



# Семинар 2

Введение в программирование на Python

Папулин С.Ю.  
papulin\_hse@mail.ru

2015

1. Переменные в Python
2. Алгоритмы. Блок-схемы
3. Условные выражения
4. Примеры задач
5. Самостоятельная работа

1. Знакомство со средой PyCharm
2. Загрузка программ на портал дистанционного обучения
3. Функции `print`, `input`:



`print("Text")` – функция вывода текста на экран

`text=input("Enter your text: ")` – функция, записывающая введенную с клавиатуры строку в переменную ***text***.

# Семинар 1

Если необходимо вывести переменные, то можно использовать следующие подходы.

```
name = "Alex" #Переменная типа строка (string)
bal = 40000 #Переменная целого типа (integer)
```

- 1 *#Вывод строки*  
`print(name, bal)`  Результат "Alex 40000"
- 2 *#Вывод строки*  
`print("Hello, " + name + "! Your balance is $" + str(bal))`
- 3 *#Вывод строки*  
`print("Hello, %s! Your balance is $%i" % (name, bal))`  
  
"Hello, Alex! Your balance is \$40000"

# Типы данных (data types) и переменные (variables)

- |                                 |                                 |
|---------------------------------|---------------------------------|
| 1. Целые числа (integer)        | Пример: -346, -34, 2, 3, 5, 665 |
| 2. Действительные числа (float) | Пример: -3.454, 0.743, 65.223   |
| 3. Строка (string)              | Пример: "hello", "text"         |

## Объявление и присвоение значения переменной:

*имя\_переменной = значение*

```
var_1 = 3 #Переменная типа integer
var_2 = -43 #Переменная типа integer
var_3 = 3.0 #Переменная типа float
var_4 = -6.32 #Переменная типа float
var_5 = "text" #Переменная типа string
var_6 = "333" #Переменная типа string
```

# Операции над числами

Обозначение	Операция	Пример
$x + y$	Сумма	$3+5=8$
$x - y$	Разность	$5-2=3$
$x * y$	Произведение	$4*5=20$
$x / y$	Деление	$8/3=2.666(6)$
$x // y$	Деление с округление вниз	$8//3=2$
$x \% y$	Остаток от деления	$8\%3=2$ $8\%2=0$
$x^{**}y$	Возведение в степень	$2^{**}3=8$
$abs(x)$	Абсолютное значение	$abs(-3)=3$ $abs(4)=4$
$divmod(x,y)$	Пара $(x//y, x\%y)$	$divmod(8,3)=(2,2)$

# Операции над строками

Обозначение	Операция	Пример
$x + y$	Объединение	"text1"+"text2"="text1text2"
$x * y$	Повторение y раз	"For"*3="ForForFor"

```
str_1 = "It's a "  
str_2 = "string."  
str_1 = str_1 + str_2  
print(str_1)
```



"It's a string."

```
str_1 = "Wow!"  
str_1 = str_1 * 3  
print(str_1)
```



"Wow!Wow!Wow!"

# Сокращенная форма записи операций

Оригинал	Сокращение
$x = x + y$	$x += y$
$x = x - y$	$x -= y$
$x = x * y$	$x *= y$
$x = x / y$	$x /= y$
$x = x // y$	$x //= y$
$x = x \% y$	$x \% = y$
$x = x ** y$	$x ** = y$



# Преобразование типов переменных

Функция	Описание	Пример
<b>int(x)</b>	Конвертирует переменную x в целое число	<code>int("3")=3</code> <code>int(3.7)=3</code>
<b>float(x)</b>	Конвертирует переменную x в действительное число	<code>float("3.5")=3.5</code> <code>float(3)=3.0</code>
<b>str(x)</b>	Конвертирует переменную x в строку	<code>str(3.5)="3.5"</code> <code>str(3)="3"</code>
<b>len(string)</b>	Количество символов в строке string	<code>len("hello")=5</code>

