

```

# compress_mp4.py

import inspect
import os.path
import glob
import ffmpeg_utils
import stat

# This variable stores a unique sequence of file names. For example,
# { 'wavey_particles_1.mp4' : 'wavey_particles_1.mp4',
#   'wavey_particles_2.MP4' : 'wavey_particles_2.MP4',
# }
namesDB = {}

def get_parent_dir():
    scriptpath = inspect.getframeinfo(inspect.currentframe()).filename
    parent_dir = os.path.dirname(os.path.abspath(scriptpath))
    return parent_dir

#-----
def get_fullfile_names(parent_dir):
    all_images = []
    glob_pattern = parent_dir + "/*.mp4"
    mp4s = glob.glob(glob_pattern)
    if len(mp4s) > 0:
        all_images.extend(mp4s)
    glob_pattern = parent_dir + "/*.MP4"
    MP4s = glob.glob(glob_pattern)
    if len(MP4s) > 0:
        all_images.extend(MP4s)
    return all_images

#-----
def add_name_to_DB(name):
    if name not in namesDB:
        namesDB[name] = name

#-----
# The main part of the script begins here...
#-----

parent_dir = get_parent_dir()
full_paths = get_fullfile_names(parent_dir)

for full_path in full_paths:
    filename = os.path.basename(full_path)
    add_name_to_DB(filename)
    #print(filename)

if len(namesDB) == 0:
    print("Sorry cannot continue - could not find any MP4 files.")
else:
    if os.name == 'posix':
        out_path = parent_dir + '/compressMP4'
    elif os.name == 'nt':
        out_path = parent_dir + '/compressMP4.bat'

```

```
file_out = open(out_path, 'w')
keys = namesDB.keys()
for key in keys:
    if os.name == 'nt':
        command = ffmpeg_utils.WIN_compress_mp4(parent_dir, key)
    else:
        command = ffmpeg_utils.POSIX_compress_mp4(parent_dir, key)
    file_out.write(command)
file_out.close()
if os.name == 'posix':
    os.chmod(out_path, stat.S_IRWXU)
```