



python

Лекция 3

Условные операторы



Условный оператор

Условный оператор - оператор, конструкция языка программирования, обеспечивающая выполнение определенной команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд (наборов команд) в зависимости от значения некоторого выражения.



Условный оператор в Python

В Python условный оператор задается с помощью оператора **if**

if Логическое условие:

Набор операторов, которые выполнятся

Логическое условие — это сравнение между собой двух операндов (чаще всего переменных и литералов). Логическое условие возвращает один из двух возможных результатов:

True — условие правдиво с точки зрения математики.

False — с точки зрения математики условие ложно.

Если логическое условие вернет True, то блок операторов, следующий за двоеточием, выполнится.

Операторы сравнения

Логическое условие — один или более операторов сравнения (для переменных числового типа или символьного) или использования переменной типа `boolean`.

Запись оператора	Описание
<code>==</code>	Сравнение на равенство
<code>!=</code>	Проверка на неравенство
<code><</code>	Меньше
<code>></code>	Больше
<code><=</code>	Меньше или равно
<code>>=</code>	Больше или равно



О важности четырех пробелов в Python

В отличие от других языков разработки, в Python, для описания логического блока используются **отступы в виде четырех пробелов**. Т.е. код, отстоящий от начала строки на одинаковое количество пробелов, принадлежит одному и тому же логическому блоку.



Пример использования условного оператора

```
orange_price = 17.5
```

```
my_money = 20
```

Логическое выражение



```
if my_money > orange_price:
```

```
    print("I bay orange")
```



**Выполнится, если
логическое условие
вернет True**



четыре пробела для указания на принадлежность к блоку if



Оператор if с блоком else

Если возникает необходимость выполнить один блок операторов, когда условие истинно, и другой блок операторов когда оно ложно, то можно оператор if дополнить блоком **else**.

if Логическое условие:

Набор операторов, которые выполнятся, если True

else:

Набор операторов, которые выполнятся, если False

Внимание! Блок else для оператора if может быть только один. Блок else без оператора if не существует.



Пример использования if и else

```
orange_price = 17.5  
my_money = 10
```

```
if my_money > orange_price:  
    print("I'll buy orange")  
else:  
    print("Bad day")
```

Теперь выполнится блок программы после else, так как условие вернет False. И на экран будет выведено «Bad day».




Использование вложенных операторов if

При необходимости можно в блок условного оператора добавить еще один оператор if со своим блоком. Такие блоки называются вложенными if.

```
orange_price = 17.5
my_money = 10
tea_price = 5
if my_money > orange_price:
    print("I bay orange")
else:
    if my_money > tea_price:
        print ("Not orange, just tea")
    else:
        print("Bad day")
```

**Вложенный условный
оператор if со своим
блоком**





Оператор множественного выбора

Иногда нужно выбрать одно условие из предложенного множества условий. В таком случае можно использовать оператор множественного выбора, который в Python реализован с помощью операторов `if..elif..else`



Синтаксис оператора множественного выбора

`if` Логическое условие:

Набор операторов, которые выполнятся, если `True`

`elif` Логическое условие:

Набор операторов, которые выполнятся, если `True`

`elif` Логическое условие:

Набор операторов, которые выполнятся, если `True`

`elif` Логическое условие:

Набор операторов, которые выполнятся, если `True`

.....

`else:`

Набор операторов которые выполнятся если не подошло ни одно из условий, описанных выше.



Описание работы оператора множественного выбора

- Логические условия оператора множественного выбора начинают проверяться сверху вниз. Как только какое-то из условий даст True, то выполнится блок кода с ним связанный. Остальные участки кода будут пропущены.
- Если при проверке окажется, что ни одно из условий не является истинным, то выполнится последний блок else. Это, так называемый, блок действий по умолчанию.



Пример работы оператора множественного выбора

```
day = int(input("Enter day please \n"))
if day == 1:
    print("Monday")
elif day == 2:
    print("Tuesday")
elif day == 3:
    print("Wednesday")
elif day == 4:
    print("Thursday")
elif day == 5:
    print("Friday")
elif day == 6:
    print("Saturday")
elif day == 7:
    print("Sunday")
else:
    print("Unknown")
```



Можно ли проверить число на принадлежность диапазону заданных чисел

В случае, когда нужно проверить принадлежность переменной числового типа на принадлежность диапазону заданных чисел, можно использовать стандартную математическую нотацию.

```
number = int(input("Enter number "))
```

Проверка принадлежности к диапазону

```
if 5 < number < 10:
    print("inside (5,10)")
else:
    print("outside")
```



Использование булевых операторов в Python

Булевы операторы в Python используются для объединения нескольких логических условий .

