Python Programming

Unit 01 – Lecture 04: Input/Output, Escape Sequences, Operators, Precedence

Tofik Ali

School of Computer Science, UPES Dehradun

February 14, 2026

Repository: `https://github.com/tali7c/Python-Programming`

Quick Links

Core Concepts Demo Interactive Summary

Agenda

1 Core Concepts

2 Demo

3 Interactive

4 Summary

Learning Outcomes

- Read input and convert to correct types

Learning Outcomes

- Read input and convert to correct types
- Format output using `print()` options and escape sequences

Learning Outcomes

- Read input and convert to correct types
- Format output using `print()` options and escape sequences
- Apply operators: arithmetic, relational, logical, bitwise, membership, identity

Learning Outcomes

- Read input and convert to correct types
- Format output using `print()` options and escape sequences
- Apply operators: arithmetic, relational, logical, bitwise, membership, identity
- Evaluate expressions using precedence and parentheses

Input and Output

- `input(prompt)` reads text from keyboard (returns `str`)

```
a = int(input("Enter a: "))
b = int(input("Enter b: "))
print("a =", a, "b =", b, sep=" | ")
```

Input and Output

- `input(prompt)` reads text from keyboard (returns `str`)
- `print(...)` displays output

```
a = int(input("Enter a: "))
b = int(input("Enter b: "))
print("a =", a, "b =", b, sep=" | ")
```

Input and Output

- input(prompt) reads text from keyboard (returns str)
- print(...) displays output
- Useful print options: sep, end

```
a = int(input("Enter a: "))
b = int(input("Enter b: "))
print("a =", a, "b =", b, sep=" | ")
```

Escape Sequences

- New line: \n

```
print("Line1\nLine2")
print("A\tB\tC")
print("C:\\\\Users\\\\tofik.ali")
```

Escape Sequences

- New line: \n
- Tab: \t

```
print("Line1\nLine2")
print("A\tB\tC")
print("C:\\\\Users\\\\tofik.ali")
```

Escape Sequences

- New line: `\n`
- Tab: `\t`
- Backslash: `\\`

```
print("Line1\nLine2")
print("A\tB\tC")
print("C:\\\\Users\\\\tofik.ali")
```

Escape Sequences

- New line: \n
- Tab: \t
- Backslash: \\
- Quotes inside strings: \" or \’

```
print("Line1\nLine2")
print("A\tB\tC")
print("C:\\\\Users\\\\tofik.ali")
```

Operator Categories

Category	Examples
Arithmetic	+ - * / % // **
Relational	< <= > >= == !=
Logical	and or not
Bitwise	& | ^ ~ << >>
Membership	in, not in
Identity	is, is not

Precedence (Why Parentheses Matter)

- Multiplication happens before addition

```
print(2 + 3 * 4)      # 14
print((2 + 3) * 4)    # 20
```

Precedence (Why Parentheses Matter)

- Multiplication happens before addition
- Use parentheses to make intention clear

```
print(2 + 3 * 4)     # 14
print((2 + 3) * 4)   # 20
```

Membership and Identity

- Membership: check presence in a sequence/collection

```python
nums = [10, 20, 30]
print(20 in nums)  # True

a = [1, 2]
b = a
c = [1, 2]
print(a is b)      # True (same object)
print(a is c)      # False (different objects)
```

Membership and Identity

- Membership: check presence in a sequence/collection
- Identity: check if two names refer to the same object

```
nums = [10, 20, 30]
print(20 in nums)  # True

a = [1, 2]
b = a
c = [1, 2]
print(a is b)      # True (same object)
print(a is c)      # False (different objects)
```

Demo: Operator Precedence + Truth Tables

- File: `demo/operator_precedence_playground.py`

Demo: Operator Precedence + Truth Tables

- File: `demo/operator_precedence_playground.py`
- Prints examples for:

Demo: Operator Precedence + Truth Tables

- File: `demo/operator_precedence_playground.py`
- Prints examples for:
  - precedence surprises

Demo: Operator Precedence + Truth Tables

- File: `demo/operator_precedence_playground.py`
- Prints examples for:
  - precedence surprises
  - bitwise truth table (&, —, ^)

Demo: Operator Precedence + Truth Tables

- File: `demo/operator_precedence_playground.py`
- Prints examples for:
  - precedence surprises
  - bitwise truth table (&, —, ˆ)
  - membership and identity checks

Checkpoint 1

Question: Evaluate and explain:

- `2 + 3 * 4`
- `(2 + 3) * 4`

Checkpoint 2

Question: What is the difference between == and `is`?

Think-Pair-Share

Discuss:

- Why can wrong input types break programs?
- How would you validate age so that it is a non-negative integer?

Key Takeaways

- Convert input strings to numbers using `int`/`float`

Key Takeaways

- Convert input strings to numbers using `int`/`float`
- Escape sequences help print clean output

Key Takeaways

- Convert input strings to numbers using `int`/`float`
- Escape sequences help print clean output
- Know operator categories and choose the right one

Key Takeaways

- Convert input strings to numbers using `int/float`
- Escape sequences help print clean output
- Know operator categories and choose the right one
- Precedence matters; parentheses improve correctness and readability

Exit Question

Give one example each:

- membership operator
- identity operator