

# Precedence and Associativity In Python

## PRECEDENCE OPERATOR

Good day everyone,

In this module you will learn about precedence and associativity of python operators.

Let's start with the definition of what precedence is.

Operator precedence is used to determine the order of operators evaluated in an expression.

When we group a set of values, variables, operators or function calls that turn out as an expression.

In Python programming every operator has it's precedence (priority).

If more than one operator is used in an expression, the higher precedence operator is evaluated first and the operator with the least precedence is evaluated last.

Let's have a look at an example.

Example

A=1

B=2

C=3

Answer =  $a + b * c$  (We assume that first the addition will happen( $1+2$ ) followed by multiplication with the result of addition which in this case is  $3*3 = 9$ )

Output =  $a + b * c$  (But Computer evaluates in the order where multiplication happens first followed by addition with its result that is  $1 + 6 = 7$ )

Python gives precedence to multiplication over addition.

Let's look at the order in which python gives first preference to an operator.

The easy way to remember is the acronym PEMDAS (Parentheses, Exponentiation, Multiplication, Division, Addition, Subtraction)

Parentheses have the highest precedence and can be used to force an expression to evaluate in the order you want.

Since expressions in parentheses are evaluated first,  $2 * (3-1)$  is 4

and  $(1+1)**(5-2)$  is 8.

Exponentiation has the next highest precedence, so  $1 + 2^{**}3$  is 9, not 27

And  $2 * 3^{**}2$  is 18, not 36.

Multiplication and Division have higher precedence than Addition and Subtraction. So  $2*3-1$  is 5, not 4

And  $6+4/2$  is 8, not 5.

I hope the order of precedence in python is clear. One additional example will help to solidify the concept.

In this example

$1+2^{**}3$

Since exponentiation has the higher precedence, therefore  $2^{**}3$  will happen first with the output 8.

Addition has lower precedence therefore this will happen second which is  $1+8$  resulting into 9.

Let's take a look at all of the operators and their meanings in order of priority.

Precedence	Operator	Operator Meaning
1	( )	Parenthesis Or Function Call
2	**	Exponentiation
3	+x	Unary Plus
	-x	Unary Minus
	~x	Bitwise Negation
4	*	Multiplication
	/	Division
	//	Floor Division
	%	Modulus
5	+	Addition
	-	Subtraction

6	<<	Left Shift
	>>	Right Shift



Precedence	Operator	Operator Meaning
7	&	Bitwise AND
8	^	Bitwise XOR
9		Bitwise OR
10	==	Equal to
	!=	Not Equal to
	<	Less Than
	<=	Less Than or Equal to
	>	Greater Than
	>=	Greater Than or Equal to
	is	Identity Operator

	<b>is not</b>	<b>Identity Operator</b>
--	---------------	--------------------------

<b>Precedence</b>	<b>Operator</b>	<b>Operator Meaning</b>
<b>11</b>	<b>not</b>	<b>Boolean NOT</b>
<b>12</b>	<b>and</b>	<b>Boolean AND</b>
<b>13</b>	<b>or</b>	<b>Boolean OR</b>

In Python, the acronym PEMDAS is useful for remembering operator preferences. Wasn't that simple to recall?



vityarthi

# ORDER OF EVALUATION/ASSOCIATIVITY

Back to the subject of operators is a pleasure. You will learn about Order of Evaluation/Associativity in this module. Wonder what these heavy words are? To understand this you will have to remember the operator precedence. Keep in mind that Python prioritises operators in a specific order.

When there are many operators with equal precedence in an expression, we use associativity to determine the order in which they are evaluated.

It can either be Left to Right or from Right to Left.

Operator associativity is used to determine the order of operators with equal precedence are evaluated in an expression.

When an expression contains multiple operators with equal precedence, we use associativity to determine the order of evaluation of those operators.

Let us understand this with an example.

$100+200/10-3*10$

Now here we have multiple operators.

Now division and multiplication both have the same precedence but division will be done first because of Left to Right (LTR) Associativity

Similarly multiplication will happen second.

It has higher precedence than addition & subtraction.

It has the same precedence as division but has lower associativity.

Now we have got the result of division and multiplication.

Which is 20 and 30 respectively.

Now addition will happen third.

It has lower precedence than division & multiplication.

It has the same precedence as subtraction but has higher associativity.

Result of this operation is 120.

Last will be subtraction that will happen.

It has lower precedence than division & multiplication.

It has the same precedence as addition but has lower associativity.

Notice that all the operators which have the same precedence follow left to right or LTR associativity.

Remember

Almost all operators except the exponent (\*\*) support the left-to-right associativity.

I hope the concept of precedence and associativity is clear with all the examples discussed in the module.

We shall touch upon many such amazing topics in the upcoming modules. Stay connected.



vityarthi