

[Python Course](#)[Python Tutorial](#)[Interview Questions](#)[Python Quiz](#)[Python Glossary](#)[Python Projects](#)**Certified Courses****Data
Science****Data
Analytics****Machine
Learning****Earn 3 Certifi****Learn More →**

Python Exercise with Practice Questions and Solutions

Last Updated : 26 Sep, 2024



Python Exercise for Beginner: Practice makes perfect in everything, and this is especially true when learning Python. If you're a beginner, regularly practicing Python exercises will build your confidence and sharpen your skills. To help you improve, try these Python exercises with solutions to test your knowledge.

You might have seen various [Python tutorials](#) that explain the concepts in detail but that might not be enough to get hold of this language. The best way to learn is by practicing it more and more.

List of Python Programming Exercises

In the below section, we have gathered **chapter-wise Python exercises with solutions**. So, scroll down to the relevant topics and try to solve the Python program practice set.

Table of Content

- [Conditional Statement Exercises](#)
- [List Exercises](#)
- [String Exercises](#)
- [Tuple Exercises](#)
- [Dictionary Exercises](#)
- [Set Exercises](#)
- [Matrix Exercises](#)
- [Functions Exercises](#)
- [OOPS Exercises](#)
- [Regex Exercises](#)

- [LinkedList Exercises](#)
- [Searching Exercises](#)
- [Sorting Exercises](#)
- [DSA Exercises](#)
- [File Handling Exercises](#)
- [JSON Exercises](#)

To complement the Python exercises, you can further test your skills on the [GeeksforGeeks Practice Platform](#), which includes numerous Python-related challenges and solutions.



Python Practice Exercises

Let's start with the **Conditional Statement Exercise**.

Python Conditional Statement Exercises

In this Section, we will explore various exercises involving [Python's conditional statements](#) and loops, providing you with hands-on experience and practical examples to enhance your coding skills.

- [Mark Even and Odd](#)
- [Check the status of Integer](#)
- [Trouble with Angry Friends](#)
- [Check for Equal Occurrences of 'cat' and 'hat' in a String](#)
- [Conditional Boolean Evaluation](#)
- [Print Characters at Even Indices in a String](#)
- [Print Numbers in Decreasing Order](#)
- [Jumping through While](#)
- [Zero Converter](#)

Programs

- [Check if a Number is Odd or Even](#)
- [Program to Print the Natural Numbers](#)
- [Check Prime Number](#)
- [Printing Patterns in Python](#)

Python List Exercises

In this section, we have compiled a variety of [Python list](#) exercises that range from basic to advanced levels. These exercises are designed to help you master the fundamentals of Python lists, including list creation, indexing, slicing, and common operations like appending, inserting, and removing elements.

- [Min and Max in Array](#)
- [Union of two arrays](#)
- [Intersection of two sorted arrays](#)

Programs

- [Maximum of two numbers in Python](#)
- [Minimum of two numbers in Python](#)
- [Interchange first and last elements in a list](#)
- [Swap two elements in a list](#)
- [Ways to find length of list](#)

>> [More Programs on List](#)

Python String Exercises

In this section, we provide a variety of [Python string](#) exercises designed to enhance your understanding of string manipulation and operations. Explore these exercises and upsacle your Python programming skills.

- [Exploring String Methods](#) (swapcase, find, strip...)
- [Exploring String Methods](#) (upper, lower, startswith...)
- [Convert String to LowerCase](#)
- [Reverse Words](#)
- [String Validation and Formatting](#)

- [Slicing in String](#)
- [Repeat the Strings](#)
- [Palindrome String](#)
- [Remove all duplicates from a given string](#)
- [Check Anagram String](#)

Programs

- [Reverse words in a given String](#)
- [Find length of a string](#)
- [Print even length words in a string](#)
- [Remove i'th character from string](#)
- [Check whether the string is Symmetrical or Palindrome](#)

>> [More Programs on String](#)

Python Tuple Exercises

Here, you'll find a range of [Python tuple](#) exercises designed to help you understand and master this essential data structure. Tuples are immutable sequences used to store collections of items, and they're a fundamental part of Python programming.

