nützliche Python Skripte

System

Bilddateien sortieren

Alle in einem Verzeichnis liegenden Bilddateien (*.jpg) mit dem gleichen Erstellungsdatum in jeweils einen Unterordner pro Tag verschieben:

sort_pix.py

```
#!/usr/bin/python
import sys, os, getopt, time, shutil

input_folder = ""
output_folder = ""
file_count = 0
folder_count = 0
verbose = False
dry_run = False
short_year = False

def print_usage():
    print("""
usage: <Scriptname>.py <options>

Options:
 [-i|--ifile] <input_folder>  Folder with files to process
 [-o|--ofile] <output_folder> Output folder, when no output folder is
specified the input folder will be used.
 [-d|--dry]                   Dry run - no changes
 [-v|--verbose]               Verbose output
 [-y|--shortyear]             use only two digits for year like 18 for
2018
 [-h|--help]                  Show this message
 """)
    sys.exit()

def checkargs(argv):
   global input_folder
   input_folder = ''
   global output_folder
   output_folder = ''
   global verbose
   global dry_run
   global short_year
   try:
```

```
      opts, args = getopt.gnu_getopt(argv, "i:o:hdvy", ["ifile=",
"ofile=", "help", "dry", "verbose", "shortyear"])
   except getopt.GetoptError as err:
       print("ERROR: Unsupported option!")
       print_usage()

   if len(sys.argv) == 1 or sys.argv[1] == "-" or sys.argv[1] == "--":
       print_usage()

   for opt, arg in opts:
      if opt in ("-h", "--help"):
         print_usage()
         sys.exit()
      elif opt in ("-d", "--dry"):
         #print("Option -d, --dry")
         dry_run = True
      elif opt in ("-v", "--verbose"):
         #print("Option -v, --verbose")
         verbose = True
      elif opt in ("-y", "--shortyear"):
         # print("Option -v, --verbose")
         short_year = True
      elif opt in ("-i", "--ifile"):
         input_folder = arg
      elif opt in ("-o", "--ofile"):
         output_folder = arg

def sortfiles():
    global file_count
    file_count = 0
    global folder_count
    folder_count = 0
    new_folder = ""
    last_folder = ""

    files = os.listdir('.')
    if dry_run:
        print("Performing dry run - no changes will be made!")
    for file in files:
        if file.lower().endswith(('.jpg', '.jpeg')) and not
os.path.isdir(file):
            if short_year == True:
                mod_year = time.strftime('%y',
time.gmtime(os.path.getmtime(file)))
            else:
                mod_year = time.strftime('%Y',
time.gmtime(os.path.getmtime(file)))
            mod_month = time.strftime('%m',
time.gmtime(os.path.getmtime(file)))
            mod_day = time.strftime('%d',
time.gmtime(os.path.getmtime(file)))
```

```
            new_folder = mod_year + mod_month + mod_day
            if new_folder != last_folder:
                last_folder = new_folder
                folder_count = folder_count + 1

            newpath = output_folder + "\\" + new_folder
            if not os.path.exists(newpath):
                if verbose:
                    print("New output folder is: " + newpath)
                if not dry_run:
                    os.makedirs(newpath)
            if verbose:
                print("       move " + file + " to folder " + newpath)
            if not dry_run:
                shutil.move(file, newpath)
                file_count = file_count + 1
            else:
                file_count = file_count + 1
    if verbose:
        print("sortfiles done!")

if __name__ == "__main__":
    if len(sys.argv) > 1:
        checkargs(sys.argv[1:])

    if input_folder == "":
        input_folder = os.getcwd()
        if verbose:
            print("Input folder is: " + input_folder)

    if output_folder == "":
        output_folder = input_folder
        if verbose:
            print("Output folder is not set -> using input folder: " +
output_folder)

    if not os.path.isdir(output_folder):
        print("Output folder does not exsist -> exit now.")
        sys.exit()

    if os.path.isdir(input_folder):
        if verbose:
            print("Input folder \"" + input_folder + "\" exists!")
        os.chdir(input_folder)
        sortfiles()
    else:
        print("ERROR: Input folder \"" + input_folder + "\" does not
exist!")
        sys.exit()

    print("Job done!")
```

```
    print("created folders: " + str(folder_count))
    print("moved files: " + str(file_count))
```

From:
https://www.von-thuelen.de/ - **Christophs DokuWiki**

Permanent link:
https://www.von-thuelen.de/doku.php/wiki/linux/python_commands

Last update: **2018/08/15 11:33**