fileHandling

January 25, 2026

1 File Handling

4 Opening a File (open())

Syntax

```
file_object = open("filename", "mode")

Common File Modes (VERY IMPORTANT)

Mode    Meaning

r   Read (default)

w   Write (overwrite)

a   Append

x   Create new file

r+  Read + Write

w+  Write + Read

a+  Append + Read

rb+ Read + Write in binary(mostly preferred for positioing in file handling)

ab+ Append + Read in binary



-- to close a file --
file_object.close()

-- to read from a file --

content = file_object.read() -- reads entire file
```

line = file_object.readline() -- reads one line

lines = file_object.readlines() -- reads all lines into a list

-- to write to a file --

w mode destroys existing content

write() overwrites file

file_object.write("text to write")

file_object.writelines(list_of_strings)

-- creating a file (using x mode)--

file_object = open("newfile.txt", "x")

file_object.write("This is a new file.")
file_object.close()

[1]:
```
# read() -- read full file
f=open("sample.txt","r")
content=f.read()
print(content)
f.close()
```

```
Here are a few random sentences:
The old map led to a forgotten treasure hidden beneath the willow tree.
Sometimes I stare at a door or a wall and I wonder what is this reality, why am
I alive, and what is this all about?.
She moved forward only because she trusted that the ending she now was going
through must be followed by a new beginning.
The first thing I remembered this morning after I woke up was my purple nose in
my dream.
```

[2]:
```
# readline() -- read single line
f=open("sample.txt","r")
line1=f.readline()
print(line1)
line2=f.readline()
```

```
Here are a few random sentences:

```

[3]:
```
# readlines() -- read all lines into a list
f=open("sample.txt","r")
print(f.readlines())
```

```
f.close()
```

```
['Here are a few random sentences:\n', 'The old map led to a forgotten treasure
hidden beneath the willow tree.\n', 'Sometimes I stare at a door or a wall and I
wonder what is this reality, why am I alive, and what is this all about?.\n',
'She moved forward only because she trusted that the ending she now was going
through must be followed by a new beginning.\n', 'The first thing I remembered
this morning after I woke up was my purple nose in my dream. ']
```

[ ]:
```
# writing to a file
f=open("data.txt","w") # w destroys old content(if the file exists) else␣
  ↪creates a new file
f.write("Hello Python\n")
f.write("File Handling in Python\n")
f.close()
```

[6]:
```
# appending to a file (a mode )
f=open("data.txt","a") # a mode appends to the existing content
f.write("Appending new line 1\n")
f.write("Appending new line 2\n")
f.close()
```

[ ]:
```
# create a file (x mode)
f=open("newfile.txt","x") # x mode creates a new file, if the file exists it␣
  ↪raises an error
f.write("This is a new file created using x mode.\n")
f.close()
```

[13]:
```
# file pointer tell()and seek()
# tell() returns the current position of the file pointer
f=open("sample.txt","r")
print(f.tell())
print(f.read(10))  # read first 10 characters
print("Current file pointer position:",f.tell())

# seek() changes the file pointer position
f.seek(0)  # move pointer to the beginning
print(f.read(10))  # read first 10 characters again
f.close()

f=open("sample.txt","r")
f.seek(10) # move pointer to the 10th position and file starts reading from␣
  ↪there
print(f.read(10))
f.close()
```

```
0
The old li
```

```
Current file pointer position: 10
The old li
ghthouse s
```

[19]:
```
# With statement for file handling(best practice)
with open("sample.txt","r") as f:
    content=f.read()
    # here l1 & ll are empty bcz f.read() takes the file pointer to end

    #so read again we should change the file pointer
    f.seek(0)
    l1=f.readline()
    ll=f.readlines()
    print(content)
    print(l1)
    print(ll)

    f.seek(10)
    print(f.read(10))
```

```
The old lighthouse stood firmly against the crashing waves, a silent guardian
for passing ships.
Seagulls circled overhead, their cries barely audible over the roaring storm.
Inside, the keeper diligently polished the lens, ensuring the light would cut
through the darkness.
The old lighthouse stood firmly against the crashing waves, a silent guardian
for passing ships.

['Seagulls circled overhead, their cries barely audible over the roaring storm.
\n', 'Inside, the keeper diligently polished the lens, ensuring the light would
cut through the darkness.']
ghthouse s
```

