

Вложенные условия и множественный выбор. Конструкция if-elif-else

Мы продолжаем изучение условных операторов языка Python. На предыдущем занятии мы познакомились с работой оператора if проверки условий. И говорили, что внутри блока этого оператора могут быть любые конструкции языка Python, в том числе и другие условия. Давайте посмотрим на примерах, как это работает.



Предположим, что сначала в первом условии нам нужно проверить число на четность, а затем, определить, является ли это число цифрой или каким-либо другим числом:

```
x = int(input())
if x % 2 == 0:
    if 0 <= x <= 9:
        print("x - четная цифра")
    else:
        print("x - четное число")
```

Как видите, это удобно сделать с помощью вложенного условия. И, обратите внимание, оператор else здесь относится именно ко второму условию, так как записан на одном уровне с ним. А вот, если бы мы его прописали без отступов:

```
if x % 2 == 0:
    if 0 <= x <= 9:
        print("x - четная цифра")
else:
    print("x - четное число")
```

то он стал бы относиться к первому условию. Программа стала бы работать иначе и, вообще говоря, некорректно для данной задачи.

Я верну отступы для блока else и добавлю еще один такой блок для первого условия:

```

x = int(input())
if x % 2 == 0:
    if 0 <= x <= 9:
        print("x - четная цифра")
    else:
        print("x - четное число")
else:
    print("x - нечетное число")

```

В результате, у нас обрабатываются все возможные исходы для числа x.

Другой пример – поиск наибольшего среди трех чисел a, b и c:

```

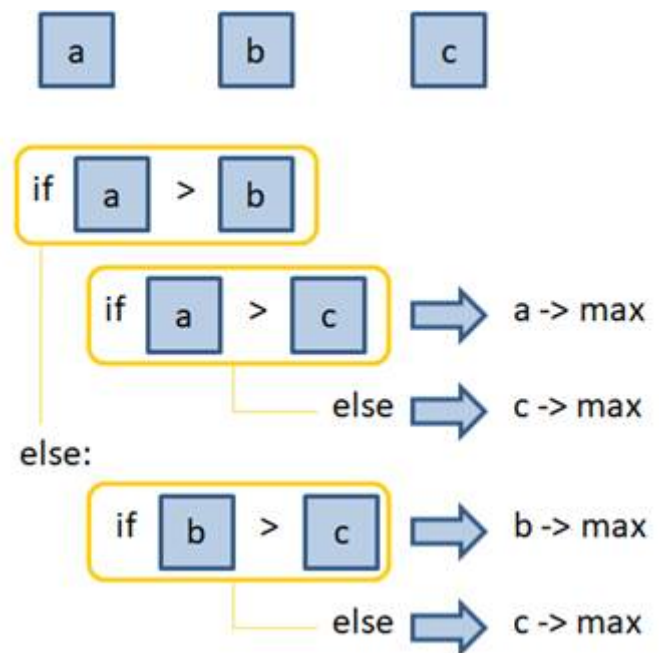
a = int(input("a: "))
b = int(input("b: "))
c = int(input("c: "))

```

```

if a > b:
    if a > c:
        print("a -> max")
    else:
        print("c -> max")
else:
    if b > c:
        print("b -> max")
    else:
        print("c -> max")

```



Думаю, из этих примеров вам стало понятно, как использовать вложенные условия. Конечно, степень вложенности может быть любой, но все будет работать по аналогии. В практике реального программирования лучше избегать больших вложенностей. Нормальным считается до трех вложений. Если у вас выходит больше, то, скорее всего, что-то делаете не так и структуру программы следует пересмотреть.

Во второй части этого занятия я хочу рассказать вам о способе реализации множественного выбора. Что это такое? Представьте, что мы просим пользователя выбрать один из пунктов меню:

1. Курс по Python
2. Курс по C++
3. Курс по Java
4. Курс по JavaScript

Используя текущие знания, мы можем реализовать это так:

```
item = int(input())

if item == 1:
    print("Выбран курс по Python")
else:
    if item == 2:
        print("Выбран курс по C++")
    else:
        if item == 3:
            print("Выбран курс по Java")
        else:
            if item == 4:
                print("Выбран курс по JavaScript")
            else:
                print("Неверный пункт")
```

Но эту же логику можно реализовать проще и нагляднее, используя еще один условный оператор

elif = else if

То есть, все конструкции с else if мы можем просто заменить на elif, получим:

```
if item == 1:
    print("Выбран курс по Python")
elif item == 2:
    print("Выбран курс по C++")
elif item == 3:
    print("Выбран курс по Java")
elif item == 4:
    print("Выбран курс по JavaScript")
else:
    print("Неверный пункт")
```

Видите, как это удобнее, когда нам в программе нужно реализовать выбор какого-то одного значения из множества доступных. Причем, последовательность операторов должна быть именно такой: if – elif – else

Конечно, помимо проверок на равенство после elif можно прописывать любые условия. Например, вводим с клавиатуры целое положительное x и для него определяем число десятков:

```
x = int(input())

if x < 0:
    print("x должно быть положительным")
elif 0 <= x <= 9:
    print("x - цифра")
elif 10 <= x <= 99:
    print("x - двузначное число")
elif 100 <= x <= 999:
    print("x - трехзначное число")
```