

# File Handling in Python :-

- 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.*

## 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, and deleting files.
- **Flexibility** : File handling in Python is highly flexible, as it allows you to work with different file types (e.g. text files, binary files, CSV files, etc.), 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.

## Example of File Handling in Python :-

*# a file named "geek", will be opened with the reading mode.*

```
file = open('geek.txt', 'r')
```

*# This will print every line one by one in the file*

```
for each in file:
```

```
    print (each)
```

## Output:-

Hello world

GeeksforGeeks

123 456