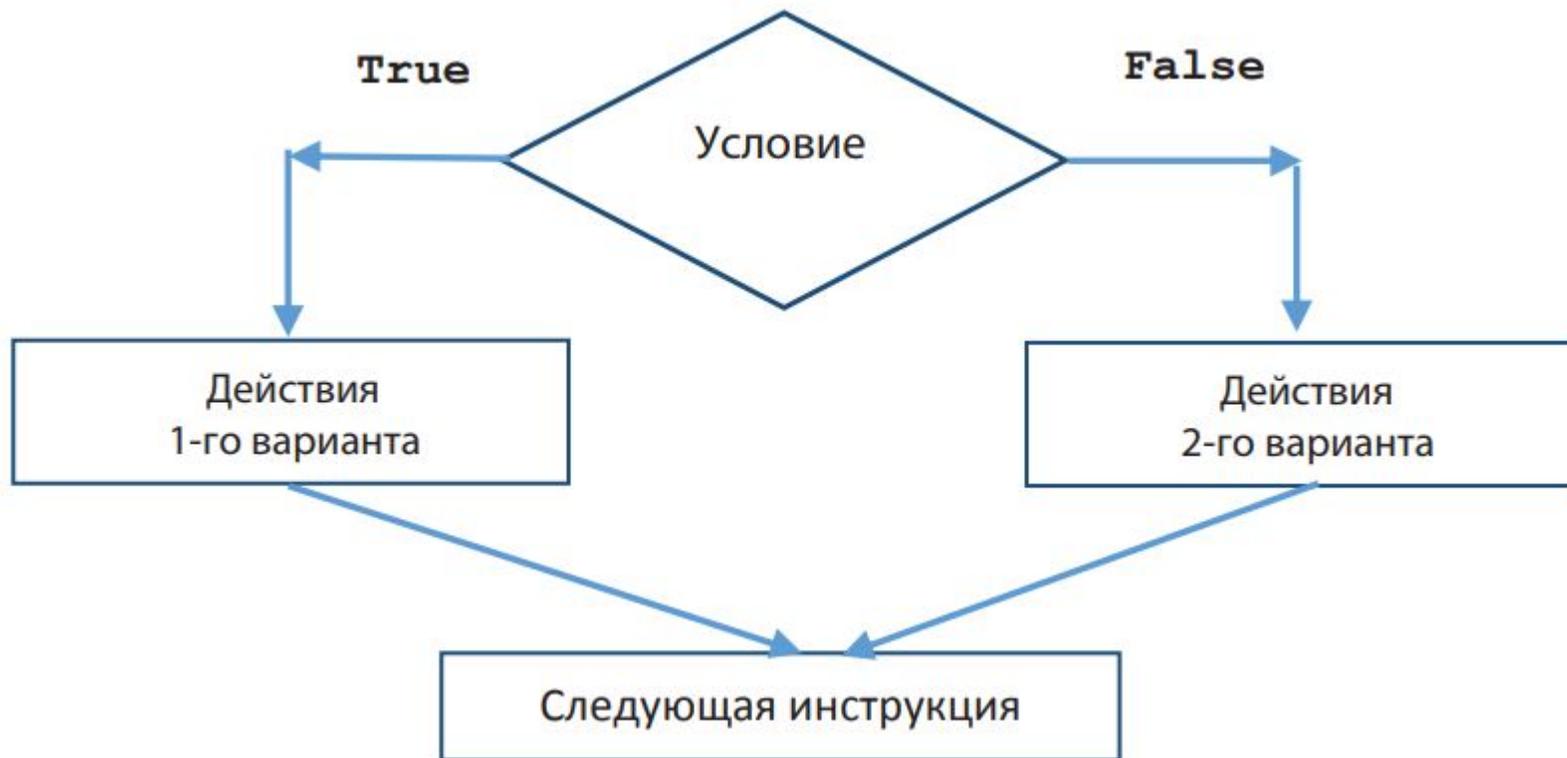


# Условный оператор

---

Задача: **изменить порядок действий** в зависимости от выполнения некоторого условия.



# Структура полного условного оператора

---

if <условие>:

    <Действия 1-го варианта (1-я серия инструкций)>

else:

    <Действия 2-го варианта (2-я серия инструкций)>

В самом простом случае – это два арифметических выражения между которыми записан знак операции сравнения.

В языке Python есть 6 операций сравнения:

	<b>Знак операции</b>	<b>Означает</b>
1	<	Меньше
2	<=	Меньше либо равно
3	>	Больше
4	>=	Больше либо равно
5	==	Равно
6	!=	Не равно

## Сложные условия

Задача: набор сотрудников в возрасте **25-40 лет**  
(включительно).

