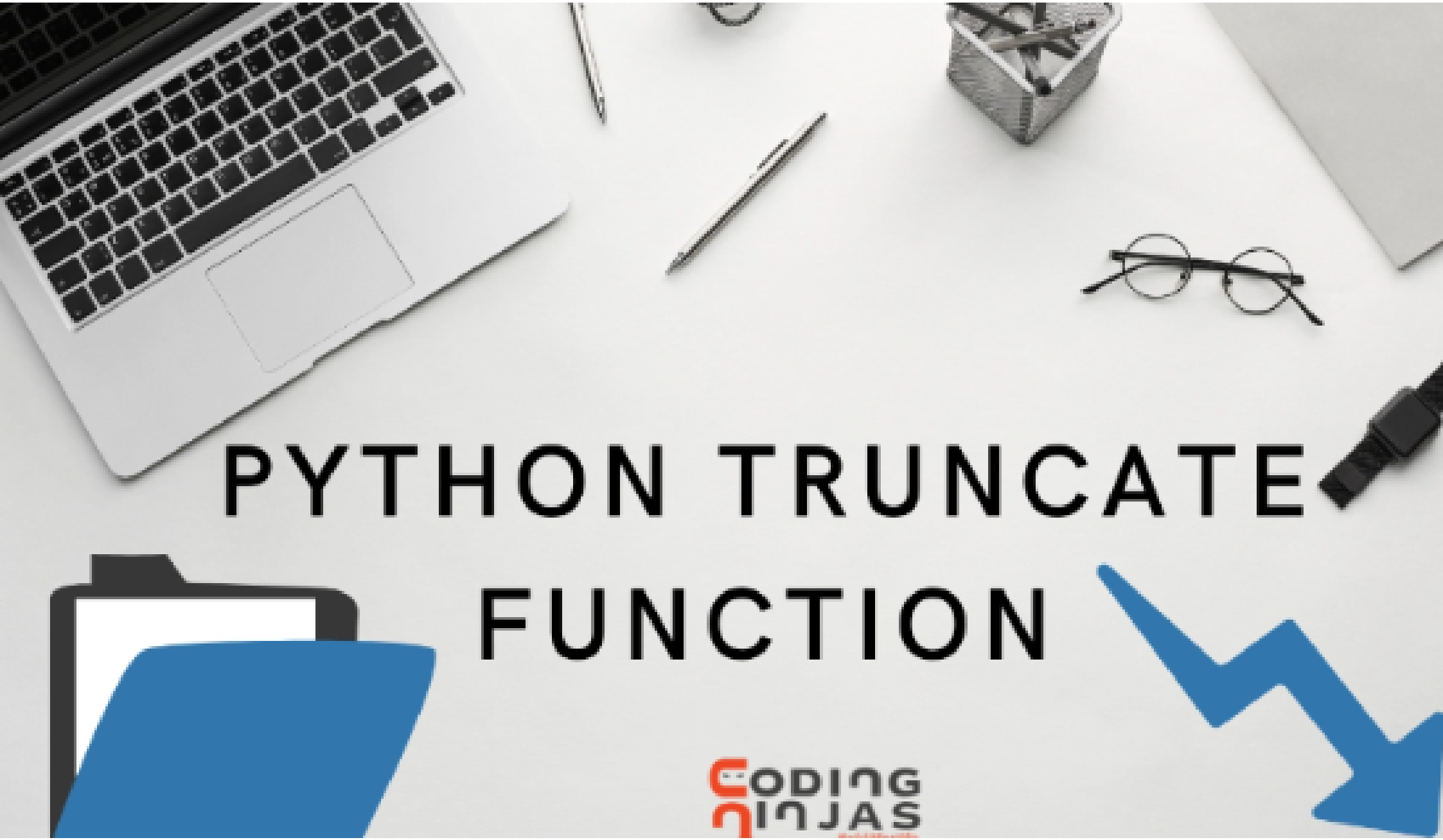
Python


PYTHON TRUNCATE
FUNCTION
CODING NINJAS

Python File truncate() Method

Prerequisites:

