

Практическая работа 3 Структура ветвление в Python

Цель работы: познакомиться со структурой ветвление (if, if-else, if-elif-else).
Научиться работать с числами и строками используя данную структуру.

Условный оператор ветвления if, if-else, if-elif-else

Оператор ветвления if позволяет выполнить определенный набор инструкций в зависимости от некоторого условия. Возможны следующие варианты использования.

1. Конструкция if

Синтаксис оператора if выглядит так:

if логическое выражение:

```
команда_1  
команда_2  
...  
команда_n
```

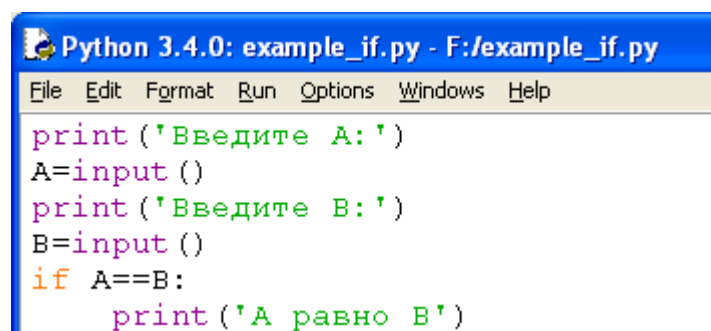
После оператора if записывается логическое выражение.

Логическое выражение — конструкция [языка программирования](#), результатом вычисления которой является «истина» или «ложь».

Если это выражение истинно, то выполняются инструкции, определяемые данным оператором. Выражение является истинным, если его результатом является число не равное нулю, непустой объект, либо логическое True. После выражения нужно поставить двоеточие ":".

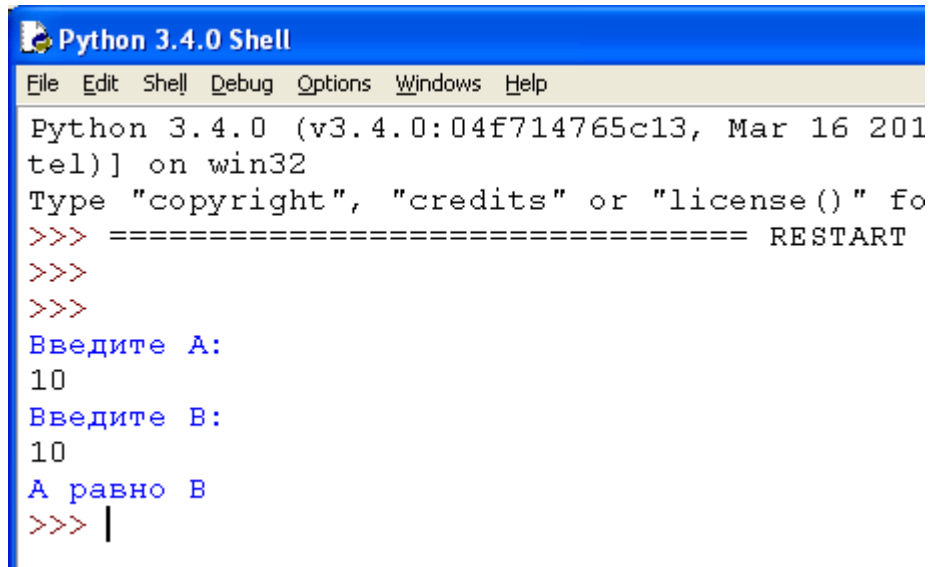
ВАЖНО: блок кода, который необходимо выполнить, в случае истинности выражения, отделяется **четырьмя** пробелами слева!

Программа запрашивает у пользователя два числа, затем сравнивает их и если числа равны, то есть логическое выражение A==B истинно, то выводится соответствующее сообщение.



```
Python 3.4.0: example_if.py - F:/example_if.py  
File Edit Format Run Options Windows Help  
print ('Введите A: ')  
A=input ()  
print ('Введите B: ')  
B=input ()  
if A==B:  
    print ('A равно B')
```