сложное условие

```
if v >= 25 and v <= 40 :  
    print ("подходит")  
else:  
    print ("не подходит")
```

**and** «И»

**or** «ИЛИ»

**not** «НЕ»

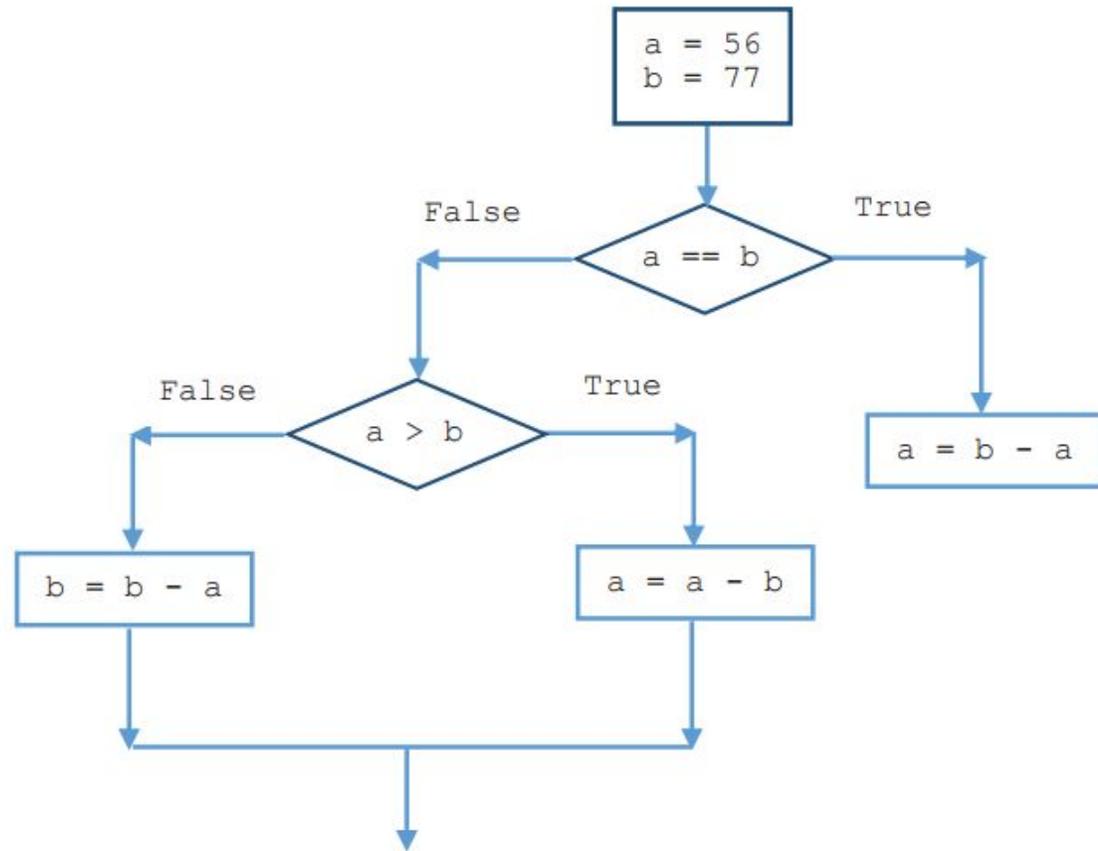
### Приоритет :

- 1) отношения (<, >, <=, >=, ==, !=)
- 2) **not** («НЕ»)
- 3) **and** («И»)
- 4) **or** («ИЛИ»)

**Пример.** Даны два вещественных числа  $a$  и  $b$ . Если первое больше второго, то увеличить каждое число в 2 раза, иначе – уменьшить в два раза.

**Соответствующая программа:**

```
a = float(input('a = '))
b = float(input('b = '))
if a > b:
    a = a * 2
    b = b * 2
else:
    a = a/2
    b = b/2
print('a =', a)
print('b =', b)
```



# Таблица истинности

---

<i>A</i>	<i>B</i>	Not <i>A</i>	<i>A</i> And <i>B</i>	<i>A</i> Or <i>B</i>	<i>A</i> Xor <i>B</i>
T	T	F	T	T	F
T	F	F	F	T	T
F	F	T	F	F	F
F	T	T	F	T	T

## Свойства условий:

### 1. Сокращение длинных условий

- использовать обратный слэш («\»):

if  $v < 400$  and  $v \neq 2$  and  $v \neq 3$  and  $v \neq 12$  and  $v \neq 13$  and  $v \neq 22$  and  $v \neq 23$ : ...