- File Objects in Python
- Reading and Writing to files in Python

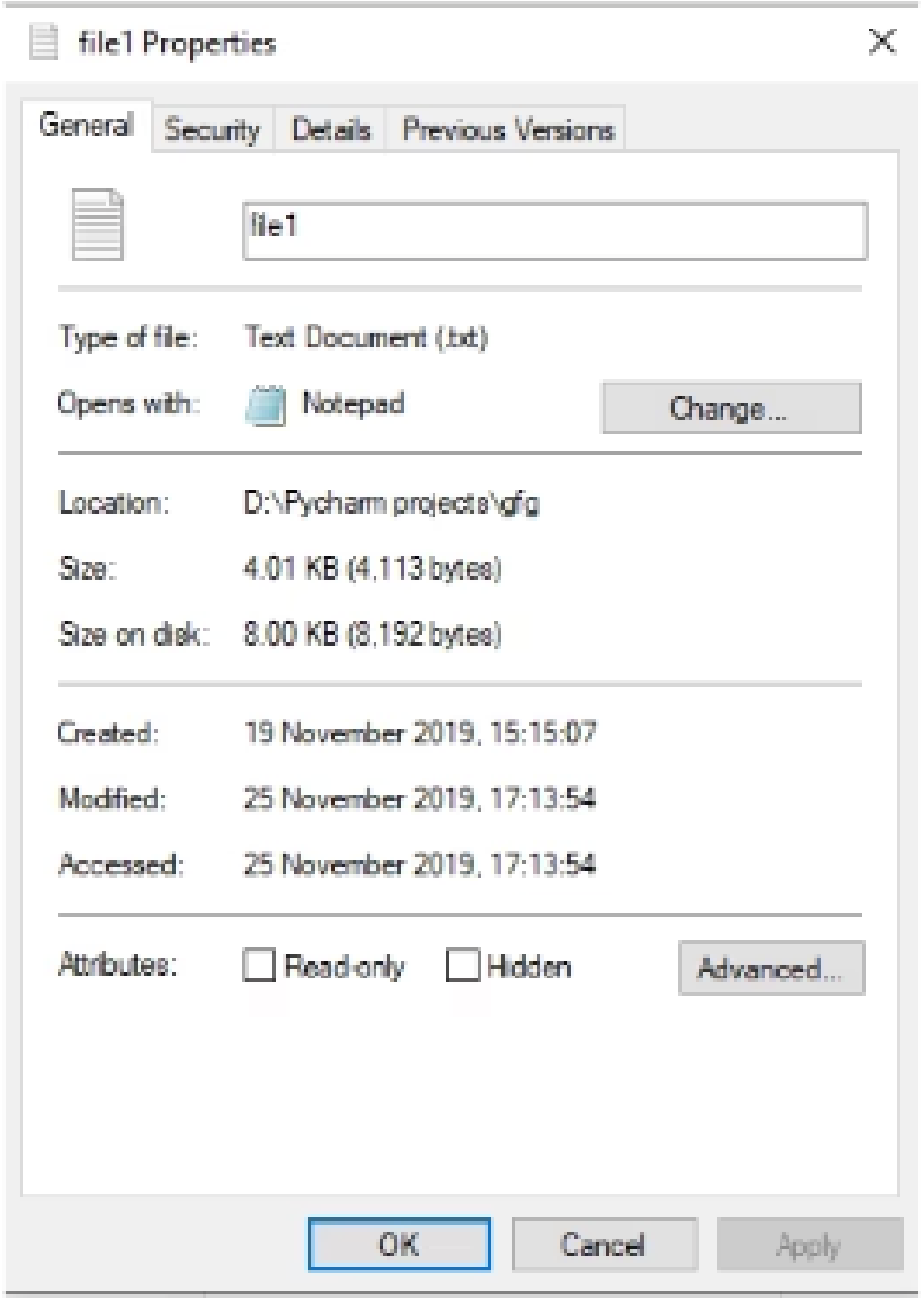
`Truncate()` method truncate the file's size. If the optional size argument is present, the file is truncated to (at most) that size. The size defaults to the current position. The current file position is not changed. Note that if a specified size exceeds the file's current size, the result is platform-dependent: possibilities include that the file may remain unchanged, increase to the specified size as if zero-filled, or increase to the specified size with undefined new content.
To truncate the file, you can open the file in append mode or write mode.

Syntax:

```
fileObject.truncate(size)
```

Example:

See the below image for file size.



Let's change the file size to 100 bytes.