Пример программы на Python



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 201
tel)] on win32
Type "copyright", "credits" or "license()" fo
>>> ===== RESTART
>>>
>>>
Введите A:
10
Введите B:
10
A равно B
>>> |
```

Результат выполнения программы с использованием условного оператора if

2. Конструкция if – else

Бывают случаи, когда необходимо предусмотреть альтернативный вариант выполнения программы. Т.е. при истинном условии нужно выполнить один набор инструкций, при ложном – другой. Для этого используется конструкция if – else.

Синтаксис оператора if – else выглядит так:

if логическое выражение:

команда_1

команда_2

...

команда_n

else:

команда_1

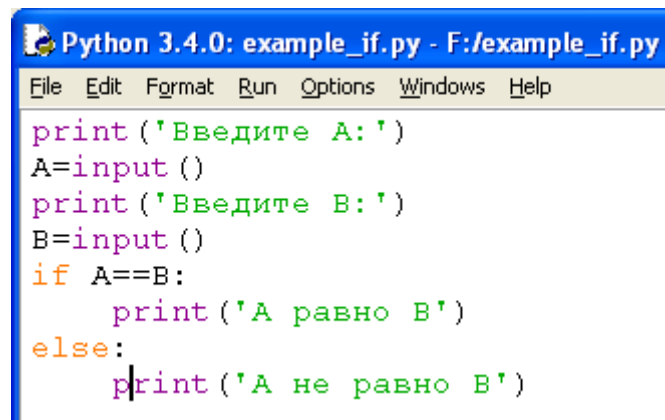
команда_2

...

команда_n

Программа запрашивает у пользователя два числа, затем сравнивает их и если числа равны, то есть логическое

выражение $A==B$ истинно, то выводится соответствующее сообщение. В противном случае выводится сообщение, что числа не равны.

A screenshot of a Python 3.4.0 IDE window titled "Python 3.4.0: example_if.py - F:/example_if.py". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The code in the editor is as follows:

```
print ('Введите A:')
A=input ()
print ('Введите B:')
B=input ()
if A==B:
    print ('A равно B')
else:
    print ('A не равно B')
```

Пример программы на Python

```
>>> ===== RESTART
>>>
Введите A:
10
Введите B:
5
A не равно B
>>> |
```

Результат выполнения программы с использованием
условного оператора if-else

3. Конструкция if – elif – else

Для реализации выбора из нескольких альтернатив можно использовать конструкцию if – elif – else. Синтаксис оператора if – elif – else выглядит так:

if логическое выражение_1:

команда_1

команда_2

...

команда_n

elif логическое выражение_2:

команда_1

команда_2

...

команда_n

elif логическое выражение_3:

команда_1

команда_2

...

команда_n

else:

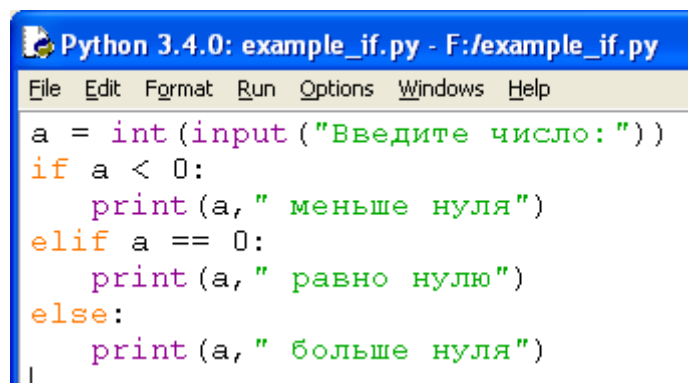
команда_1

команда_2

...

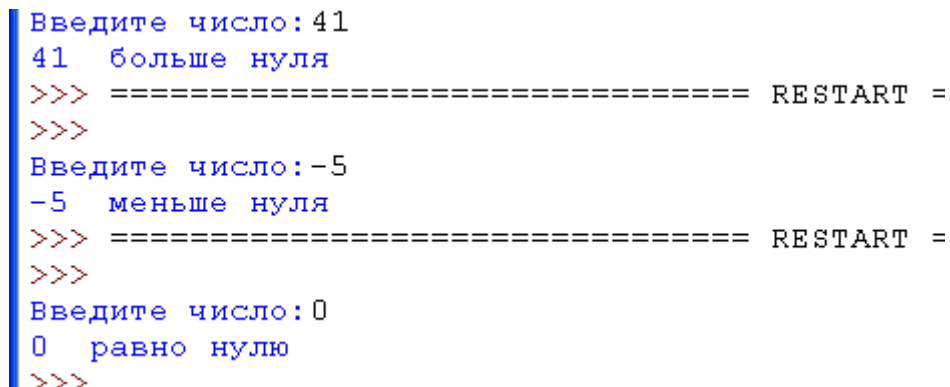
команда_n

Программа запрашивает число у пользователя и сравнивает его с нулём $a < 0$. Если оно меньше нуля, то выводится сообщение об этом. Если первое логическое выражение не истинно, то программа переходит ко второму - $a == 0$. Если оно истинно, то программа выведет сообщение, что число равно нулю, в противном случае, если оба вышеуказанных логических выражения оказались ложными, то программа выведет сообщение, что введённое число больше нуля.



