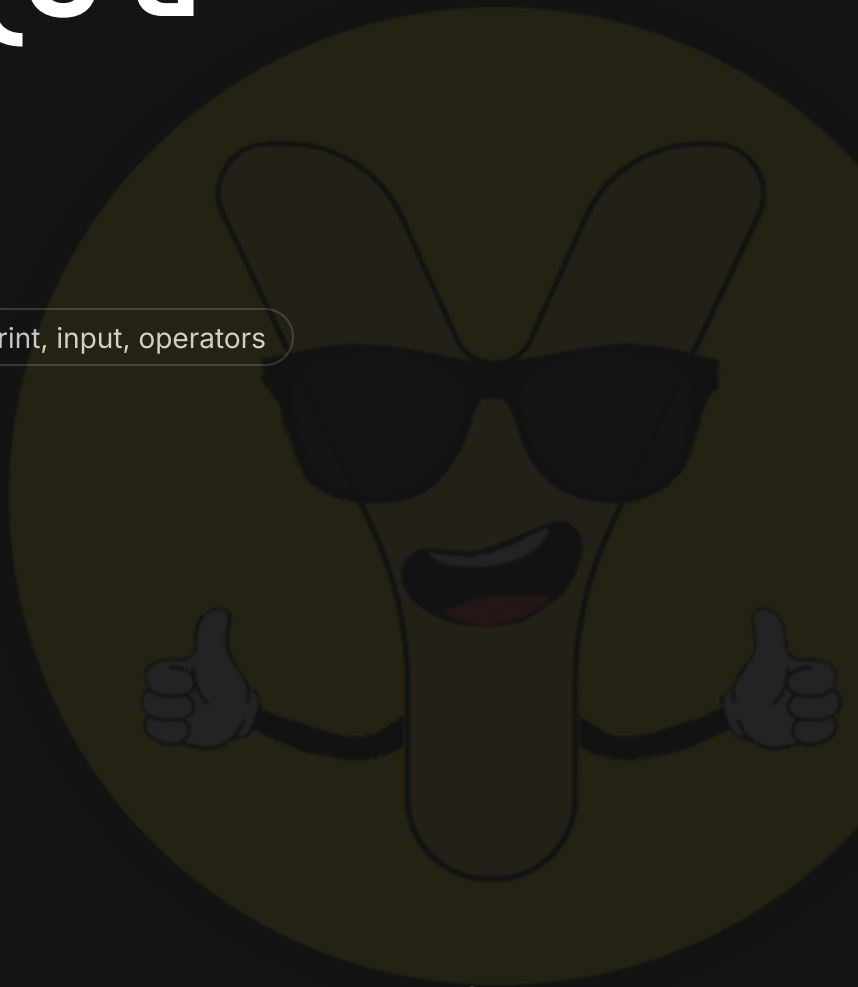
AKTU · 1ST YEAR · SEMESTER 1 · ACS101

Programming Languages (C & Python)

Unit 4 — Python Introduction

- Why Python — where it is used (AI, Data Science, Web, Automation)
- Python basics — variables, data types, print, input, operators
- Python control flow — if/else, for loop, while loop
- Python functions — def, return, default arguments
- Real connect: Python is #1 in AI/ML, Data Science, and automation — same syntax, more power, less boilerplate.



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Unit 4 Overview: Python Introduction

This unit introduces Python, a high-level, interpreted programming language known for its simplicity and vast ecosystem.

- Understanding the philosophy of Python and why it is widely adopted.
- Core syntax, variables, data types, and basic I/O operations.
- Control flow mechanisms including conditional statements and loops.
- Modular programming using functions, default arguments, and return values.
- Real-world relevance in AI, Data Science, and automation.

What is Python?

