## Operators

Operators are the special symbols, which carry out **arithematic or logical processes/actions.** The variables/objects on which operator operates are called **operands**.

**Types of Operators in Python:**

1. Arithmetic Operators
2. Comparison Operators / Relational Operators
3. Logical Operators
4. Bitwise Operators
5. Assignment Operators
6. Special Operators
    - Identity Operators
    - Membership Operators

## Arithmetic Operators

| | | |
|---|---|---|
| % | **Modulus: returns remainder of the division of left operand by right operand** | a%b |
| ** | **Power/Exponent: returns left operand raised to the power right operand** | a**b |

## Arithmetic Operators

| Operator | Illustration | Syntex |
|---|---|---|
| + | **Addition: add two operands** | a+b |
| - | **Subtraction: subtract right operand from left operand** | a-b |
| * | **Multiplication: multiply two operands** | a*b |
| / | **Division(float): divides left operand by right operand** | a/b |
| // | **Division(floor): divides left operand by right operand and returns integer value** | a//b |

## + Operator

```
>>> 5+7
12
```

```
In [1]:a = -15
In [2]:b = 7
In [3]:c = a+b
In [4]:print(c)
-8
```

? **Questions:**

1. Can we use + operator on strings? If yes, then, what will it do?
2. Can we use + operator on operands where one is numeric and other is string?

## - Operator

```
In [7]: 10-5
5
```

```
>>>a = 15
>>>b = -5
>>>a-b
20
```

**Questions:**

1 Can we use - operator on strings? If yes, then, what will it do?

2 Can we use - operator on operands where one is numeric and other is string?

## / Operator

```
>>> 25/5
5
```

```
In [8]:a = 7
In [9]:b = 3
In [10]:print(a/b)
2.3333333333333335
```

**Questions:**

1 Can we use / operator on strings? If yes, then, what will it do?

2 Can we use / operator on operands where one is numeric and other is string?

## * Operator

```
>>> 15*5
75
```

```
In [8]:a = 4
In [9]:b = -5
In [10]:print(a*b)
-20
```

**Questions:**

1 Can we use * operator on strings? If yes, then, what will it do?

2 Can we use * operator on operands where one is numeric and other is string?

## // Operator

```
>>> 25//5
5
```

```
In [8]:a = 7
In [9]:b = 3
In [10]:print(a//b)
2
```

**Questions:**

1 Can we use // operator on strings? If yes, then, what will it do?

2 Can we use // operator on operands where one is numeric and other is string?

## % Operator

```
>>> 13%2
1
```

```
In [8]:a = 26.8
In [9]:b = 7
In [10]:print(a%b)
5.8
```

**Questions:**

1. Can we use % operator on strings? If yes, then, what will it do?
2. Can we use % operator on operands where one is numeric and other is string?

## Comaparison Operators/Relational Operators

Relational Operators compare similar type of obects. A relational operator either returns a **True** or **False** value according to the condition.

## ** Operator

```
>>> 2**6
64
```

```
In [8]:a = 2.5
In [9]:b = 6
In [10]:print(a**b)
244.140625
```

**Questions:**

1. Can we use ** operator on strings? If yes, then, what it will do?
2. Can we use ** operator on operands where one is numeric and other is string?

## Relational Operators

| Operator | Illustration | Syntex |
|---|---|---|
| < | **Less than: True if left operand is less than right operand, otherwise, False** | a<b |
| > | **Greater than: True if left operand is greater than righ operand, otherwise, False** | a>b |
| == | **Equal to: True if left operand and right operand are equal, otherwise, False** | a==b |
| != | **Not equal to: True if operands are not equal, otherwise, False** | a!=b |

## Relational Operators

| | | |
|---|---|---|
| ≤ | **Less than or equal to: True if left operand is less than equal to right operand, otherwise, False** | a≤b |
| ≥ | **Greater than or equal to: True if left operand is greater than equal to right operand, otherwise, False** | a≥b |

## Logical operators

Logical operators perform **Logical AND**, **Logical OR** and **Logical NOT** operations. Returns either **True** or **False** according to boolean algebra.

| AND | | |
|---|---|---|
| A | B | A and B |
| False | False | False |
| False | True | False |
| True | False | False |
| True | True | True |

| OR | | |
|---|---|---|
| A | B | A or B |
| False | False | False |
| False | True | True |
| True | False | True |
| True | True | True |

| NOT | |
|---|---|
| A | not A |
| False | True |
| True | False |

## Examples of Relational Operators

```
ln[1]:a = 7
ln[2]:b = 11
ln[3]:print('a<b is',a<b)
a<b is True
ln[4]:print('a>b is',a>b)
a>b is False
ln[5]:print('a==b is',a==b)
a==b is False
```

```
ln[6]:print('a!=b is',a!=b)
a!=b is True
ln[7]:print('a<=b is',a<=b)
a<=b is True
ln[8]:print('a>=b is',a>=b)
a>=b is False
```

