

4. Операторы сравнения и логические операторы.

Инструкция if...else.

Операторы сравнения, логические операторы, if...else - все это связано с понятием ветвящихся (нелинейных) алгоритмов.

В отличие от линейных, в них возможны разные действия, зависящие от данных на каком-либо этапе выполнения программы.

Оператор условного перехода if используется как раз для перехода по определенной ветки в зависимости от данных.

Полный синтаксис оператора сравнения if:

```
if условие1:  
    действие1..  
elif условие2:  
    действие2..  
else:  
    действие3..
```

Если условие1 истинно, то выполняется действие1

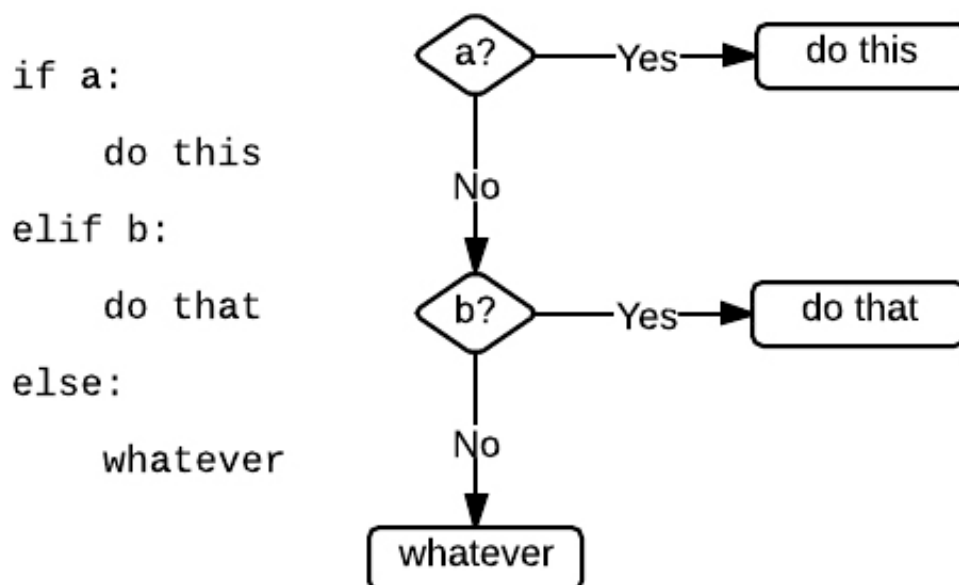
Иначе (если условие1 ложно) происходит проверка на истинность условия2

Если оно истинно, то выполняется действие2

Если и условие2 ложно, то происходит выполнение действия3

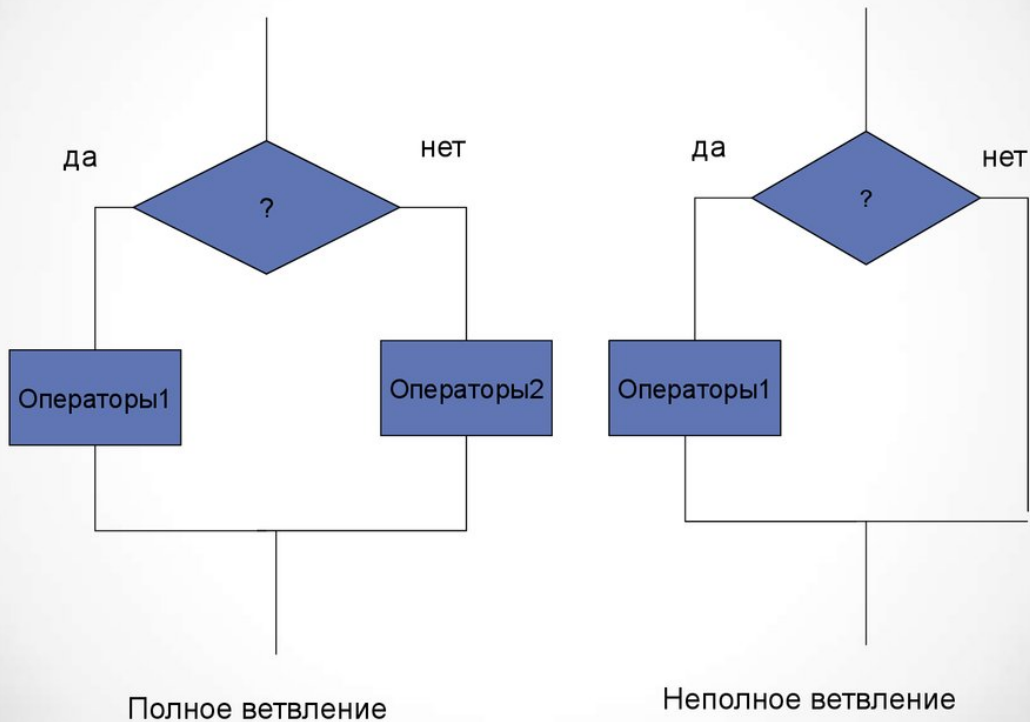
Количество if и elif в условии может быть более одного, что не скажешь об else

