

[Courses @90% Refund](#) [Free Python 3 Tutorial](#) [Data Types](#) [Control Flow](#) [Functions](#) [List](#) [String](#) [S](#)

File Handling in Python

[Read](#)[Courses](#)[Practice](#)[Jobs](#)

File handling in Python is a powerful and versatile tool that can be used to perform a wide range of operations. However, it is important to carefully consider the advantages and disadvantages of file handling when writing Python programs, to ensure that the code is secure, reliable, and performs well.

In this article we will explore ***Python File Handling, Advantages, Disadvantages and How open, write and append functions works in python file.***

Python File Handling

Python supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files. The concept of file handling has stretched over various other languages, but the implementation is either complicated or lengthy, like other concepts of Python, this concept here is also easy and short. [Python](#) treats files differently as text or binary and this is important. Each line of code includes a sequence of characters, and they form a text file. Each line of a file is terminated with a special character, called the **EOL** or **End of Line** characters like **comma {,}** or **newline character**. It ends the current line and tells the interpreter a new one has begun. Let's start with the reading and writing files.

Advantages of File Handling in Python

- **Versatility:** File handling in Python allows you to perform a wide range of operations, such as creating, reading, writing, appending, renaming

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

[Got It !](#)

and to perform different operations on files (e.g. read, write, append, etc.).

- **User-friendly:** Python provides a user-friendly interface for file handling, making it easy to create, read, and manipulate files.
- **Cross-platform:** Python file-handling functions work across different platforms (e.g. Windows, Mac, Linux), allowing for seamless integration and compatibility.

Disadvantages of File Handling in Python

- **Error-prone:** File handling operations in Python can be prone to errors, especially if the code is not carefully written or if there are issues with the file system (e.g. file permissions, file locks, etc.).
- **Security risks:** File handling in Python can also pose security risks, especially if the program accepts user input that can be used to access or modify sensitive files on the system.
- **Complexity:** File handling in Python can be complex, especially when working with more advanced file formats or operations. Careful attention must be paid to the code to ensure that files are handled properly and securely.
- **Performance:** File handling operations in Python can be slower than other programming languages, especially when dealing with large files or performing complex operations.

For this article, we will consider the following “**geeks.txt**” file as an example.

```
Hello world  
GeeksforGeeks  
123 456
```

Python File Open

Before performing any operation on the file like reading or writing, first, we have to open that file. For this, we should use Python’s inbuilt function [open\(\)](#) but at the time of opening, we have to specify the mode, which

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



```
f = open(filename, mode)
```

Where the following mode is supported:

1. **r**: open an existing file for a read operation.
2. **w**: open an existing file for a write operation. If the file already contains some data, then it will be overridden but if the file is not present then it creates the file as well.
3. **a**: open an existing file for append operation. It won't override existing data.
4. **r+**: To read and write data into the file. The previous data in the file will be overridden.
5. **w+**: To write and read data. It will override existing data.
6. **a+**: To append and read data from the file. It won't override existing data.

Working in Read mode

There is more than one way to [How to read from a file in Python](#). Let us see how we can read the content of a file in read mode.

Example 1: The open command will open the Python file in the read mode and the for loop will print each line present in the file.

Python3

```
# a file named "geek", will be opened with the reading mode.  
file = open('geek.txt', 'r')
```



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)

Output:

```
Hello world  
GeeksforGeeks  
123 456
```

Example 2: In this example, we will extract a string that contains all characters in the Python file then we can use **file.read()**.

Python3

```
# Python code to illustrate read() mode  
file = open("geeks.txt", "r")  
print (file.read())
```

Output:

```
Hello world  
GeeksforGeeks  
123 456
```

Example 3: In this example, we will see how we can read a file using the with statement in Python.

Python3

```
# Python code to illustrate with()  
with open("geeks.txt") as file:  
    data = file.read()  
  
print(data)
```



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)

```
Hello world  
GeeksforGeeks  
123 456
```

Example 4: Another way to read a file is to call a certain number of characters like in the following code the interpreter will read the first five characters of stored data and return it as a string:

Python3

```
# Python code to illustrate read() mode character wise  
file = open("geeks.txt", "r")  
print (file.read(5))
```

Output:

```
Hello
```

Example 5: We can also split lines while reading files in Python. The split() function splits the variable when space is encountered. You can also split using any characters as you wish.