- взять все условие в скобки (перенос внутри скобок разрешен):

if ( $v < 400$  and  $v \neq 2$  and  $v \neq 3$  and  $v \neq 12$  and  $v \neq 13$  and  $v \neq 22$  and  $v \neq 23$ ): ...

### 2. Разрешены двойные неравенства,

if  $A < B < C$ : ... означает то же самое, что и if  $A < B$  and  $B < C$ :

3. **Условие - логические функции**, то есть функции, возвращающие результат логического типа

```
n = int(input('Введите целое число '))
```

```
if Chet(n):
```

```
    print('Это число четное')
```

```
else:
```

```
    print('Это число нечетное')
```

где Chet() – функция, возвращающая результат True, если ее параметр (значение, указанное в скобках) является четным числом, и False – в противном случае;

4. **оператор in** (оператор проверки принадлежности), который проверяет, принадлежит ли некоторый объект (число, символ, переменная и т. п.) набору значений (списку, строке, диапазону чисел и т. п.):

```
a = 3
```

```
if a in range(10):
```

```
    sim = input('Введите символ ')
```

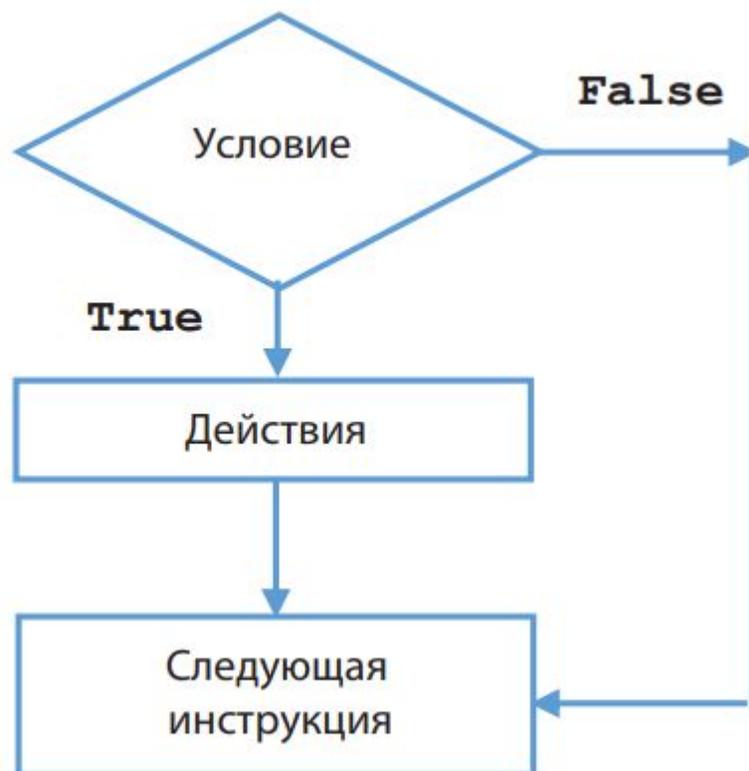
```
    s = input('Введите строку символов ')
```

```
if sim in s:
```

```
    Zvet = 'Зеленый'
```

```
if Zvet in Raduga:
```

## Неполная форма



```
M = a
if b > a:
    M = b
```

Решение в стиле Python:

```
M = max(a, b)
```

```
M = a if a > b else b
```

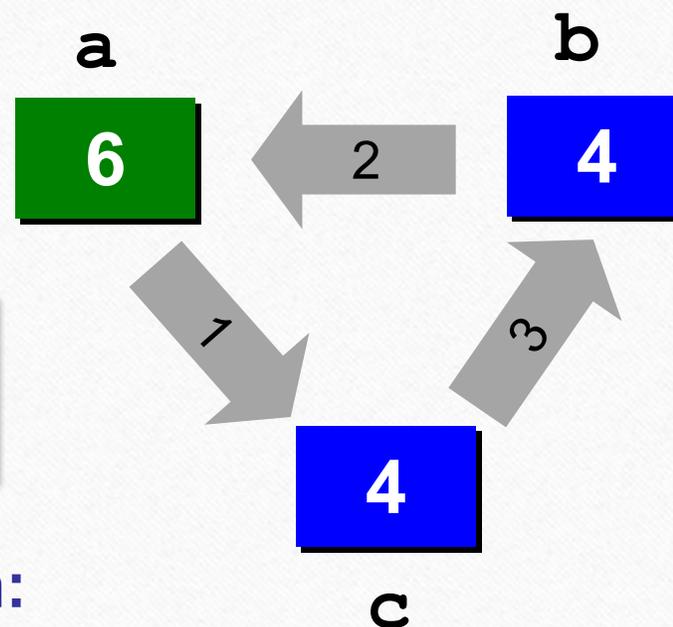
```
if a > b:
```

```
    c = a
```

```
    a = b
```

```
    b = c
```