```
Python 3.4.0: example_if.py - F:/example_if.py
File Edit Format Run Options Windows Help
a = int(input("Введите число:"))
if a < 0:
    print(a, " меньше нуля")
elif a == 0:
    print(a, " равно нулю")
else:
    print(a, " больше нуля")
|
```

Пример программы на Python



```
Введите число: 41
41 больше нуля
>>> ===== RESTART =
>>>
Введите число: -5
-5 меньше нуля
>>> ===== RESTART =
>>>
Введите число: 0
0 равно нулю
>>>
```

Результат выполнения программы с использованием
условного оператора if-elif-else

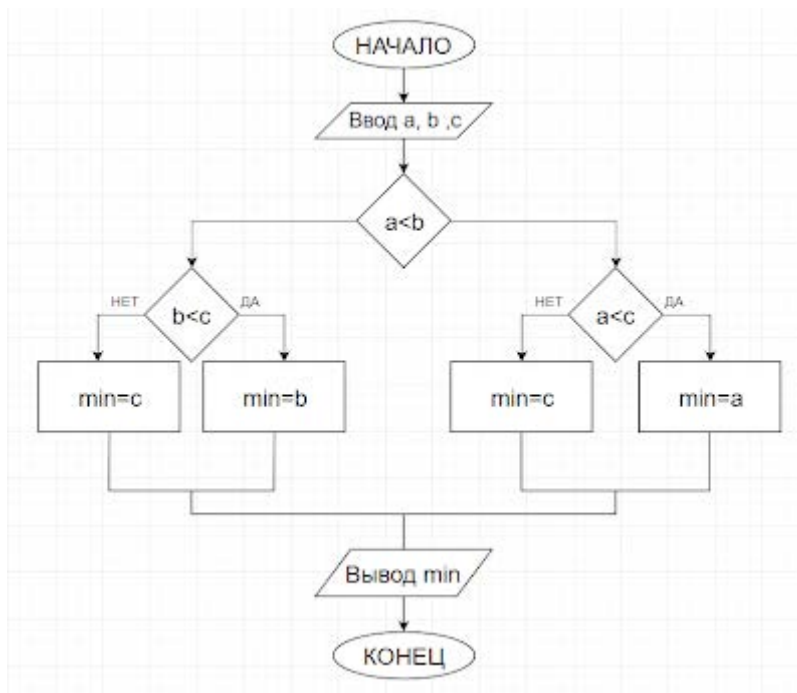
Пример

Задание

Дано 3 числа. Найти минимальное среди них и вывести на экран.

Решение

Для простоты построим блок-схему задачи.



Командами

```
a=input("")
```

```
b=input("")
```

```
c=input("")
```

введём три числа, присвоив значения переменным a, b, c.

Условной конструкцией if-else проверим на истинность логическое выражение $a < b$. Если оно истинно, то переходим на проверку логического выражения $a < c$. Если оно истинно, то переменной "y" присвоим значение переменной "a", т.е. "a" будет минимальным, а иначе "y" присвоится значение переменной "c".

Если в начале логическое выражение $a < b$ оказалось ложным, то переходим на проверку другого логического выражения $b < c$.

Если оно истинно, то "y" присвоится значение переменной "b", иначе "c".

Командой `print()` выводим минимальное значение.

```

#нахождение минимального из 3-х чисел
a=input('Введите целое число \n')
b=input('Введите целое число \n')
c=input('Введите целое число \n')
if a<b:
    if a<c:
        y=a
    else:
        y=c
else:
    if b<c:
        y=b
    else:
        y=c
print('Минимальное:', y)

```

Пример программы

```

Введите целое число
2
Введите целое число
5
Введите целое число
1
Минимальное: 1

```

Результат выполнения программы

Задания для самостоятельной работы

Задание

Даны три целых числа. Выбрать из них те, которые принадлежат интервалу [1,3].