



# Python

Условный оператор IF

# Условный оператор ветвления *if*

- Оператор ветвления *if* позволяет выполнить определенный набор инструкций в зависимости от некоторого условия. Возможны следующие варианты использования.
- **Синтаксис оператора *if* выглядит так:**

```
if выражение:  
    инструкция_1  
    инструкция_2  
    ...  
    инструкция_n
```

После оператора *if* записывается выражение. 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! Ну или клавиша «Tab» решит этот вопрос однократным нажатием.

```
a = 3
if a == 3:
    print("hello 2")
```

Напечатает: *hello 2*

```
a = 3
if a > 1:
    print("hello 3")
```

# 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*.

```
if выражение:  
    инструкция_1  
    инструкция_2  
    ...  
    инструкция_n  
else:  
    инструкция_a  
    инструкция_b  
    ...  
    инструкция_x
```

```
a = 3  
if a > 2:  
    print("H")  
else:  
    print("L")
```

Напечатает: *H*

# Конструкция *if – elif – else*

- Для реализации выбора из нескольких альтернатив можно использовать конструкцию *if – elif – else*

```
if выражение_1:  
    инструкции_(блок_1)  
elif выражение_2:  
    инструкции_(блок_2)  
elif выражение_3:  
    инструкции_(блок_3)  
else:  
    инструкции_(блок_4)
```

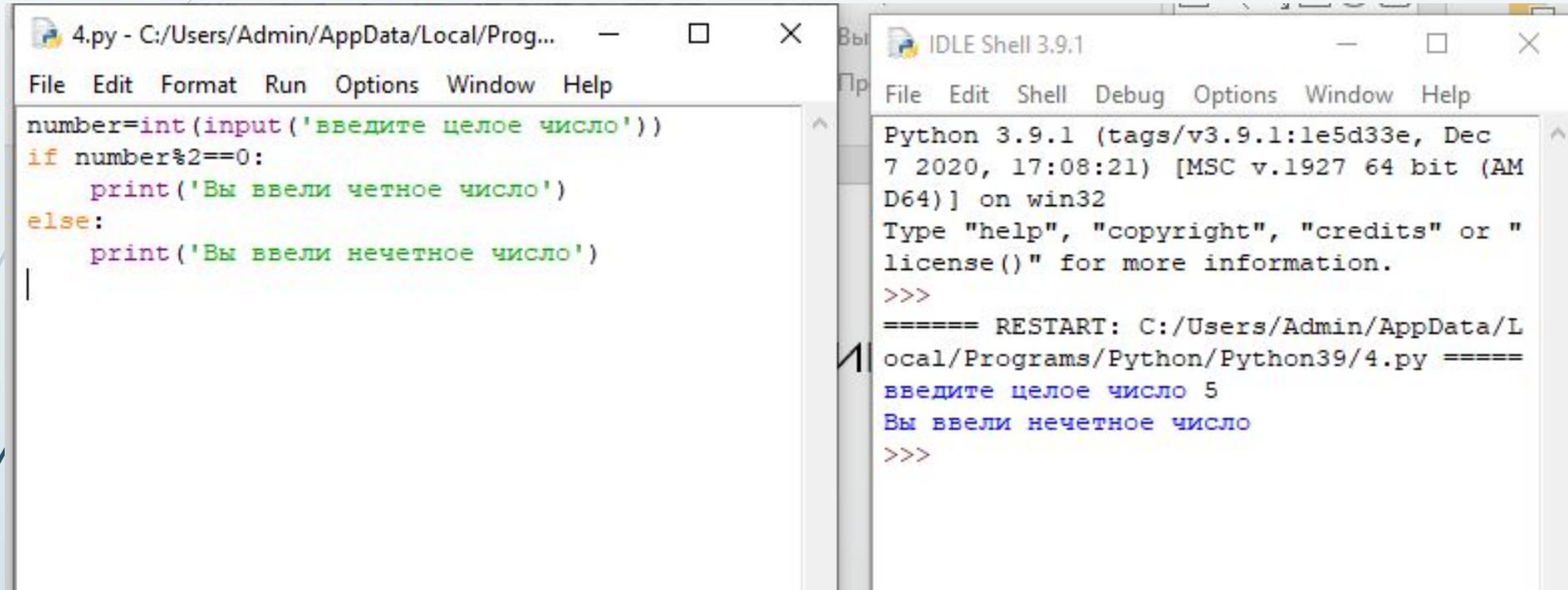
```
a = int(input("введите число:"))  
if a < 0:  
    print("Neg")  
elif a == 0:  
    print("Zero")  
else:  
    print("Pos")
```

# Рассмотрим для примера программу по нахождению максимума из трех чисел:

```
3.py - C:/Users/Admin/AppData/Local/Programs/Python/Pyt
File Edit Format Run Options Window Help
a=int(input('введите первое число'))
b=int(input('введите второе число'))
c=int(input('введите третье число'))
if a>b and a>c:
    print('максимум =',a)
elif b>a and b>c:
    print('максимум =',b)
else:
    print('максимум =',c)
|
```

```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec
D64) ] on win32
Type "help", "copyright", "credits" or
>>>
===== RESTART: C:/Users/Admin/AppData
введите первое число 23
введите второе число 11
введите третье число 2
максимум = 23
>>> |
```

И еще один пример:



The image shows two overlapping windows from a Python IDE. The left window displays a Python script named '4.py' with the following code:

```
number=int(input('введите целое число'))
if number%2==0:
    print('Вы ввели четное число')
else:
    print('Вы ввели нечетное число')
```

The right window shows the IDLE Shell 3.9.1 output, including the Python version information and the execution of the script with the input '5':

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python39/4.py =====
введите целое число 5
Вы ввели нечетное число
>>>
```

# А теперь самостоятельно:

- 1) Напишите программу по нахождению минимального числа
- 2) Напишите программу по нахождению нечетного числа
- 3) Напишите программу по нахождению минимального нечетного числа