**?** Что делает?



**?** Можно ли обойтись без переменной **c**?

Решение в стиле Python:

```
a, b = b, a
```

## Вложенные условные операторы

**Задача:** в переменных **a** и **b** записаны возрасты Андрея и Бориса. Кто из них старше?

```
if a > b:  
    print("Андрей старше")  
else:  
    if a == b:  
        print("Одного возраста")  
    else:  
        print("Борис старше")
```

вложенный  
условный оператор

## Каскадное ветвление

```
if a > b:  
    print("Андрей старше")  
elif a == b:  
    print("Одного возраста")  
else:  
    print("Борис старше")
```



**elif = else if**

## Каскадное ветвление

```
cost = 1500
if cost < 1000:
    print ( "Скидок нет." )
elif cost < 2000:
    print ( "Скидка 2%." )
elif cost < 5000:
    print ( "Скидка 5%." )
else:
    print ( "Скидка 10%." )
```

первое сработавшее  
условие



Что выведет?

Скидка 2%.

## Задача 1

Николай написал функцию `is_alive(health)`, которая проверяет здоровье персонажа в игре. Если оно равно или меньше нуля, то функция возвращает **False**, в противном случае **True**. К сожалению, функция не работает, так как ученик допустил в ней ряд ошибок. Исправьте их и проверьте работоспособность программы (в качестве аргумента всегда передается число).

```
def is_alive(health):  
    ___if:  
    ___health < 0  
    _____False  
    ___else:  
    _____return true
```

## Задача 2

Составьте функцию `season_events(number_of_month)`, которая принимает номер месяца вашего рождения и в зависимости от сезона печатает на выходе следующее:  
**Вы родились в <НАЗВАНИЕ\_МЕСЯЦА>.**  
**<ОПИСАНИЕ СОБЫТИЙ>.**

В качестве **ОПИСАНИЯ СОБЫТИЙ** будет характеристика сезона:

- для зимы **За окном падал белый снег,**
- для весны **Птицы пели прекрасные песни,**
- для лета **Солнце светило ярче чем когда-либо,**
- для осени **Урожай был невероятным.**

Важно учесть, что пользователи могут ввести любой тип данных в качестве аргумента (не попадитесь на этом и предупредите о том, что **Требуется ввести реальный номер месяца**).

Николай написал функцию `is_alive(health)`, которая проверяет здоровье персонажа в игре. Если оно равно или меньше нуля, то функция возвращает **False**, в противном случае **True**. К сожалению, функция не работает, так как ученик допустил в ней ряд ошибок. Исправьте их и проверьте работоспособность программы (в качестве аргумента всегда перед

```
def is_alive(health):  
    if:  
        health < 0  
        False  
    else:  
        return true
```

В функции Николая есть ряд ошибок:

- **True** – пишется с большой буквы
- проверка здоровья должна писаться после выражения **if** до двоеточия и включать в себя нулевое значение
- в теле оператора **if** значение **False** должно возвращаться, а не просто объявляться.

```
months = {
    1: 'Январе',
    2: 'Феврале',
    3: 'Марте',
    4: 'Апреле',
    5: 'Мае',
    6: 'Июне',
    7: 'Июле',
    8: 'Августе',
    9: 'Сентябре',
    10: 'Октябре',
    11: 'Ноябре',
    12: 'Декабре'
}

def season_events(number_of_month):
    if not isinstance(number_of_month, int) and 1 <= number_of_month <= 12:
        print('Требуется ввести реальный номер месяца')
        return
    if number_of_month in range(3, 6):
        print(f'Вы родились в {months[number_of_month]}. Птицы пели прекрасные песни')
    elif number_of_month in range(6, 9):
        print(f'Вы родились в {months[number_of_month]}. Солнце светило ярче чем когда-либо')
    elif number_of_month in range(9, 12):
        print(f'Вы родились в {months[number_of_month]}. Урожай был невероятным')
    else:
        print(f'Вы родились в {months[number_of_month]}. За окном падал белый снег')

# Тесты
season_events(33)
season_events('wq')
season_events(1)
season_events(7)
season_events(10)
season_events(5)
```