# Računalna forenzika

## Lab 2

import os

import pandas as pd

import hashlib

import magic

import mimetypes

dir\_path = '.'

file\_names = []

extensions = []

md5s = []

sha1s = []

sha256s = []

magic\_numbers = []

mag\_obj = magic.Magic(mime=True)

extension\_matches = []

for file in os.listdir(dir\_path):

# check if the file is a regular file (i.e., not a directory)

if os.path.isfile(os.path.join(dir\_path, file)):

# if so, add the file name to the list

(file\_name, extension) = os.path.splitext(file)

file\_names.append(file\_name)

extensions.append(extension)

with open(file, 'rb') as f:

data = f.read()

magic\_number = mag\_obj.from\_file(os.path.join(dir\_path, file))

magic\_numbers.append(magic\_number)

md5hash = hashlib.md5(data).hexdigest()

md5s.append(md5hash)

sha1hash = hashlib.sha1(data).hexdigest()

sha1s.append(sha1hash)

sha256hash = hashlib.sha256(data).hexdigest()

sha256s.append(sha256hash)

if extension.lower() == '':

extension\_matches.append(False)

elif mimetypes.guess\_type('test'+extension.lower())[0] in magic\_number.lower():

extension\_matches.append(True)

else:

extension\_matches.append(False)

df = pd.DataFrame({'file\_name': file\_names, 'extension': extensions,

'md5': md5s, 'sha1': sha1s, 'sha256': sha256s, 'magic\_numbers': magic\_numbers, 'extension\_matches': extension\_matches})

# print the dataframe

print(df)

print(df['sha1'])