PYTHON PROGRAMMING LANGUAGE

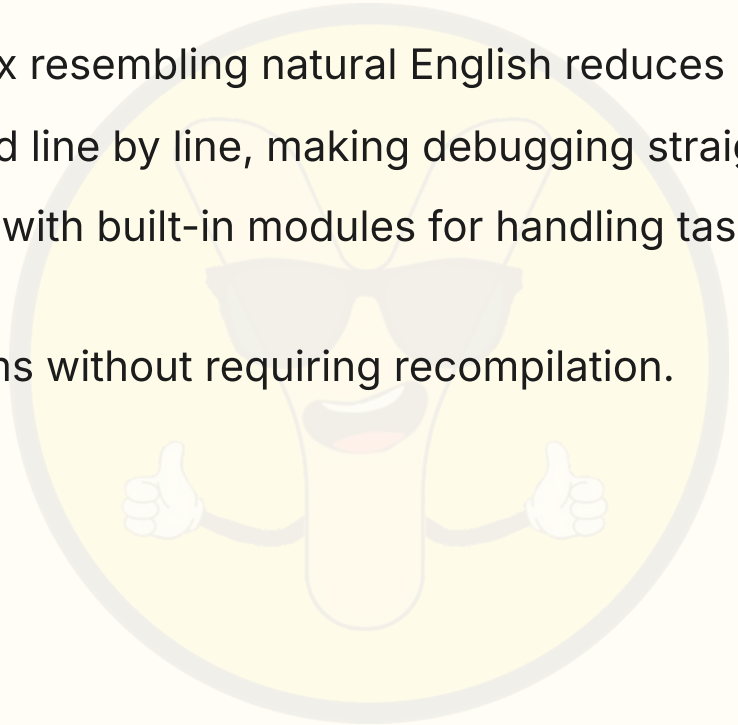
An interpreted, high-level, general-purpose programming language created by Guido van Rossum and first released in 1991.

- Emphasizes code readability with the notable use of significant whitespace.
- Dynamically typed and garbage-collected memory management.
- Supports multiple programming paradigms including procedural, object-oriented, and functional programming.

Why Python? Key Characteristics

Python has become a dominant language in software development due to several defining features.

- Easy to learn and read: Clean syntax resembling natural English reduces maintenance cost.
- Interpreted nature: Code is executed line by line, making debugging straightforward.
- Extensive Standard Library: Comes with built-in modules for handling tasks from web protocols to mathematics.
- Portability: Runs on various platforms without requiring recompilation.



Python Application Domains

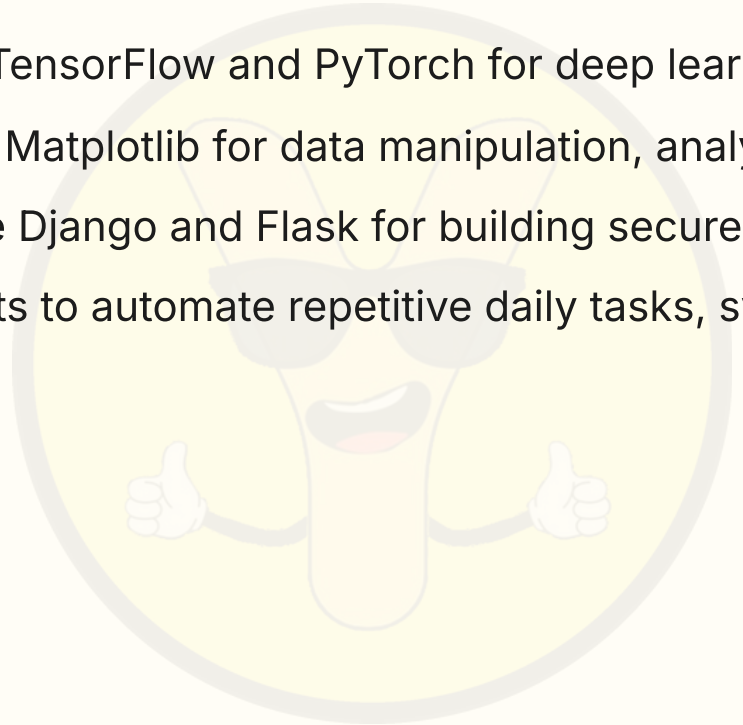
- Artificial Intelligence and Machine Learning
- Data Science and Big Data Analytics
- Web Development and APIs
- Task Automation and Scripting

Primary application domains where Python dominates modern industry requirements.