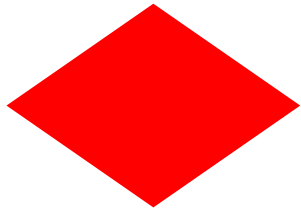
# Алгоритмы. Блок-схемы



Блок начало-конец



Блок действие



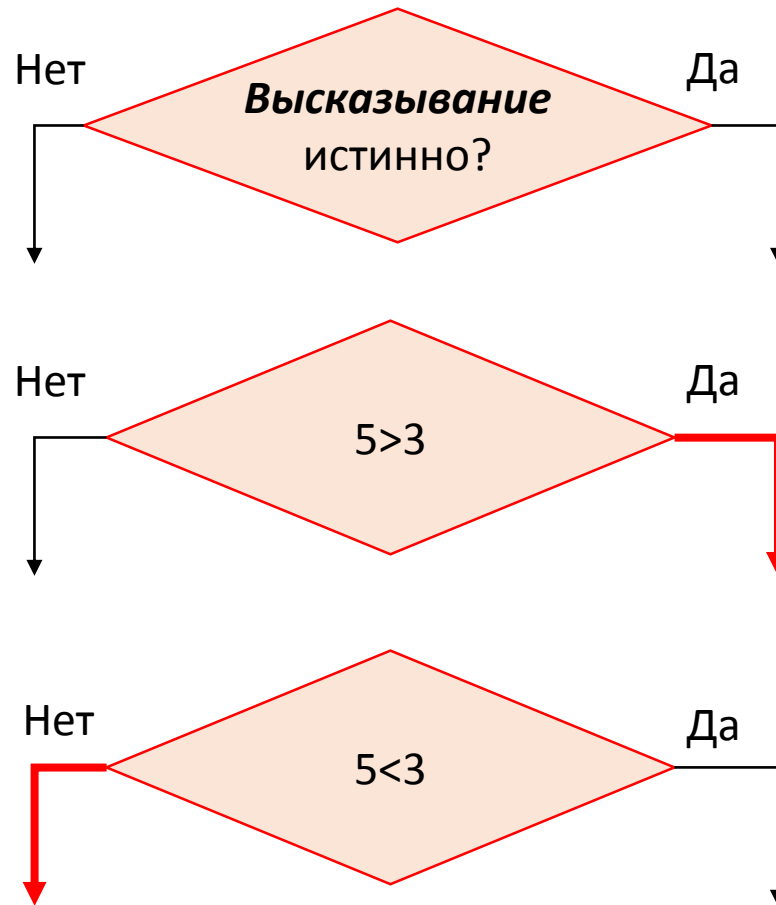
Блок условие



Блок ввода-вывода данных

# Условные выражения

**Высказывание** может быть либо истинным (TRUE), либо ложным (FALSE). За счет этого свойства можно формировать условие для ветвления выполнения операций в программе.



# Условные выражения. Синтаксис

## Тип 1: простое

```
if условие_1:  
    операция_1
```

## Тип 2: с альтернативой

```
if условие_1:  
    операция_1  
else:  
    операция_2
```

## Тип 3: последовательность

```
if условие_1:  
    операция_1  
elif условие_2:  
    операция_2  
  
...  
elif условие_n:  
    операция_n  
else:  
    операция
```

