# File_Handling_Python

April 19, 2020

[15]:
```
#File should opened using open statement
file = open("sample.txt", "w")
```

[16]:
```
#File should always be closed using close method
file.close()
```

[21]:
```
#Ensuring a file is closed even if an error occurrs
file = open("sample.txt")
try:
    pass

finally:
    file.close()
```

[22]:
```
with open('sample.txt') as file_obj:
    pass
#File is automatically closed. #Context manager with
```

[24]:
```
with open('sample.txt') as file:
    content = file.read(5)
    print(content) #5 bytes space is invisible
```

```
This
```

## 0.1 reads a single line from a file

[25]:
```
with open('sample.txt') as file:
    content = file.readline()
    print(content)
```

```
This is the first line.
```

[26]:
```
with open('sample.txt') as file:
    content = file.readline() #reads a single line from a file
    print(content)
    content = file.readline() #reads a single line from a file
    print(content)
```

```
This is the first line.

This is the second line.
```

## 0.2 readlines method reads the content of the file as a list

[28]:

```
with open('sample.txt') as file: #File opened in read mode if no arguments are␣
 ↪given and it is default mode.
    content = file.readlines()
    print(content)

print(file.mode)
```

```
['This is the first line.\n', 'This is the second line.\n', 'This is the last
line.']
r
```

[30]:

```
import os
folder_path = os.path.join("Users", "a", "Downloads")
print(folder_path) #backlash
```

```
Users\a\Downloads
```