# **Chaining comparison operators in Python**

Checking more than two conditions is very common in Programming Languages. Let’s say we want to check the below condition:

a < b < c

The most common syntax to do it is as follows:

if a < b and b < c :

{...}

In Python, there is a better way to write this using **Comparison operator Chaining**. The chaining of operators can be written as follows:

if a < b < c :

{.....}

According to [associativity and precedence in Python](https://www.geeksforgeeks.org/precedence-and-associativity-of-operators-in-python/), all comparison operations in Python have the same priority, which is lower than that of any arithmetic, shifting, or bitwise operation. Also unlike C, expressions like a < b < c have the interpretation that is conventional in mathematics. List of comparison operators in Python:

">" | "<" | "==" | ">=" | "<=" | "!=" | "is" ["not"] | ["not"] "in"

**Chaining in Comparison Operators:**

1. Comparisons yield boolean values: True or False.
2. Comparisons can be chained arbitrarily. For example:

x < y <= z is equivalent to x < y and y <= z,

1. except that y is evaluated only once. (but in both cases z is not evaluated at all when x < y is found to be false).
2. Formally, if a, b, c, …, y, z are expressions and op1, op2, …, opN are comparison operators, then a op1 b op2 c … y opN z is equivalent to a op1 b and b op2 c and … y opN z, except that each expression is evaluated at most once.
3. Also,

a op1 b op2 c

It doesn’t imply any kind of comparison between a and c, so

a < b > c

is perfectly legal.

|  |
| --- |
| # Python code to illustrate  # chaining comparison operators  x =5  print(1< x < 10)  print(10< x < 20)  print(x < 10< x\*10< 100)  print(10> x <=9)  print(5==x > 4) |

**Output:**

True

False

True

True

True

**Another Example:**

|  |
| --- |
| # Python code to Illustrate Chaining Comparison Operators    a, b, c, d, e, f =0, 5, 12, 0, 15, 15  exp1 =a & lt  =b & lt  c & gt  d isnote isf  exp2 =a isd & gt  f isnotc    print(exp1)  print(exp2) |

**Output:**

True

False

**Reference**: [Python 3 Documentation](https://docs.python.org/3/reference/expressions.html) This article is contributed by [**Pratik Chhajer**](https://www.linkedin.com/in/pratik-chhajer-4a102213b/). If you like GeeksforGeeks and would like to contribute, you can also write an article using [write.geeksforgeeks.org](https://write.geeksforgeeks.org/) or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or if you want to share more information about the topic discussed above.