Module 1k: Introduction to Problem Solving and Python Fundamentals

Premanand S

Assistant Professor,
School of Electronics and Engineering,
Vellore Institute of Technology, Chennai

premanand.s@vit.ac.in

July 19, 2025

What are Operators in Python?

- Operators are **special symbols or keywords** used to perform operations on variables and values.
- Python supports several categories of operators to handle arithmetic, logic, comparisons, etc.

Why Do We Need Operators?

- To perform **calculations**, **compare values**, and **build logical expressions**.
- Operators help create meaningful expressions like:
  - `a + b`, `a > b`, `not True`, `a  b`
- They are essential in all decision-making and computation-based programs.

Types of Operators in Python

1. Arithmetic Operators
2. Assignment Operators
3. Comparison (Relational) Operators
4. Logical Operators
5. Bitwise Operators
6. Identity Operators
7. Membership Operators

1. Arithmetic Operators

Operator	Name	Example
+	Addition	5 + 2 = 7
-	Subtraction	5 - 2 = 3
*	Multiplication	5 * 2 = 10
/	Division	5 / 2 = 2.5
//	Floor Division	5 // 2 = 2
%	Modulus	5 % 2 = 1
**	Exponent	2 ** 3 = 8

Arithmetic Operators

Example (Understanding)

```
# Define two numbers
a = 10
b = 3.5

# Perform arithmetic operations
print("a =", a)
print("b =", b)
print("Addition (a + b):", a + b)
print("Subtraction (a - b):", a - b)
print("Multiplication (a * b):", a * b)
print("Division (a / b):", a / b)
print("Floor Division (a // b):", a // b)
print("Modulus (a % b):", a % b)
print("Exponentiation (a ** b):", a ** b)
```

2. Assignment Operators

Operator	Meaning	Equivalent
=	Assign	x = 5
+=	Add and Assign	x += 2 (x = x + 2)
-=	Subtract and Assign	x -= 2
*=	Multiply and Assign	x *= 3
/=	Divide and Assign	x /= 2
//=	Floor Divide and Assign	x //= 2
%=	Modulus and Assign	x %= 2
**=	Exponent and Assign	x **= 2

Assignment Operators

Example (Understanding)

```
# Initial value
x = 10
print("Initial x:", x)

# Basic assignment
x = 5
print("x = 5 →", x)

# Add and assign
x += 3  # x = x + 3
print("x += 3 →", x)

# Subtract and assign
x -= 2  # x = x - 2
print("x -= 2 →", x)
```

Assignment Operators

Example (Understanding)

```
# Multiply and assign
x *= 4  # x = x * 4
print("x *= 4 →", x)

# Divide and assign
x /= 3  # x = x / 3
print("x /= 3 →", x)

# Floor divide and assign
x //= 2  # x = x // 2
print("x //= 2 →", x)

# Modulus and assign
x %= 3  # x = x % 3
print("x %= 3 →", x)
```

3. Comparison Operators

Operator	**Description**	**Example**
==	Equal to	5 == 5 (True)
!=	Not equal to	5 != 3 (True)
>	Greater than	5 > 3 (True)
<	Less than	3 < 5 (True)
>=	Greater than or equal to	5 >= 5 (True)
<=	Less than or equal to	3 <= 4 (True)

Comparision Operators

Example (Understanding)

```
# Define two variables
a = 10
b = 5

print("a =", a)
print("b =", b)

# Equal to
print("a == b:", a == b)

# Not equal to
print("a != b:", a != b)

# Greater than
print("a > b:", a > b)
```

Comparision Operators

Example (Understanding)

```
# Less than
print("a < b:", a < b)

# Greater than or equal to
print("a >= b:", a >= b)

# Less than or equal to
print("a <= b:", a <= b)
```

4. Logical Operators

- `and` — True if both statements are true: `(5 > 3 and 2 < 4)`
- `or` — True if one of the statements is true: `(5 > 10 or 2 < 4)`
- `not` — Reverses the result: `not(5 > 3)` is False

Logical Operators

Example (Understanding)

```
a = True
b = False
print("a =", a)
print("b =", b)

# Logical AND
print("a and b:", a and b)

# Logical OR
print("a or b:", a or b)

# Logical NOT
print("not a:", not a)
print("not b:", not b)
```

5. Bitwise Operators

Operator	**Name**	**Example**
&	AND	5 & 3 = 1
|	OR	5||3 = 7
^	XOR	5 ^ 3 = 6
~	NOT	~5 = -6
<<	Left Shift	5 << 1 = 10
>>	Right Shift	5 >> 1 = 2

Bitwise Operators

Example (Understanding)

```
# Define two integers
a = 5   # Binary: 0101
b = 3   # Binary: 0011

print("a =", a, "(", bin(a), ")")
print("b =", b, "(", bin(b), ")")

# Bitwise AND
print("a & b:", a & b)

# Bitwise OR
print("a | b:", a | b)

# Bitwise XOR
print("a ^ b:", a ^ b)
```

Bitwise Operators

Example (Understanding)

```
# Bitwise NOT
print("~a:", ~a)

# Left Shift
print("a << 1:", a << 1)

# Right Shift
print("a >> 1:", a >> 1)
```

6. Identity Operators

- `is` — True if two variables point to the same object in memory.
- `is not` — True if they do not point to the same object.

