Introduction

Python is a high-level programming language and is widely being used among the developers' community. Python was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets developers work quickly and integrate systems more efficiently.

It's simple to use, packed with features and supported by a wide range of libraries and frameworks. Its clean syntax makes it beginner-friendly.

Advantages of Learning Python

- Requires fewer lines of code compared to other programming languages like Java
- Provides frameworks such as Django and Flask for web development
- Offers libraries like Pandas, TensorFlow, and Scikit-learn for AI and data science
- Cross-platform and works on Windows, macOS, and Linux
- Used by companies such as Google, Netflix, and NASA
- Strong job opportunities in software development, data science, and AI

Indentation in Python

In Python, Indentation is used to define blocks of code. It indicates to the Python interpreter that a group of statements belongs to the same block. All statements with the same level of indentation are treated as part of the same code block. Indentation is created using whitespace at the beginning of each line, and the commonly accepted convention is to use four spaces per indentation level.

Print Function

At its core, printing output in Python is straightforward, thanks to the print() function. This function allows us to display text, variables and expressions on the console.

Variables

variables are used to store data that can be referenced and manipulated during program execution. A variable is essentially a name that is assigned to a value.

Python Operators

Operators in general are used to perform operations on values and variables.

Operators

Operators: Special symbols like -, + , * , /, etc.

Operands

Operands: Value on which the operator is applied.

Keyword

Keywords in Python are special reserved words that are part of the language itself. They define the rules and structure of Python programs which means you cannot use them as names for your variables, functions, classes or any other identifiers.

Data Type
Data types in Python are a way to classify data items. They represent the kind of value, which determines what operations can be performed on that data. Since everything is an object in Python programming, Python data types are classes and variables are instances (objects) of these classes.

Conditional Statements in Python
Conditional statements in Python are used to execute certain blocks of code based on specific conditions. These statements help control the flow of a program, making it behave differently in different situations.

Loops in Python
Loops in Python are used to repeat actions efficiently. The main types are For loops (counting through items) and While loops (based on conditions).

Functions
In this section of Python 3 tutorial we'll explore Python function syntax, parameter handling, return values and variable scope. Along the way, we'll also introduce versatile functions like range(), map, filter and lambda functions.

Python Functions
Python Functions are a block of statements that does a specific task. The idea is to put some commonly or repeatedly done task together and make a function so that instead of writing the same code again and again for different inputs, we can do the function calls to reuse code contained in it over and over again.

Python pass Statement
The pass statement in Python is a placeholder that does nothing when executed.
It is used to keep code blocks valid where a statement is required but no logic is needed yet. Examples situations where pass is used are empty functions, classes, loops or conditional blocks.

Global and Local Variables in Python
variables play a key role in storing and managing data. Their behavior and accessibility depend on where they are defined in the program. In this article, we'll explore global and local variables

Data Structures
Python offers versatile collections of data types, including lists, string, tuples, sets, dictionaries and arrays. In this section, we will learn about each data types in detail.

- Strings
- List
- Tuples
- Dictionary
- Sets
- Arrays

- List Comprehension

Python's collections module offers essential data structures, including the following

- Counters
- Heapq
- Deque
- OrderedDict
- Defaultdict

OOP Concepts

In this section, we'll explore the core principles of object-oriented programming (OOP) in Python. From encapsulation to inheritance, polymorphism, abstract classes and iterators, we'll cover the essential concepts that helps you to build modular, reusable and scalable code.

Python OOP

Object Oriented Programming is a fundamental concept in Python, empowering developers to build modular, maintainable and scalable applications.

OOP is a way of organizing code that uses objects and classes to represent real-world entities and their behavior. In OOP, object has attributes thing that has specific data and can perform certain actions using methods.

- Classes and Objects
- Polymorphism
- Inheritance
- Abstraction
- Encapsulation
- Iterators