File IO

You will learn about Python file operations. More specifically, opening a file, reading from it, writing into it, closing it and various file methods.

Write a Cell to File in Jupytor Notebook

Jupyter Notebook provides a magic function `%%file` to export the content in a cell to a file.

- Note: Any magic function staring with `%%` must be at first line of the cell, and it is applied to all lines below it.

Try Code:

Create a file `quote.txt` by running following code.

```
%%file 'quote.txt'
Give a man a program, frustrate him for a day.
Teach a man to program, frustrate him for a lifetime.
```

In [2]:
```
%%file 'quote.txt'
Give a man a program, frustrate him for a day.
Teach a man to program, frustrate him for a lifetime.
```

```
Writing quote.txt
```

1. Introduction

What is File?

File is a named location on disk to store related information.

- It uses non-volatile memory, e.g. hard disk, to store data permanently.

A file operation takes place in following order:

- Open a file
- Read or Write (perform operations)
- Close the file

A file can be **text** or **binary** format.

Open File

Python has a built-in function `open(file_path)` to open a file.

- The `open()` function returns a file object, also called a file handler, as it is used to read or modify the file accordingly.

To read data from a file object, use its `read()` method.

To close a file object, uses its `close()` method.

Exercise:

Read and print out content in file `quote.txt`, which is created in previous step.

In [3]:
```
f = open('quote.txt')    # open file in current directory
s = f.read()
print(s)
f.close()
```

```
Give a man a program, frustrate him for a day.
Teach a man to program, frustrate him for a lifetime.
```

Question:

What will be the returned value of `read()` function when the function is called second time?

In [11]:
```
f = open('quote.txt')    # open file in current directory
s = f.read()
# f.seek(0)  # Move pointer to start of file
s = f.read()
print(s)
f.close()
```

Filepath

The `filepath` can be a relative or absolute path.

- If only file name is specified, Python assume the file is in the same folder as current Python kernel
- When specifying full path, use `/` instead of `\`, which is used as escape character in string

In [10]:
```
import os
path = os.path.join(os.path.abspath(''), 'quote.txt')
print(path)

f = open(path)  # specifying full path
f.close()
```

```
D:\GoogleDrive\Learn-Python\Python-MOE-Teacher-Training-2020\09 File IO\quo
te.txt
```

Question:

What if I foget to close the file?

```
For Read operation, unclosed file objects will be garbage collected.

But for Write operation, close() function will flush content in buffer to the file before closing it. The content may not be written to file if the file is not closed.
```

2. Operation Mode

You can specify the mode used to open a file by applying a second argument to the open function.

- `r` / `w` / `a` : Are you reading, writing or appending a file?
- `t` / `b` : Is it a text or binary file?

```
f = open(filepath, mode)
```

Read / Write / Append

The `mode` specifies how you want to work with the file.

- `'r'` : read mode, which is the default.
- `'w'` : write mode, for overwriting the contents of a file. Existing file content will be lost.
- `'a'` : append mode, for appending new content to the end of the file. Existing content in the file will not be lost.

Exercise:

Complete following operations using `with` code block:

- Write `"Alexa, "` to a file `test.txt` . This operator will overwrite any content in the file.
- Append `"Good morning!\n"` to the file `test.txt` .
- Append `"What time is it?"` to the file `test.txt` .
- Read and print out content from the file `test.txt` .

In [44]:
```
# write/create a file
with open("test.txt", "w") as f:
    f.write("Alexa, ")

# append/create a file
with open("test.txt", "a") as f:
    f.write("Good morning!\n")
    f.write("What time is it?")

# read a file
with open("test.txt", "r") as f:
    s = f.read()
#     print(repr(s))
    print(s)
```

```
Alexa, Good morning!
What time is it?
```

Write a String

The `write()` method returns number of characters written to the file.

- Note: It counts special characters too.

In [1]:
```
s = "Alexa\tGood morning!\nWhat time is it?"
print(len(s))
with open("test.txt", "w") as f:
    x = f.write(s)
    print(x)
```

```
36
36
```

Open the file `test.txt` and examine text inside it. All special characters are handled properly.

Try Code:

```
!notepad test.txt
```

In [38]:
```
!notepad test.txt
```

Read by Lines

Compared to `read()` function, which return all content in a single string, the `readlines()` function returns a list, where each item contains a line.

NOTE: No character is removed, e.g. new line character `\n` at the end of a line.

In [54]:
```
with open('test.txt') as f:
    s = f.read()
    print(repr(s))

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

```
'HelloWorld\nfrom\nSingapore'
['HelloWorld\n', 'from\n', 'Singapore']
```

In fact, file object can be passed directly to `for-loop`, which will iterate through the file line by line.

In [8]:
```
f = open('test.txt')

for i in f:
    print(i)
```

```
Alexa   Good morning!

What time is it?
```

Question:

- Why above there is an empty line between `'Alexa   Good morning!'` and `'What time is it?'`?
- How to remove trailing `\n` from each line?

```
1
```

Write Multiple Lines

To write a list of stirngs to a file, method `writelines()` can be used.

NOTE: No character, e.g. `\n`, will be added or removed.

Try Code:

```
s = ['Hello', 'World', '\nfrom', '\nSingapore']
with open('test.txt', 'w') as f:
    f.writelines(s)
```

Use `!notepad test.txt` to examine the content in the file.

In [1]:
```
s = ['Hello', 'World', '\nfrom', '\nSingapore']
with open('test.txt', 'w') as f:
    f.writelines(s)
```

In [2]:
```
!notepad test.txt
```

3. Text File / Binary File

By default, `open()` assumes the file is a **text** file. To work with **binary** files, e.g. images and sound files, adding `"b"` to the `mode`.

- Use `rb` to read a binary file
- Use `wb` to write binary content to a file

Excercise:

Write code to copy an image file `test.jpg` in `./images/` folder to current folder with name `dup.jpg`.

In [15]:
```
## Read binary content of a image and create a duplicate image
with open('./images/test.jpg', 'rb') as f:
    s = f.read()

with open('dup.jpg', 'wb') as f:
    f.write(s)
```