

What is short-circuiting with respect to compound conditional expressions in Python? X

Include a brief pseudocode example. X

Provide at least one reference to support your findings. X

Peer responses:

- Constructive feedback on the structures and circumstances posted by your peers
- Provide examples to support your response

Hello,

Short circuiting entails ending the execution of an conditional expression if a true value has been evaluated (Singh, 2019). The Python interpreter will rely on this process given boolean and conditional operators.

To determine an object's truth value (The Python Language Reference), Python utilizes `bool()` which calls a special method `__bool__()` (The Python Language Reference) in the Python data model. Special methods are intended to be called by the Python interpreter. We can expect an object to be true by default (if not an empty sequence or collection, a zero of any numeric type, None, or False) unless its class defines `__bool__()` returning False or `__len__()` returning 0.

Example:

"""

If free is true, execution will short-circuit and "use service" will be printed.

If free is false, `cost_effective()` and `available()` will be evaluated together. If they are both true, "use service" will be printed.

"""

```
if (free or cost_effective() and available())  
    print("use service")
```

Thanks,
Lauren

References

Singh, M. (2019, June 12). Short circuiting techniques in Python. Retrieved from <https://www.geeksforgeeks.org/short-circuiting-techniques-python>

The Python Language Reference. (n.d.). *Built-in Types: Truth Value Testing*.
<https://docs.python.org/3/library/stdtypes.html#truth-value-testing>

The Python Language Reference. (n.d.). *Data model: Basic customization*.
https://docs.python.org/3/reference/datamodel.html#object.__bool__

Yes, the *and* operator has higher precedence than the *or* operator (The Python Language Reference). Python groups the operands for the operator with the highest precedence first. So writing “a or b and c” means “a or (b and c).”

References

The Python Language Reference. (n.d.). *Expressions: Operator precedence*.
<https://docs.python.org/3/reference/expressions.html#operator-precedence>

Python evaluates first value then second value

<https://www.pythontutorial.net/python-basics/python-logical-operators/>

https://www.programiz.com/python-programming/keyword-list#and_or_not

If they both evaluate to true

Then the program would recommend using a service that is both free and cost-effective and offers a trial.

A true

B true

Result of a or b = true is true.

If the compiler short circuits

Professor,

I think one way of eliciting a different outcome is not taking side effects into consideration when short-circuiting occurs (Gennari, 2018). For instance, the code in my example can short-circuit but the application may be dependent on having both `cost_effective()` and `trial_available()` evaluated within that expression for future calculations.

References

Gennari, J. (2018). *Be aware of the short-circuit behavior of the logical AND and OR operators.*

<https://wiki.sei.cmu.edu/confluence/display/c/EXP02-C.+Be+aware+of+the+short-circuit+behavior+of+the+logical+AND+and+OR+operators>

This is a great example of short circuiting Rizart. I like that you incorporated the concept of probability as it pertains to chance of condition returning "true" or "false".

David,

I think the algorithm's first if statement is a good example of this optimization. As noted by Padmaja, if there are one or more defects, instead of running the algorithm a number of times to log a single defect, the algorithm can be modified to allow the user to supply a list of multiple defects at once. Here is what that could look like:

Lauren

I'd say this perhaps comes from the left to right reading of Python code (SINGH, 2019). Interested though that Python groups operands in highest precedent first. Does this mean that the example "a or b and c" is synthetically read all the way through to realize the and operator is there, then goes back to the start to execute?

From what I understand Python would group "b and c" together so that the interpreter would effectively read the expression as "(b and c)." This means the expression would be read as a single value that needs to be evaluated. Execution is when the code is being read. Often you'll hear 'run' (running an application) and 'execution' used interchangeably.