



## 8.5. Precedence of Operators

Arithmetic operators take precedence over logical operators. Python will always evaluate the arithmetic operators first (`**` is highest, then multiplication/division, then addition/subtraction). Next comes the relational operators. Finally, the logical operators are done last. This means that the expression `x*5 >= 10 and y-6 <= 20` will be evaluated so as to first perform the arithmetic and then check the relationships. The `and` will be done last. Many programmers might place parentheses around the two relational expressions, `(x*5 >= 10) and (y-6 <= 20)`. It is not necessary to do so, but causes no harm and may make it easier for people to read and understand the code.

The following table summarizes the operator precedence from highest to lowest. A complete table for the entire language can be found in the [Python Documentation](#).

Level	Category	Operators
7(high)	exponent	<code>**</code>
6	multiplication	<code>*</code> , <code>/</code> , <code>//</code> , <code>%</code>
5	addition	<code>+</code> , <code>-</code>
4	relational	<code>==</code> , <code>!=</code> , <code>&lt;=</code> , <code>&gt;=</code> , <code>&gt;</code> , <code>&lt;</code>
3	logical	<code>not</code>
2	logical	<code>and</code>
1(low)	logical	<code>or</code>

### Note

This workspace is provided for your convenience. You can use this activecode window to try out anything you like.

Save & Run

5/9/2021, 9:21:09 AM - 2 of 2

Show in CodeLens

1

2

3

Activity: 1 -- ActiveCode (ac7\_5\_1)

### Common Mistake!

Students often incorrectly combine the `in` and `or` operators. For example, if they want to check that the letter `x` is inside of either of two variables then they tend to write it the following way:

```
'x' in y or z
```

Written this way, the code would not always do what the programmer intended. This is because the `in` operator is only on the left side of the `or` statement. It doesn't get implemented on both sides of the `or` statement. In order to properly check that `x` is inside of either variable, the `in` operator must be used on both sides which looks like this:

```
'x' in y or 'x' in z
```

### Check your understanding

condition-5-1: Which of the following properly expresses the precedence of operators (using parentheses) in the following expression: `5*3 > 10 and 4+6==11`

- ☒ A. `((5*3) > 10) and ((4+6) == 11)`
- ☐ B. `(5*(3 > 10)) and (4 + (6 == 11))`
- ☐ C. `(((((5*3) > 10) and 4)+6) == 11)`
- ☐ D. `((5*3) > (10 and (4+6))) == 11`

Check me

Compare me

✔ Yes, `*` and `+` have higher precedence, followed by `>` and `==`, and then the keyword `"and"`

Activity: 2 -- Multiple Choice (question7\_5\_1)

Here is an animation for the above expression:

Next Step

Reset

5 \* 3 > 10 and 4 + 6 == 11

5 \* 3 > 10 and 4 + 6 == 11  
15 > 10 and 4 + 6 == 11  
True and 4 + 6 == 11  
True and 10 == 11  
True and False  
False

Activity: 3 -- ShowEval (se\_ac7\_5\_1)

You have attempted 4 of 3 activities on this page

8.4. The in and not in operators">

in and not in operators">

✓ Completed. Well Done!

8.6. Conditional Execution: Binary Selection">

>