Identity Operators

Example (Understanding)

```
# Case 1: Same object
a = [1, 2, 3]
b = a

print("a is b:", a is b)
print("a == b:", a == b)

# Case 2: Different object with same content
c = [1, 2, 3]

print("a is c:", a is c)
print("a == c:", a == c)

# Case 3: is not operator
print("a is not c:", a is not c)
```

7. Membership Operators

- `in` — True if a value is present in a sequence.
- `not in` — True if a value is not present.

Membership Operators

Example (Understanding)

```
# Define a list
fruits = ["apple", "banana", "cherry"]

# Using 'in' operator
print("apple in fruits:", "apple" in fruits)
print("grape in fruits:", "grape" in fruits)

# Using 'not in' operator
print("mango not in fruits:", "mango" not in fruits)
print("banana not in fruits:", "banana" not in fruits)

# It works with other sequences too
text = "hello world"
print("'h' in text:", 'h' in text)
print("'z' not in text:", 'z' not in text)
```

Operator Precedence (Order of Evaluation)

- Python follows specific rules to decide the **order** in which parts of expressions are evaluated.
- From highest to lowest precedence:

1. `()` – Parentheses
2. `**` – Exponentiation
3. `+`, `-`, `~` – Unary operators
4. `*`, `/`, `//`, `%`
5. `+`, `-` (binary addition and subtraction)
6. `<<`, `>>` – Bitwise shifts
7. `&`, `^`, `|` – Bitwise operations
8. `==`, `!=`, `>`, `<`, `>=`, `<=`
9. `not`, `and`, `or`

Assignments – Operators in Python

- Take two numbers and perform all arithmetic operations.
- Check whether two numbers are equal or not.
- Use logical operators to check multiple conditions.
- Use bitwise operations on two integers and print the result.
- Evaluate a complex expression and explain the order of execution.
- Compare the behavior of `is` vs == with integers and lists.
- Accept a list of numbers from user, check which are divisible by both 3 and 5 using logical and modulo operators.

Problem3:

Example (Understanding)

Write a program that adds the digits in a 2 digit number.
Ex: if the input was 35, then the output should be 3+5 = 8?

Answer3:

Example (Understanding)

```
two_digit_number = input('type a two digit number')
print(type(two_digit_number))
first_digit = int(two_digit_number[0])
second_digit = int(two_digit_number[1])
result = int(first_digit) + int(second_digit)
print(result)
```

Problem5

Example (Understanding)

```
Write a program that calculates the Body Mass Index (BMI)
from a user, weight and height?

BMI = weight (kg) / height * height (m.m)
```

Number manipulation

Example (Understanding)

```
print(8/3)
print(int(8/3))
print(round(8/3))
print(round(8/3,2))
print(round(2.6666666666,2))
print(8//3)
print(type(8//3))
print(type(8/3))
print(type(4/4))
```

Number manipulation

Example (Understanding)

```
score = 0
print(score)

score = score + 1
print(score)

score += 1
print(score)

score -= 1              # score *= 1    score /= 1
print(score)
```

Number manipulation

Example (Understanding)

```
score = 0
print('Your score is' + score)
print('Your score is' + str(score))
```

f-string

- formatted string data
- A way to embed expressions inside string literals, using curly braces .
- The expressions are evaluated at runtime and formatted using the f prefix before the string.
- F-strings provide a concise and readable way to include variables and expressions within strings.

f-string

Example (Understanding)

```
name = "Python"
age = 33

print(f"Hello, my name is {name} and I am {age} years old.")
```

f-string

Example (Understanding)

```
score = 0
height = 1.8
isWinning = True
print(f'Your score is {score}')
print(f'your score is {score} and height is {height}
         and winning is always {isWinning} ')
```

Problem6

Example (Understanding)

Create a problem using maths and f-strings that tells us how many weeks we have left, if we live until 85 years old? Get the input from user!

```
Output:
You have @@@@ weeks left
```

Solution6

Example (Understanding)

```
age = input("What is your current age?")
age = int(age)
no_of_weeks_per_year = 52
no_of_years_left = 85 - age
no_of_weeks_we_live = no_of_years_left * no_of_weeks_per_year
print(f"You have {no_of_weeks_we_live} weeks left")
```

Quiz3

Example (Try manual)

1. You are a computer. what will this line of code print?

```
print(6+4/2-(1*2))
```

a.3

b.6.0

c.8.0

d.5

Quiz3

Example (Try manual)

```
2. What is the data type of the result of the variable 'a' in
the following line of code?
a = int('5')/int(2.7)
a. int
b. float
c. str
d. bool
```

Quiz3

Example (Try manual)

```
3. Which of the following lines of code gives error?
a. age = 12
   print('you are' + age + 'years old')

b. age = 12
   print(f'you are {age} years old')

c. name = input('whats your name?')
   print('hello' + name)

d. name = input('whats your name?')
   print(f'hello, {name}')
```

Problem7

Example (Understanding)

Write a Python program that calculates the total bill amount per person after adding a tip and GST. The program should prompt the user to enter the total bill amount, the percentage of tip they would like to give (10, 20, or 50/-), and the number of people among whom the bill will be split. The program should then calculate an 18% GST on the original bill amount and add it to the bill along with the tip. Finally, the program should output the total bill amount (including tip and GST) and how much each person needs to pay.