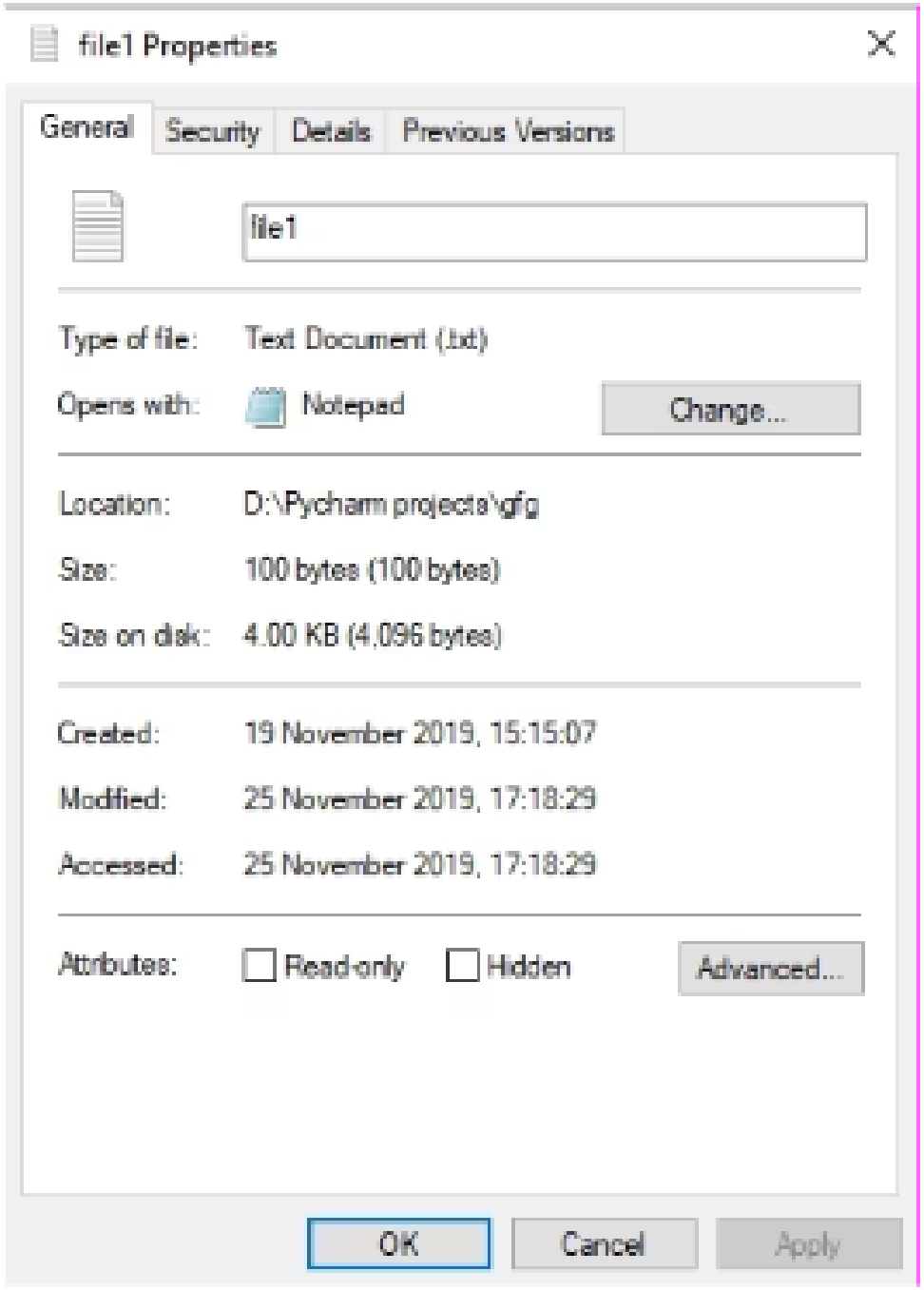
```
# Python program to demonstrate
# truncate() method

fp = open('file1.txt', 'w')

# Truncates the file to specified
# size
fp.truncate(100)

# Closing files
fp.close()
```

Output:


file1 Properties
General
Security
Details
Previous Versions
file1
Type of file:
Text Document (.txt)
Opens with:
Notepad
Change...
Location:
D:\Pycharm projects\gfg
Size:
100 bytes (100 bytes)
Size on disk:
4.00 KB (4,096 bytes)
Created:
19 November 2019, 15:15:07
Modified:
25 November 2019, 17:18:29
Accessed:
25 November 2019, 17:18:29
Attributes:
Read-only
Hidden
Advanced...
OK
Cancel
Apply

