Operators

In this lesson, we will learn about operators in Python.

WE'LL COVER THE FOLLOWING

Operators are used to perform *arithmetic* and *logical* operations on data. They enable us to manipulate and interpret data to produce useful outputs.

Arithmetic operators

Operator	Purpose
`()`	Parentheses
`**`	Exponent
`%`, `*`, `/`, `//`	Modulo, Multiplication, Division, Floor Division
`+`, `-`	Addition, Subtraction

```
x = 24
y = 12
z = 3

print("x + y =", x + y )    # addition
```

```
print("x - y =", x - y )    # subtraction
print("y * z =", y * z )    # multiplication
print("x / y =", x / y )    # division

print("y**2 =", y**2 )      # exponential
print("x % 5 =", x % 5 )    # modulo
print("x // 5 =", x // 5 )     # floor division
```

Precedence

An arithmetic expression containing different operators will be computed on the basis of operator precedence. Whenever operators have equal precedence, the expression is computed from the left side. Precedence can be altered using parentheses. An expression which is enclosed inside parentheses will be computed first, regardless of operator precedence:

```
# Different precendence
print ("10 - 3 * 2 =", 10 - 3 * 2) # Multiplication computed first, followed by subtraction

# Same precedence
print ("3 * 20 / 5 =", 3 * 20 / 5) # Multiplication computed first, followed by division

# Using Parenthesis
print ("(10 - 3) * 2 =", (10 - 3)  * 2) # subtraction computed first, followed by multiplicat
```

Comparison operators

Comparison operators can be used to compare values in mathematical terms.

Operator	Purpose
`>`	Greater Than
`<`	Less Than
`>=`	Greater Than or Equal To
`<=`	Less Than or Equal To

`==`	Equal to
`is`	Equal to
`is not`	Not Equal To

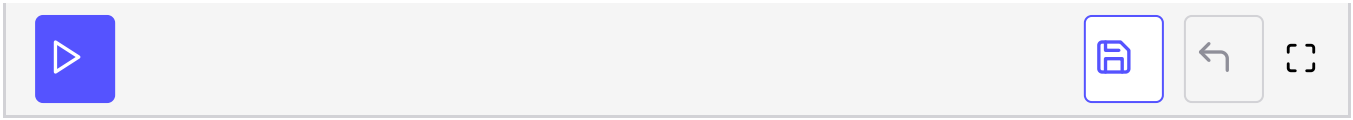
The comparison operator always returns a bool. If the comparison is correct, `True` is returned, otherwise `False`.

Let's look at a few examples:

```
x = 5
y = 10
z = 10
print("y > x =",y > x) # 10 is greater than 5
print("x > y =", x > y) # 5 is not greater than 10

print("y is z =", y is z) # Both have the same value
print("y is not x =", y is not x) # Both have different values

print (3 + 10 == 5 + 5) # Both are not equal
print (3 <= 2) # 3 is not less than or equal to 2
```



Logical operators

Logical operators are used to manipulate the logic of *Boolean expressions*.

Operator	Purpose
`and`	AND
`or`	OR
`not`	NOT

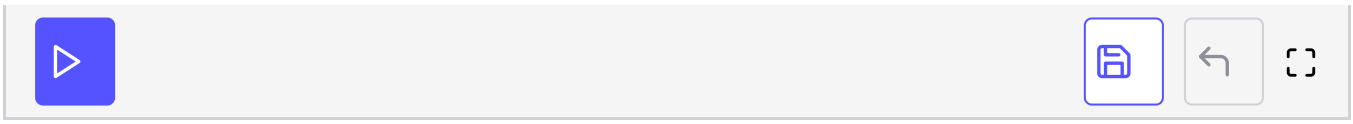
Logical expressions are formed using Booleans and logical operators. Below,

we can find some examples:

```
# OR Expression
my_bool = True or False
print (my_bool)

# AND Expression
my_bool = True and False
print (my_bool)

# NOT expression
my_bool = False
print (not my_bool)
```



In the next lesson, we will learn about conditional statements in Python.