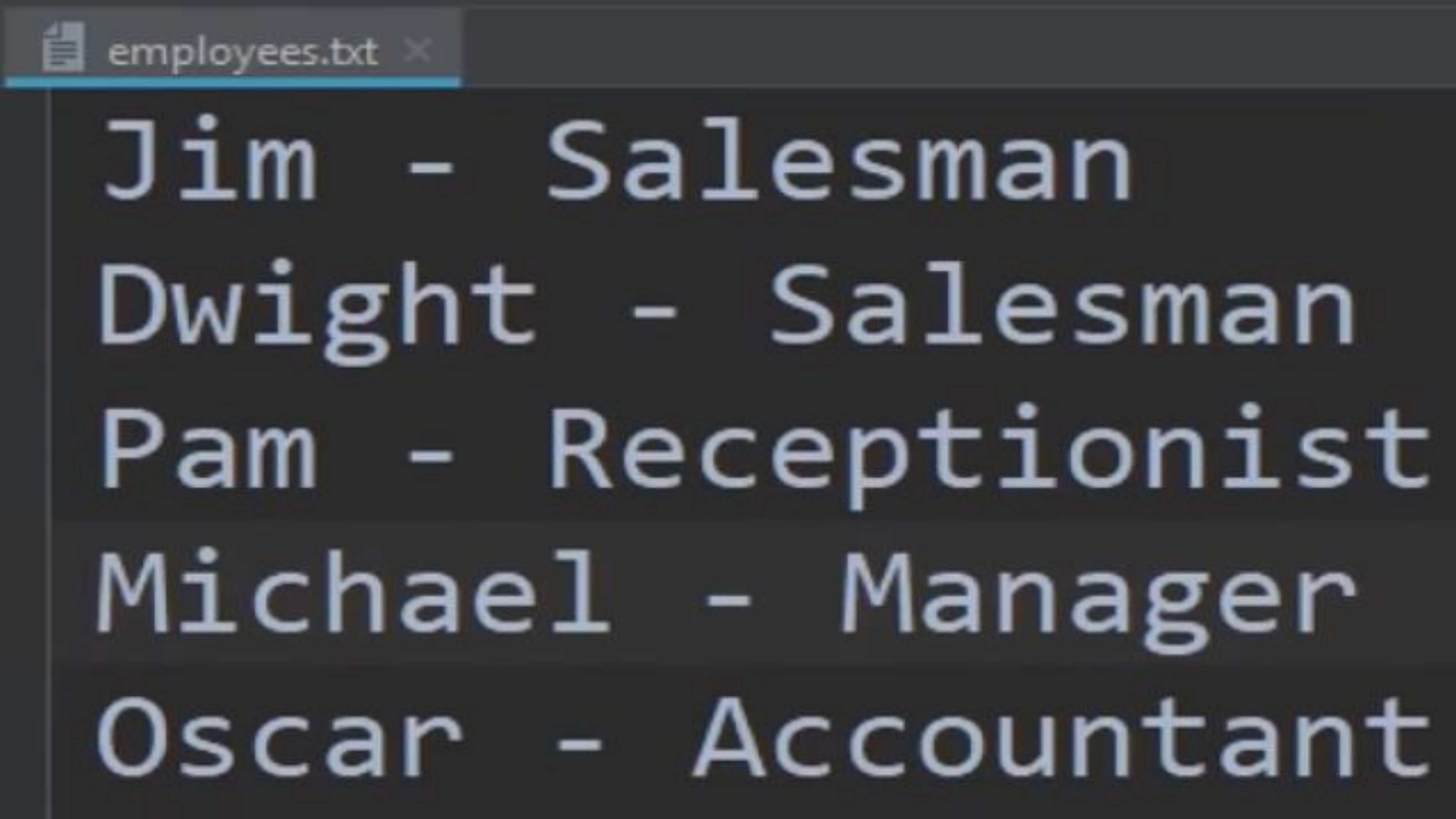
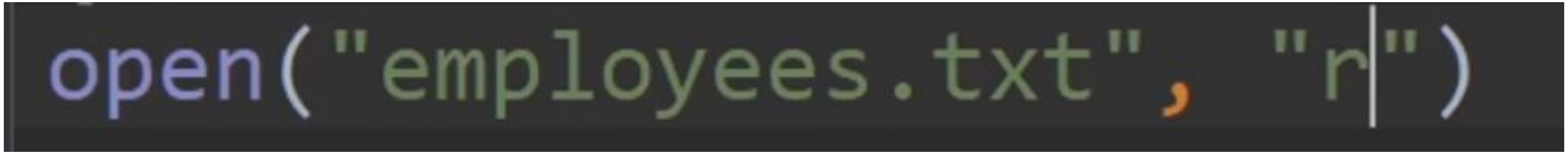
Reading Files



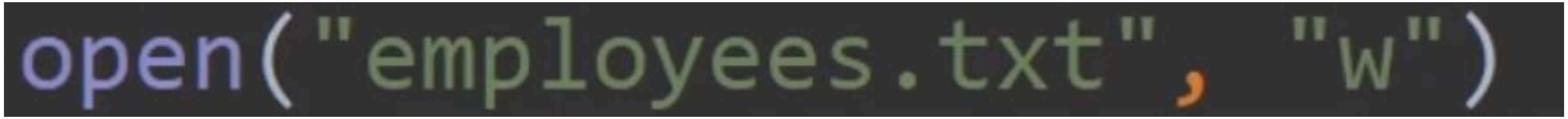
```
Jim - Salesman
Dwight - Salesman
Pam - Receptionist
Michael - Manager
Oscar - Accountant
```

Ways of opening files to work with them



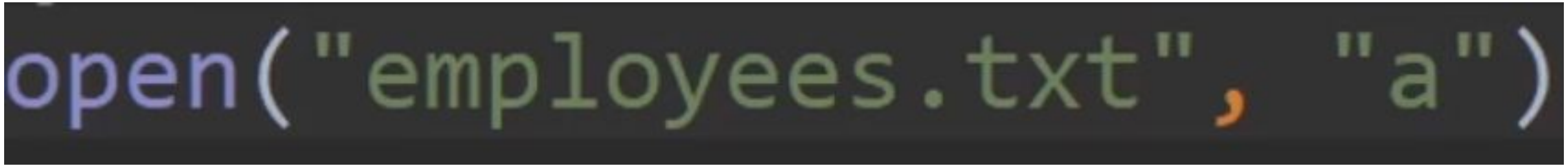
```
open("employees.txt", "r")
```

r = read files



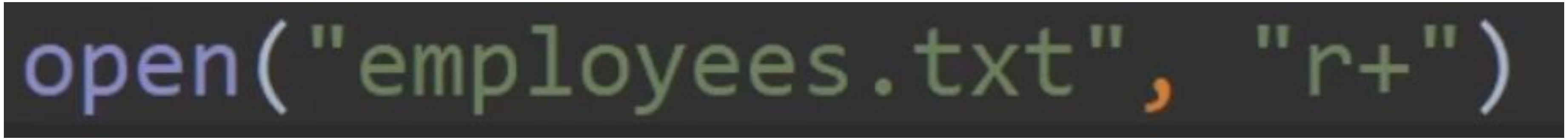
```
open("employees.txt", "w")
```

w = write files



```
open("employees.txt", "a")
```

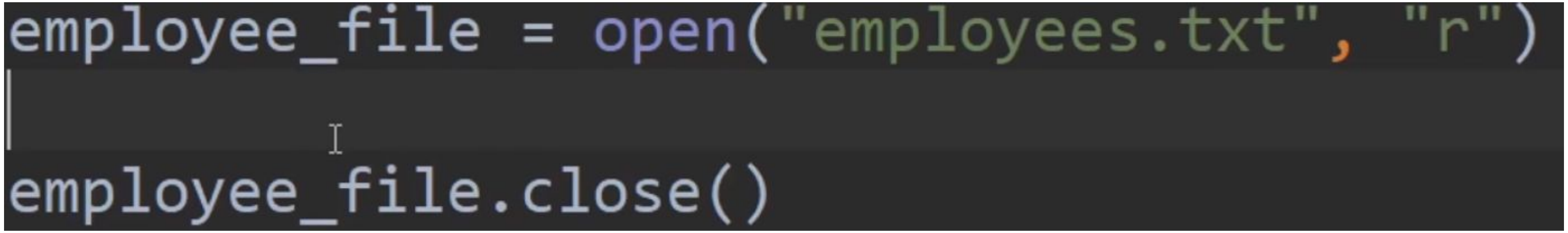
a = append files , append information on to the end of the file



```
open("employees.txt", "r+")
```

