Operators

1. What is an Operator?

Operators

- Symbols that perform operations on variables and values.

Operands

- Variables or values which are passed as inputs to an operator.

For example: `10 - 4`

- `-` is an operator which performs `minus` operation
- `10` and `4` are operands

Similar to many programming languages, Python reserves some special characters for acting as operators.

Python operators are grouped into following categories:

- Arithmetic operators
- Comparison operators
- Identity operators
- Membership operators
- Logical operators
- Assignment operators
- Bitwise operators

2. Operators

2.1 Arithmetic Operators

Arithmetic operators perform common mathematical operations on values.

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
//	Floor division	x // y
%	Modulus	x % y
**	Exponentiation	x ** y

Addition `+`, Subtraction `-`

In [1]:

```
x = 5
y = 3
x + y
x - y
```

Out[1]:

2

Multiplication `*`

In [2]:

```
x * y
```

Out[2]:

```
15
```

Division `/`

In [3]:

```
x / y
```

Out[3]:

```
1.6666666666666667
```

Floor Division `//`

In [4]:

```
x // y
```

Out[4]:

```
1
```

Modulus Division `%`

In [5]:

```
x % y
```

Out[5]:

```
2
```

Exponential `**`

In [6]:

```
x ** y
```

Out[6]:

```
125
```

How to Round a number?

Use `round()` function. *Note: https://stackoverflow.com/questions/10825926/python-3-x-rounding-behavior*

In [7]:

```
round(2.5)
```

```
round(3.5)
round(4.5)
round(5.5)
```

Out[7]:

```
6
```

2.2 Comparison Operators

Comparison operators are used to compare two values.

- It returns a boolean value.

Operator	Name	Example
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y
==	Equal	x == y
!=	Not equal	x != y

Greater than `>`, Greater than or equal to `>=`

In [8]:

```
x, y = 5, 3
x > y
x >= y
```

Out[8]:

```
True
```

Euqal `==`, Not equal `!=`

In [9]:

```
x == y
x != y
```

Out[9]:

```
True
```

2.3 Logical Operators

Sometimes we need to make decisions based on multiple conditions. Logical operators are used to combine conditional statements.

- Operands shall be conditions which can result in a boolean value.
- The outcome of such an operation is either true or false too.

Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

In [10]:

```
x = 7
x < 5 and x < 10
```

Out[10]:

```
False
```

In [11]:

```
x < 5 or x > 10
```

Out[11]:

```
False
```

In [12]:

```
not( x == 7 )
```

Out[12]:

```
False
```

2.4 Identity Operators

Identity Operators check whether two objects/variables are identical, i.e. whether they point to same memory locations.

Operator	Description	Example
is	Returns true if both variables are the same object	x is y
is not	Returns true if both variables are not the same object	x is not y

In [13]:

```
x = 1
y = x
x is y
```

Out[13]:

```
True
```

In [14]:

```
x = 1000
y = 1000
x is y
```

Out[14]:

```
False
```

Question:

- What the output of `x is y` if both `x` and `y` are set to a small integer value, e.g. `1`.
- Are they pointing to same object? Use `id()` function to examine them.

In [15]:

```
x = 1
```

```
y = 1
x is y
```

Out[15]:

```
True
```

Identity Operator can also be used to determine whether a value is of a specific class or type.

In [16]:

```
x = 1
type(x) is int
```

Out[16]:

```
True
```

2.5 Membership Operators

Membership operators enable us to test whether a value is a member of other Python objects such as strings, lists, or tuples.

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

In [17]:

```
x = 5
y = [1,3,5,7]
x in y
```

Out[17]:

```
True
```

In [18]:

```
x = 5
y = set([1,3,5,7])
x not in y
```

Out[18]:

```
False
```

In [19]:

```
x = 5
y = {'five':5, 'two':2}
x in y.values()
```

Out[19]:

```
True
```

In [20]:

```
10 in y
```

Out[20]:

```
False
```

String is a collection of characters. It behaves very much like a list.

In [21]:

```
x = 'Hello'
'o' in x
```

Out[21]:

```
True
```

In [22]:

```
'h' not in x
```

Out[22]:

```
True
```

2.6 Bitwise Operators

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
|	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and remove leftmost bits
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left and remove rightmost bits

In [23]:

```
x = 8
y = 4
z = x | y
print(bin(x), bin(y), bin(z))
x & y
```

```
0b1000 0b100 0b1100
```

Out[23]:

```
0
```

2.7 Assignment Operators

Assignment operators are used to assign values to variables.

Following are compound operator which perform operation and then assign value.

Operator	Example	Same As
+=	x+=3	x=x+3
-=	x-=3	x=x-3
*=	x*=3	x=x*3
/=	x/=3	x=x/3

Operator	Example	Same As
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3

In [24]:

```
x = 10
# x++ # Not working
x += 1
```

2.8 Operator Precedence

Operators have precedence so that expressions are evaluated in predefined order.

Following are some common operators with descending order of precedence.

- Expressions in `()` operator are given higher precedences.
- Product, division, remainder
  - `*`, `/`, `//`, `%`
- Addition, subtraction
  - `+`, `-`
- Comparisons, membership, identity
  - `in`, `not in`, `is`, `is not`, `<`, `<=`, `>`, `>=`, `!=`, `==`

If not sure, always use parentheses `()` to group expressions.

In [25]:

```
10 * (1 + 2) % 3
```

Out[25]:

```
0
```

Recap

- What's the operator to check if two values are NOT equal?
- How to check if two variables are referring to the same object?
- What's the operator to check whether a list contains a particular value?
- What are the operators used to join multiple conditions?

Reference

- https://www.w3schools.com/python/python_operators.asp