Where Python is Used

Python powers applications across diverse industrial sectors.

- Artificial Intelligence: Libraries like TensorFlow and PyTorch for deep learning and neural networks.
- Data Science: Pandas, NumPy, and Matplotlib for data manipulation, analysis, and visualization.
- Web Development: Frameworks like Django and Flask for building secure and scalable web applications.
- Automation & Testing: Writing scripts to automate repetitive daily tasks, system administration, and software testing.



Python Variables

VARIABLE

A named memory location used to store data values in a program.

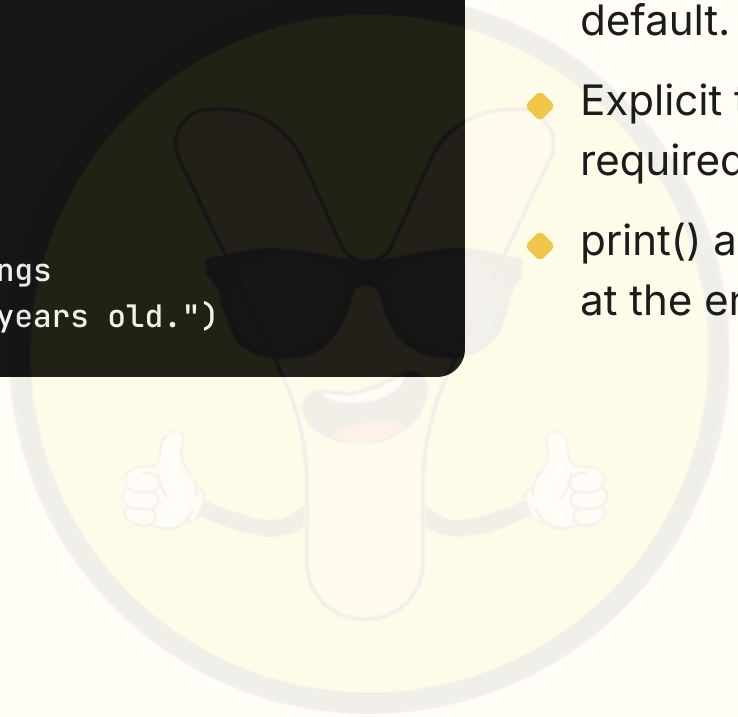
- Unlike C, Python has no command for declaring a variable; it is created the moment you first assign a value to it.
- Variable names are case-sensitive and must start with a letter or an underscore.
- Python uses dynamic typing, meaning the type of a variable is determined at runtime based on the assigned value.

Python Built-in Data Types

Data Type	Description	Example
int	Whole numbers without decimals	x = 10
float	Numbers with decimal points	pi = 3.14
str	Sequence of characters	name = "AKTU"
bool	Boolean truth values	is_passed = True
list	Ordered, mutable collection	data = [1, 2, 3]

Basic I/O: print() and input()