Python3

```
# Python code to illustrate split() function  
with open("geeks.txt", "r") as file:  
    data = file.readlines()  
    for line in data:  
        word = line.split()  
        print (word)
```

Output:

```
['Hello', 'world']  
[GeeksforGeeks]
```



We use cookies to ensure you have the best browsing experience on our website. By

using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)

Creating a File using the write() Function

Just like reading a file in Python, there are a number of ways to [Writing to file in Python](#). Let us see how we can write the content of a file using the write() function in Python.

Working in Write Mode

Let's see how to create a file and how the write mode works.

Example 1: In this example, we will see how the write mode and the write() function is used to write in a file. The close() command terminates all the resources in use and frees the system of this particular program.

Python3

```
# Python code to create a file
file = open('geek.txt','w')
file.write("This is the write command")
file.write("It allows us to write in a particular file")
file.close()
```

Output:

This is the write commandIt allows us to write in a particular file

Example 2: We can also use the written statement along with the with() function.

Python3

```
# Python code to illustrate with() alongwith write()
with open("file.txt", "w") as f:
    f.write("Hello World!!!!")
```



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)

Hello World!!!

Working of Append Mode

Let us see how the append mode works.

Example: For this example, we will use the Python file created in the previous example.

Python3

```
# Python code to illustrate append() mode
file = open('geek.txt', 'a')
file.write("This will add this line")
file.close()
```

Output:

This is the write commandIt allows us to write in a particular fileThis will add this line

There are also various other commands in Python file handling that are used to handle various tasks:

`rstrip()`: This function strips each line of a file off spaces from the right-hand side.

`lstrip()`: This function strips each line of a file off spaces from the left-hand side.

It is designed to provide much cleaner syntax and exception handling when you are working with code. That explains why it's good practice to use them with a statement where applicable. This is helpful because using this method any files opened will be closed automatically after one is done, so auto-cleanup.

Implementing all the functions in File Handling



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

[Got It!](#)

[Privacy Policy](#)

Python3

```
import os

def create_file(filename):
    try:
        with open(filename, 'w') as f:
            f.write('Hello, world!\n')
        print("File " + filename + " created successfully.")
    except IOError:
        print("Error: could not create file " + filename)

def read_file(filename):
    try:
        with open(filename, 'r') as f:
            contents = f.read()
            print(contents)
    except IOError:
        print("Error: could not read file " + filename)

def append_file(filename, text):
    try:
        with open(filename, 'a') as f:
            f.write(text)
        print("Text appended to file " + filename + " successfully.")
    except IOError:
        print("Error: could not append to file " + filename)

def rename_file(filename, new_filename):
    try:
        os.rename(filename, new_filename)
        print("File " + filename + " renamed to " + new_filename + " ")
    except IOError:
        print("Error: could not rename file " + filename)

def delete_file(filename):
    try:
        os.remove(filename)
        print("File " + filename + " deleted successfully.")
    except IOError:
        print("Error: could not delete file " + filename)

if __name__ == '__main__':
    filename = "example.txt"
    new_filename = "new_example.txt"
```

We use cookies to ensure you have the best browsing experience on our website. By

using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

[Privacy Policy](#)

Got It!



```
rename_file(filename, new_filename)
read_file(new_filename)
delete_file(new_filename)
```

Output:

```
File example.txt created successfully.
Hello, world!
Text appended to file example.txt successfully.
Hello, world!
This is some additional text.
File example.txt renamed to new_example.txt successfully.
Hello, world!
This is some additional text.
File new_example.txt deleted successfully.
```

Don't miss your chance to ride the wave of the data revolution! Every industry is scaling new heights by tapping into the power of data. Sharpen your skills and become a part of the hottest trend in the 21st century.

Dive into the future of technology - explore the [Complete Machine Learning and Data Science Program](#) by GeeksforGeeks and stay ahead of the curve.

Three 90 Challenge ending on 31st Jan! Last chance to get 90% refund by completing 90% course in 90 days. [Explore](#) offer now.

Last Updated : 26 Dec, 2023

390

Previous

Next

[NZEC error in Python](#)

[Open a File in Python](#)



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

[Got It !](#)

Similar Reads

[Inventory Management with File handling in Python](#)

[Handling File Uploads via CGI](#)

[reStructuredText | .rst file to HTML file using Python for Documentations](#)

[Python - Copy contents of one file to another file](#)

[Python program to reverse the content of a file and store it in another file](#)

[Create a GUI to convert CSV file into excel file using Python](#)

[Python - Get file id of windows file](#)

[How to save file with file name from user using Python?](#)

[How to convert PDF file to Excel file using Python?](#)

[How to convert CSV File to PDF File using Python?](#)

Complete Tutorials

[Python Crash Course](#)

[Python API Tutorial: Getting Started with APIs](#)

[Advanced Python Tutorials](#)

[Python Automation Tutorial](#)

[OpenAI Python API - Complete Guide](#)



GeeksforGeeks

Article Tags : [python](#) , [python-io](#) , [Python](#)

Practice Tags : [python](#), [python](#)



We use cookies to ensure you have the best browsing experience on our website. By

using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

[Privacy Policy](#)

Got It !



