File Handling

Although most of the libraries we will use in the course have inbuilt functionality to save and load files, it is sometimes useful to being able to handle files manually.

In Python, `open()` will open a file, whether it is for reading or for writing, and calling `close()` on the variable referencing the file (this is called a *file handler*) will close it again. It's important to make sure that a file is closed after usage to avoid stale file handlers and memory hogging. `open()` has many arguments it can take, we'll just look at the most important ones here. One way to open a file for writing is the following, where the argument `"w"` is provided to indicate that the file should be *overwritten*:

In [10]:

```
textlist = ["some", "text in form of", "a", "list"]

fh = open("test.txt", "w")

fh.write("testfile\n")
fh.write("some more text\n")
fh.writelines(textlist)

fh.close()
```

Since `test.txt` did not exist before executing the above cell, it was created. The `write` statements write the provided strings into the file, *without* a newline at the end. `writelines` write a sequence (here, a list) of strings into a file line by line. Let's see what happened by opening the file in read mode with `"r"` :

In [11]:

```
filereader = open("test.txt", "r")

print(filereader.read())

filereader.close()
```

```
testfile
some more text
sometext in form ofalist
```

`read` gets every line separately. We can also use `readlines` to get a list of all the lines in the file:

In [12]:

```
filereader = open("test.txt", "r")

print(filereader.readlines())

filereader.close()
```

```
['testfile\n', 'some more text\n', 'sometext in form ofalist']
```

For appending to a file instead of overwriting it, the option "a" has to be given for `open()` :

In [15]:

```
fh = open("test.txt", "a")

fh.write("A new line\n")
fh.write("without changing the rest\n")

fh.close()
```

Let's check the result:

In [16]:

```
filereader = open("test.txt", "r")

print(filereader.read())

filereader.close()
```

```
testfile
some more text
sometext in form ofalistA new line
without changing the rest
A new line
without changing the rest
```

Sometimes it's useful to iterate oover all the lines in a file:

In [19]:

```
filereader = open("test.txt", "r")

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

filereader.close()
```

```
testfile
some more text
sometext in form ofalistA new line
without changing the rest
A new line
without changing the rest
```

Sometimes it's not clear whether a file already exists and overwriting it would yield catastrophic data loss. For creating new files without overwriting existing files, the option "x" is useful. In this case, an error is thrown since "test.txt" already exist:

In [20]:

```
fh = open("test.txt", "x")

fh.write("New file\n")
fh.write("but only if it didn't exist before\n")
fh.writelines(textlist)

fh.close()
```

```
---------------------------------------------------------------------------
-----
FileExistsError                           Traceback (most recent call
 last)
<ipython-input-20-32c15536a01c> in <module>
----> 1 fh = open("test.txt", "x")
      2 
      3 fh.write("New file\n")
      4 fh.write("but only if it didn't exist before\n")
      5 fh.writelines(textlist)

FileExistsError: [Errno 17] File exists: 'test.txt'
```