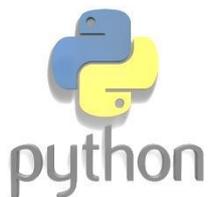


Обработка целых чисел

Программирование на языке
Python

Ключевые понятия:

- Арифметические операции
- Арифметические выражения
- Остаток
- Частное
- Приоритет операций
- **Форматный вывод**



Числовые типы данных

Стандартные функции языка Python:

Функция	Назначение	Тип аргумента	Тип результата
<code>abs(x)</code>	Модуль x	int, float	Такой же, как у аргумента
<code>round(x)</code>	Округление вещественного x до заданного количества знаков после запятой	float	int, float
<code>int(x)</code>	Преобразование вещественного или строкового x к целому	str, float	int

Прочие математические функции собраны в модуле `math`, подключение модуля:

`from math import *`

<code>sqrt(x)</code>	Квадратный корень из x	int, float	float
----------------------	--------------------------	------------	-------

Датчик случайных чисел: **`from random import *`**

<code>randint(a,b)</code>	Целое случайное число в интервале $[a, b]$	int	int
<code>random ()</code>	Вещественное случайное число в интервале $[0, 1)$		float



Исследование функций `round`, `int`

```
print ('Исследование функций round, int ')
x = float(input (' Введите x>>'))
print ('Округление - ', round( x))
print ('Целая часть - ', int(x))
```

Выполните программу несколько раз для

$x \in \{10,2; 10,8; -10,2; -10,8\}$.

Какой будет тип результата каждой из этих функций?



Целочисленный тип данных

Операции над целыми числами:

Операция	Обозначение	Тип результата
Сложение	+	int
Вычитание	-	int
Умножение	*	int
Возведение в степень	**	int
Получение целого частного	//	int
Получение целого остатка деления	%	int
Деление	/	float

Операции // и %

Трёхзначное число можно представить в виде следующей суммы: $x = a \cdot 100 + b \cdot 10 + c$, где a, b, c - цифры числа.

Программа нахождения суммы цифр вводимого с клавиатуры целого трёхзначного числа:

```
print ('Нахождение суммы цифр трёхзначного числа');  
x = int (input ('Введите исходное число>>'))  
a = x // 100  
b = x % 100 // 10  
c = x % 10  
s = a + b + c  
print ('s= ', s)
```



Логический тип данных

Величины логического типа принимают всего два значения:

False и **True**;

False < **True**.

Логические значения получаются в результате выполнения операций сравнения числовых, строковых и логических выражений.

В языке Python логической переменной можно присваивать результат операции сравнения.



Логический тип данных

Пусть **ans** - логическая переменная,

n - целая переменная.

В результате выполнения оператора присваивания

ans = n % 2 == 0

переменной **ans** будет присвоено значение **True** при любом чётном *n* и **False** в противном случае.

```
# является ли число чётным?
```

```
n = int (print ('Введите исходное число>>'))
```

```
ans = n% 2 == 0
```

```
print ('Число', n, 'является чётным -', ans)
```



Логический тип данных

Логическим переменным можно присваивать значения логических выражений, построенных с помощью логических функций и (**and**), или (**or**), не (**not**).

Логическая операция в Python	Название операции
and	конъюнкция (логическое умножение)
or	дизъюнкция (логическое сложение)
not	отрицание (инверсия)



Логический тип данных

```
# Является ли треугольник равнобедренным?  
a = int(input('Введите длину стороны a>>'))  
b = int(input('Введите длину стороны b>>'))  
c = int(input('Введите длину стороны c>>'))  
ans = a == b or a == c or b == c  
print ('Треугольник со сторонами', a, ', ', b, ', ', c, ' является  
равнобедренным –', ans)
```



Порядок выполнения операций

3 1 2 4 5 6

```
a = (c + b**5*3 - 1) / 2 * d
```

Приоритет (*старшинство*):

- 1) скобки
- 2) возведение в степень **
- 3) умножение и деление
- 4) сложение и вычитание

$$a = \frac{c + b^5 \cdot 3 - 1}{2} \cdot d$$

```
a = (c + b**5*3 - 1) \
```

```
      / 2 * d
```

