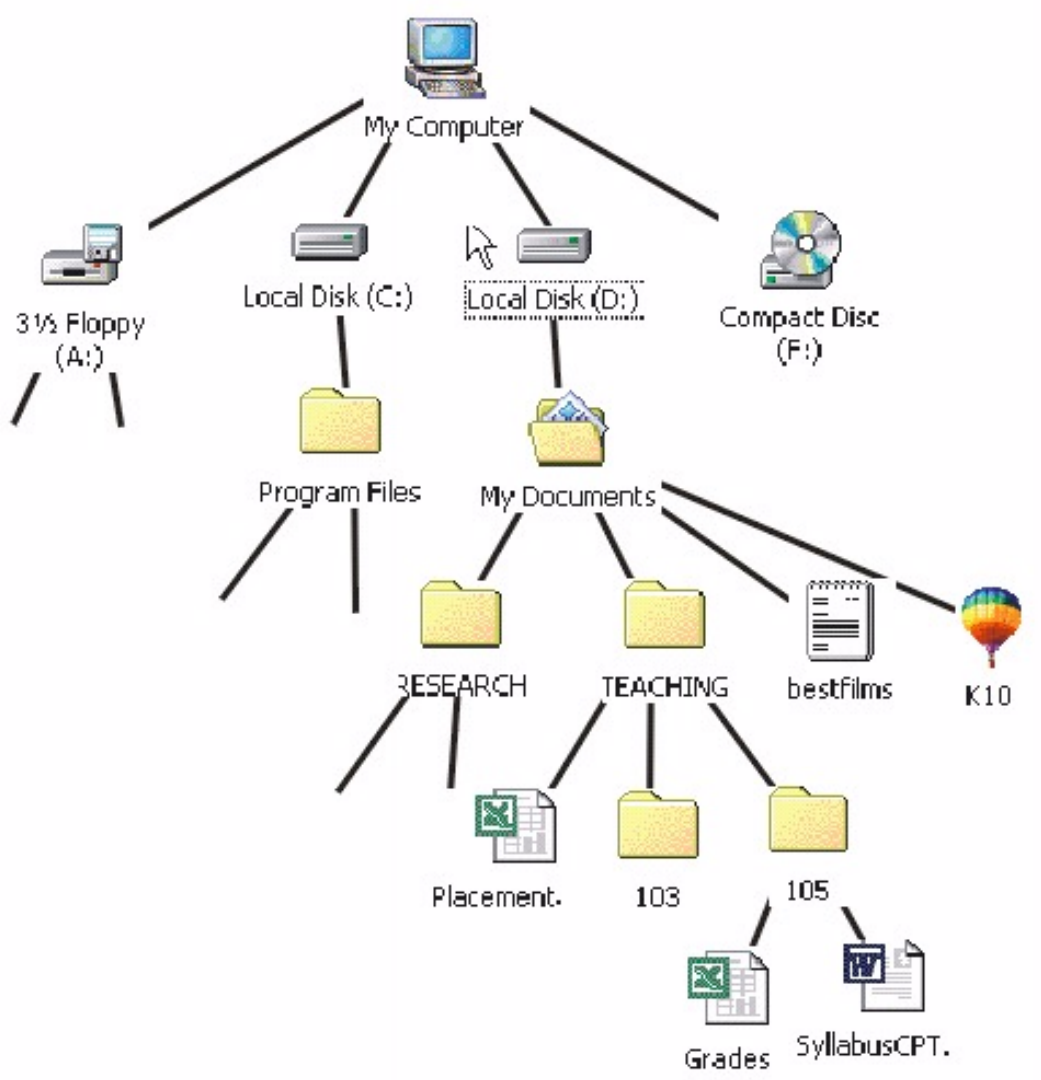
Files I/Os

Files, folders and working directory



Files, folders, and working directories

- On a smartphone, files and folders are often hidden
- Even computers start hiding them
- It is easier for some users
- However, it becomes harder when you:
  - want to attach a file to an email – where is the file?
  - want your files to stay organized – how do we do that?
  - want to learn software developments: it relies heavily on file structures!