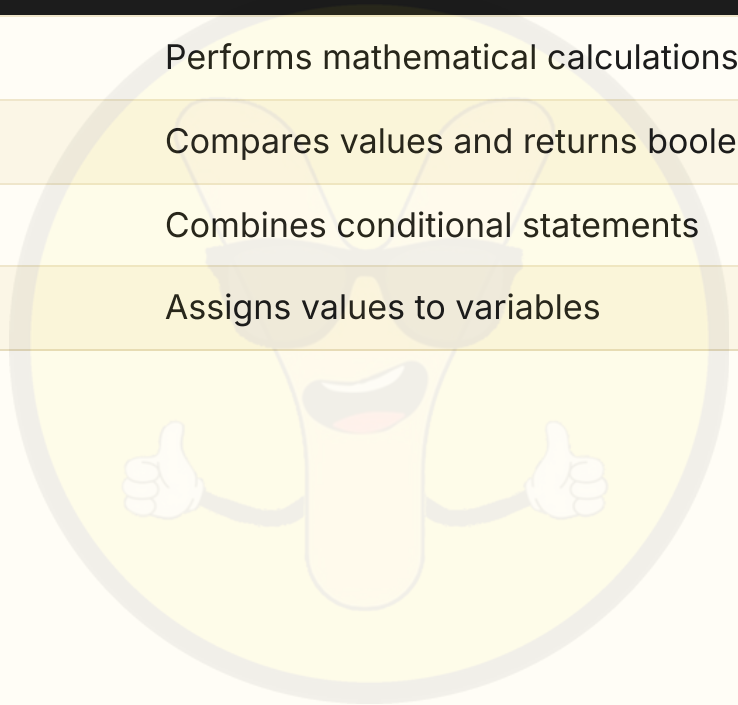
```
# Taking input from user
name = input("Enter your name: ")

# Type casting input string to integer
age = int(input("Enter your age: "))

# Printing output using formatted strings
print("Hello", name, "You are", age, "years old.")
```



- input() always returns data of string type by default.
- Explicit type casting (like int() or float()) is required for numerical operations.
- print() automatically adds a newline character at the end by default.

Python Operators

Category	Operators	Description
Arithmetic	+, -, *, /, //, %, **	Performs mathematical calculations (// is floor division)
Comparison	==, !=, >, <, >=, <=	Compares values and returns boolean result
Logical	and, or, not	Combines conditional statements
Assignment	=, +=, -=, *=	Assigns values to variables