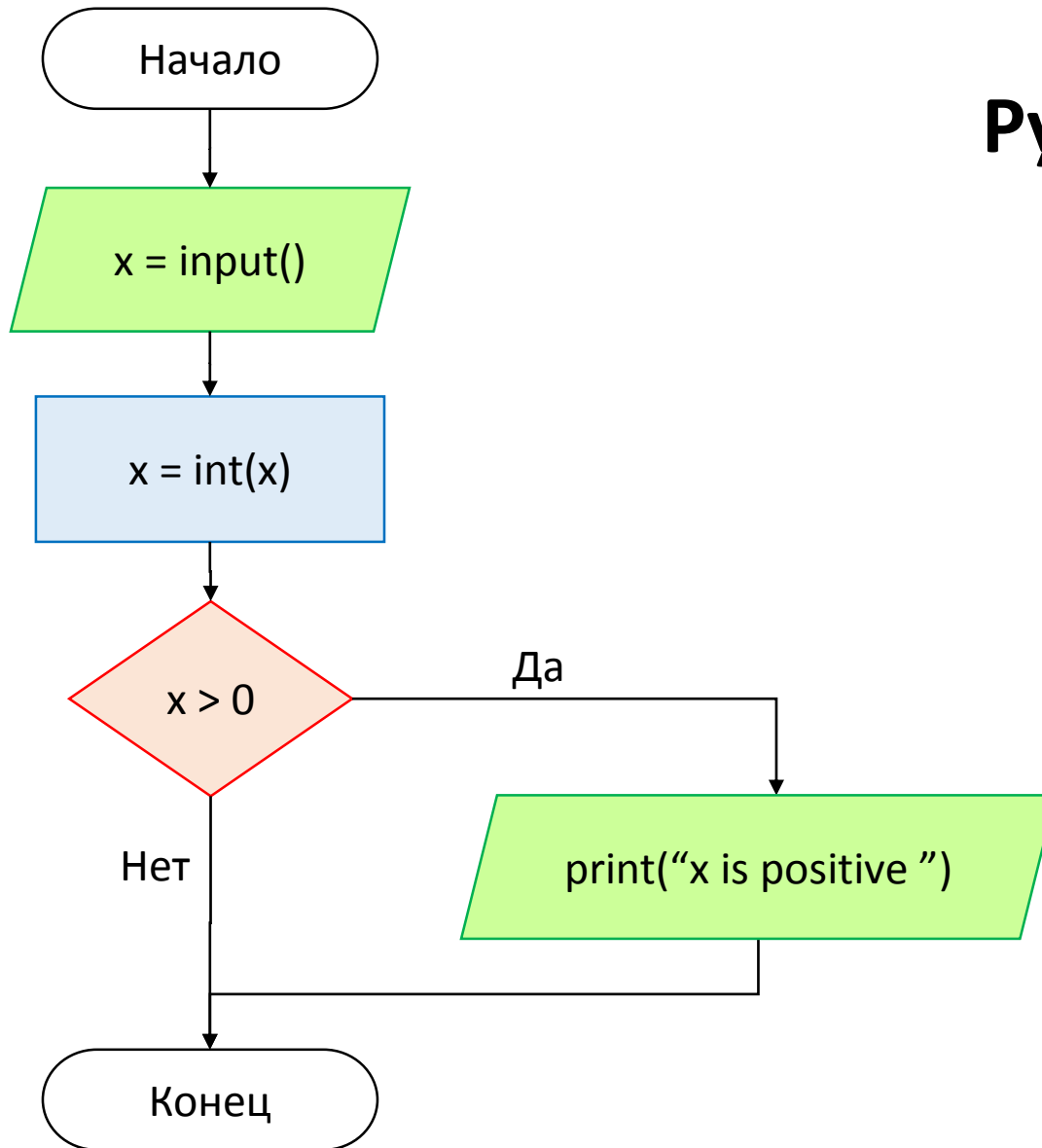
## Тип 4: вложенные условные выражения

```
if условие_1:  
    условное_выражение_1  
else:  
    условное_выражение_2
```

# Операции сравнения

Операция	Описание	Пример
<	Строго меньше чем	1<4 – TRUE 3>5 – FALSE
<=	Меньше или равны	4<=4 – TRUE 4<=2 – FALSE
>	Строго больше	5>5 – FALSE 5>3 – TRUE
>=	Больше или равны	7>=7 – TRUE 4>=8 – FLASE
==	Равны	5==5 – TRUE 5==4 – FALSE
!=	Неравны	5!=5 – FALSE 5!=4 – TRUE
is	Идентичные объекты	
is not	Различные объекты	