[32]:
```
# seek(offset,whence)
# whence=0 --> beginning of the file
# whence=1 --> current position
# whence=2 --> end of the file
with open("sample.txt","r") as f:
    print(f.tell()) # initially 0

    f.seek(6) # navigates to 12th position from beginning
    print(f.tell())

    # this doesnt work in r mode bcz Why Python forbids this
# In text mode:

# Newlines may be translated (\r\n   \n)
# Characters may be multi-byte (UTF-8)

# So Python cannot reliably move "3 bytes forward" relative to current position.
    # f.seek(3,1) # navigates 3 positions ahead from current position
    print(f.tell())

# so we should open in binary mmode
with open("sample.txt","rb+") as f:
    print(f.tell()) # initially 0

    f.seek(12) # navigates to 6th position from beginning
    print(f.tell())

    f.seek(3,1) # navigates 3 positions ahead from current position
    print(f.tell())

    f.seek(-3,2) # navigates 3 positions back from end of the file
    print(f.tell())
```

```
0
6
6
0
12
15
275
```

[20]:
```
# reading file line by line using for loop
with open("sample.txt","r") as f:
    for line in f:
        print(line)
```

```
The old lighthouse stood firmly against the crashing waves, a silent guardian
for passing ships.

Seagulls circled overhead, their cries barely audible over the roaring storm.

Inside, the keeper diligently polished the lens, ensuring the light would cut
through the darkness.
```

[21]:
```
# check if file exists
import os
if os.path.exists("sample.txt"):
    print("File exists")

else:
    print("File does not exist")
```

```
File exists
```

[22]:
```
# file handling with exception handling
try:
    with open("test.txt") as f:
        print(f.read())
except FileNotFoundError:
    print("File not found. Please check the file path.")
```

```
File not found. Please check the file path.
```

[ ]:
```
# writing multiple lines
lines=["Python\n","Java\n","C++\n","JavaScript\n"]
with open("data.txt","w") as f:
    f.writelines(lines) # replaces old content, to append use a
```

[25]:
```
# copying content from one file to another
with open("sample.txt","r") as source:
    with open("copy.txt","w") as dest:
        for line in source:
            dest.write(line)
```

[26]:
```
# count lines,words,characters in a file
with open("sample.txt","r") as f:
    text=f.read()
    lines=text.splitlines()
    words=text.split()
    characters=len(text)

print("Lines:",len(lines))
print("Words:",len(words))
print("Characters:",characters)
```

```
Lines: 3
Words: 41
Characters: 276
```

1.0.1 Examples

[ ]:
```
# print line numbers that are even
with open("data.txt","r") as f:
    for i,line in enumerate(f,start=1):
        if i%2==0:
            print(line.strip())
```

```
Java
JavaScript
Java
JavaScript
```

[35]:
```
# count number of vowels in a file
count=0
with open("sample.txt","r") as f:
    text=f.read()
    for ch in text:
        if ch in 'aeiouAEIOU':
            count+=1
print("Number of vowels:",count)
```

```
Number of vowels: 82
```

[37]:
```
# copy only lines containing a specific word from one file to another
word="Python"
with open("input.txt","r") as src,open("output.txt","w") as dest:
    for line in src:
        if word in line:
            dest.write(line)
```

[38]:
```
# reverse the entire file content
with open("data.txt","r") as f:
    lines=f.readlines()

with open("data.txt","w") as f:
    for line in reversed(lines):
        f.write(line)
```

[40]:
```
# merge two files into a third file
with open("file1.txt","r") as f1,open("file2.txt","r") as f2:
    content=f1.read()+"\n"+f2.read()
with open("merged.txt","w") as f:
    f.write(content)
```

[41]:
```
# remove duplicate lines from a file
s=set()
with open("data.txt" ,"r") as f:
    lines=f.readlines() # first read

with open("data.txt","w") as f:
    for line in lines:
        if line not in s:
            f.write(line)
            s.add(line)
```

[42]:
```
# copy  a file
with open("sample.txt","r") as src,open("sample_copy.txt","w") as dest:
    dest.write(src.read())
```

[43]:
```python
# print longest line in a path
with open("data.txt", "r") as f:
    longest = max(f.readlines(), key=len)

print(longest.strip())
```

```
JavaScript
```