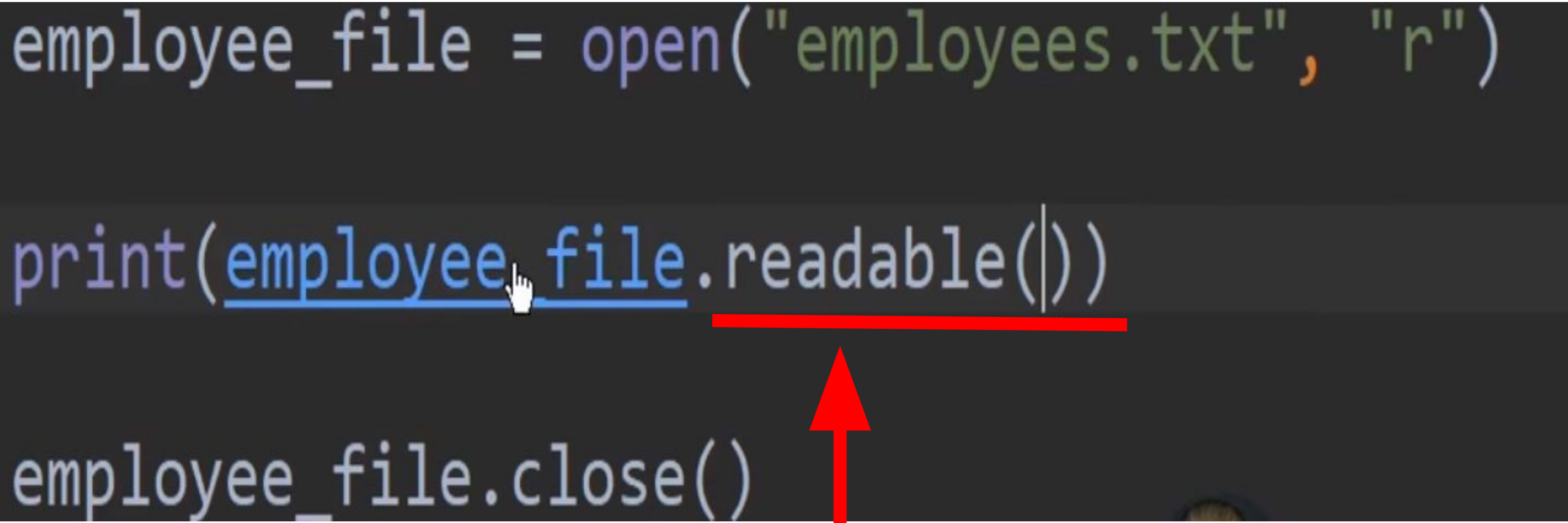
r+ = means you can read and write files

Generally we want to store this open file inside of a variable
And remember to close it



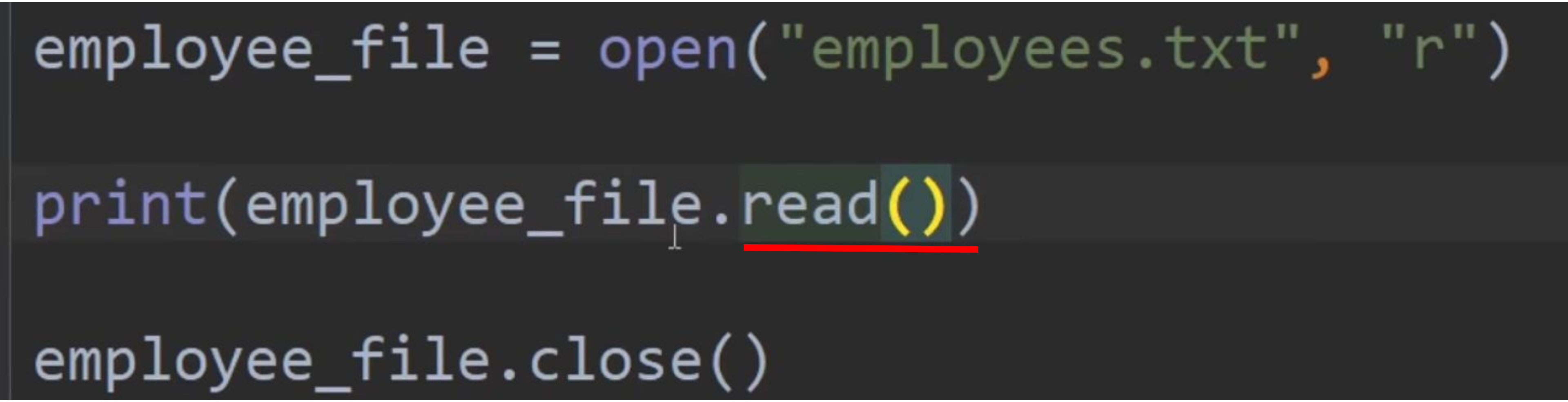
```
employee_file = open("employees.txt", "r")

employee_file.close()
```

Make sure you close the file



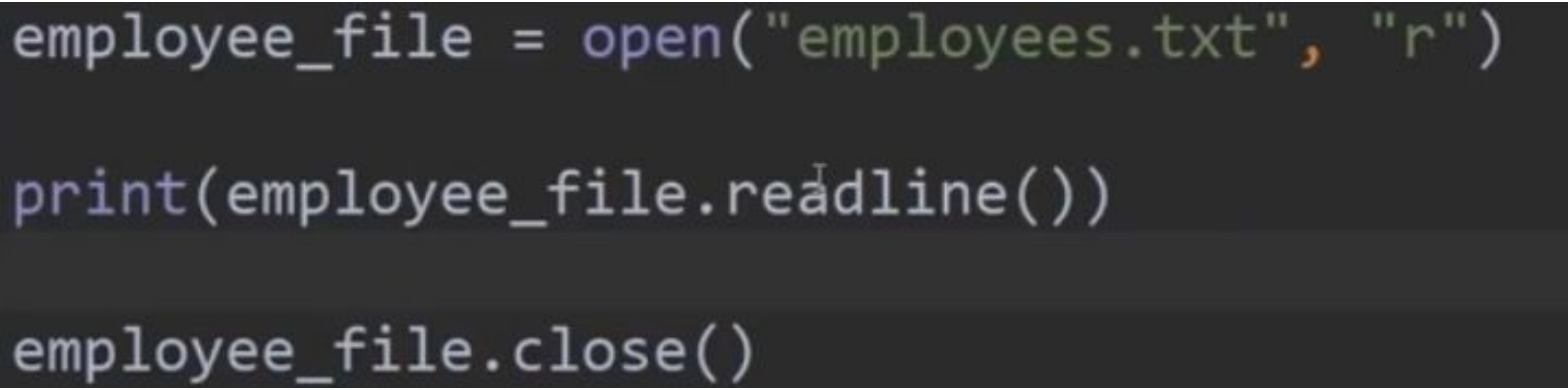
```
employee_file = open("employees.txt", "r")

print(employee_file.readable())

employee_file.close()
```

Returns a boolean value and tells us whether or not we can read from this file



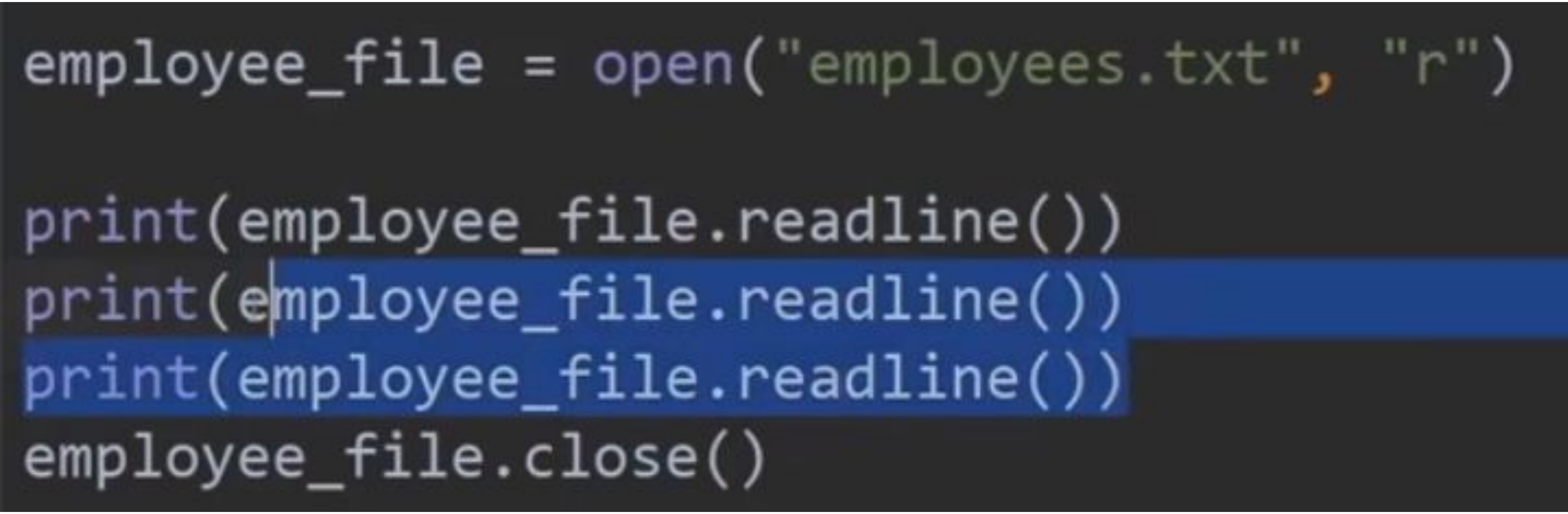
```
employee_file = open("employees.txt", "r")

print(employee_file.read())

employee_file.close()
```