# Условные выражения. Тип 1



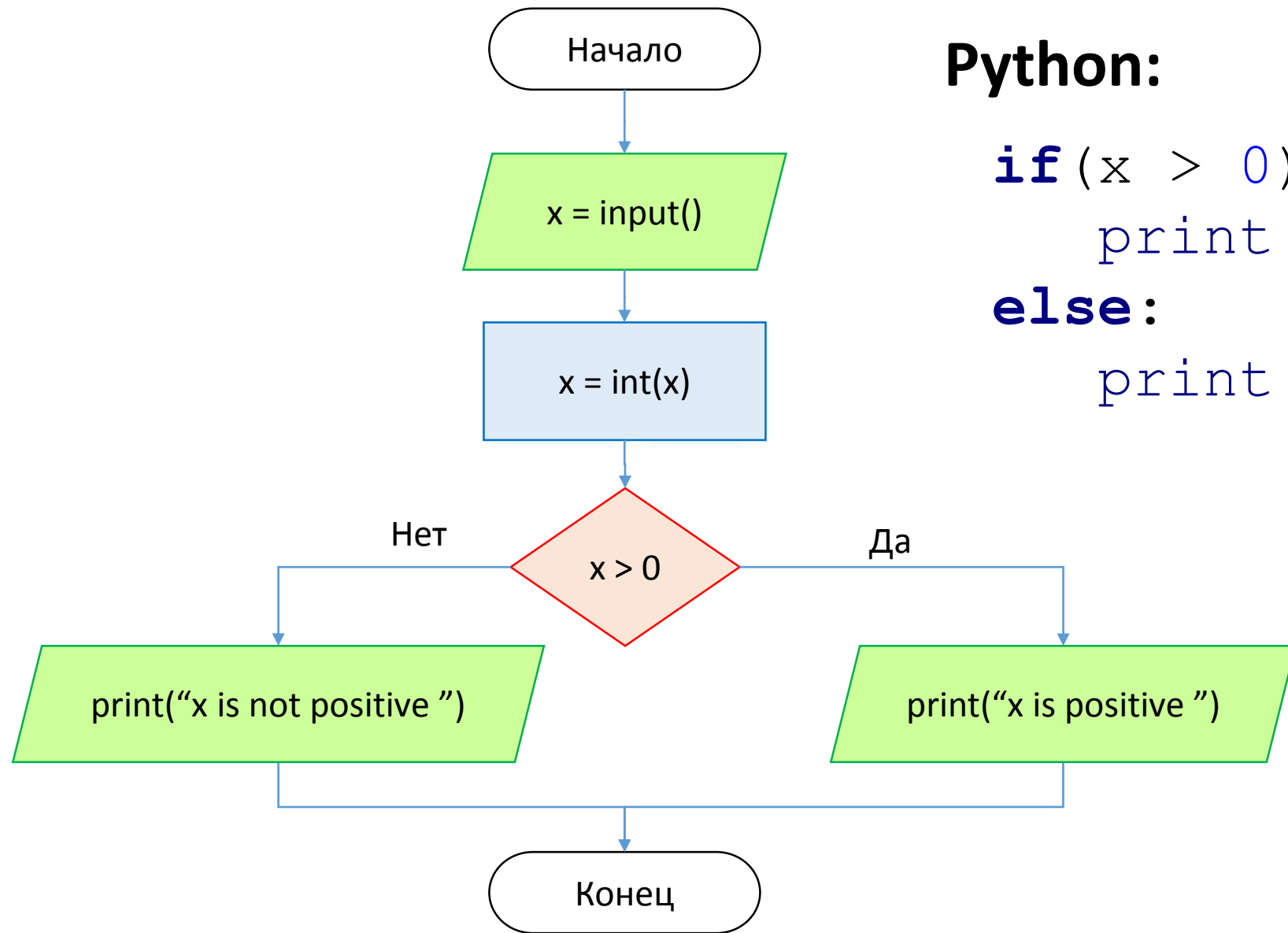
**Python:**

```
x = int(input())  
if (x > 0):  
    print("x is positive")
```

## Условные выражения. Тип 2

**Python:**

```
if (x > 0) :  
    print ("x is positive")  
else:  
    print ("x is not positive")
```



## 1 Стандартный вариант

```
if условие_1:  
    переменная = операция_1  
else:  
    переменная = операция_2
```

```
if (x > 0) :  
    y = "yes"  
else:  
    y = "no"
```

