



Operators



B. Ramar(Ph.D)

Resource Faculty,

AU TVS Centre for Quality Management, Anna University.

Content



- 1. What are operators in Python?**
- 2. Arithmetic Operators**
- 3. Comparison (Relational) Operators**
- 4. Logical (Boolean) Operators**
- 5. Bitwise Operators**
- 6. Assignment Operators**
- 7. Special Operators**
 - a) Identity Operator**
 - b) Membership Operator**

Arithmetic operators

Operator	Meaning	Example
+	Add two operands or unary plus	$x + y$ $+2$
-	Subtract right operand from the left or unary minus	$x - y$ -2
*	Multiply two operands	$x * y$
/	Divide left operand by the right one (always results into float)	x / y
%	Modulus - remainder of the division of left operand by the right	$x \% y$ (remainder)
//	Floor division - division that results into whole number adjusted to the left in the number line	$x // y$
**	Exponent - left operand raised to the power of right	$x ** y$ (x to the power y)

Arithmetic operators in Python

```
x = 15
```

```
y = 4
```

```
# Output: x + y = 19
```

```
print('x + y =',x+y)
```

```
# Output: x - y = 11
```

```
print('x - y =',x-y)
```

```
# Output: x * y = 60
```

```
print('x * y =',x*y)
```

```
# Output: x / y = 3.75
```

```
print('x / y =',x/y)
```

```
# Output: x // y = 3
```

```
print('x // y =',x//y)
```

```
# Output: x ** y = 50625
```

```
print('x ** y =',x**y)
```

Comparision operators

Operator	Meaning	Example
>	Greater that - True if left operand is greater than the right	$x > y$
<	Less that - True if left operand is less than the right	$x < y$
==	Equal to - True if both operands are equal	$x == y$
!=	Not equal to - True if operands are not equal	$x != y$
>=	Greater than or equal to - True if left operand is greater than or equal to the right	$x >= y$
<=	Less than or equal to - True if left operand is less than or equal to the right	$x <= y$

Comparision operators

```
x = 10
```

```
y = 12
```

```
# Output: x > y is False
```

```
print('x > y is',x>y)
```

```
# Output: x < y is True
```

```
print('x < y is',x<y)
```

```
# Output: x == y is False
```

```
print('x == y is',x==y)
```

```
# Output: x != y is True
```

```
print('x != y is',x!=y)
```

```
# Output: x >= y is False
```

```
print('x >= y is',x>=y)
```

```
# Output: x <= y is True
```

```
print('x <= y is',x<=y)
```

Logical operators

Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

```
x = True  
y = False
```

```
# Output: x and y is False  
print('x and y is',x and y)
```

```
# Output: x or y is True  
print('x or y is',x or y)
```

```
# Output: not x is False  
print('not x is',not x)
```

Bitwise operators

In the table below: Let $x = 10$ (0000 1010 in binary) and $y = 4$ (0000 0100 in binary)

Operator	Meaning	Example
&	Bitwise AND	$x \& y = 0$ (0000 0000)
	Bitwise OR	$x y = 14$ (0000 1110)
~	Bitwise NOT	$\sim x = -11$ (1111 0101)
^	Bitwise XOR	$x \wedge y = 14$ (0000 1110)
>>	Bitwise right shift	$x >> 2 = 2$ (0000 0010)
<<	Bitwise left shift	$x << 2 = 40$ (0010 1000)

Assignment operators

Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
=	x = 5	x = x 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5

Assignment operators

Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
=	x = 5	x = x 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5

Special operators

Python language offers some special type of operators like the identity operator or the membership operator.

Identity operators

Operator	Meaning	Example
is	True if the operands are identical (refer to the same object)	x is True
is not	True if the operands are not identical (do not refer to the same object)	x is not True

```
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]
```

```
# Output: False
print(x1 is not y1)
```

```
# Output: True
print(x2 is y2)
```

```
# Output: False
print(x3 is y3)
```

Special operators

Python language offers some special type of operators like the identity operator or the membership operator.

Membership operators

Operator	Meaning	Example
in	True if value/variable is found in the sequence	5 in x
not in	True if value/variable is not found in the sequence	5 not in x

```
x = 'Hello world'
y = {1:'a',2:'b'}
```

```
# Output: True
print('H' in x)
```

```
# Output: True
print('hello' not in x)
```

```
# Output: True
print(1 in y)
```

```
# Output: False
print('a' in y)
```

Content



- 1. What are operators in Python?**
- 2. Arithmetic Operators**
- 3. Comparison (Relational) Operators**
- 4. Logical (Boolean) Operators**
- 5. Bitwise Operators**
- 6. Assignment Operators**
- 7. Special Operators**
 - a) Identity Operator**
 - b) Membership Operator**