Read just spits out all the information in the file



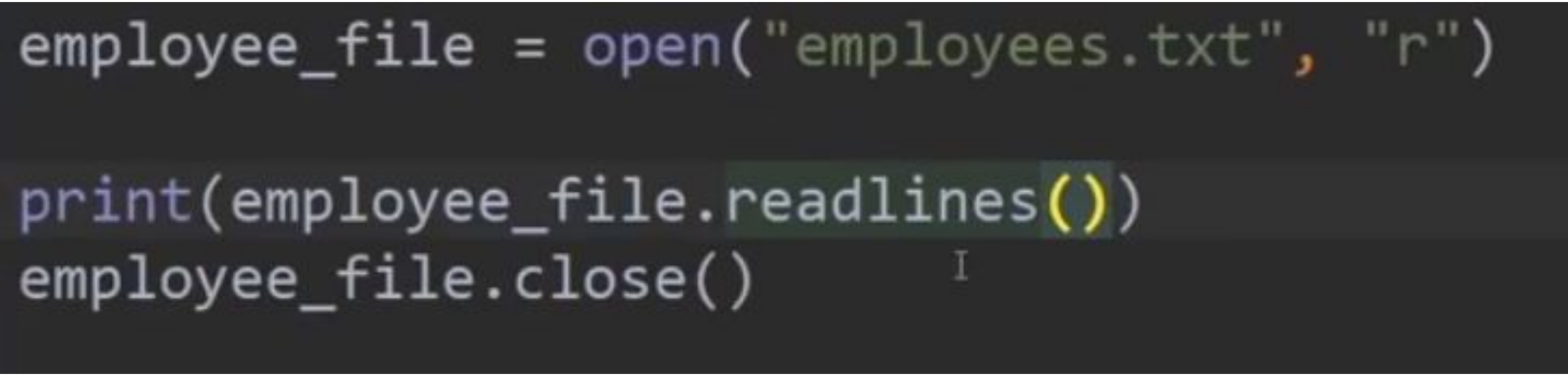
```
employee_file = open("employees.txt", "r")

print(employee_file.readline())

employee_file.close()
```

Readline reads only an individual line in the file



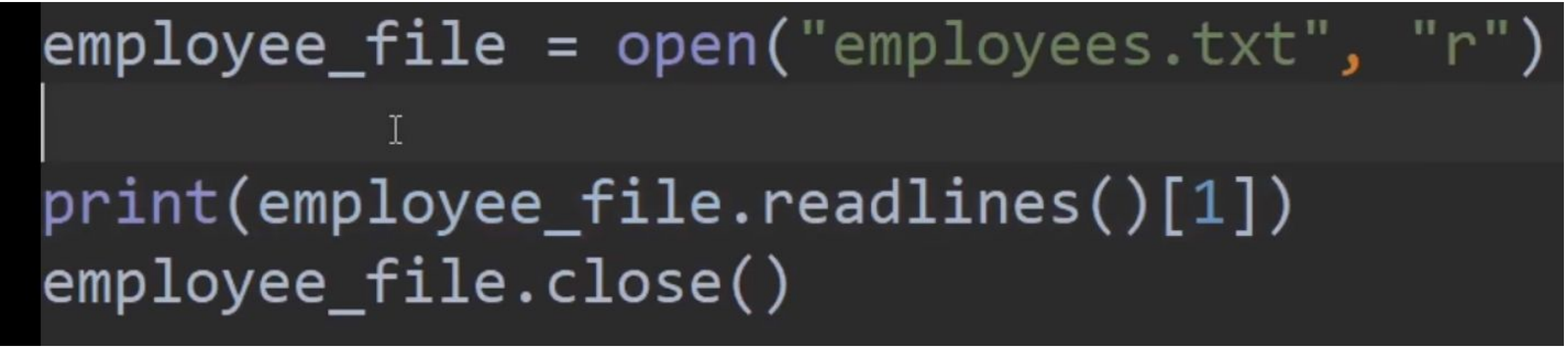
```python
employee_file = open("employees.txt", "r")

print(employee_file.readline())
print(employee_file.readline())
print(employee_file.readline())
employee_file.close()
```

Now I can read multiple lines --- in this, we are reading the first three lines



```
employee_file = open("employees.txt", "r")

print(employee_file.readlines())
employee_file.close()
```

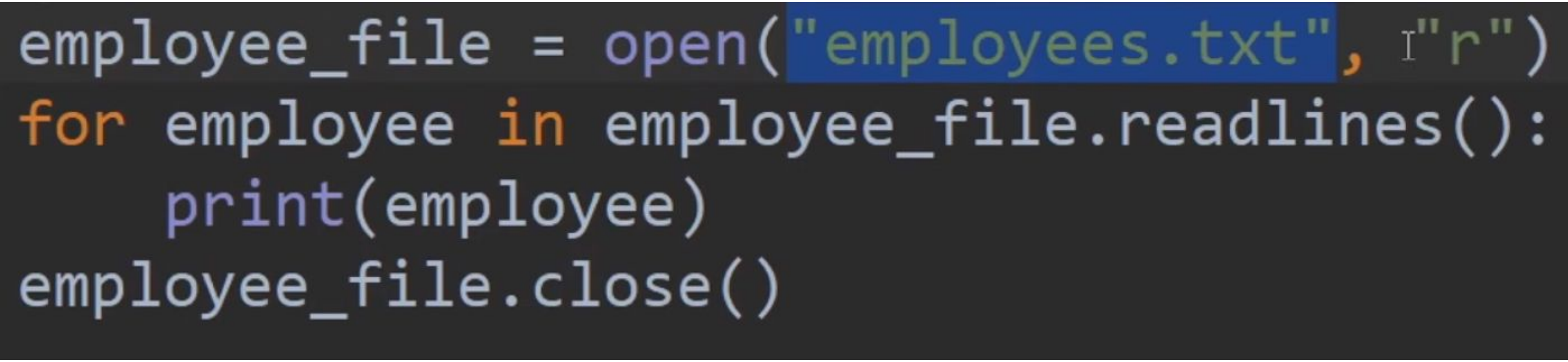
Readlines takes the multiple lines inside of our file and Puts them inside of an array



```python
employee_file = open("employees.txt", "r")

print(employee_file.readlines()[1])
employee_file.close()
```

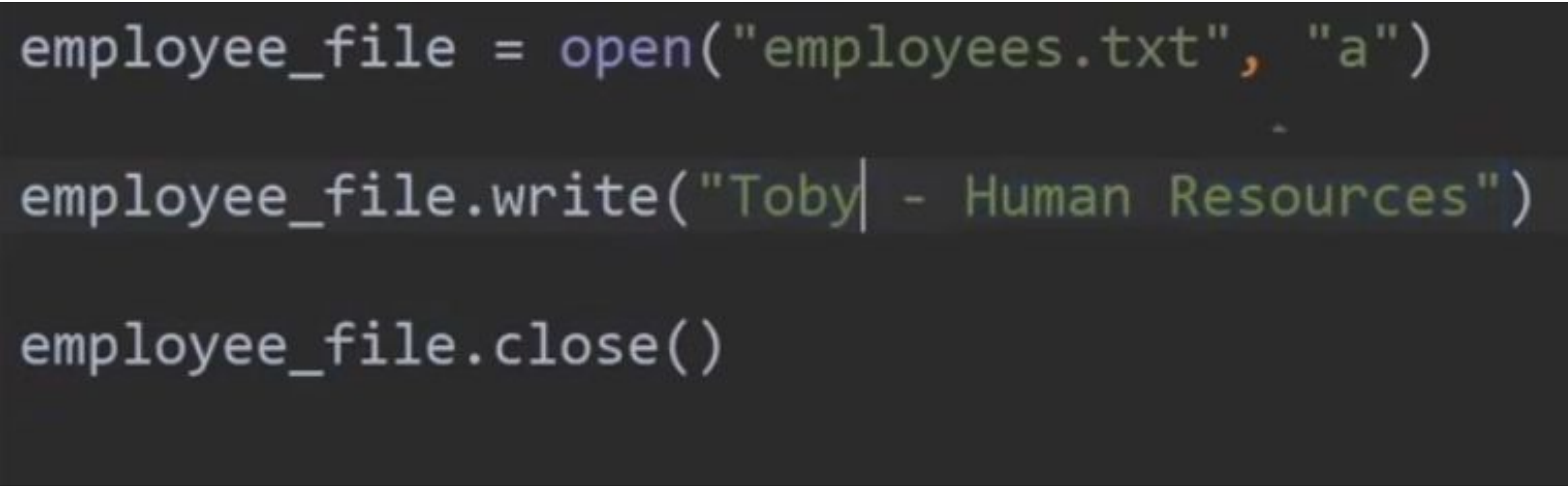
What will this do?

