Reading from a file

Week 7

Opening a file for reading

- Use the `open()` function to open a file.
- You need to specify the **file name** and the **mode** in which you want to open the file (*read*, *write*, *append*).

Example. `file = open("Example.txt", "r")`

- The variable `file` is called a **file object** or **file handle**.
- It acts as a reference to `Example.txt`, allowing you to interact with it.
- The command gives an error message if the file does not exist:
  `No such file or directory: 'example.txt'`

Example.txt

Let us consider the following file:

```
Hello, this is a sample text file.
It contains multiple lines of text.
Python can read this file.
This is the fourth line.
```

`readline()`

- The `readline()` method reads one line at a time from the file. This is useful if you want to process the file line by line.
- If there are **no more lines** to read, `readline()` will return an empty string: `''` (two quotation marks)

```
>>> file = open("Example.txt", "r")
>>> file.readline()
'Hello, this is a sample text file.\n'
>>> file.readline()
'It contains multiple lines of text.\n'
>>> file.readline()
'Python can read this file.\n'
>>> file.readline()
'This is the fourth line.\n'
>>> file.readline()
''
>>> file.close()
```

readline()

readline1.py - C:\Users\Jouni\OneDrive - LUT University\Desktop\readline1.py (3.11.9)

File Edit Format Run Options Window Help

```
file = open("Example.txt", "r")
line = file.readline()
while line != "":
    print(line)
    line = file.readline()
file.close()
```

```
Hello, this is a sample text file.

It contains multiple lines of text.

Python can read this file.

This is the fourth line.

>>>
```

You can use loop to read the content of the file line-by-line.

Then you can print each line.

There seems to be an additional empty lines between the text.

`strip()` method

- The `strip()` method in Python is used to remove any **leading** and **trailing** whitespace characters from a string.
- This includes spaces, tabs, and newline characters.
- This is useful when reading lines from a file and to get rid of extra newline characters.

```
>>> text = "Hello, World! \n"
>>> text
'Hello, World! \n'
>>> stripped_text = text.strip()
>>> stripped_text
'Hello, World!'
```

readline() + strip()

readline2.py - C:/Users/Jouni/OneDrive - LUT University/Desktop/readline2.py (3.11.9)

File Edit Format Run Options Window Help

```
file = open("Example.txt", "r")
line = file.readline().strip()
while line != "":
    print(line)
    line = file.readline().strip()
file.close()
```

```
= RESTART: C:/Users/Jouni/OneDrive - LUT Ur
Hello, this is a sample text file.
It contains multiple lines of text.
Python can read this file.
This is the fourth line.
>>>
```

`readlines()`

- In Python, the `readlines()` method reads all lines from a file and returns them as a **list of strings**.
- Each string in the list represents a line from the file, including the newline character `\n` at the end of each line.
- The **last line** of the file may not have a newline character.

```
>>> file = open("Example.txt", "r")
>>> lines = file.readlines()
>>> lines
['Hello, this is a sample text file.\n', 'It contains multiple lines of text.\n',
'Python can read this file.\n', 'This is the fourth line.\n']
>>> file.close()
```

readlines()

readlines.py - C:\Users\Jouni\OneDrive - LUT University\Desktop\readlines.py (3.11.9)

File Edit Format Run Options Window Help

```
file = open("Example.txt", "r")
lines = file.readlines()

stripped_lines = []

# Iterate over each line and strip whitespace
for line in lines:
    stripped_lines.append(line.strip())

# Print the stripped lines
for line in stripped_lines:
    print(line)

file.close()
```