- [Find the size of a Tuple](#)
- [Maximum and Minimum K elements in Tuple](#)
- [Sum of tuple elements](#)
- [Row-wise element Addition in Tuple Matrix](#)
- [Create a list of tuples from given list having number and its cube in each tuple](#)

>> [More Programs on Tuple](#)

Python Dictionary Exercises

In this section, you'll find a variety of [Python dictionary](#) exercises designed to help you master the use of dictionaries, one of Python's most powerful and versatile data structures.

- [Understanding Python Dictionaries](#)
- [Exploring Dictionary Operations](#)

- [Counting Element Frequencies](#)

Programs

- [Sort Python Dictionaries by Key or Value](#)
- [Handling missing keys in Dictionaries](#)
- [Python dictionary with keys having multiple inputs](#)
- [Find the sum of all items in a dictionary](#)
- [Find the size of a Dictionary](#)

>> [More Programs on Dictionary](#)

Python Set Exercises

This section offers a range of [Python set](#) exercises that will help you build a strong understanding of set operations in Python. You'll practice adding and removing elements, performing operations like union, intersection, and difference, and using set comprehensions.

- [Important functions in Set](#)
- [Operations on Sets](#)

Programs

- [Find the size of a Set in Python](#)
- [Iterate over a set in Python](#)
- [Maximum and Minimum in a Set](#)
- [Remove items from Set](#)
- [Check if two lists have atleast one element common](#)

>> [More Programs on Sets](#)

Python Matrix Exercises

Here, in this section you'll find a collection of [Python matrix](#) exercises tailored for beginners and advanced Python programmers. These exercises focus on essential skills such as creating and manipulating matrices, performing operations like addition, subtraction, multiplication, and more.

- [Matrix Operations](#)
- [Last cell in a Matrix](#)
- [Rotate a Matrix](#)

Programs

- [Assigning Subsequent Rows to Matrix first row elements](#)
- [Adding and Subtracting Matrices in Python](#)
- [Group similar elements into Matrix](#)
- [Row-wise element Addition in Tuple Matrix](#)
- [Create an n x n square matrix, where all the sub-matrix have the sum of opposite corner elements as even](#)

>> [More Programs on Matrices](#)

Python Functions Exercises

This section offers a variety of exercises focused on [Python functions](#) to help you master this essential programming concept. You'll learn how to define functions, work with parameters and return values, and explore advanced topics like lambda functions, decorators, and recursion.

- [How to get list of parameters name from a function in Python?](#)
- [How to Print Multiple Arguments in Python?](#)
- [Python program to find the power of a number using recursion](#)
- [Sorting objects of user defined class in Python](#)
- [Functions that accept variable length key value pair as arguments](#)

>> [More Programs on Functions](#)

Python OOPS Exercises

Here in this practice section, you'll find exercises focused on [Object-Oriented Programming \(OOP\) concepts in Python](#). These exercises are designed to help you understand and implement key OOP principles such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction.

- [Python program to build flashcard using class in Python](#)
- [Shuffle a deck of card with OOPS in Python](#)

- [How to create an empty class in Python?](#)
- [Student management system in Python](#)

>> [More Programs on Python OOPS](#)

Python Regex Exercises

[Python Regex](#) exercises to help you master the art of pattern matching and text manipulation. Regular expressions, or Regex, are powerful tools used to search, match, and manipulate strings based on specific patterns.

- [Extracting Numbers from a String](#)
- [Password Validation with Regex](#)

Programs

- [Python program to find the type of IP Address using Regex](#)
- [Python program to find Indices of Overlapping Substrings](#)
- [Python program to extract Strings between HTML Tags](#)
- [Check if String Contain Only Defined Characters using Regex](#)
- [Python program to find files having a particular extension using RegEx](#)

>> [More Programs on Python Regex](#)

Python LinkedList Exercises

In this section, we've compiled a series of exercises focused on implementing and manipulating [linked lists](#) using Python. These exercises cover various operations, such as inserting nodes, deleting nodes, reversing linked lists, and detecting cycles, allowing you to practice and master linked list concepts.