You can also readlines with a for loop

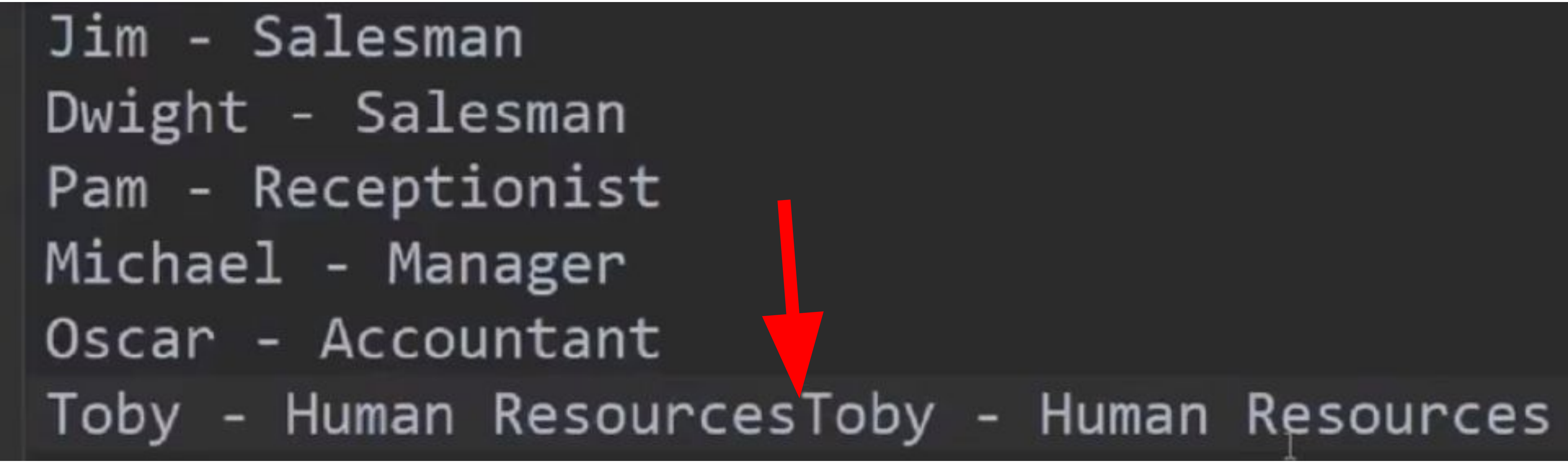


```
employee_file = open("employees.txt", "r")
for employee in employee_file.readlines():
    print(employee)
employee_file.close()
```

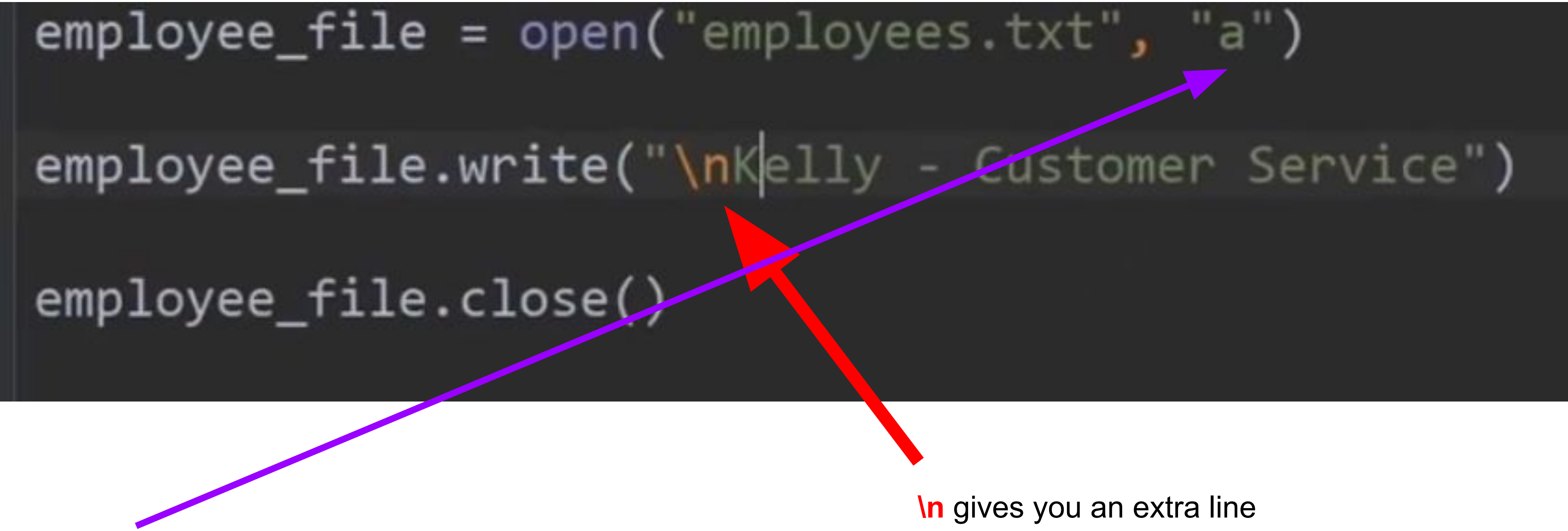
Writing to Files



```
employee_file = open("employees.txt", "a")

employee_file.write("Toby - Human Resources")

employee_file.close()
```



```
Jim - Salesman
Dwight - Salesman
Pam - Receptionist
Michael - Manager
Oscar - Accountant
Toby - Human ResourcesToby - Human Resources
```

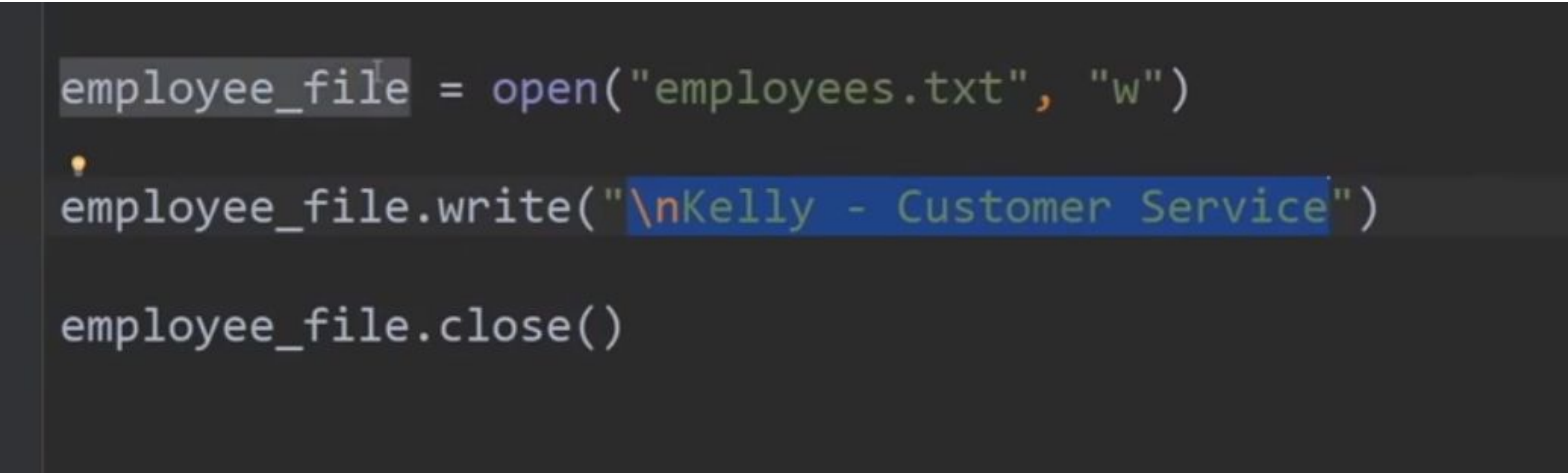


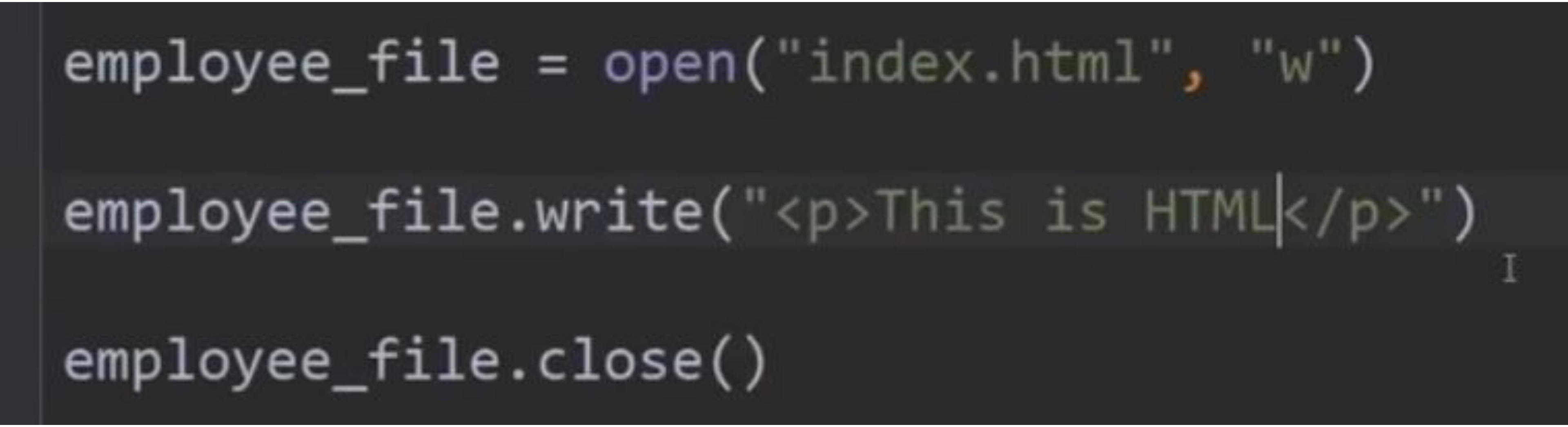
```
employee_file = open("employees.txt", "a")

employee_file.write("\nKelly - Customer Service")

employee_file.close()
```