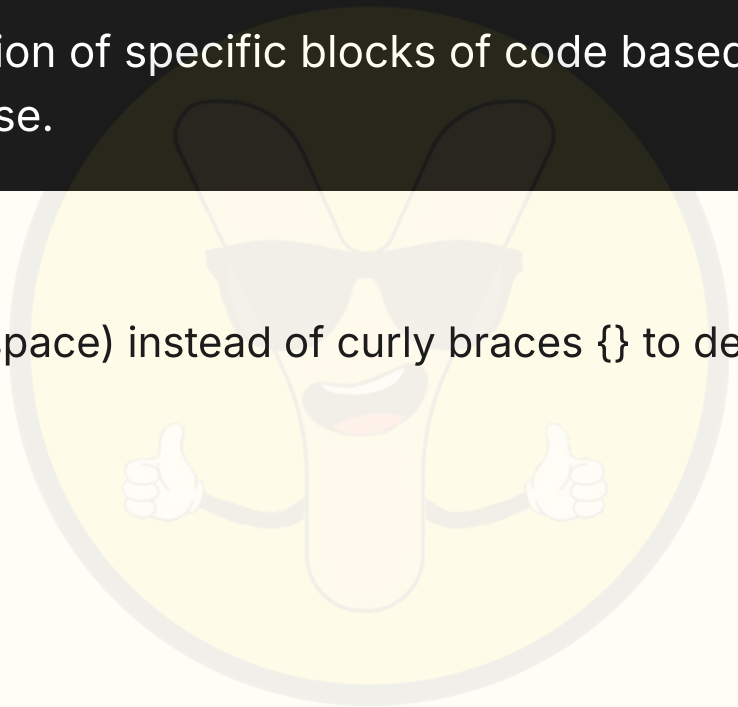


Control Flow: Conditional Statements

DECISION MAKING

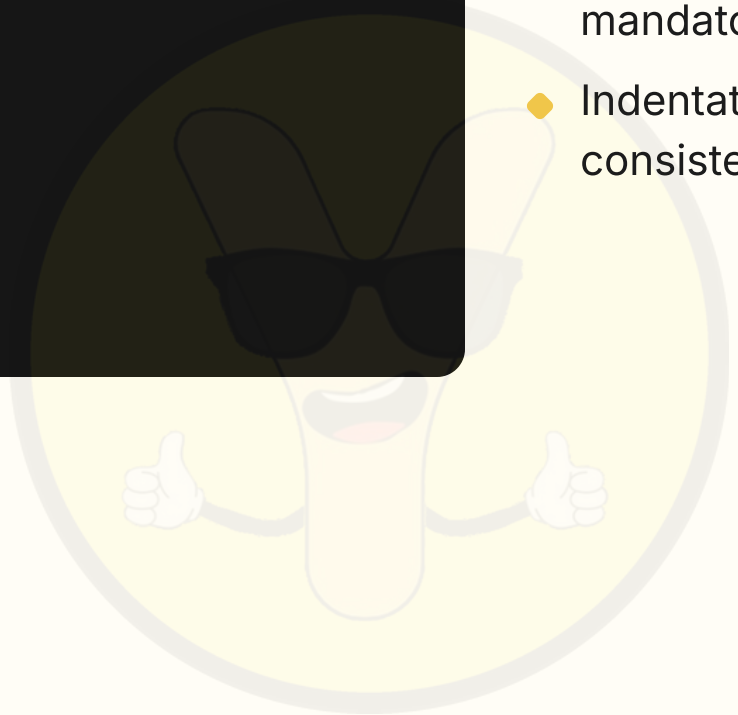
Mechanisms that allow the execution of specific blocks of code based on whether a boolean expression evaluates to true or false.

- Uses if, elif, and else keywords.
- Python relies on indentation (whitespace) instead of curly braces {} to define code blocks.



Conditional Statements Syntax

```
marks = int(input("Enter marks: "))

if marks ⩾ 75:
    print("Grade: Distinction")
elif marks ⩾ 50:
    print("Grade: First Division")
else:
    print("Grade: Needs Improvement")
```



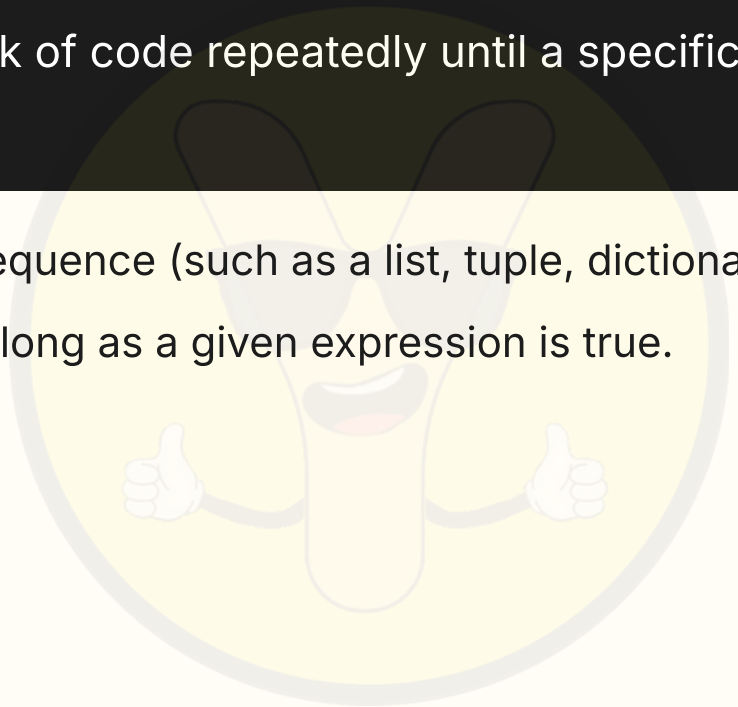
- The colon (:) at the end of the if/elif/else line is mandatory.
- Indentation inside the block must be consistent (typically 4 spaces).

Control Flow: Loops

ITERATIVE STATEMENTS

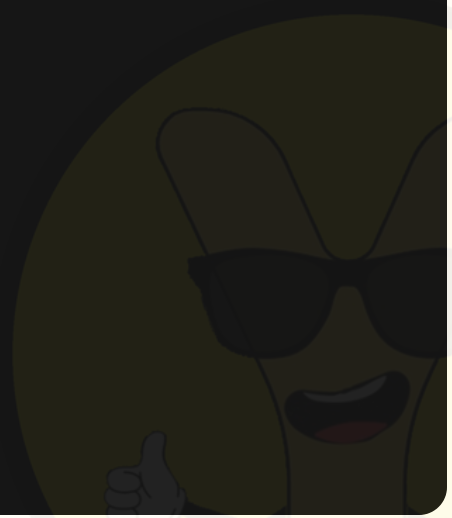
Constructs used to execute a block of code repeatedly until a specific condition is met or through a sequence.