В Python существуют следующие булевы операторы:

and — Условный оператор И. Вернет True только в случае, если оба объединяемых условия вернут True. В остальных случаях вернет False.

or — Условный оператор ИЛИ. Вернет False только в случае, если оба объединяемых условия вернут False. В остальных случаях вернет True.

not — Условный оператор НЕ. Инвертирует результат условного оператора.



Использование оператора **and**

Булев оператор **and** используется для объединения логических условий. Результат объединения вернет True, если оба логических условия вернут True. В остальных случаях результатом будет False.

if Логическое_условие_1 **and** Логическое_условие_2:
операторы

Логическое_условие_1	True	False	False	True
Логическое_условие_2	False	True	False	True
Результат	False	False	False	True



Пример использования оператора **and**

```
number_a = 17
number_b = 21

if number_a > 15 and number_b
> 20:
    print("Yes")
else:
    print("No")
```

В результате работы программы на экран будет выведено слово «Yes», так как оба эти условия вернут True.



Использование оператора **or**

Булев оператор **or** используется для объединения логических условий. Результат объединения вернет False, если оба логических условия вернут False. В остальных случаях результатом будет True.

if Логическое_условие_1 **or** Логическое_условие_2:
операторы

Логическое_условие_1	True	False	False	True
Логическое_условие_2	False	True	False	True
Результат	True	True	False	True



Пример использования оператора **or**

```
number_a = 17
number_b = 21

if number_a > 20 or number_b > 20:
    print("Yes")
else:
    print("No")
```

В результате работы программы на экран будет выведено слово «Yes». И хотя первое условие вернет False, но второе вернет True и результат объединения вернет True.



Использование оператора **not**

Булев оператор **not** используется для инвертирования логических условий.

if **not** Логическое_условие_1:
операторы

Логическое_условие_1	True	False
Результат	False	True



Пример использования оператора **not**

```
number_a = 17

if not number_a > 20:
    print("Yes")
else:
    print("No")
```

В результате работы программы на экран будет выведено слово «Yes». И хотя первое условие вернет False, но оператор `not` его инвертирует и результатом будет True.



Объединение более двух логических условий

Могу ли я объединить более двух логических условий?

Да. Используя булевы операторы вы, можете объединить произвольное количество логических условий. Также приоритет логических условий можно менять с помощью круглых скобок.

```
number_a = 3  
number_b = 7
```

Изменение приоритета

```
if (number_a < 5 and number_b > 0) or number_b < 5:
```

```
    print("Yes")  
else:  
    print("No")
```

И еще один булев оператор



Список использованной литературы

- 1) Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011 - 376 - 390 с.
- 2) Саммерфилд М. Программирование на Python 3. Подробное руководство. - Пер. с англ. - СПб.:Символ-Плюс, 2009. - 189 - 190 с.

Домашнее задание

- 1) Написать программу, которая считает 4 числа с клавиатуры и выведет на экран самое большое из них.
- 2) Есть девятиэтажный дом, в котором 4 подъезда. Номер подъезда начинается с единицы. На одном этаже 4 квартиры. Напишите программу которая, получит номер квартиры с клавиатуры, и выведет на экран, на каком этаже, какого подъезда расположена эта квартира. Если такой квартиры нет в этом доме, то нужно сообщить об этом пользователю.
- 3) Определить количество дней в году, который вводит пользователь. В високосном году - 366 дней, тогда как в обычном их 365. Високосный год определяется по следующему правилу:
Год високосный, если он делится на четыре без остатка, но если он делится на 100 без остатка, это не високосный год. Однако, если он делится без остатка на 400, это високосный год.
- 4) Треугольник существует только тогда, когда сумма любых двух его сторон больше третьей. Дано: a , b , c – стороны предполагаемого треугольника. Напишите программу, которая укажет, существует ли такой треугольник или нет.



Дополнительное домашнее задание.

- 1) Есть круг с центром в начале координат и радиусом 4. Пользователь вводит с клавиатуры координаты точки x и y . Написать программу, которая определит, лежит ли эта точка внутри круга или нет.
- 2) Дан треугольник координаты вершин которого $A(0,0)$, $B(4,4)$, $C(6,1)$. Пользователь вводит с клавиатуры координаты точки x и y . Написать программу, которая определит, лежит ли эта точка внутри треугольника или нет.
- 3) Дано четырехзначное число. Проверить, является ли оно «счастливым билетом». Примечание: счастливым билетом называется число, в котором, при четном количестве цифр в числе, сумма цифр его левой половины равна сумме цифр его правой половины. Например, рассмотрим число 1322. Его левая половина равна 13, а правая 22, и оно является счастливым билетом (т. к. $1 + 3 = 2 + 2$).
- 4) С клавиатуры вводится шестизначное число. Проверить, является ли оно палиндромом. Примечание: палиндромом называется число, слово или текст, которые одинакового читаются слева направо и справа налево. Например, это числа 143341, 5555, 7117 и т. д.