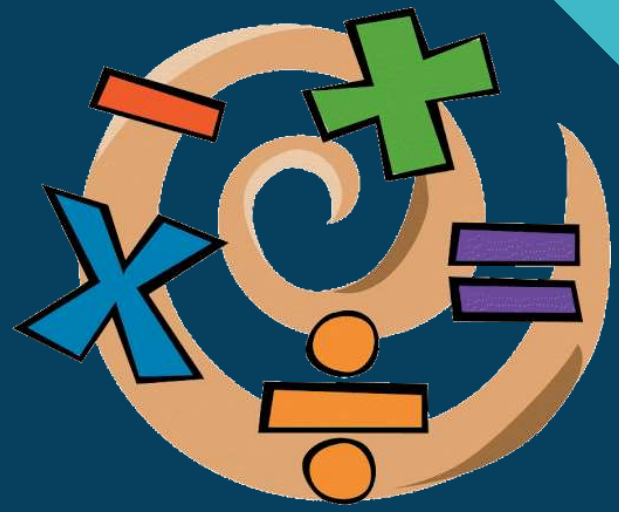
Operators in Python

Prof. Kriti Jaiswal
Assistant Professor
Computer Science & Engineering
School of Engineering & Technology

NAVRACHANA UNIVERSITY
a UGC recognized University

Table of Contents

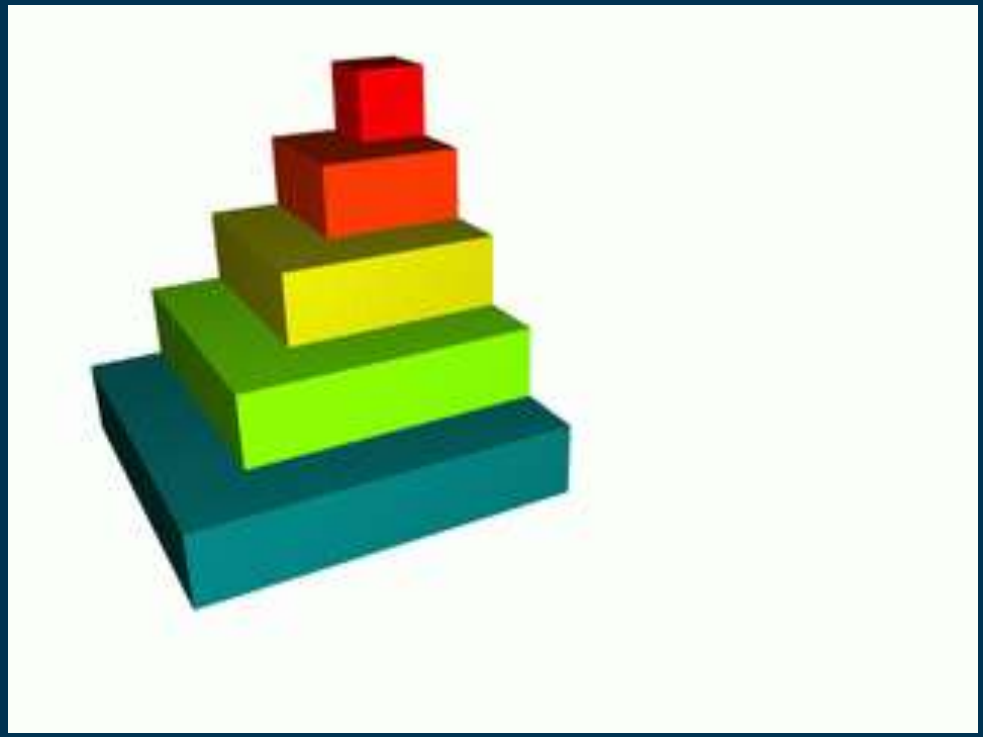


Operators

NAVRACHANA UNIVERSITY
a UGC recognized University

- Symbol that performs an operation.
- An operator acts on some variables are **operands**.

1. Unary Operator
2. Binary Operator
3. Ternary Operator



Arithmetic Operators

Operator	Meaning
+	Addition Operator
-	Subtraction Operator
*	Multiplication Operator
/	Division Operator
%	Modulus Operator
**	Exponent Operator
//	Integer Division or Floor Division

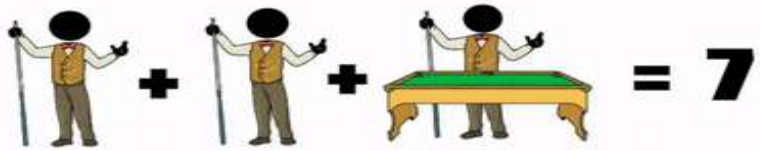
Table 1

Arithmetic Operators

NAVRACHANA UNIVERSITY
a UGC recognized University

Order of Evaluation

1. Parenthesis
2. Exponentiation
3. Multiplication, Division, modulus and floor divisions are at equal priority.
4. Addition and Subtraction
5. Assignment Operator



Assignment Operators

NAVRACHANA UNIVERSITY
a UGC recognized University

- These operators are useful to store the right side value into a left side variable.