\n gives you an extra line

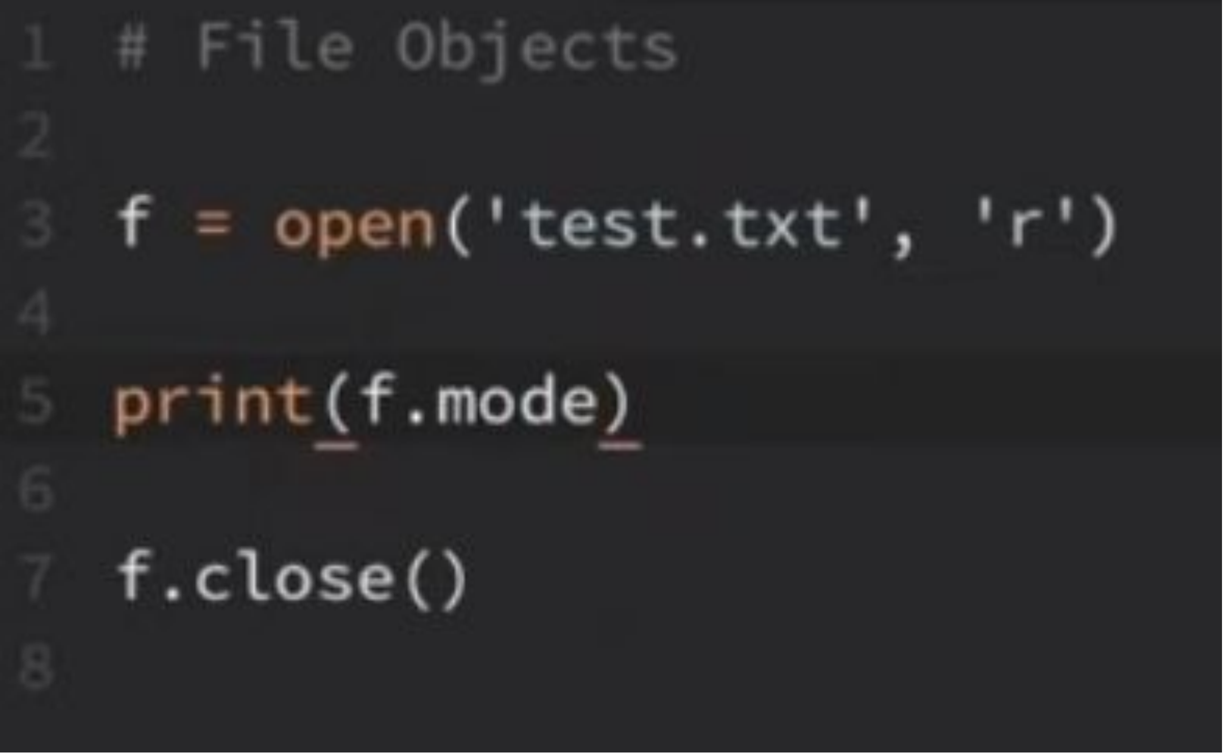
a = lets you append to the file

```
Kelly - Customer Service
```


```python
employee_file = open("employees.txt", "w")

employee_file.write("\nKelly - Customer Service")

employee_file.close()
```



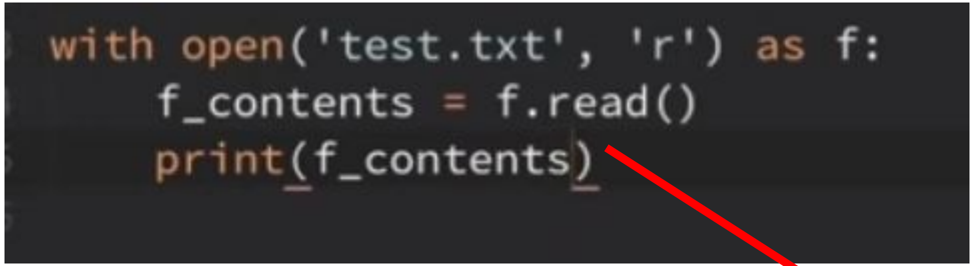
```
employee_file = open("index.html", "w")

employee_file.write("<p>This is HTML</p>")

employee_file.close()
```



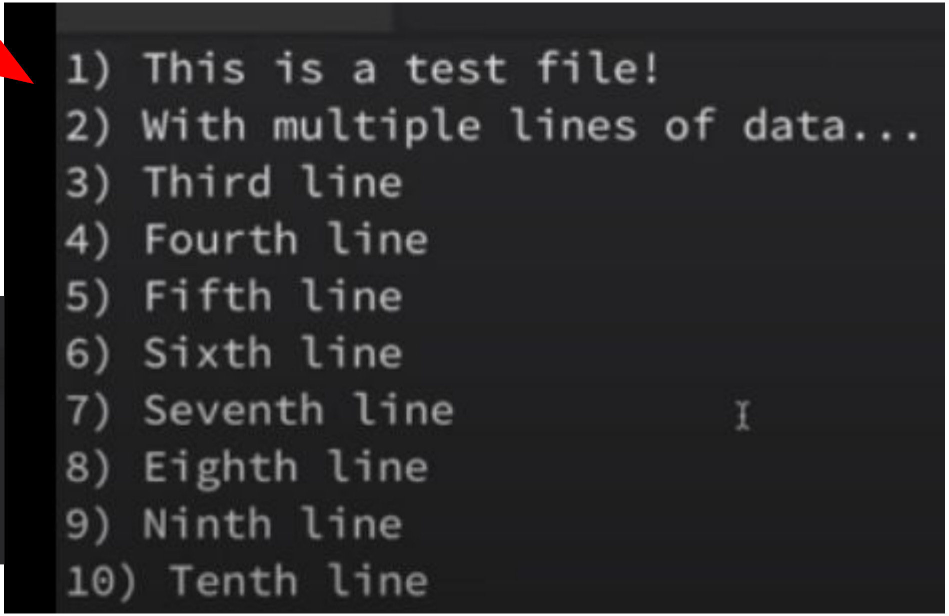
```python
# File Objects

f = open('test.txt', 'r')

print(f.mode)

f.close()

```

You can use this method of opening files ….

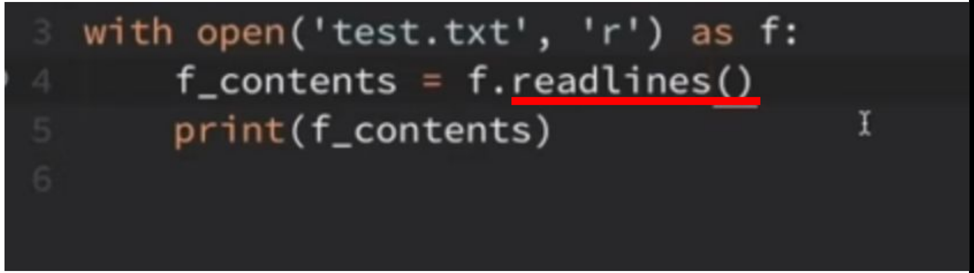
Or you can use a context manager to open files



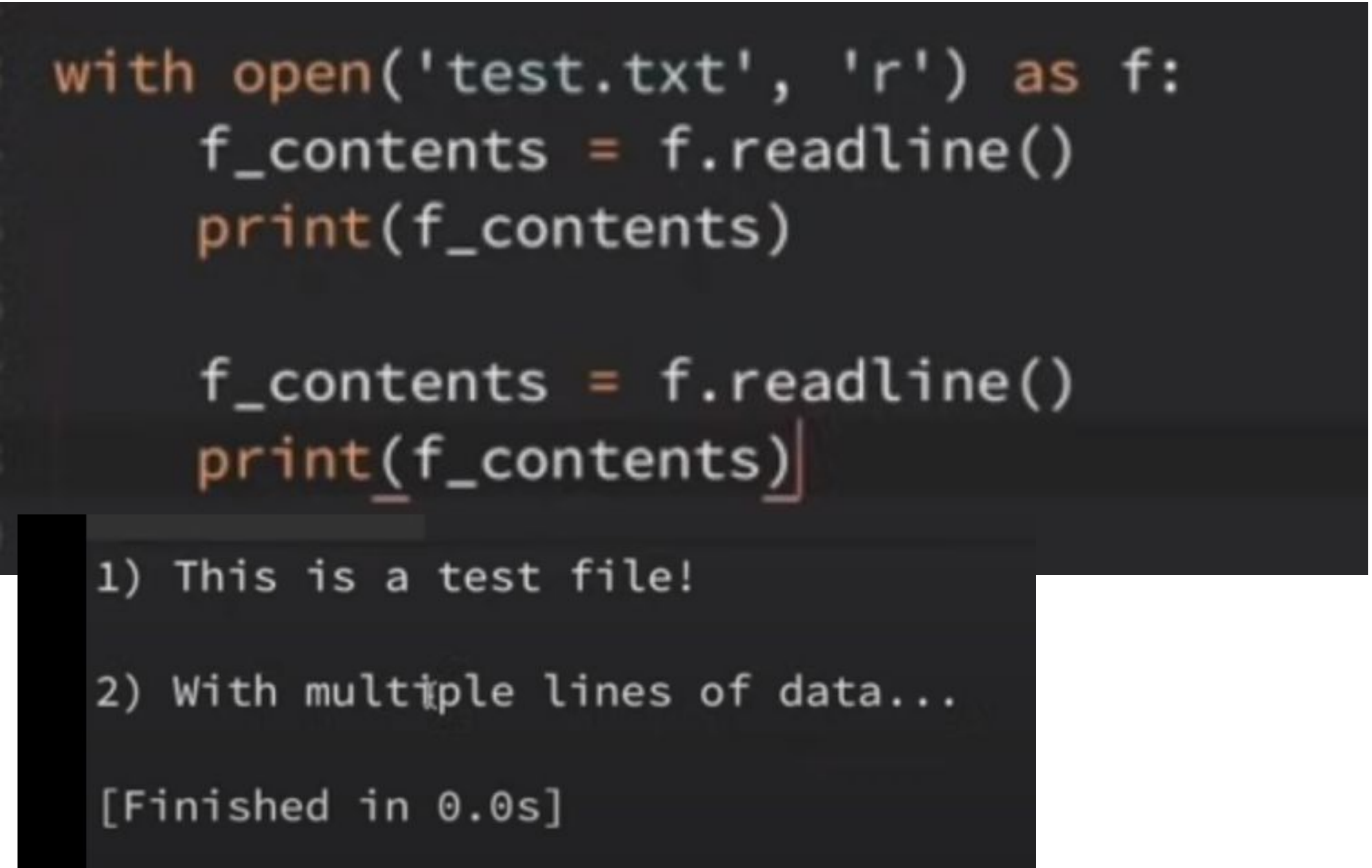
```
with open('test.txt', 'r') as f:
    f_contents = f.read()
    print(f_contents)
```



```
1) This is a test file!
2) With multiple lines of data...
3) Third line
4) Fourth line
5) Fifth line
6) Sixth line
7) Seventh line
8) Eighth line
9) Ninth line
10) Tenth line
```