перенос на
следующую строку

```
a = (c + b**5*3
```

```
      - 1) / 2 * d
```

перенос внутри
скобок разрешён



Деление

Классическое деление:

```
a = 9; b = 6
x = 3 / 4      # = 0.75
x = a / b      # = 1.5
x = -3 / 4     # = -0.75
x = -a / b     # = -1.5
```

Целочисленное деление (округление «вниз»!):

```
a = 9; b = 6
x = 3 // 4     # = 0
x = a // b     # = 1
x = -3 // 4    # = -1
x = -a // b    # = -2
```



Частное и остаток

// – деление нацело (остаток отбрасывается)

% – остаток от деления

175 сек = 2 мин 55 сек



Как получить 2 и 55?

```
t = 175
```

```
m = t // 60
```

```
s = t % 60 # 55
```



Частное и остаток



Что получится?

```
n = 123
```

```
d = n // 10
```

```
k = n % 10
```

При делении на 10 нацело отбрасывается последняя цифра числа.

Остаток от деления на 10 – это последняя цифра числа.



Операторы // и %

```
a = 1234
d = a % 10; print( d )
a = a // 10 # 123
d = a % 10; print( d )
a = a // 10 # 12
d = a % 10; print( d )
a = a // 10 # 1
d = a % 10; print( d )
a = a // 10 # 0
```

4

3

2

1



Сокращенная запись операций

```
a += b # a = a + b
a -= b # a = a - b
a *= b # a = a * b
a /= b # a = a / b
a //= b # a = a // b
a %= b # a = a % b
```

```
a += 1
```

увеличение на 1



Ввод двух значений в одной

```
a, b = map ( int, input() .split() )
```

21 33

`input()`

ввести строку с клавиатуры

21 33

`input().split()`

целые

применить

разделить строку на части по пробелам

21 33

`map (int, input().split())`

эту операцию

к каждой части

```
a, b = map ( int, input() .split() )
```



ФОРМАТНЫЙ ВЫВОД

```
a = 1; b = 2; c = 3  
print( a, b, c )
```

1 2 3

форматная строка

```
print ( "{ } { } { } " .format ( a , b , c ) )
```

123

тут нужно что-то
вывести

```
print ( "{ } { :3 } { :5 } " .format ( a , b , c ) )
```

количество знаков
на вывод числа

1 2 3
3 5

?

Сколько знаков для вывода *a*?



ФОРМАТНЫЙ ВЫВОД

```
a = 1; b = 2
```

```
print("{}+{}={}".format(a,b,c))
```



```
1+2=3
```



ФОРМАТНЫЙ ВЫВОД

```
x=12.345678
```

```
print("x={}" .format(x))
```

```
x=12.345678
```

всего на
число

в дробной
части

```
print("x={:10.3f}" .format(x))
```

```
_____12.346
```

3

10

```
print("x={:8.2f}" .format(x))
```

```
____12.34
```



ФОРМАТНЫЙ ВЫВОД

```
print ("x={ :2.2f} ".format(x))
```

→ 12.34

```
print ("x={ :.2f} ".format(x))
```

→ 12.34

МИНИМАЛЬНО
ВОЗМОЖНОЕ

```
print ("x={ :0.1f} ".format(x))
```

→ 12.3



Задачи

«3»: Ввести число, обозначающее количество секунд. Вывести то же самое время в минутах и секундах.

Пример:

Введите число секунд: 175

2 мин. 55 с.

«4»: Ввести число, обозначающее количество секунд. Вывести то же самое время в часах, минутах и секундах.

Пример:

Введите число секунд: 8325

2 ч. 18 мин. 45 с.

«5»: Занятия в школе начинаются в 8-30. Урок длится 45 минут, перерывы между уроками – 10 минут. Ввести номер урока и вывести время его окончания.

Пример:

Введите номер урока: 6

13-50



- Арифметические выражения могут содержать константы (постоянные значения), имена переменных, знаки арифметических операций, круглые скобки (для изменения порядка действий).
- Порядок выполнения действий определяется **приоритетом** операций: сначала выполняется возведение в степень, затем – умножение и деление, затем – сложение и вычитание. Для изменения порядка действий используются скобки.
- Для деления нацело используется оператор `//`, для взятия остатка от деления – оператор `%`.
- Форматный вывод данных выполняется с помощью функции ***format***.

