Logical Operators

In this notebook we introduce logical operators and study various examples of how boolean expressions can be combined together. Logical operators are used to combine boolean expressions. The three logical operators are `and`, `or`, `not`.

The order of precedence is as follows when evalutating expressions containing arithmetic, comparison and logical operators is:

1. PEMDAS comes first.
2. Followed by comparison operators all of which have equal precedence and are evaluated from left to right
3. Last are the logical operators. Among them, `not` is evaluated first, then `and` and finally `or`

We begin by first looking at the truth tables associated with the logical operators. The term 'Truth Tables' refers to the values that will result when two boolean operands are combined together with each one of the logical operators.

The **not** operator

X	**Not** X
True	False
False	True

The **or** operator

X	Y	X **Or** Y
True	True	True
True	False	True
False	True	True
False	False	False

The **and** operator

X	Y	X **And** Y
True	True	True
True	False	False
False	True	False
False	False	False

In [1]:

```
'''
Logical operators are used to combine boolean expressions
The three logical operators are  'and', 'or', 'not'
'''
x, y, z = 3, 5, 7
print(x >= 0 and x < 10)
print(x < 0 and x < 10)
print(x >= 0 and x < 2)
print(x < 0 or x < 10)
print(x > 0 or x < 10)
```

```
print(x > 0 or x < 10)
print(x < 0 or x > 10)
print(not (x == 3))
```

```
True
False
False
True
True
False
False
```

In [2]:

```
x, y, z = 3, 5, 7
print(x < 1 and y > 8 or z < 10)
print(x < 1 and (y > 8 or z < 10))
```

```
True
False
```

In [3]:

```
x, y = 3, 5
b1, b2, b3, b4 = True, False, x == 3, y < 3
print('b1: {}'.format(b1))
print('b2: {}'.format(b2))
print('b3: {}'.format(b3))
print('b4: {}'.format(b4))
print('not (not b1 or not b2 or not b3): {}'.format(not (not b1 or not b2 or not b3)))
print('not b1 and b2 and b3: {}'.format(not b1 and b2 and b3))
print('not (b1 and b2 and b3): {}'.format(not (b1 and b2 and b3)))
```

```
b1: True
b2: False
b3: True
b4: False
not (not b1 or not b2 or not b3): False
not b1 and b2 and b3: False
not (b1 and b2 and b3): True
```