How to Check file size in Python

In this article, we will learn to check the size of a file in Python. We will use some built-in functions and some custom codes as well. Let's first have a quick look over why we need file size and how can we calculate the file size in Python.

Check the File Size in Python

It is important to get file size in Python in case of ordering files according to file size or in many use case scenarios. The output of file size is always in bytes. The value could be passed as multiples of the file system block size so that further computation is easy.

We will learn four ways to check file size using the path and os module.

1. path.stat() function
2. os.stat() function
3. os.path.getsize() function
4. seek() and tell() function

Check File Size using Path.stat() function in Python

Python language has `os` module which helps python programs to interact with the operating system and provide the user with functionality. Here `stat()` is a function of the os module. For this, here we are using `pathlib` library. In the below example, we have used `st_size()` function to find the size of any given file.

Syntax

```
Path(filename).stat().st_size()
```

Example

It returns an object which contains so many headers including file created time and last modified time etc. among them st_size gives the exact size of the file.


```
from pathlib import Path

var1 = Path('filename.txt').stat()
var2 = Path('filename.txt').stat().st_size
print("Output of stat()- ", var1)
print("File size- ", var2)
```

OUTPUT:

```
Output of stat()- os.stat_result(st_mode=33206, st_ino=4503599627421738, st_dev=47883412, st_nlink=1, st_uid=0, st_g:
File size- 93
```

Explanation: The first path is imported from pathlib library which is an easy way to perform file-related operations. The filename is passed with `stat()` function to get details of file and then st_size() is used to return file size in bytes.

Check File Size using os.stat() function in Python

Comparing with the above example, instead of using pathlib, we have used os module. Thereby performing `os.stat()` function. st_size() property of the object is returned by os.stat() function.

Example

```
import os

var1 = os.stat('filename.txt')
```

```
var2 = os.stat('filename.txt').st_size

print("Output of stat()- ", var1)
print("File size- ", var2)
```

OUTPUT:

```
Output of stat()- os.stat_result(st_mode=33206, st_ino=4503599627421738, st_dev=47883412, st_nlink=1, st_uid=0, st_g:

File size- 93
```

Check File Size using os.path.stat() function in Python

The third way of finding the size of the file is by using `os.path.getsize()`. It also involves the os module.

Implementation of os.path.getsize() is simple and easy to process as compared to os.stat(file).st_size(). It raises `os.error` if the file does not exist or is inaccessible.

Syntax

```
os.path.getsize("file path/file name")
```

Example

In this, we have to provide the exact file path(absolute path), not a relative path.

```
import os

var1 = os.path.getsize('filename.txt')
print("File size- ", var1)
```

OUTPUT:

```
File size- 93
```


Check File Size using seek() and tell() function in Python

The above-given methods work for real files, but if we need something that works for "file-like objects", the solution is using seek/tell file handling functions. It works for real files and StringIO's.

In this, seek() will take the cursor from beginning to end, and then tell() will return the size of the file.

`seek()`- This function is used to change the cursor position of the file to a given specific position. The cursor defines where the data has to be read or written in the file.

`tell()`- This function returns the current file position in a file stream.

Let us look at the below example and see how the seek() and tell() gives file size.

```
import os

with open('filename.txt') as f:
    f.seek(0, os.SEEK_END)
    size = f.tell()
print("File size- ", size)
```

OUTPUT:

```
File size- 93
```

Explanation-

In the above example, f is a file type object made while opening the file. f is used to perform the seek function. As we can see 0 and os.SEEK.END is used in the parameters. First, the pointer is placed at the beginning of the file i.e. 0, and then `SEEK_END()` will place the pointer at the end of the file. Further, in the next line, f.tell() is used to tell the current position which is equivalent to the number of bytes the cursor has moved. This will store the size into the size variable starting from 0 to end.

The difference between seek/tell and os.stat() is that you can stat() a file even if you don't have permission to read it. So, the seek/tell approach won't work unless you have read permission.

Conclusion

In this article, we learned how to check the file size by using several built-in functions such as `seek()`, `tell()`, `st_size()`, and `os.path.getsize()`. We used some custom codes and file handling concepts as well. For example, we used open() function to open the file and then used functions to check file size.

← Read JSON File

List all files →