



PYTHON OPERATORS



Operators are symbols (sometimes keywords) that are predefined to perform a certain most commonly required operations on one or more operands.

What are the Types of Operators?

Types of Operators

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

Arithmetic Operators

Operator	Name	Example
+	Addition	$a + b = 30$
-	Subtraction	$a - b = -10$
*	Multiplication	$a * b = 200$
/	Division	$b / a = 2$
%	Modulus	$b \% a = 0$
**	Exponent	$a ** b = 10 ** 20$
//	Floor Division	$9 // 2 = 4$

used to perform basic mathematical operations.

Arithmetic Operators

```
a=10 b=20
```

```
print ("Addition of two integers")
```

```
print ("a =",a,"b =",b,"addition =",a+b)
```

Output:

```
Addition of two integers
```

```
a = 10 b = 20 addition = 30
```

Arithmetic Operators

```
a=10
```

```
b=20.5
```

```
print ("Addition of integer and float")
```

```
print ("a =",a,"b =",b,"addition =",a+b)
```

Output:

```
Addition of integer and float
```

```
a = 10 b = 20.5 addition = 30.5
```

Arithmetic Operators

```
a=10+5j
```

```
b=20.5
```

```
print ("Addition of complex and float")
```

```
print ("a=",a,"b=",b,"addition=",a+b)
```

Output:

```
Addition of complex and float
```

```
a= (10+5j) b= 20.5 addition= (30.5+5j)
```


Comparison Operators

Operator	Name	Example
<code>==</code>	Equal	<code>(a == b)</code> is not true.
<code>!=</code>	Not equal	<code>(a != b)</code> is true.
<code>></code>	Greater than	<code>(a > b)</code> is not true.
<code><</code>	Less than	<code>(a < b)</code> is true.
<code>>=</code>	Greater than or equal to	<code>(a >= b)</code> is not true.
<code><=</code>	Less than or equal to	<code>(a <= b)</code> is true.

compare the values on either sides of them and decide the relation among them.

Assignment Operators

Operator	Example	Same As
<code>=</code>	<code>a = 10</code>	<code>a = 10</code>
<code>+=</code>	<code>a += 30</code>	<code>a = a + 30</code>
<code>-=</code>	<code>a -= 15</code>	<code>a = a - 15</code>
<code>*=</code>	<code>a *= 10</code>	<code>a = a * 10</code>
<code>/=</code>	<code>a /= 5</code>	<code>a = a / 5</code>
<code>%=</code>	<code>a %= 5</code>	<code>a = a % 5</code>

used to assign values to variables.

Assignment Operators

<code>**=</code>	<code>a **= 4</code>	<code>a = a ** 4</code>
<code>//=</code>	<code>a //= 5</code>	<code>a = a // 5</code>
<code>&=</code>	<code>a &= 5</code>	<code>a = a & 5</code>
<code> =</code>	<code>a = 5</code>	<code>a = a 5</code>
<code>^=</code>	<code>a ^= 5</code>	<code>a = a ^ 5</code>
<code>>>=</code>	<code>a >>= 5</code>	<code>a = a >> 5</code>
<code><<=</code>	<code>a <<= 5</code>	<code>a = a << 5</code>

used to assign values to variables.

Bitwise Operators

Operator	Name	Example
&	AND	<code>a & b</code>
	OR	<code>a b</code>
^	XOR	<code>a ^ b</code>
~	NOT	<code>~a</code>
<<	Zero fill left shift	<code>a << 3</code>
>>	Signed right shift	<code>a >> 3</code>

works on bits
and performs
bit by bit
operation.

Logical Operators

Operator	Name	Example
and	AND	a and b
or	OR	a or b
not	NOT	not(a)

used to combine two or more conditions and check the final result.

Membership Operators

Operator	Description	Example
in	Returns True if it finds a variable in the specified sequence, false otherwise.	a in b
not in	returns True if it does not finds a variable in the specified sequence and false otherwise.	a not in b

test for membership in a sequence, such as strings, lists, or tuples.

Identity Operators

Operator	Description	Example
is	Returns True if both variables are the same object and false otherwise.	a is b
is not	Returns True if both variables are not the same object and false otherwise.	a is not b

compare the memory locations of two objects.

Python Operators Precedence

Sr.No.	Operator & Description
1	** Exponentiation (raise to the power)
2	~ + - Complement, unary plus and minus (method names for the last two are +@ and -@)
3	* / % // Multiply, divide, modulo and floor division

Python Operators Precedence

4	+ - Addition and subtraction
5	>> << Right and left bitwise shift
6	& Bitwise 'AND'
7	^ Bitwise exclusive 'OR' and regular 'OR'

Python Operators Precedence

8

<= < > >=

Comparison operators

9

<> == !=

Equality operators

10

= %= /= //= -= += *= **=

Assignment operators

Python Operators Precedence

11

is is not

Identity operators

12

in not in

Membership operators

13

not or and

Logical operators



Applications???

Any
questions???

REFERENCES:

- Learn Python Programming. (2023). <https://www.tutorialsteacher.com/python>
- Python Tutorial. (2022). <https://www.w3resource.com/python/python-tutorial.php>
- Python Tutorial. (n.d.). <https://www.tutorialspoint.com/python/index.htm>
- Python Tutorial. (n.d.). <https://www.w3schools.com/python/default.asp>