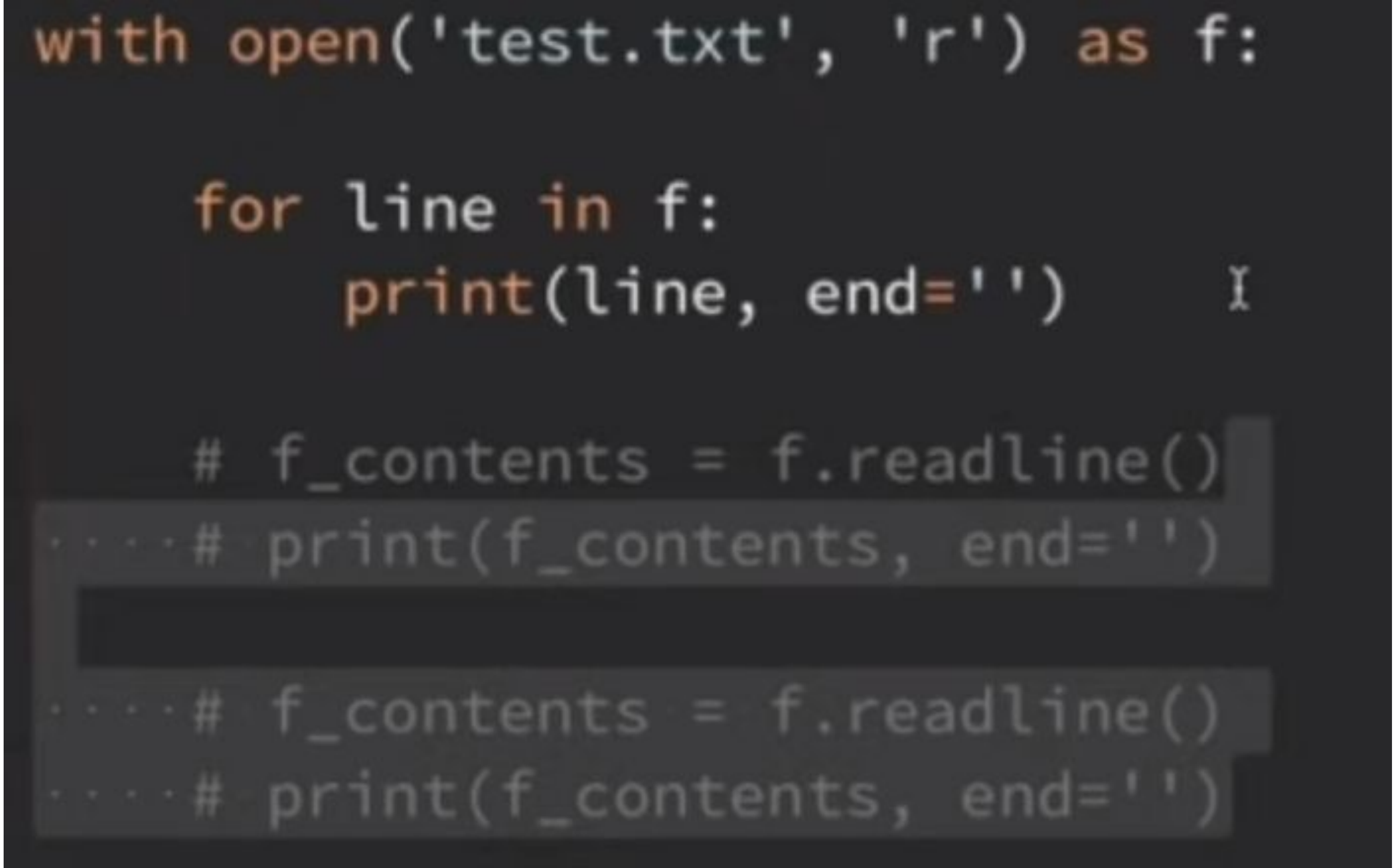


```
with open('test.txt', 'r') as f:
    f_contents = f.readlines()
    print(f_contents)
```

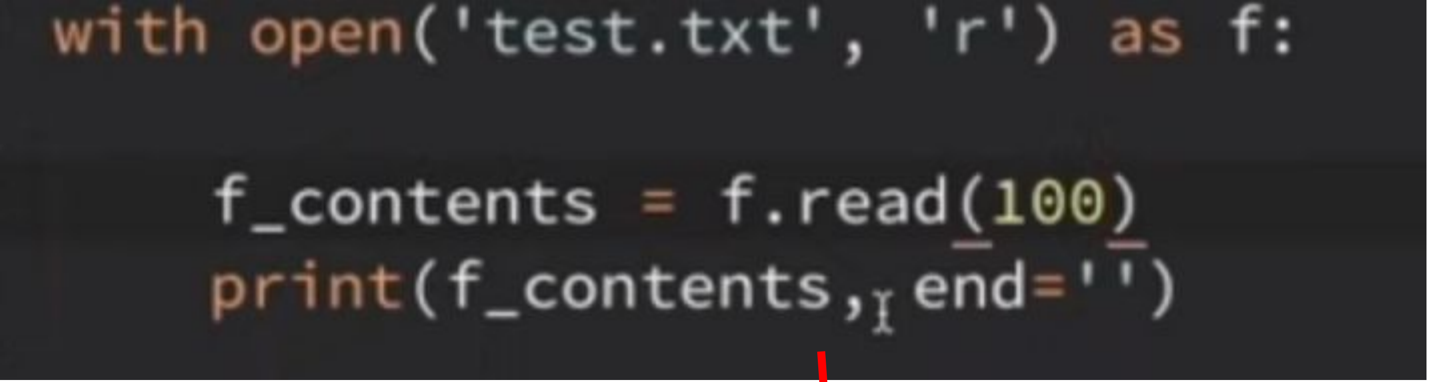


```python
with open('test.txt', 'r') as f:
    f_contents = f.readline()
    print(f_contents)

    f_contents = f.readline()
    print(f_contents)
```

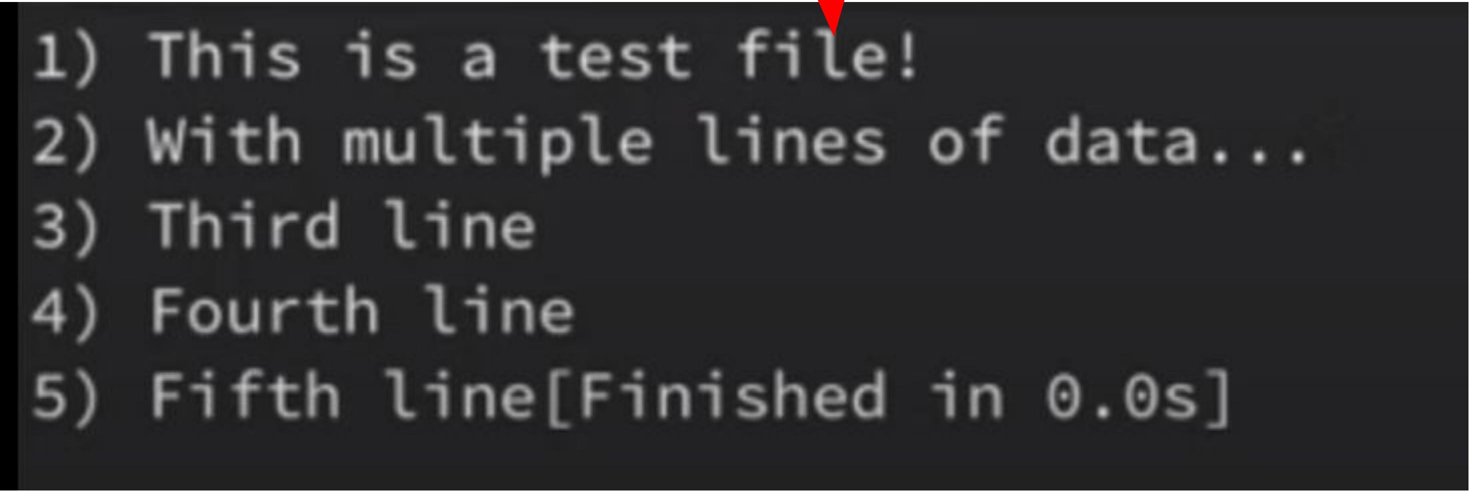
```
1) This is a test file!

2) With multiple lines of data...

[Finished in 0.0s]
```



