

PY0101EN-4-2-WriteFile

October 17, 2019

[

](https://cocl.us/PY0101EN_edx_add_top)

Write and Save Files in Python

Welcome! This notebook will teach you about write the text to file in the Python Programming Language. By the end of this lab, you'll know how to write to file and copy the file.

Table of Contents

- [Writing Files](#)
- [Copy a File](#)

Estimated time needed: **15 min**

Writing Files

We can open a file object using the method `write()` to save the text file to a list. To write the mode, argument must be set to write `w`. Let's write a file `Example2.txt` with the line: "This is line A"

```
[1]: # Write line to file

with open('/resources/data/Example2.txt', 'w') as writefile:
    writefile.write("This is line A")
```

We can read the file to see if it worked:

```
[2]: # Read file

with open('/resources/data/Example2.txt', 'r') as testwritefile:
    print(testwritefile.read())
```

This is line A

We can write multiple lines:

```
[3]: # Write lines to file

with open('/resources/data/Example2.txt', 'w') as writefile:
```

```
writefile.write("This is line A\n")
writefile.write("This is line B\n")
```

The method `.write()` works similar to the method `.readline()`, except instead of reading a new line it writes a new line. The process is illustrated in the figure, the different colour coding of the grid represents a new line added to the file after each method call.

You can check the file to see if your results are correct

```
[4]: # Check whether write to file

with open('/resources/data/Example2.txt', 'r') as testwritefile:
    print(testwritefile.read())
```

```
This is line A
This is line B
```

By setting the mode argument to append **a** you can append a new line as follows:

```
[5]: # Write a new line to text file

with open('/resources/data/Example2.txt', 'a') as testwritefile:
    testwritefile.write("This is line C\n")
```

You can verify the file has changed by running the following cell:

```
[6]: # Verify if the new line is in the text file

with open('/resources/data/Example2.txt', 'r') as testwritefile:
    print(testwritefile.read())
```

```
This is line A
This is line B
This is line C
```

We write a list to a `.txt` file as follows:

```
[7]: # Sample list of text

Lines = ["This is line A\n", "This is line B\n", "This is line C\n"]
Lines
```

```
[7]: ['This is line A\n', 'This is line B\n', 'This is line C\n']
```

```
[8]: # Write the strings in the list to text file

with open('Example2.txt', 'w') as writefile:
    for line in Lines:
```

```
print(line)
writefile.write(line)
```

This is line A

This is line B

This is line C

We can verify the file is written by reading it and printing out the values:

```
[9]: # Verify if writing to file is successfully executed

with open('Example2.txt', 'r') as testwritefile:
    print(testwritefile.read())
```

This is line A

This is line B

This is line C

We can again append to the file by changing the second parameter to a. This adds the code:

```
[10]: # Append the line to the file

with open('Example2.txt', 'a') as testwritefile:
    testwritefile.write("This is line D\n")
```

We can see the results of appending the file:

```
[11]: # Verify if the appending is successfully executed

with open('Example2.txt', 'r') as testwritefile:
    print(testwritefile.read())
```

This is line A

This is line B

This is line C

This is line D

Copy a File

Let's copy the file Example2.txt to the file Example3.txt:

```
[12]: # Copy file to another

with open('Example2.txt','r') as readfile:
    with open('Example3.txt','w') as writefile:
```

```
for line in readfile:
    writefile.write(line)
```

We can read the file to see if everything works:

```
[13]: # Verify if the copy is successfully executed

with open('Example3.txt','r') as testwritefile:
    print(testwritefile.read())
```

```
This is line A
This is line B
This is line C
This is line D
```

After reading files, we can also write data into files and save them in different file formats like **.txt**, **.csv**, **.xls (for excel files) etc.** Let's take a look at some examples.

Now go to the directory to ensure the **.txt** file exists and contains the summary data that we wrote.

The last exercise!

Congratulations, you have completed your first lesson and hands-on lab in Python. However, there is one more thing you need to do. The Data Science community encourages sharing work. The best way to share and showcase your work is to share it on GitHub. By sharing your notebook on GitHub you are not only building your reputation with fellow data scientists, but you can also show it off when applying for a job. Even though this was your first piece of work, it is never too early to start building good habits. So, please read and follow this article to learn how to share your work.

Get IBM Watson Studio free of charge!

<p><img src="https://s3-api.us-geo.objectstorage

About the Authors:

Joseph Santarcangelo is a Data Scientist at IBM, and holds a PhD in Electrical Engineering. His research focused on using Machine Learning, Signal Processing, and Computer Vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Other contributors: Mavis Zhou

Copyright © 2018 IBM Developer Skills Network. This notebook and its source code are released under the terms of the MIT License.