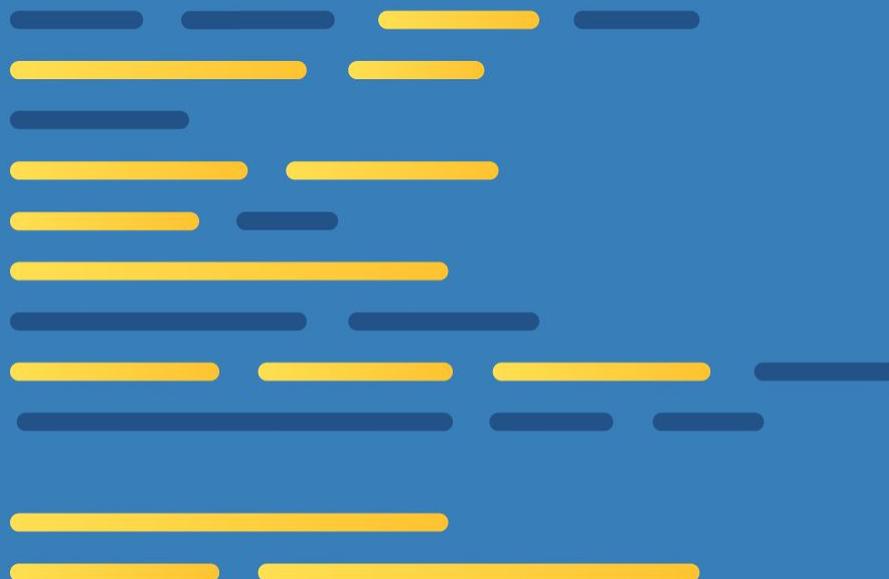
$n$  школьников делят  $k$  мандарин поровну, неделящийся остаток остается в корзине. Сколько мандарин достанется каждому школьнику? Сколько мандарин останется в корзине?

Ввод	Вывод
10	2
25	5

В курсе: урок 2.5, стэп 8



# Обработка цифр числа



# Цифры двузначного числа

При помощи операций `//` и `%` можно вычислять цифры числа:

```
num = 17
a = num % 10
b = num // 10
print(a)
print(b)
```



```
7
1
```



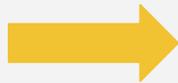
получить последнюю цифра числа **n**: `n % 10`

отщепить последнюю цифру от числа **n**: `n // 10`

# Цифры трехзначного числа

При помощи операций `//` и `%` можно вычислять цифры числа:

```
num = 754
a = num % 10
b = (num % 100) // 10
c = num // 100
print(a)
print(b)
print(c)
```



```
4
5
7
```



получить последнюю цифра числа **n**: `n % 10`

отщепить последнюю цифру от числа **n**: `n // 10`

# Алгоритм нахождения цифр

Алгоритм нахождения цифр  $n$ -значного числа `num`:

- последняя цифра: `(num % 10**1) // 10**0;`
- предпоследняя цифра: `(num % 10**2) // 10**1;`
- предпредпоследняя цифра: `(num % 10**3) // 10**2;`
- .....
- вторая цифра: `(num % 10**n-1) // 10**n-2;`
- первая цифра: `(num % 10**n) // 10**n-1.`

# Задача «Трехзначное число»

39



Напишите программу, в которой рассчитывается сумма и произведение цифр положительного трехзначного числа.

Ввод	Вывод
333	Сумма цифр = 9 Произведение цифр = 27

В курсе: урок 2.5, стэп 13

# Над созданием курса работали:

40

«Поколение Python»: курс для начинающих» разработан специально для  
Благотворительного фонда развития образования «Айкью Опшн»  
авторским коллективом **школы BEEGEEK**.

Авторы методических указаний для преподавателей:

- **Ольга Скобина** (Ассоциация Российских Тренеров, АРТa)
- **Тимур Гуев** (ОАНО Новая школа)
- **Роман Белых** (БФ развития образования «Айкью Опшн»)

Дизайнеры:

- **Юлия Уебельман**
- **Ольга Галак** (БФ развития образования «Айкью Опшн»)

Редактор:

- **Наталья Белых**



**iq option**  
благотворительный фонд



**ПОКОЛЕНИЕ  
PYTHON**