# Learning Objectives - Boolean Operators

- Describe the difference between = and ==
- Explain how boolean statements are evaluated
- Describe how to use the AND and OR operators

# **Equal To & Not Equal To**

Boolean operators are operators that return a boolean value (true or false).

## **Equal To**

Python uses the == operator to determine equality. Beginners often confuse the = and the == operators. Remember, = is the assignment operator.

```
a = 5
b = 5
print(a == b)
```

challenge

# What happens if you:

- Change b to 1?
- Change a to True and b to 1?
- Change a to True and b to False?

## **Not Equal To**

The != operator checks to see if two values are not equal.

```
a = 5
b = 5
print(a != b)
```

- Change b to 1?
- Change a to True and b to 1?
- Change a to True and b to False?

# **Less Than & Less Than or Equal To**

#### **Less Than**

The < operator is used to check if one value is less than another value.

```
a = 5
b = 7
print(a < b)</pre>
```

challenge

#### What happens if you:

- Change b to 1?
- Change b to 5?
- Change b to False?

#### Less Than or Equal To

The <= operator is used to check if one value is less than or equal to another value.

```
a = 5
b = 7
print(a <= b)</pre>
```

challenge

- Change b to 1?
- Change b to 5?
- Change a to False and b to True?

# **Greater Than & Greater Than or Equal To**

#### **Greater Than**

The > operator is used to check if one value is greater than another value.

```
a = 9
b = 17
print(a > b)
```

challenge

## What happens if you:

- Change b to 1?
- Change b to 9?
- Change b to False?

## **Greater Than or Equal To**

The >= operator is used to check if one value is greater than or equal to another value.

```
a = 9
b = 17
print(a >= b)
```

- Change b to 1?
- Change b to 9?
- Change a to True and b to False?

#### **And**

#### The and Operator

The and operator allows for compound (more than one) boolean expressions. All boolean expressions **must** be true in order for the whole thing to be true. If only one boolean expressions is false, then the whole thing is false.

```
a = True
b = True
c = False
print(a and b)
```

challenge

## What happens if you:

- Change print to print(a and c)?
- Change print to print(c and b)?

#### **Multiple and Statements**

You can chain several and statements together. They are evaluated in a left-to-right manner.

```
a = True
b = True
c = False
print(a and b and c)
```

- Change print to print(a and b and a and b and a)?
- Change print to print(a and b and a and b and c)?

## The or Operator

The or operator allows for compound (more than one) boolean expressions. If only one boolean expressions is true, then the whole thing is true. To be false, all boolean expressions **must** be false.

```
a = True
b = True
c = False
d = False
print(a or b)
```

challenge

## What happens if you:

- Change print to print(a or c)?
- Change print to print(c or d)?

#### **Multiple or Statements**

You can chain several or statements together. They are evaluated in a left-to-right manner.

```
a = True
b = True
c = False
print(a or b or c)
```

- Change print to print(a or c or c or c or c)?
- Change print to print(c and c and c and c)?

## Not

## The not Operator

The not operator produces the opposite of the boolean expression that it modifies.

print(not True)

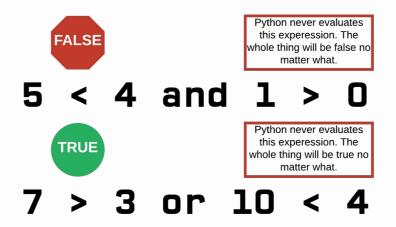
challenge

- Change print to print(not True and False)?
- Change print to print(not (True and False))?
- Change print to print(not not True)?

# **Short Circuiting**

## **Short Circuiting**

If Python can determine the result of a boolean expression before evaluating the entire thing, it will stop and return the value.



**Short Circuiting** 

# **Formative Assessment 1**

# **Formative Assessment 2**