- for loop: Used for iterating over a sequence (such as a list, tuple, dictionary, set, or string).
- while loop: Executes statements as long as a given expression is true.



For and While Loops

```
# Using for loop with range()
print("For loop output:")
for i in range(1, 4):
    print(i)

# Using while loop
print("While loop output:")
count = 3
while count > 0:
    print(count)
    count -= 1
```



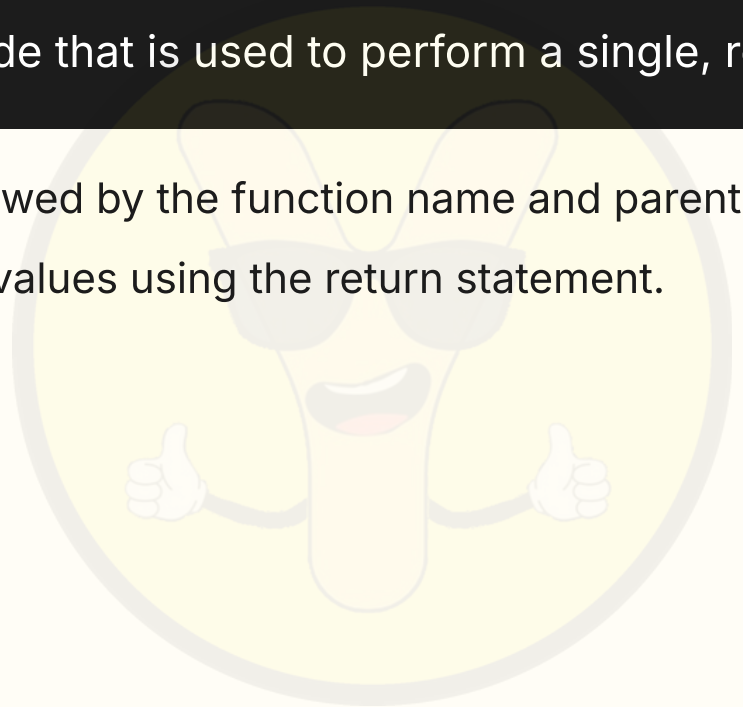
- range(start, stop) generates numbers starting from start up to stop-1.
- Infinite loops occur if the loop condition in a while loop never becomes false.

Python Functions

FUNCTION DEFINITION

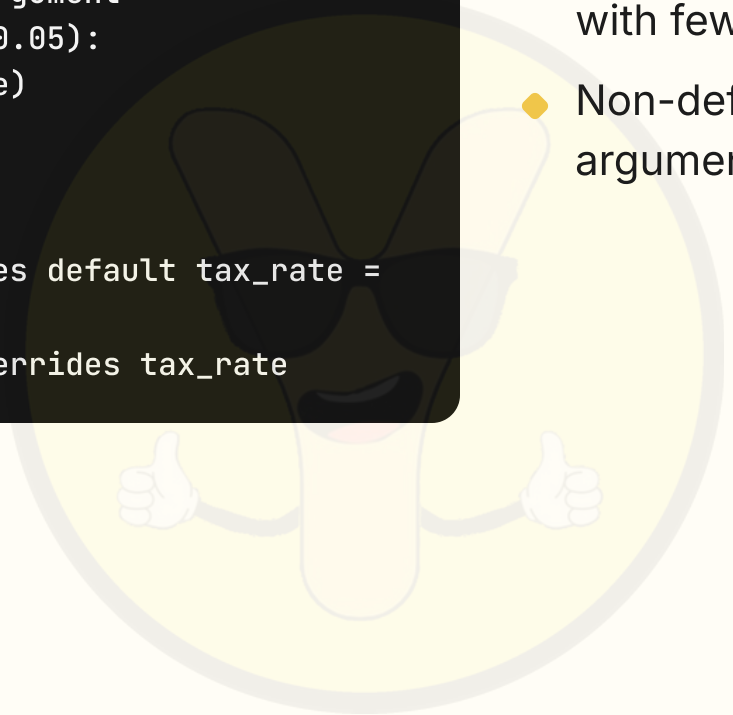
A block of organized, reusable code that is used to perform a single, related action.

