# Processing photos to add date to filename and add date to photo

**A) To rename the files with the date first, use the DatetoName.py program**

For Small groups of photos, they can be copied to

C:Python>DatetoName>in\_name

Then run the **DatetoName**.py program with

path = 'in\_name'

A1) For larger groups of photos on another drive, on another drive. If you have a backup of the named photos, you can rename then in on the other drive. For E:\Temp\Photos and run DatetoNamePy with

path = 'E:\\Temp\\Photos'

**B) To put the date from the filename into image and resave the .jpg file, use NametoJPG.py**

Before running ‘**NametoJPG**.py’ check that all photos have been renamed and manually rename any that haven’t been renamed or have been given the wrong date (usually only if the jpg date is missing).

The default is to use ‘in\_name’ is the input directory and ‘out’ as the output directory.

**C) Notes**

Note 1: Both programs should handle all subdirectories. So you can start with directories that contain multiple levels of subdirectories in ‘in\_name’ and the directory structure should be preserved in ‘out’

Note 2: DatetoName.py skips filenames that start with ‘19’ or ‘20’ as it assumes they have already been renamed. This means that running DatetoName.py a second time on the processed files should not change them.

Note 3: ‘bSkipShorten = 0’ in **DatetoName.py**. Initially the files are renamed with ‘yyyy-mm-dd hh-mm-ss’ added in front of the old name. If bSkipShorten == 0, a second pass shartens the –mm-ss part to ‘ ‘, ‘a’-‘z’ for the first 26 and then ‘aa’, ‘ab’ … ‘zz’ for the next 26\*25 photos taken during the same hour. This keeps the names shorter. There is also code that can change the ‘hour’ on photos named by my old Panasonic camera that used ‘gps’ to get the time, but often didn’t change the time for new time zones until it was on long enough to get a good gps signal. You can ignore this.

Note 4: Be sure to check that they are right side up after photo transfer (photo info is lost with write). If not, just rotate them in the out directory. (I think python may miss some of the encoded rotations when it rewrites the image). This applies to photos that have been rotated by changing jpg encoding.

Note 5: DatetoName.py also includes the hour from the 24 hour

**D) Test case**

For a test case, copy the files and directory from ‘ShortTest’ into ‘in\_name’ and then run **DatetoName.py** and then run **NametoJPG.py**. The output directory ‘out’ should contain jpg files that have their names changed (date added to the filename) and have the date visible in the new .jpg image.

Also, the directory structure in ‘in\_name’ should be preserved in ‘out’