Files, folders, and working directories

- Every program has a working directory
- It is usually the folder in which the python/Jupyter file is saved
- Being aware of this active directory will help you open or save files from your programs

In Python, we'll use the ‘open` function to open a file. This requires two arguments, the path of the file to read as well as the opening mode. We will use a variable that will store the information.

```
1 file_in = open("example.txt","w")
```

An essential parameter of the ‘open` function is the opening mode. this is a character string that specifies how you want to open the file. The mode chosen will determine what you can do with the file. Here are the commonly used opening modes:

- 'w': used to read a file;
- 'r': used to read a file;
- 'x': used to create and write to a new file;
- 'a': used to append text to an existing file;
- 'r +': used to read and write to a file;

The `close` instruction is used to close the open file.

```
1 file_in.close()
```

It is important to note that when creating a text file, if no absolute path is specified, the file will be created at the root (in the same directory) as the notebook running the command.

Opening and closing a text file

My file won't open!

In the previous example, we opened "example.txt"

This can only work if example.txt is a file in the working directory of our program

This means that example.txt has to be in the same folder as our program or notebook

What if my file is elsewhere?

- If your file is **not** in the working directory of your program, you need to access it using the file's path
- The file's absolute path is something like:
  - "c:/my_folder/my_sub_folder/my_file.txt" on windows
  - "/my_folder/my_sub_folder/my_file.txt" on mac/linux
- The file's relative path looks like:
  - "../../my_cousin_folder/its_subfolder/my_file.txt"
  - '../' means "one folder up
  - It is realtive to the working directory

Writing in a file

- Once you open a file with “w” or “a” options, you can write on it
- “w” mode will write at the beginning of the file
- “a” (append) mode will write at the end of the file
- file.write(“my text”) (Remember what happens when a function is written as “.write(…)” or “.pop()” ?)
- Don’t forget to close !

Examples

```
1 file_in = open("example_write.txt","w")
2 file_in.write("First line")
3 file_in.close()
```

```
1 file_in = open("example_write.txt","a")
2 file_in.write("Next line")
3 file_in.close()
```

```
1 file_in = open("example_write.txt","a")
2 file_in.write("Writing next line\n")
3 file_in.write("Writing text in the next line")
4 file_in.close()
```

Reading a file

- .readlines() reads all the lines at once and returns them as a list of lines
- .read() reads all the lines and returns them as a string with \n's
- .readline() (no s) allors you to read one line at a time. It can be used in a loop!

Readlines

```
1  # Writing the text file
2  file = open("example_read.txt","x")
3  file.write("Margaux\nThéo\nSimon\nSophie")
4  file.close()
5
6  # Reading of the text file
7  file = open("example_read.txt","r")
8  content = file.readlines()
9  file.close()
10 print(content)
11
12 # Deleting the text file
13 import os
14 os.remove("example_read.txt")
```

```
['Margaux\n', 'Théo\n', 'Simon\n', 'Sophie']
```

Read

```python
# Write the text file
file = open("example_read.txt","x")
file.write("Margaux\nThéo\nSimon\nSophie")
file.close()

# Reading the text file
file = open("example_read.txt","r")
contenu = file.read()
file.close()
print(contenu)

#
import os
os.remove("example_read.txt")
```

```
Margaux
Théo
Simon
Sophie
```

Readline

```
# Writing the text file
file = open("example_read.txt","x")
file.write("Margaux\nThéo\nSimon\nSophie")
file.close()

# Reading the text file
file = open("example_read.txt","r")
contenu = file.readline()
file.close()
print(contenu)

# Deleting the text file
import os
os.remove("example read.txt")
```

```
Margaux
```

The “with” keyword

With allows for a “cleaner” way to write code

It wraps the open function so that you don’t have to use close at the end

```
1  # Writing text file
2  file = open("example_read.txt","x")
3  file.write("Margaux\nThéo\nSimon\nSophie")
4  file.close()
5
6  # Reading of text file
7  with open("example_read.txt", 'r') as filin:
8      lignes = filin.readlines()
9      for ligne in lignes:
10         print(ligne)
11
12 # Deleting the text file
13 import os
14 os.remove("example_read.txt")
```

```
Margaux

Théo

Simon

Sophie
```