Problem7

Example (Expected Output)

```
Welcome to Tip Calculator!
What was the total bill? 5648
What percentage tip would you like to give? 10, 20, or 50? 50
How many people to split the bill? 5
Total bill amount with tip and GST is Rs: 6714.64/-
Each person should pay Rs: 1342.93/-
```

Answer7

Example (Python snippet)

```
print('Welcome to Tip Calculator!')

# Get the inputs
bill = input('What was the total bill? ')
bill = float(bill)
tip = input('What percentage tip would you like to give?
                                 10, 20, or 50? ')
tip = int(tip)
people = input('How many people to split the bill? ')
people = int(people)

# Calculate the GST amount (18%)
gst = 0.18 * bill
```

Answer7

Example (Python snippet)

```
# Calculate the total bill including tip and GST
bill_with_tip_gst = bill + tip + gst

# Print the total bill amount with tip and GST
print(f'Total bill amount with tip and GST is Rs:
                                {bill_with_tip_gst}/-')

# Calculate how much each person should pay
bill_per_person = bill_with_tip_gst / people
final_amount = round(bill_per_person, 2)

# Print the amount each person should pay
print(f'Each person should pay Rs: {final_amount}/-')
```

Problem8

Example (Understanding)

Write a Python program that calculates the final price of a product after applying a discount and sales tax. The program should prompt the user to enter the original price of the product, the discount percentage, and the sales tax percentage. The program should first apply the discount to the original price and then calculate the sales tax on the discounted price. Finally, it should output the final price after applying both the discount and the tax.

For example, if the discount percentage is 15% and the sales tax percentage is 8%, the program should calculate the correct final price based on these inputs.

Problem8

Example (Expected Output)

```
What is the original price of the product? 12545
What is the discount percentage? 15
What is the sales tax percentage? 8
The final price after discount and tax is Rs: 11516.31/-
```

Answer8

Example (Python snippet)

```
# enter the original price of the product
original_price = float(input("What is the original price
                                    of the product? "))

# enter the discount percentage
discount_percentage = float(input("What is the discount
                                        percentage? "))

# enter the sales tax percentage
sales_tax_percentage = float(input("What is the sales
                                    tax percentage? "))

# Calculate the discount amount
discount_amount = (discount_percentage / 100)
                                        * original_price
```

Answer8

Example (Python snippet)

```
# Calculate the price after discount
price_after_discount = original_price - discount_amount

# Calculate the sales tax on the discounted price
sales_tax = (sales_tax_percentage / 100) *
                                price_after_discount

# Calculate the final price
final_price = price_after_discount + sales_tax
final = round(final_price, 2)

# Output the final price after applying both
print(f'The final price after discount and
                        tax is Rs: {final_price}/-')
```

Problem9

Example (Understanding)

Write a Python program that calculates the total amount to be paid on a loan after adding the interest. The program should prompt the user to enter the principal loan amount, the annual interest rate, and the number of years the loan will be held. The program should calculate the total amount to be paid after applying the interest for the given period. Additionally, it should calculate how much needs to be paid per month.

For example, if the annual interest rate is 9.5% and the loan period is 2 years, the program should accurately calculate the total amount to be paid and the monthly payment.

Problem9

Example (Expected Output)

```
What is the principal loan amount? 500000
What is the annual interest rate (in %)? 9.5
For how many years will the loan be held? 2
Total amount to be paid after 2 years is: 595000.0
Monthly payment is: 24791.666666666668
```

Answer9

Example (Python snippet)

```
# enter the principal loan amount
principal = float(input("What is the principal loan
                                            amount? "))

# enter the annual interest rate (in %)
annual_interest_rate = float(input("What is the annual
                                interest rate (in %)? "))

# enter the number of years the loan will be held
years = int(input("For how many years will
                                    the loan be held? "))

# Calculate the total interest over the entire period
total_interest = (annual_interest_rate / 100) *
                                        principal * years
```

Answer9

Example (Python snippet)

```
# Calculate the total amount to be paid (principal +
                                         total interest)
total_amount = principal + total_interest

# Calculate the monthly payment amount
monthly_payment = total_amount / (years * 12)

# Output the total amount to be paid and the
#monthly payment amount
print(f'Total amount to be paid after {years}
                            years is: {total_amount}')
print(f'Monthly payment is: {monthly_payment}')
```

mail me: er.anandprem@gmail.com / premanand.s@vit.ac.in
ring me: +91 73586 79961
Follow me: Linkedin
Medium Blogs
Analytics Vidhya: Blogs

Don't just code — think, plan, and solve