Introduction to Computer Programming

Logical Operators

Logical Operators

Logical operators modify and join expressions to create more complex statements with Boolean (`True` or `False`) value.

- **`and`**
- **`or`**
- **`not`**

`x and y`

Process:
Return `x` if the **Boolean** value of `x` is `False` ; otherwise, return `y` .

Output if `x` and `y` are both Booleans:
`True` if `x` **and** `y` are both `True` .
Otherwise `False` .

Examples: `and`

In [2]:
```
True and False
```

Out[2]: `False`

Test if both expressions are True

$10 < 9$ `False`

$20 <= 20$ `True`

In [3]:
```
print(10<9 and 20<=20)
```

```
False
```

Test if c is equal to b and a is greater than c

In [4]:
```
a = 5
b = -2
c = 4

print( c == b and a > c )
```

```
False
```

`x or y`

Process:
Return `x` if the **Boolean** value of `x` is `True`; otherwise, return `y`.

Output if `x` and `y` are both Booleans:
`True` if at least one of `x`, `y` has the value `True`.
Otherwise `False`.

Examples: `or`

In [6]:
```
False or True
```

Out[6]:
```
True
```

Test if at least one is True

$10 < 9$ `False`

$20 <= 20$ `True`

In [7]:
```
print(10<9 or 20<=20)
```

```
True
```

Test if c is equal to b or a is greater than c

In [8]:
```
a = 5
b = -2
c = 4

print(c == b or a > c)
```

```
True
```

The `not` operator negates the Boolean value of a statement

Examples: `not`

In [11]:
```
print( not 10 < 9 )
```

```
True
```

In [13]:
```
a = 12

print(a < 0)

print(not a < 0)
```

```
False
True
```

Stacking Comparison Operators

The following statement tests if both the outcome of the left comparison and the right comparison are True.

Both comparisons iclude the same variable, b

```
a < b and b < c
```

We can rewrite, *stacking* the comparison operators:

```
a < b < c
```

Rewrite

```
a < b and c < b
```

In [16]:
```
a, b, c = 0, 0, 1

print(a < b and c < b)

print(a < b > c )
```

```
False
False
```

Rewrite

```
a == b and b < c
```

In [17]:
```
a, b, c = 0, 0, 1

print(a == b and b < c)

print(a == b < c)
```

```
True
True
```

Operator Precedence

1. Parentheses
2. Arithmetic operators (top to bottom)
   - `**` Exponent
   - `/`, `*`, `//`, `\%` Division, multiplication, floor division, modulo (evaluated left to right)
   - `+`, `-` Addition, subtraction (evaluated left to right)
3. Comparison operators: `<`, `<=`, `>`, `>=`, `!=`, `==` (evaluated left to right)
4. Assignment operators `=`, `/=`, `*=`, `//=`, `\%=`, `+=`, `-=` ....
5. Identity operators `is`, `is not`
6. Logical `not`
7. Logical `and`
8. Logical `or`

Need to see some more examples?

https://www.w3schools.com/python/python_operators.asp (https://www.w3schools.com/python/python_operators.asp)
https://www.geeksforgeeks.org/python-operators/ (https://www.geeksforgeeks.org/python-operators/)
https://www.programiz.com/python-programming/operators (https://www.programiz.com/python-programming/operators)
https://pynative.com/python-operators/ (https://pynative.com/python-operators/)

Want to take a quiz?

https://realpython.com/quizzes/python-operators-expressions/ (https://realpython.com/quizzes/python-operators-expressions/)
https://pynative.com/python-operators-and-expression-quiz/ (https://pynative.com/python-operators-and-expression-quiz/)

Want some more advanced information on this topic?

https://realpython.com/python-operators-expressions/ (https://realpython.com/python-operators-expressions/)

In [ ]: