

Laboratory 2: Dealing with files and directories

Python allows us to interact with the file system. Key is the module `os`, which you need to import.

Remember that Windows and Unix machines use different conventions for filenames and paths. Unix uses forward slashes, whereas windows uses backward slashes. In addition, an absolute path in windows starts with a drive number, as in

```
'c:\Users\Thomas\Documents\PythonClass\lab.tex'
```

When using Windows pathnames, it is always safer to always escape the backward slash, so that the preceding absolute path name becomes

```
'c:\\Users\\Thomas\\Documents\\PythonClass\\lab.tex'.
```

- (1) Use the `os.listdir` function to obtain a list of all files in a directory on your machine. Then display the file names that end in `'.py'`. You can use the `endswith` method in order to verify that a filename ends with `'.py'`.

You can obtain more data on files by using the `os.path` submodule, which you also need to import. You should consult the corresponding manual page for the `os.path` submodule, which you can find under: <https://docs.python.org/3/library/os.path.html>.

First, we want to get the MAC (Modified, Accessed, Created) times of a file. We can use the `os.path.getatime`, `os.path.getmtime`, and `os.path.getctime` methods, which given the path of a file will return the number of seconds since the last epoch, which for Unix is January 1, 1970 midnight UTC. We can use the `time` module to convert these values to a normal date-time field, using the `ctime` method.

- (2) Use the `os.path.getatime`, `os.path.getmtime`, and `os.path.getctime` to find the access times for the python files obtained previously.
- (3) Similarly, use `os.path.getsize` to add the size of the file.