# Project I

Create a software 'product' (Call it FilePro), which will be a python class that analyzes the folder1 provided to you (and all sub-folders and files within those folders) and does some operations on it.

This would be accessed from the file "main_notebook.ipynb". You may have (one or more) .py or .ipynb files to host your code, but it's the "main_notebook.ipynb", where the code will be accessed, demonstrated and also documented (using markdown cells). Also, use comments in your code to document the functionality of your code and make it more readable.

**Required functionality:**

**Note**: All methods are mentioned with the suffix parenthesis (…). But the implementation of a given method is up to you (including what arguments to pass, unless explicitly specified). You can also create more methods to support the required methods.

DOM : Date of Modification

Cwd : Current Working Directory

1. get_info(…): For every file inside folder1 (including files in sub-folders), calculate the extension of the file (.txt, .csv, .pdf etc), the size (in bytes), a hash value based on a checksum hash technique (MD5,SHA-256, your choice), the filename, the filepath and the date the file was modified on. Store this data in a file "file_info", the type of which is your choice [json,csv,txt etc].
2. __init__(…): The constructor should take the path where folder1 is located. Then, if "file_info" file is not present in cwd or if "file_info"'s DOM is older than 2 days, run get_info(…) and get the most updated version of "file_info".
3. generate_report(…) : When this is run, it generates a report on the screen with an option to save it to file (txt or csv). It would contain the following information:
    a. different types of files present in the folders (including all sub-folders)
    b. mean and median size of the files (in general and by type)
    c. the folder name (not including subfolders) with the largest amount of files by (i) number and (ii) size

    All the data required for this method will come from the "file_info" file. When this functionality is executed, check if the "file_info"'s date last modified is greater than 2 days, Call get_info() and overwrite the old "file_info" (and then generate the report).