With statement

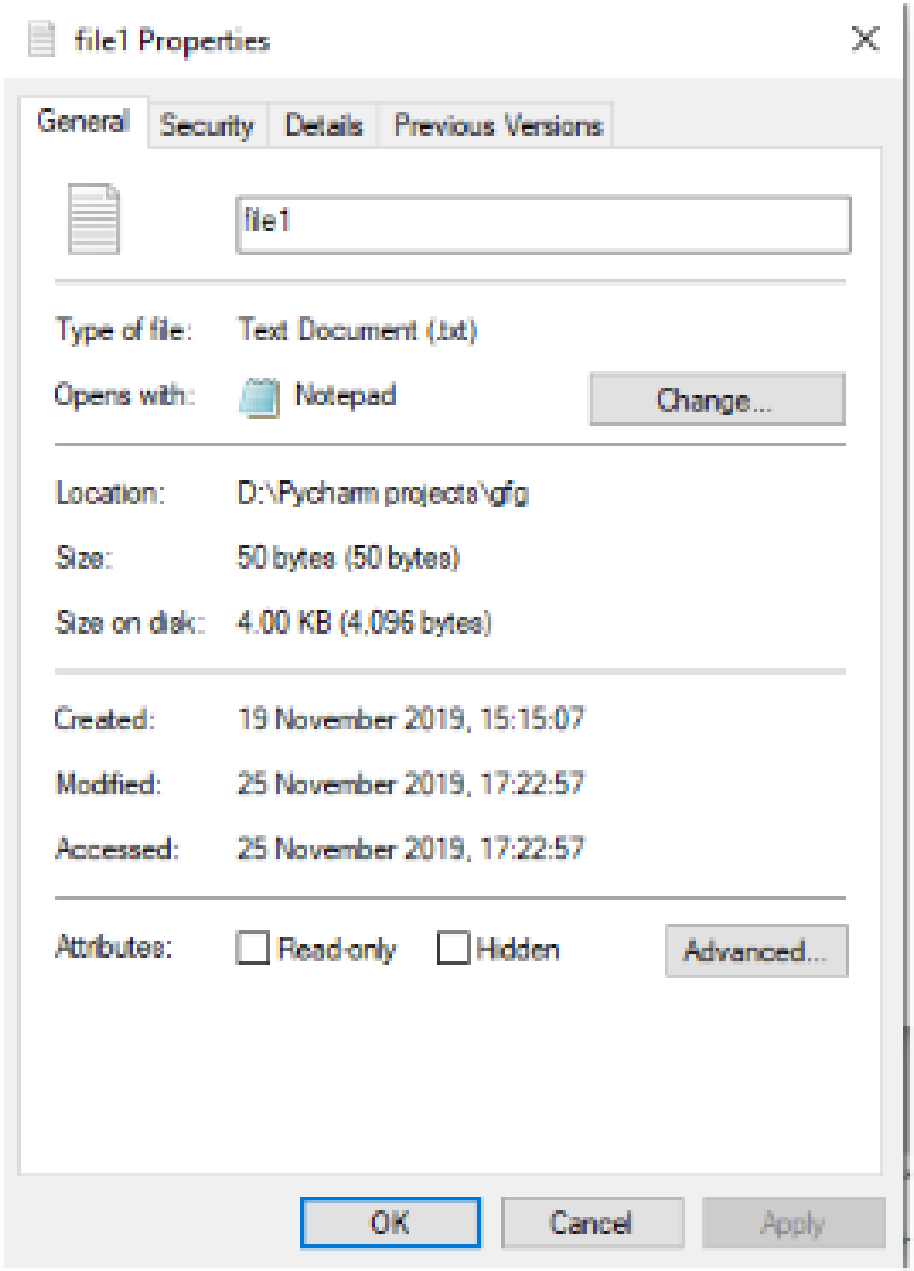
In the above approaches, every time the file is opened it is needed to be closed explicitly. If one forgets to close the file, it may introduce several bugs in the code, i.e. many changes in files do not go into effect until the file is properly closed. To prevent this `with` statement can be used. with statement in Python is used in exception handling to make the code cleaner and much more readable. It simplifies the management of common resources like file streams. Observe the following code example on how the use of `with` statement makes code cleaner. There is no need to call `file.close()` when using with statement. The `with` statement itself ensures proper acquisition and release of resources.

Let's change the above file to 50 bytes

```
# Python program to demonstrate
# truncate method using with statement

with open('file1.txt', 'w') as fp:
    fp.truncate(50)
```

Output:


file1 Properties
General
Security
Details
Previous Versions
file1
Type of file:
Text Document (.txt)
Opens with:
Notepad
Change...
Location:
D:\Pycharm projects\gfg
Size:
50 bytes (50 bytes)
Size on disk:
4.00 KB (4,096 bytes)
Created:
19 November 2019, 15:15:07
Modified:
25 November 2019, 17:22:57
Accessed:
25 November 2019, 17:22:57
Attributes:
Read-only
Hidden
Advanced...
OK
Cancel
Apply