Operator	Meaning
=	Assignment Operator
+=	Addition Assignment Operator : x+=y i.e. x=x+y
-+	Subtraction Assignment Operator
*=	Multiplication Assignment Operator
/=	Division Assignment Operator
%=	Modulus Assignment Operator
**=	Exponentiation Assignment Operator
//=	Floor Division Assignment Operator

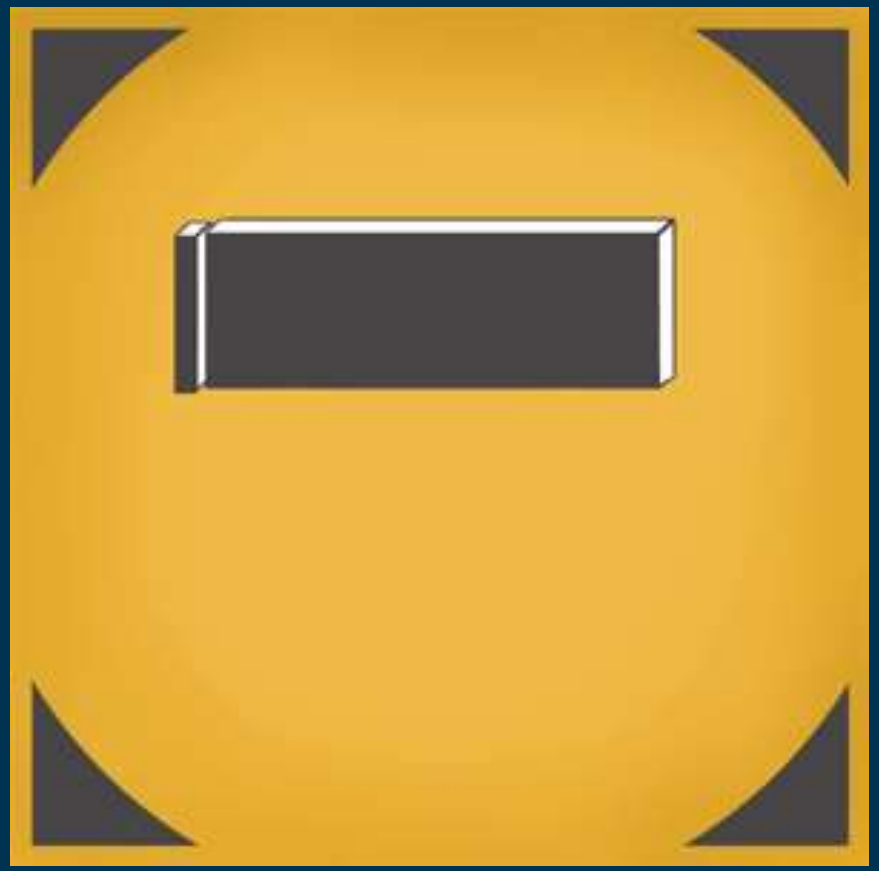
Table 2

NOTE: Python does not have increment operator (++) and decrement operator (--).

Unary Minus Operator

NAVRACHANA UNIVERSITY
a UGC recognized University

- Denoted by the symbol minus (-).
- When this operator is used before a variable, its value is negated.
- That means if the variable value is positive, it will be converted into negative and vice-versa.



Relational Operators

NAVRACHANA UNIVERSITY
a UGC recognized University

- Used to compare two quantities.
- These operators will result in True or False depending on the values compared.

Operator	Meaning
>	Greater than operator
>=	Greater than or equal operator
<	Less than operator
<=	Less than or equal operator
==	Equals Operator
!=	Not Equals Operator

Table 3

Logical Operators

NAVRACHANA UNIVERSITY
a UGC recognized University

- Useful to construct compound conditions.
- A compound condition is a combination of more than one simple condition.
- Each of the simple condition is evaluated to True or False.

1. **And** – True if both the operands are true.
2. **Or** – True of either of the operands is true.
3. **Not** – True if operand is false.

Boolean Operators

NAVRACHANA UNIVERSITY
a UGC recognized University

- **True or False**.
- Boolean operators act upon ‘bool’ type literals.
- They provide ‘bool’ type output.

1. **Boolean And Operator** – If both a and y are True, then it returns True, otherwise False.
2. **Boolean Or Operator** – If either a or y is True, then it returns True, else False.
3. **Boolean Not Operator** – If x is True, it returns False, else True.

Bitwise Operators

NAVRACHANA UNIVERSITY
a UGC recognized University

- Acts on individual bits (0 or 1).
- Use bitwise operators directly on binary numbers or on integer also.

1. Bitwise Complement Operator (~)
2. Bitwise AND Operator (&)
3. Bitwise OR Operator (|)
4. Bitwise XOR Operator (^)
5. Bitwise Left Shift Operator (<<)
6. Bitwise Right Shift Operator (>>)

Membership Operators

- Useful to test for membership in a sequence.

1. **in:** True if value is found in the sequence, otherwise False.
2. **not in:** True if the value is not found in the sequence, otherwise False.

Identity Operator

NAVRACHANA UNIVERSITY
a UGC recognized University

- Compare the memory locations of two objects.
- The memory location of the objet can be seen using the **id() function**.
- id() function returns identity number that internally represents the memory location of the object.

1. **Is**: True if the operands are identical; otherwise False.
2. **Is not**: True if the operand is not identical; otherwise False.

For your Practice

NAVRACHANA UNIVERSITY
a UGC recognized University

1. Python program to demonstrate arithmetic operators.
2. Python program to demonstrate assignment operators.
3. Python program to demonstrate comparison operators.
4. Python program to demonstrate logical operators.
5. Python program to demonstrate all bitwise operators.
6. Python program to demonstrate membership operators.
7. Python program to demonstrate identity operators.