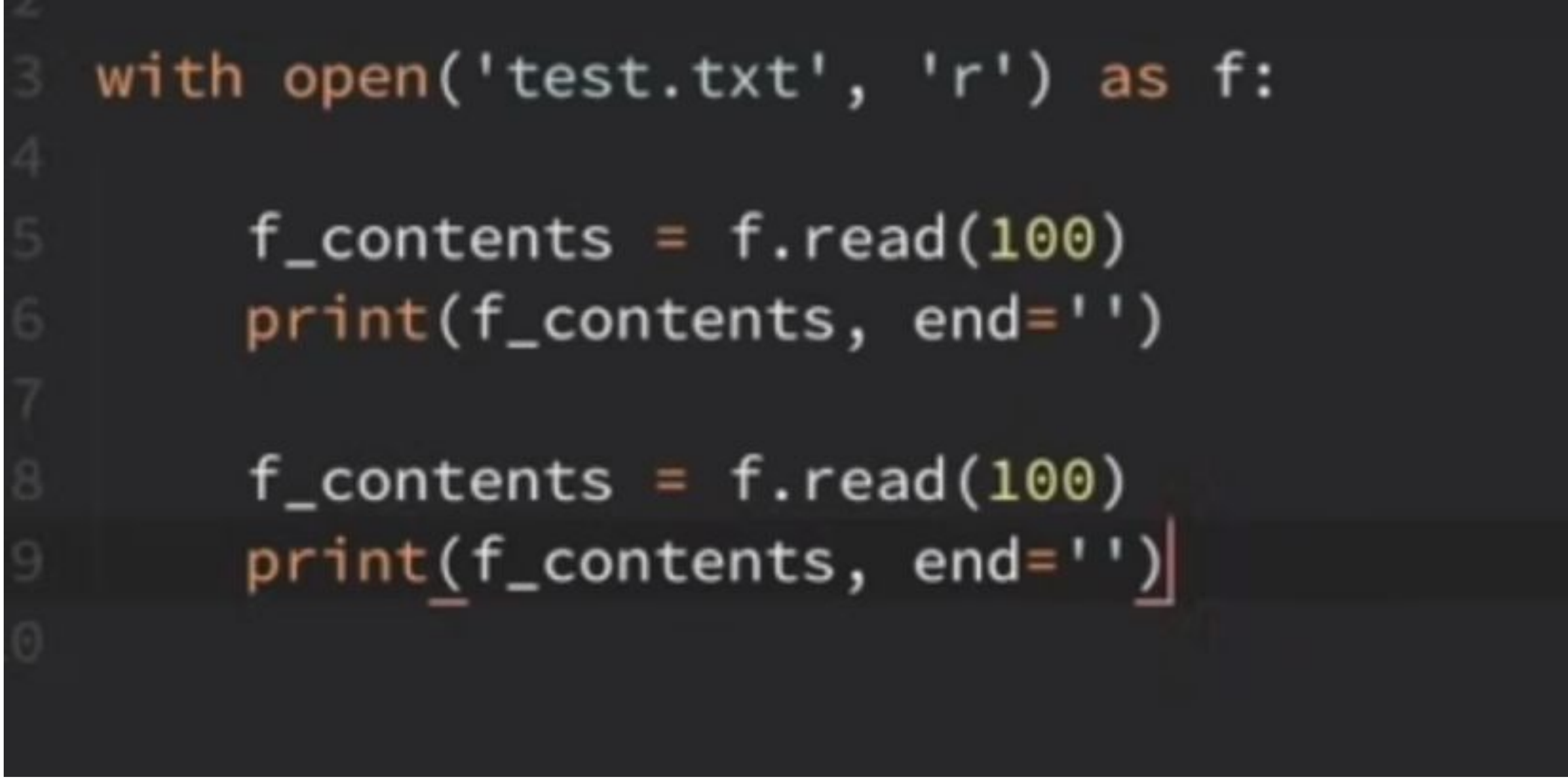
```
with open('test.txt', 'r') as f:

    for line in f:
        print(line, end='')

    # f_contents = f.readline()
    # print(f_contents, end='')

    # f_contents = f.readline()
    # print(f_contents, end='')
```



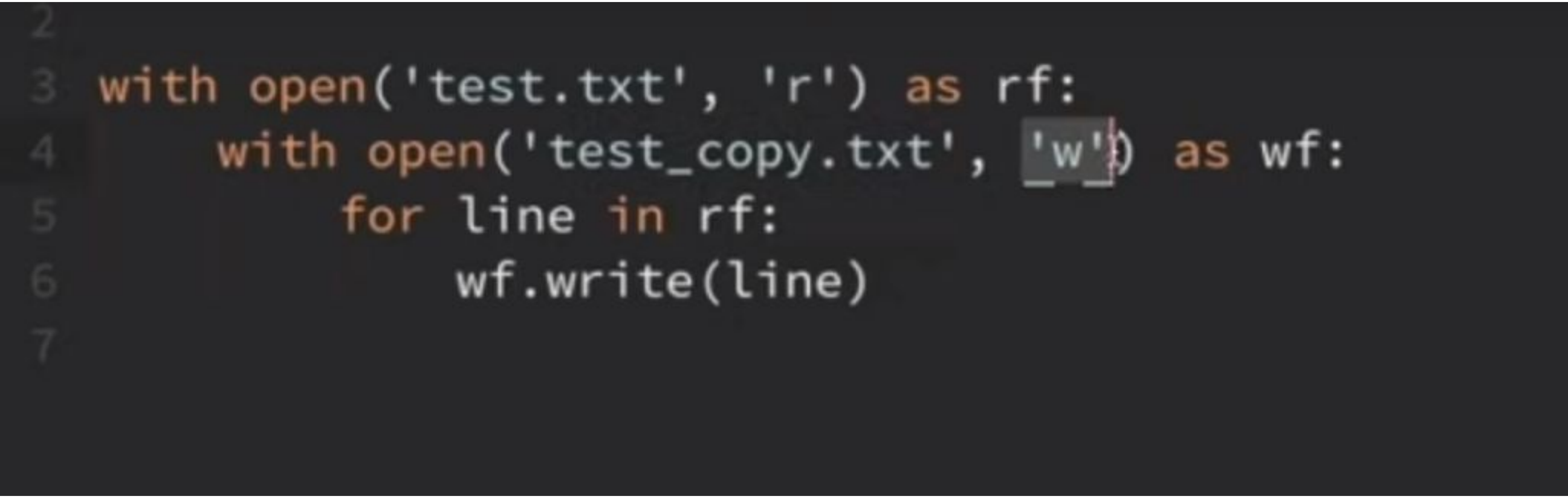
```
with open('test.txt', 'r') as f:

    f_contents = f.read(100)
    print(f_contents, end='')
```