? **Questions:**

1 Can we use relational operators on strings? If yes, then, what will they do?

2 Can we use relational operators on operands where one is numeric and other is string?

## Logical operators

| Operator | Illustration | Syntex |
|---|---|---|
| and | Logical AND: True if both the operands are true | a and b |
| or | Logical OR: True if either of the operand is true | a or b |
| not | Logical NOT: True if the operand is false (Complements the operand) | not a |

## Examples

```
ln[1]:a = False
ln[2]:b = True
ln[3]:print('a and b is',a and b)
a and b is False
ln[4]:print('a or b is',a or b)
a or b is True
ln[5]:print('not a is',not a)
not a is True
ln[6]:print('not b is',not b)
not b is False
```

## Examples

**? Questions:**

1 Do logical operators work with numerical operands (e.g. 4 and 6, 5 or 7, not 10 etc)? If yes, then, figure out their working?

2 Do logical operators work with string operands (e.g. ‘hello’ and ‘world’, ‘python’ or ‘spyder’, not ‘python’ etc)? If yes, then, figure out their working?

3 Do logical operators work with mixed operands (e.g. ‘hello’ and 13.4, 0 or ‘spyder’ etc)? If yes, then, figure out their working ?

## Bitwise Operator

Bitwise operators perform bitwise operations on binary equivalent of the operand.

$13_{10}$

$$\left.\begin{array}{rc} 2)\underline{13} & 1 \\ 2)\underline{6} & 0 \\ 2)\underline{3} & 1 \\ 2)\underline{1} & 1 \end{array}\right\} 1101$$

$156_{10}$

$$\left.\begin{array}{rc} 2)\underline{156} & 0 \\ 2)\underline{78} & 0 \\ 2)\underline{39} & 1 \\ 2)\underline{19} & 1 \\ 2)\underline{9} & 1 \\ 2)\underline{4} & 0 \\ 2)\underline{2} & 0 \\ 2)\underline{1} & 1 \end{array}\right\} 10011100$$

## Bitwise Operator

| Operator | Illustration | Syntex |
|---|---|---|
| & | **Bitwise AND** | a&b |
| \| | **Bitwise OR** | a\|b |
| ~ | **Bitwise NOT** | ~a |
| ^ | **Bitwise XOR** | a^b |
| << | **Bitwise left shift** | a<<b |
| >> | **Bitwise right shift** | a>>b |

## Examples

```
ln[1]:a = 51   # 0011 0011
ln[2]:b = 23   # 0001 0111
ln[3]:n = 2
ln[4]:print('a&b is',a&b)
a&b is 19                  # 19 =  0001 0011
ln[5]:print('a|b is',a|b)
a | is 55                                # 55 = 0011 0111
ln[6]:print('~a is',~a)
~a is -52                                #52 = 1100 1100
ln[7]:print('a^b is',a^b)
a^b is 36                                #36 = 0010 0100
ln[8]:print('a<<n is',a<<n)
a<<n is 204                                 #204 = 1100 1100
ln[9]:print('a>>n is',a>>n)
a>>n is 36                                #12 = 0000 1100
```

## Examples

**? Questions:**

1 Do bitwise operators support floating type operands?

2 Do bitwise operator support string type operands?

## Assignment Operator

Assignment operators are used to assign values to variables.

## Assignment Operator

| Operator | Illustration |
|---|---|
| = | a=b+c |
| += | a+=b $\Longleftrightarrow$ a=a+b |
| -= | a-=b $\Longleftrightarrow$ a=a-b |
| *= | a*=b $\Longleftrightarrow$ a=a*b |
| /= | a/=b $\Longleftrightarrow$ a=a/b |
| //= | a//=b $\Longleftrightarrow$ a=a//b |

## Assignment Operator

| | |
|---|---|
| %= | a%=b $\Longleftrightarrow$ a=a%b |
| **= | a**=b $\Longleftrightarrow$ a=a**b |
| &= | a&=b $\Longleftrightarrow$ a=a&b |
| \|= | a\|=b $\Longleftrightarrow$ a=a\|b |

## Examples

```
ln[11]:a = 15
ln[12]:b = 8
ln[13]:c = 184
ln[14]:c /= b
ln[15]:print('Value of c is',c)
Value of c is 23
ln[16]:c //= a
ln[17]:print('Value of c is',c)
Value of c is 1
ln[18]:c = 3
ln[19]:c **= b
ln[20]:print('Value of c is',c)
Value of c is 6561
ln[21]:c %= a
ln[22]:print('Value of c is,c)
Value of c is 6
```

**Homework**
Figure out the working of bitwise assingment operators (e.g. a&=b, a|=b).

## Examples

```
ln[1]:a = 15
ln[2]:b = 8
ln[3]:c = 0
ln[4]:c=a+b
ln[5]:print('Value of c is',c)
Value of c is 23
ln[6]:c += a
ln[7]:print('Value of c is',c)
Value of c is 38
ln[8]:c -= a
ln[9]:print('Value of c is',c)
Value of c is 23
ln[10]:c *= b
ln[11]:print('Value of c is',c)
Value of c is 184
```