

Методические указания

Урок 3.2 Условный оператор. if..elif..else. Тернарный оператор

Задачи урока:

- Познакомить с условным оператором if:elif:else, вложенными условиями
- Закрепить материал на практике

0. Подготовка к уроку

До урока учителю необходимо:

- 1) Просмотреть в классе как ученики справились с домашним заданием
- 2) Прочитать методичку

1. Знакомство с условным оператором

Учитель: На прошлом занятии мы с вами познакомились подробнее с условным оператором if (*если*), разобрали как написать составные условия с помощью операторов and, or и not.

Связка if..else, подходит только когда у нас не более двух значений, но что если у нас больше двух вариантов. Конечно же мы можем написать программу и с помощью if, но стоит запомнить, что это будут отдельные условия и может быть вариант, когда несколько из них окажутся верными.

```
x = 5
if x >= 5:
    print('YES')
if x == 5:
    print('YES')
if x != 10:
    print('YES')
else:
    print('NO')
```

В данном примере у нас все условия верны, а оператор else относится только к последнему условию.

Что же делать, когда у нас более 2 вариантов и как написать условие? Для этого существует оператор elif (*если иначе*). Он позволяет составлять длинные условия с неограниченным

количеством вариантов. Как пример представим, что пользователь вводит свой возраст и программа должна вывести отдельные временные промежутки.

```
age = int(input('Введите свой возраст: '))
if age >= 14:
    print('Вам больше 14 или 14 лет')
elif age >= 18:
    print('Вам больше или 18 лет')
elif age >= 21:
    print('Вам больше или 21 год')
elif age > 45:
    print('Вам больше 45 лет')
```

В данном примере выводится только одно значение и это одно большое условие.

Учитель: Кстати, а что нам делать если нам надо проверить значение по двум или более параметрам. Например старше ли мы 11 и младше ли мы 15 лет?

В данном случае у нас есть несколько вариантов написания подобного кода.

1. Через вложенные условия.
2. Используя операторы `and` или `or`
3. Используя проверку 'по диапазону'

Разберем все варианты.

```
age = 12
if age > 11:
    if age < 15:
        print('Да')
```

```
age = 12
if age > 11 and age < 15:
    print('Да')
```

```
age = 12
if 11 < age < 15:
    print('Да')
```

Первый и второй варианты нам понятны, но как работает третий вариант. Читается он как если age больше 11 и меньше 15. В данном случае переменная всегда будет стоять между проверочными значениями.

Учитель: С написанием условий думаю разобрались. *(учитель спрашивает у учеников есть ли вопросы)*. А как думаете можно ли как то написать условие используя if..else в одну строку?*(интересуемся у учеников)*

Есть еще одна интересная возможность записывать условия в Python, такая как тернарный оператор. Давайте посмотрим как это выглядит на практике.

Представим, что пользователь вводит число и программа выводит четное оно или нет. Вроде бы все легко, но как написать это в одну строчку?

```
num = int(input('Введите число'))
print('Четное' if num % 2 == 0 else 'Нечетное')
```

С помощью тернарного оператора, как мы видим можем записать условие в одну строку и как пример сразу вывести. Разберем синтаксис:

Если условие верно Условие Иначе Если не верно

Как и во многих языках программирования, почти любую задачу мы можем решить множеством способов.

2. Решение задач

Учитель: Пришло время попрактиковаться. Давайте решим несколько задач для закрепления материала.

Задача 1

Дано число. Если оно больше **100** или меньше **-100**, то вывести на экран символ **—**, иначе вывести на экран символ **+**.

Решение

```
num = int(input())
if num > 100 or num < -100:
    print('-')
else:
    print('+')
```

Задача 2.

Пользователь вводит три числа. Если все числа больше **10**, то вывести на экран **yes**, иначе **no**

Решение

```
num1 = int(input())
num2 = int(input())
num3 = int(input())
if num1 > 10 and num2 > 10 and num3 > 10:
    print('yes')
else:
    print('no')
```

Дополнительно

Если на уроке остается время, то ученикам можно предложить начать прорешивать домашнее задание.

Домашняя работа

Задача 1

1. Пользователь вводит порядковый номер месяца с 1 и программа выводит к какому сезону относится данный месяц(осень, лето, зима, весна)