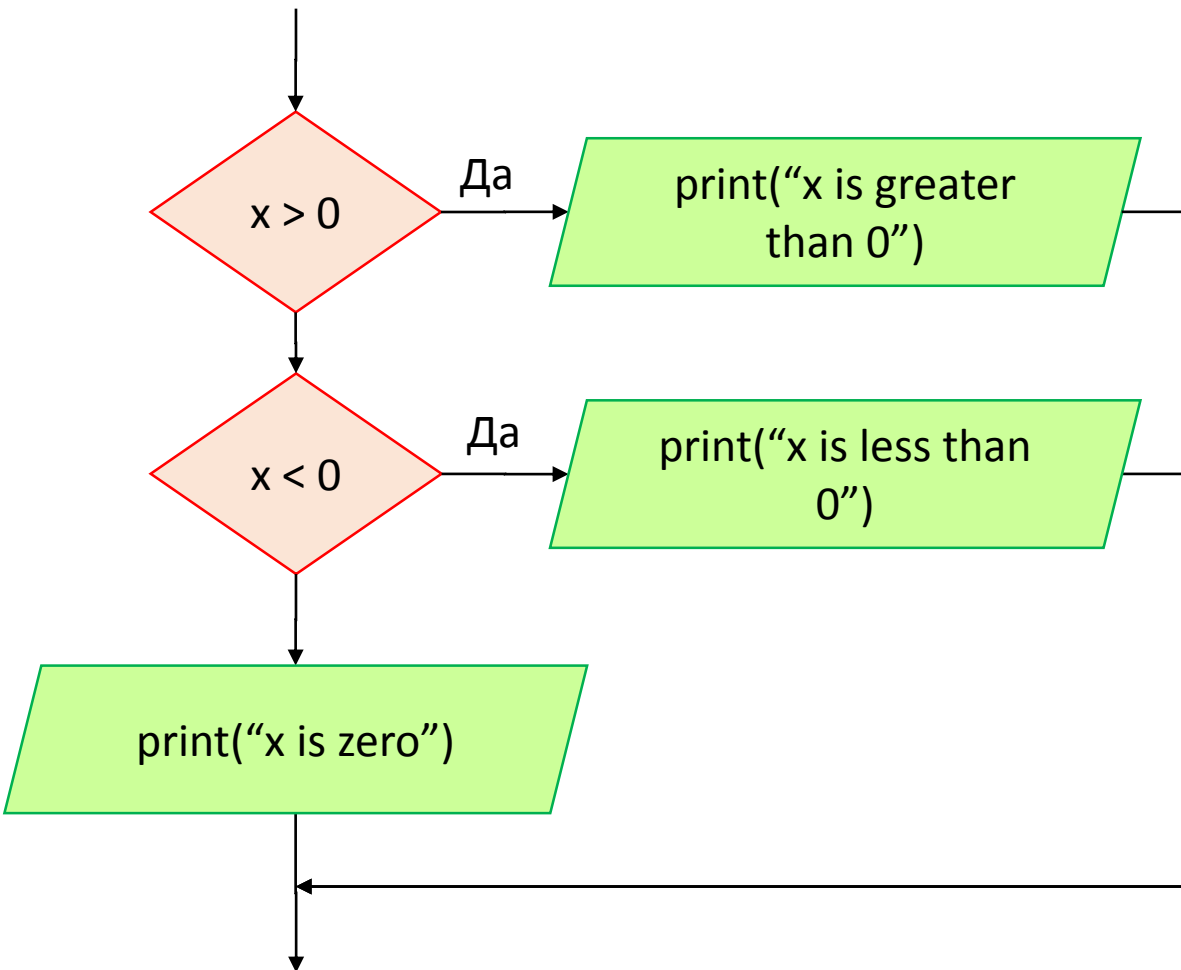
## 2 Сокращенный вариант

```
переменная = операция_1 if условие_1 else операция_2
```

```
y = "yes" if (x > 0) else "no"
```