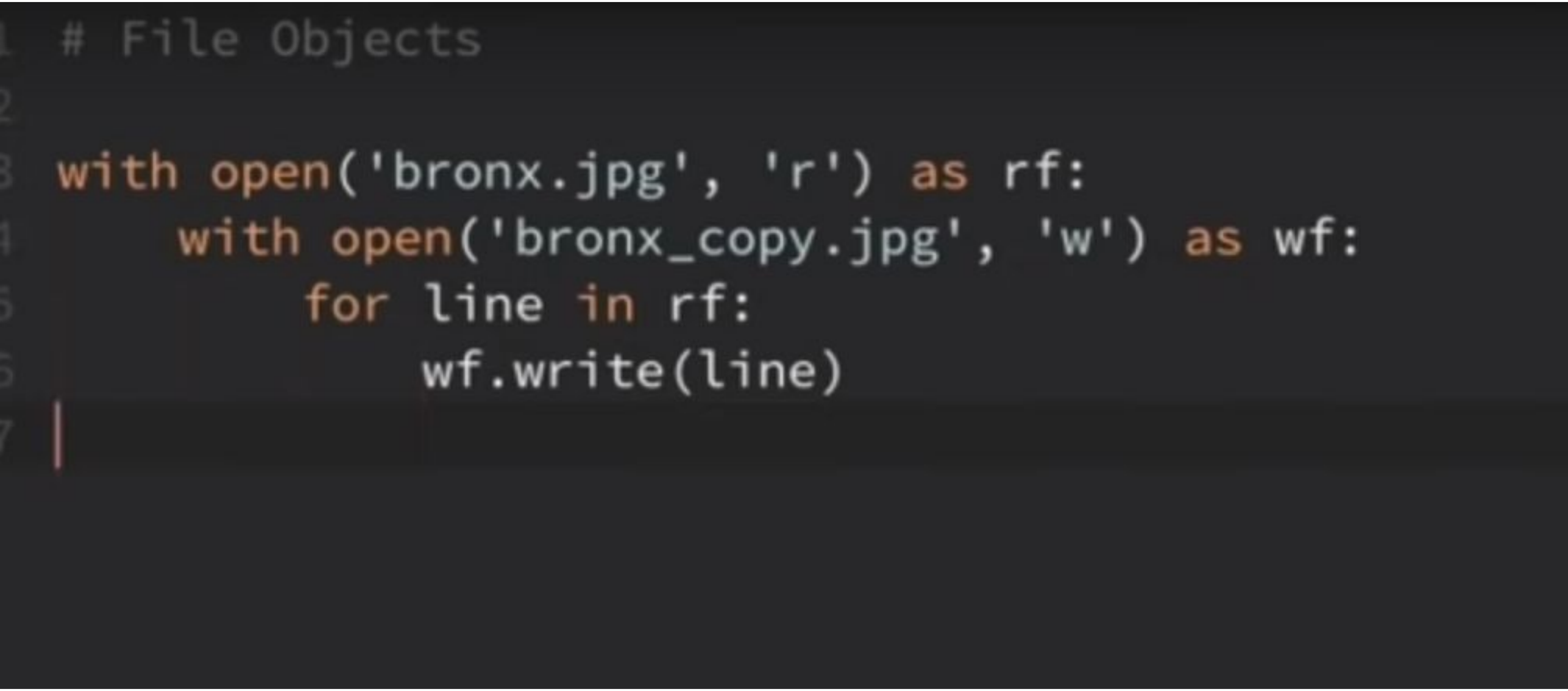
```
1) This is a test file!
2) With multiple lines of data...
3) Third line
4) Fourth line
5) Fifth line[Finished in 0.0s]
```



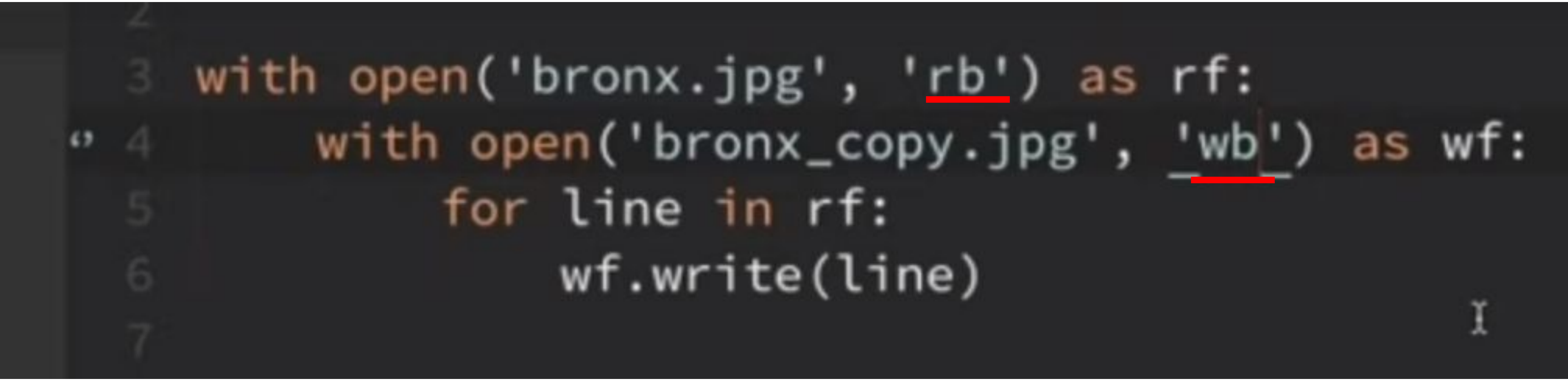
```python
with open('test.txt', 'r') as f:

    f_contents = f.read(100)
    print(f_contents, end='')

    f_contents = f.read(100)
    print(f_contents, end='')
```



```python

with open('test.txt', 'r') as rf:
    with open('test_copy.txt', 'w') as wf:
        for line in rf:
            wf.write(line)

```

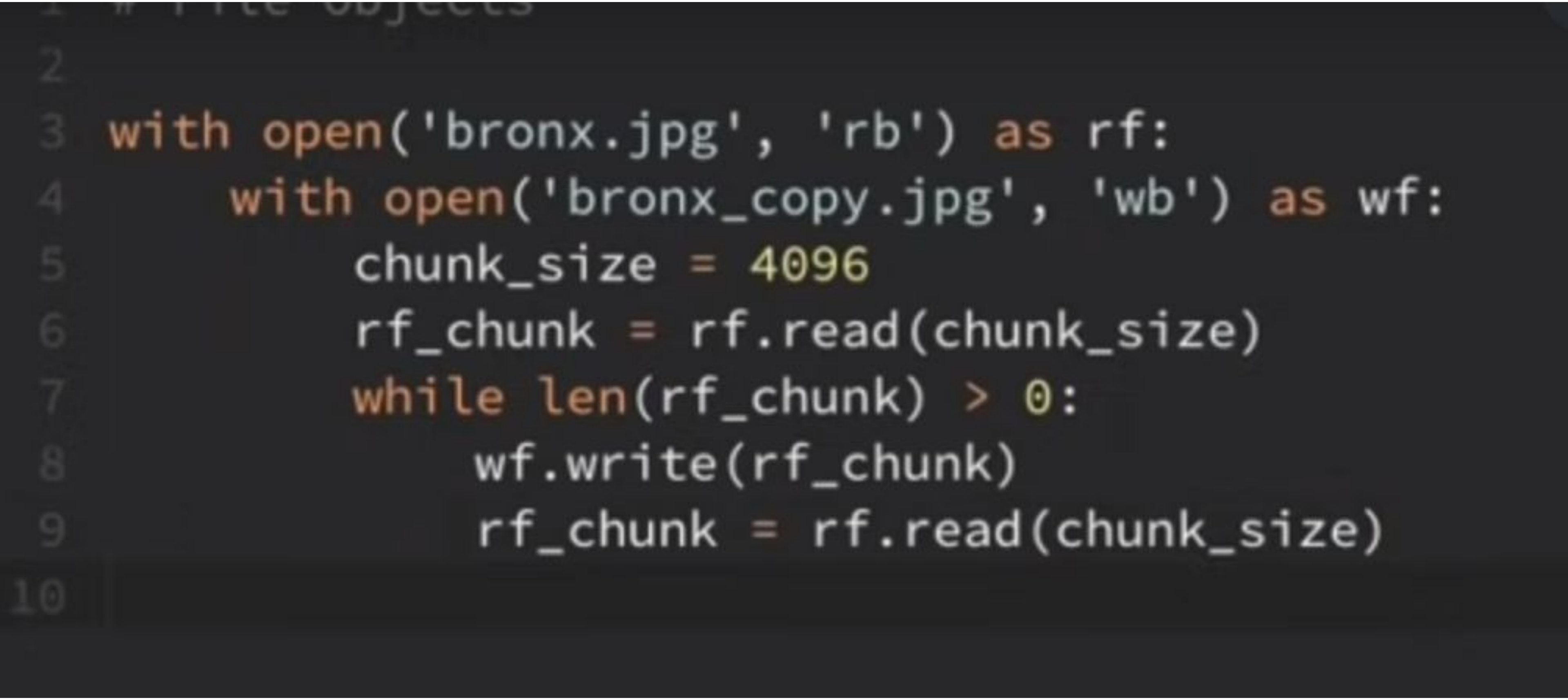


```python
# File Objects

with open('bronx.jpg', 'r') as rf:
    with open('bronx_copy.jpg', 'w') as wf:
        for line in rf:
            wf.write(line)

```



```
with open('bronx.jpg', 'rb') as rf:
    with open('bronx_copy.jpg', 'wb') as wf:
        for line in rf:
            wf.write(line)
```

b stands for binary



```
# File Objects

with open('bronx.jpg', 'rb') as rf:
    with open('bronx_copy.jpg', 'wb') as wf:
        chunk_size = 4096
        rf_chunk = rf.read(chunk_size)
        while len(rf_chunk) > 0:
            wf.write(rf_chunk)
            rf_chunk = rf.read(chunk_size)

```

1. **Read Only ('r') :** Open text file for reading. The handle is positioned at the beginning of the file. If the file does not exists, raises I/O error. This is also the default mode in which file is opened.
2. **Read and Write ('r+') :** Open the file for reading and writing. The handle is positioned at the beginning of the file. Raises I/O error if the file does not exists.
3. **Write Only ('w') :** Open the file for writing. For existing file, the data is truncated and over-written. The handle is positioned at the beginning of the file. Creates the file if the file does not exists.
4. **Write and Read ('w+')** : Open the file for reading and writing. For existing file, data is truncated and over-written. The handle is positioned at the beginning of the file.
5. **Append Only ('a')** : Open the file for writing. The file is created if it does not exist. The handle is positioned at the end of the file. The data being written will be inserted at the end, after the existing data.
6. **Append and Read ('a+') :** Open the file for reading and writing. The file is created if it does not exist. The handle is positioned at the end of the file. The data being written will be inserted at the end, after the existing data.

Exercise with reading and writing files