OPERATORS AND EXPRESSIONS

Course: Python Programming

Section: Python Basics

Topic: Operators and Expressions

📌 What You Will Learn

- What operators are
- Different types of operators in Python
- What expressions are
- How Python evaluates expressions

---

1. What is an Operator?

An **operator** is a symbol that performs an operation on one or more values.

Operators are used to:

- Perform calculations
- Compare values
- Combine conditions
- Make decisions in programs

Operators work together with variables and values to produce results.

---

2. Arithmetic Operators

Arithmetic operators are used to perform mathematical calculations.

Operator	Description
`+`	Addition

Operator	Description
`-`	Subtraction
`*`	Multiplication
`/`	Division
`%`	Modulus (remainder)
`//`	Floor division
`**`	Exponentiation

These operators are commonly used in calculations and formulas.

---

3. Comparison Operators

Comparison operators compare two values and return a **Boolean result** (`True` or `False`).

Operator	Description
`==`	Equal to
`!=`	Not equal to
`>`	Greater than
`<`	Less than
`>=`	Greater than or equal to
`<=`	Less than or equal to

Comparison operators are mostly used in conditions and decision-making.

---

4. Logical Operators

Logical operators are used to combine multiple conditions.

Operator	Description
`and`	True if both conditions are true
`or`	True if at least one condition is true
`not`	Reverses the result

Logical operators help build complex decision logic.

5. What is an Expression?

An **expression** is a combination of:

- Values
- Variables
- Operators

An expression is evaluated to produce a **single result**.

Examples of expressions:

- Arithmetic expression
- Comparison expression
- Logical expression

Expressions are used everywhere in Python programs.

---

6. Order of Evaluation (Operator Precedence)

Python follows a specific order when evaluating expressions.

General precedence:

1. Parentheses `()`
2. Exponentiation `*`
3. Multiplication, Division, Modulus
4. Addition and Subtraction
5. Comparison operators
6. Logical operators

Understanding precedence prevents logical and calculation errors.

---

7. Common Mistakes with Operators and Expressions

Beginners often face issues such as:

- Confusing `=` with `==`
- Ignoring operator precedence
- Mixing data types in expressions
- Incorrect use of logical operators

Careful use of operators ensures correct program output.

📝 Summary

- Operators perform actions on values
- Python supports arithmetic, comparison, and logical operators
- Expressions combine values and operators
- Python evaluates expressions using operator precedence
- Correct usage avoids logical errors