Company

- [About Us](#)
 - [Legal](#)
 - [Careers](#)
 - [In Media](#)
 - [Contact Us](#)
 - [Advertise with us](#)
 - [GFG Corporate Solution](#)
 - [Placement Training Program](#)
 - [Apply for Mentor](#)
-
- [Job-A-Thon Hiring Challenge](#)
 - [Hack-A-Thon](#)
 - [GfG Weekly Contest](#)
 - [Offline Classes \(Delhi/NCR\)](#)
 - [DSA in JAVA/C++](#)
 - [Master System Design](#)
 - [Master CP](#)
 - [GeeksforGeeks Videos](#)
 - [Geeks Community](#)

Explore

Languages

- [Python](#)
 - [Java](#)
 - [C++](#)
 - [PHP](#)
 - [GoLang](#)
 - [SQL](#)
 - [R Language](#)
-
- [Data Structures](#)
 - [Algorithms](#)
 - [DSA for Beginners](#)
 - [Basic DSA Problems](#)
 - [DSA Roadmap](#)
 - [Top 100 DSA Interview Problems](#)
 - [DSA Roadmap by Sandeep Jain](#)

DSA



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

[Got It !](#)

[Privacy Policy](#)

Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning Tutorial
- ML Maths
- Data Visualisation Tutorial
- Pandas Tutorial
- NumPy Tutorial
- NLP Tutorial
- Deep Learning Tutorial

HTML & CSS

- HTML
- CSS
- Web Templates
- CSS Frameworks
- Bootstrap
- Tailwind CSS
- SASS
- LESS
- Web Design

Python

- Python Programming Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Python Tutorial
- Python Interview Question

Computer Science

- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management System
- Software Engineering
- Digital Logic Design
- Engineering Maths

DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Competitive Programming

- Top DS or Algo for CP
- Top 50 Tree
- Top 50 Graph
- Top 50 Array
- Top 50 String
- Top 50 DP
- Top 15 Websites for CP

System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide

JavaScript

- JavaScript Examples
- TypeScript
- ReactJS
- NextJS

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



[System Design Bootcamp](#)[Lodash](#)[Interview Questions](#)[Web Browser](#)

NCERT Solutions

School Subjects

[Class 12](#)[Mathematics](#)[Class 11](#)[Physics](#)[Class 10](#)[Chemistry](#)[Class 9](#)[Biology](#)[Class 8](#)[Social Science](#)[Complete Study Material](#)[English Grammar](#)

Commerce

Management & Finance

[Accountancy](#)[Management](#)[Business Studies](#)[HR Management](#)[Indian Economics](#)[Income Tax](#)[Macroeconomics](#)[Finance](#)[Microeconomics](#)[Economics](#)[Statistics for Economics](#)

UPSC Study Material

SSC/ BANKING

[Polity Notes](#)[SSC CGL Syllabus](#)[Geography Notes](#)[SBI PO Syllabus](#)[History Notes](#)[SBI Clerk Syllabus](#)[Science and Technology Notes](#)[IBPS PO Syllabus](#)[Economy Notes](#)[IBPS Clerk Syllabus](#)[Ethics Notes](#)[SSC CGL Practice Papers](#)[Previous Year Papers](#)

Colleges

Companies

[Indian Colleges Admission & Campus Experiences](#)[META Owned Companies](#)[List of Central Universities - In India](#)[Alphabhet Owned Companies](#)[Colleges in Delhi University](#)[TATA Group Owned Companies](#)[IIT Colleges](#)[Reliance Owned Companies](#)[NIT Colleges](#)[IIIT Colleges](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)

Preparation for SDE
Experienced Interviews
Internship Interviews
Competitive Programming
Aptitude Preparation
Puzzles

JEE Advanced
GATE CS
NEET
UGC NET

More Tutorials

Software Development
Software Testing
Product Management
SAP
SEO - Search Engine Optimization
Linux
Excel

Write & Earn

Write an Article
Improve an Article
Pick Topics to Write
Share your Experiences
Internships

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)