Блок-схема if/else



Также ветвление может быть полным и неполным (без использования конструкции else)

ВЕТВЛЕНИЕ



С использованием инструкции if неотъемлемо связаны операторы сравнения и логические операторы.

Операторы сравнения

Широко используются для получения булевых результатов (True/False) по определенным критериям

Например, $2 < 3$ отдает True

Оператор	Описание	Примеры
==	Проверяет равны ли оба операнда. Если да, то условие становится истинным.	5 == 5 в результате будет True True == False в результате будет False "hello" == "hello" в результате будет True
!=	Проверяет равны ли оба операнда. Если нет, то условие становится истинным.	12 != 5 в результате будет True False != False в результате будет False "hi" != "Hi" в результате будет True
<>	Проверяет равны ли оба операнда. Если нет, то условие становится истинным.	12 <> 5 в результате будет True. Похоже на оператор !=
>	Проверяет больше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	5 > 2 в результате будет True. True > False в результате будет True. "A" > "B" в результате будет False.
<	Проверяет меньше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	3 < 5 в результате будет True. True < False в результате будет False. "A" < "B" в результате будет True.
>=	Проверяет больше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	1 >= 1 в результате будет True. 23 >= 3.2 в результате будет True. "C" >= "D" в результате будет False.
<=	Проверяет меньше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	4 <= 5 в результате будет True. 0 <= 0.0 в результате будет True. -0.001 <= -36 в результате будет False.

ПРОГРАММИСТ ПОМНИ! Операторы сравнения для разных типов отдадут другие результаты

"2" == 2 -> False

Int("2") == 2 -> True

НО

2 == 2.0 -> True

Логические операторы

Необходимы для сложных связей, в том числе с операторами сравнения, например:

If a != b or x <= y

Оператор	Описание	Примеры
and	Логический оператор "И". Условие будет истинным если оба операнда истина.	True and True равно True. True and False равно False. False and True равно False. False and False равно False.
or	Логический оператор "ИЛИ". Если хотя бы один из операндов истинный, то и все выражение будет истинным.	True or True равно True. True or False равно True. False or True равно True. False or False равно False.
not	Логический оператор "НЕ". Изменяет логическое значение операнда на противоположное.	not True равно False. not False равно True.

Во всех языках программирования логические операторы основаны на использовании законов Булевой логики, поэтому на Python возможна реализация тождественно истинных и ложных условий:

If `a > b and b > a`: #никогда не будет True
Можно заменить на If False: и код никогда не будет выполняться

Также в Python в if необязательно писать True или False, например

```
If a == True:
    print("А истина")
```

Аналогично записи
If a:
 print("А истина")

И как следствие:

```
If a == False:
    print("А ложь")
```

Аналогично записи
If not a:
 print("А ложь")