



ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

НАЧАЛА ПРОГРАММИРОВАНИЯ

8 класс

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)
```