- Defined using the def keyword followed by the function name and parentheses ().
- Can accept parameters and return values using the return statement.



Functions with Default Arguments

```
# Function definition with a default argument
def calculate_bill(amount, tax_rate = 0.05):
    total = amount + (amount * tax_rate)
    return total

# Function calls
print(calculate_bill(1000))       # Uses default tax_rate =
0.05
print(calculate_bill(1000, 0.12)) # Overrides tax_rate
```

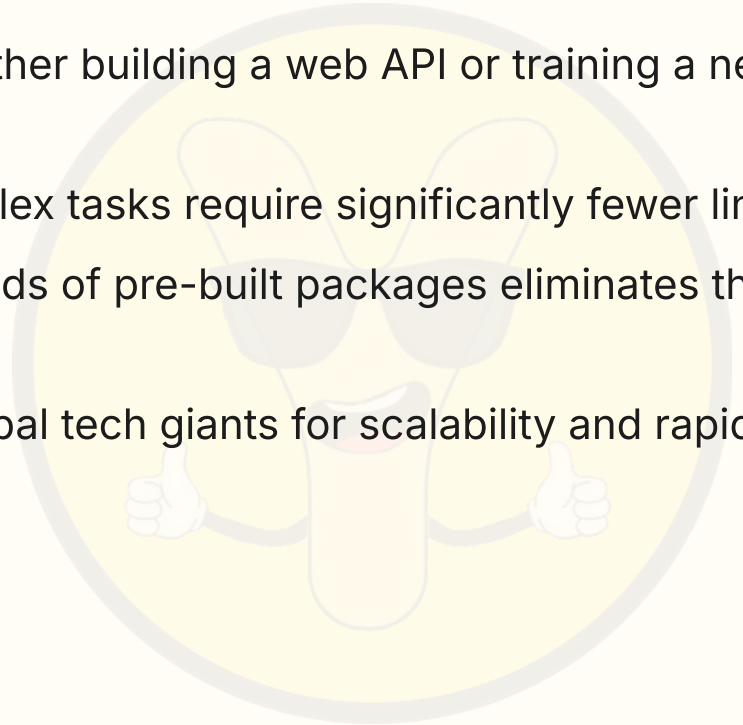


- Default arguments allow functions to be called with fewer arguments than defined.
- Non-default arguments cannot follow default arguments in the function definition.

Real Connect: Why Python is #1

Python leads the technological landscape in AI, Machine Learning, Data Science, and automation.

- Same syntax across domains: Whether building a web API or training a neural network, the base syntax remains identical.
- More power, less boilerplate: Complex tasks require significantly fewer lines of code compared to C or Java.
- Rich ecosystem: Access to thousands of pre-built packages eliminates the need to write algorithms from scratch.
- Industry standard: Preferred by global tech giants for scalability and rapid prototyping.



Exam corner

- **Q1** Define Python. Explain its key features and why it is preferred for Artificial Intelligence and Data Science.
- **Q2** Differentiate between dynamically typed and statically typed languages with suitable examples.
- **Q3** Explain the significance of indentation in Python control flow statements.
- **Q4** Write a Python program to check whether a given number is prime or not.
- **Q5** Explain the working of for loop and while loop with code examples.
- **Q6** What are default arguments in Python functions? Explain with a programming example.
- **Q7** Compare Python and C language based on syntax complexity, execution speed, and memory management.

Unit 4 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 5 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

