Day 28: File Handling

File Handling

The key function for working with files in Python is the `open()` function.

The `open()` function takes two parameters; *filename*, and *mode*.

There are four different methods (modes) for opening a file:

`r`	Read	Default value. Opens a file for reading, error if the file does not exist
`a`	Append	Opens a file for appending, creates the file if it does not exist
`w`	Write	Opens a file for writing, creates the file if it does not exist
`x`	Create	Creates the specified file, returns an error if the file exists

In addition you can specify if the file should be handled as binary or text mode.

`t`	Text	Default value. Text mode
`b`	Binary	Binary mode (e.g. images)

Syntax

To open a file for reading it is enough to specify the name of the file:

```
f = open("demofile.txt")
```

The code above is the same as:

```
f = open("demofile.txt", "rt")
```

Python Read Files

Assume we have `file_1.txt` file, located in the same folder as the python code:

```
This is a test file.
```

To open the file, use the built-in `open()` function.

The `open()` function returns a file object, which has a `read()` method for reading the content of the file:

```
f = open("file_1.txt")
print(f.read())
```

If the file is located in a different location, you will have to specify the file path, like this:

```
f = open("path_to_file/file_1.txt")
print(f.read())
```

Using the with statement

You can also use the `with` statement when opening a file:

```
with open("file_1.txt") as f:
  print(f.read())
```

Then you do not have to worry about closing your files, the `with` statement takes care of that.

Close Files

It is a good practice to always close the file when you are done with it.

If you are not using the `with` statement, you must write a close statement in order to close the file:

```
f = open("file_1.txt")
print(f.readline())
f.close()
```

Read Only Parts of the File

By default the `read()` method returns the whole text, but you can also specify how many characters you want to return:

```
with open("file_1.txt") as f:
  print(f.read(5))
```

Read Lines

You can return one line by using the `readline()` method:

```
with open("file_1.txt") as f:
  print(f.readline())
```

By calling `readline()` two times, you can read the two first lines:

```
with open("file_1.txt") as f:
  print(f.readline())
  print(f.readline())
```

By looping through the lines of the file, you can read the whole file, line by line:

```
with open("file_1.txt") as f:
  for x in f:
    print(x)
```

📝 Test