- [Remove loop in Linked List](#)
- [Flattening a Linked List](#)
- [XOR Linked List](#)

Programs

- [Python program to Search an Element in a Circular Linked List](#)
- [Pretty print Linked List in Python](#)
- [Stack using Doubly Linked List](#)
- [Queue using Doubly Linked List](#)
- [Python program to find middle of a linked list using one traversal](#)

>> [More Programs on Linked Lists](#)

Python Searching Exercises

This section offers a range of exercises designed to help you master searching algorithms in Python. You'll learn how to implement techniques like linear search and binary search, as well as more advanced methods such as interpolation and exponential search.

- [Last Match](#)
- [Pattern searching](#)
- [Binary Search](#)

Programs

- [Python Program for Linear Search](#)
- [Python Program for Binary Search \(Recursive and Iterative\)](#)
- [Python Program for Anagram Substring Search \(Or Search for all permutations\)](#)

>> [More Programs on Python Searching](#)

Python Sorting Exercises

This section provides a collection of exercises to help you practice and understand [sorting in Python](#). You'll explore various sorting algorithms, like bubble sort, merge sort, quicksort, and others, each with its unique approach to arranging data.

- [Sort a String](#)
- [Counting Sort](#)
- [Topological sort](#)

Programs

- [Python Program for Bubble Sort](#)
- [Python Program for QuickSort](#)
- [Python Program for Insertion Sort](#)
- [Python Program for Selection Sort](#)
- [Python Program for Heap Sort](#)

>> [More Programs on Python Sorting](#)

Python DSA Exercises

Strengthen your [Python skills with Data Structures and Algorithms \(DSA\)](#) exercises tailored to help you master the fundamental concepts of programming. These exercises cover a wide range of topics, including arrays, linked lists, stacks, queues, trees, graphs, and sorting algorithms, providing a hands-on approach to learning.

- [Python program to reverse a stack](#)
- [Multithreaded Priority Queue in Python](#)
- [Check whether the given string is Palindrome using Stack](#)
- [Program to Calculate the Edge Cover of a Graph](#)
- [Python Program for N Queen Problem](#)

>> [More Programs on Python DSA](#)

Python File Handling Exercises

In this section, you'll find a variety of exercises focused on [Python file handling](#) to help you master reading from and writing to files. These exercises will guide you through the essentials, such as opening, reading, writing, closing files and more.

- [Read content from one file and write it into another file](#)
- [Write a dictionary to a file in Python](#)
- [How to check file size in Python?](#)
- [Find the most repeated word in a text file](#)
- [How to read specific lines from a File in Python?](#)

>> [More Programs on Python File Handling](#)

Python JSON Exercises

In this section, we provide a variety of exercises to help you master [JSON \(JavaScript Object Notation\) in Python](#). JSON is a popular data format used for exchanging information between web clients and servers.

- [Convert class object to JSON in Python](#)
- [Convert JSON data Into a Custom Python Object](#)
- [Flattening JSON objects in Python](#)
- [Convert CSV to JSON using Python](#)

>> [More Programs on Python JSON](#)

Python Projects

- [Number guessing game in Python](#)
- [2048 Game in Python](#)
- [Get Live Weather Desktop Notifications Using Python](#)
- [8-bit game using pygame](#)
- [Tic Tac Toe GUI In Python using PyGame](#)

>> [More Projects in Python](#)

What we Offer?

The best thing about this Python practice exercise is that it helps you [learn Python](#) using sets of detailed programming questions from basic to advanced. It covers questions on core python concepts as well as applications of Python in various domains. So if you are at any stage like beginner, intermediate, or advanced this Python practice set will help you to boost your programming skills in Python.

Conclusion

In closing, we just want to say that the practice or solving Python problems always helps to clear your core concepts and programming logic. Hence, we have designed this Python exercises after deep research so that one can easily enhance their skills and logic abilities.