# Условные выражения. Тип 3



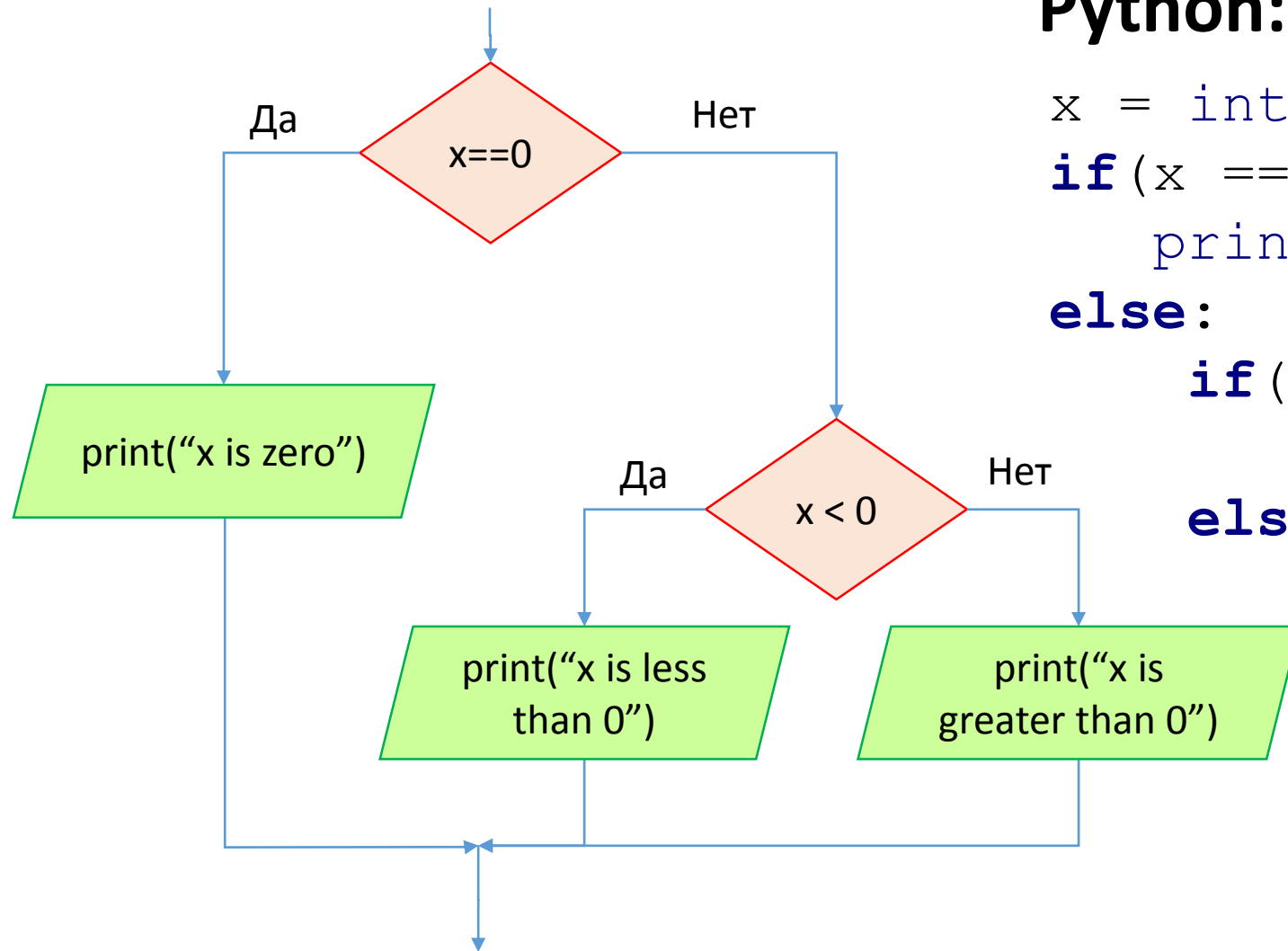
## Python:

```
x = int(input())
if (x > 0):
    print("x is greater than 0")
elif (x < 0):
    print("x is less than 0")
else:
    print("x is zero")
```

# Условные выражения. Тип 4

## Python:

```
x = int(input())  
if (x == 0):  
    print("x is zero")  
else:  
    if (x < 0):  
        print("x is less than 0")  
    else:  
        print("x is greater than 0")
```



# Логические операции

Операция	Описание	Пример
<b>e1 or e2</b>	Истинным должно быть одно из выражений (e1 или e2)	<code>if(x&gt;y or x==y):</code>
<b>e1 and e2</b>	Истинными должны быть оба выражения (e1 и e2)	<code>if(x&gt;y and x==y):</code>
<b>not e1</b>	Выражение e1 должно быть ложным	<code>if(not e1):</code>

1. [Блок-схема](#)
2. [Formatted Output](#)
3. [Charles R Severance. Python for Informatics: Exploring Information](#)