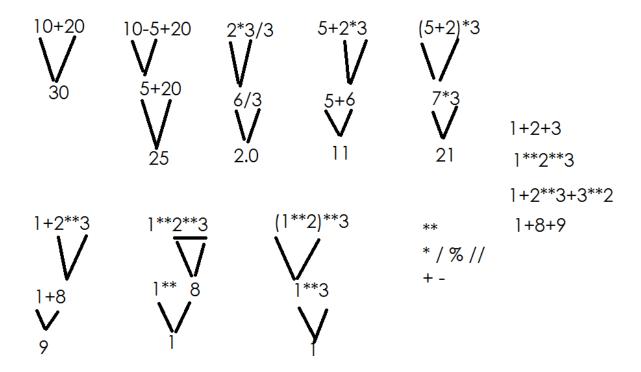
Operators Precedence

The following table summarizes the operator precedence in Python, from highest precedence (most binding) to lowest precedence (least binding). Operators in the same box have the same precedence. Unless the syntax is explicitly given, operators are binary. Operators in the same box group left to right (except for exponentiation, which groups from right to left).

Operator	Description		
(expressions), [expressions], {key: value}, {expressions}	Binding or parenthesized expression, list display, dictionary display, set display		
x[index], x[index:index], x(arguments), x.attribute	Subscription, slicing, call, attribute reference		
await x	Await expression		
**	Exponentiation		
+x, -x, ~x	Positive, negative, bitwise NOT		
*, @, /, //, %	Multiplication, matrix multiplication, division, floor division, remainder		
+, -	Addition and subtraction		
<<, >>	Shifts		
&	Bitwise AND		
۸	Bitwise XOR		
	Bitwise OR		
<u>in, not in, is, is not,</u> <, <=, >, >=, !=, ==	Comparisons, including membership tests and identity tests		
not x	Boolean NOT		
and	Boolean AND		
<u>or</u>	Boolean OR		
<u>if</u> – else	Conditional expression		
<u>lambda</u>	Lambda expression		
<u>;=</u>	Assignment expression		



Relational Operators

These operators are used to compare values.

These operators return boolean value (True/False).

Operators	Description
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
==	Equal
!=	Not Equal

These operators are applied on scalar data types and collection data types.

>>>10>=5

True

>>> 5>=5

True

>>> 4>=5

False

>>> 4<5

True

>>> 4<=5

True

>>> 4<=4

True

>>> 4==4

True

>>> 4!=4

False

>>> 4!=3

True

>>> 2>5<10

False

>>> 5>2<10

True

>>> a=5

>>> 1<=a<=10**10

True

>>> ch1='A'

>>> ch2='B'

>>> ord(ch1)

65

>>> ord(ch2)

66

>>> ch1>ch2

False

>>> ch2>ch1

True

>>> ch3='A'

>>> ch4='a'

>>> ord(ch3)

65

>>> ord(ch4)

97
>>> ch4>ch3
True
>>> ch3<ch4
True

Conditional Operators

Conditional operators are ternary operator.

Conditional operator is used to create conditional expression.

Conditional operator evaluates the expression based on condition or test.

Syntax:

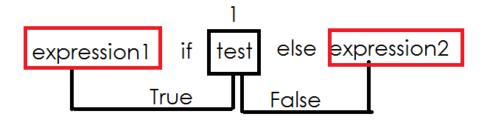
Variable=opr1 if opr2 else opr3

opr2 → condition/test/boolean-expression

opr1 → expression1

opr3 → expression2

if opr2 value is True, PVM evaluates opr1 and assign to variable if opr2 value is False, PVM evaluates opr3 and assign to variable



Example

Voter Elg application

name=input("Enter Name")
age=int(input("Enter Age"))

print(name,"Elg") if age>=18 else print(name,"not Elg")

Output

Enter Namenaresh Enter Age50 naresh Elg

Enter Namesuresh Enter Age15 suresh not Elg

Example:

Write a program to find maximum of two number

num1=int(input("Enter First Number")) # 30 num2=int(input("Enter Second Number")) # 20 num3=num1 if num1>num2 else num2 print("Maximum is ",num3)

Output:

Enter First Number30 Enter Second Number10 Maximum is 30

Enter First Number10
Enter Second Number10
Maximum is 10

Logical Operators

Logical operators are used to combine two or more boolean expressions or conditions or test. Logical operators are represented using keywords.

Operator	Description
and	and operator (binary)
or	or operator (binary)
not	not operator (unary)