

ORGANIZING FILES

Jorian Saenen & Dries Peeters

VOORWOORD

- Het kopiëren en verplaatsen van files
- In grote hoeveelheden files verwerken
- Manipuleren van ZIP files

IMPORT SHUTIL , OS

- Om deze functies te kunnen gebruiken moet shutil en os geïmporteerd worden
- `import shutil, os`

KOPIËREN VAN FILES

- `shutil.copy(source, destination)`

```
>>> import shutil, os
>>> os.chdir('C:\\')
>>> shutil.copy('C:\\spam.txt', 'C:\\delicious')
'C:\\delicious\\spam.txt'
>>> shutil.copy('eggs.txt', 'C:\\delicious\\eggs2.txt')
'C:\\delicious\\eggs2.txt'
```

- File → folder destination
- File → filename destination

KOPIËREN VAN FOLDERS

```
>>> import shutil, os  
>>> os.chdir('C:\\\\')  
>>> shutil.copytree('C:\\\\bacon', 'C:\\\\bacon_backup')  
'C:\\\\bacon_backup'
```

- Folder → folder destination
- De destinatie folder wordt aangemaakt

VERPLAATSEN FILES EN FOLDERS

```
>>> import shutil  
>>> shutil.move('C:\\bacon.txt', 'C:\\eggs')  
'C:\\eggs\\bacon.txt'
```

- File → folder
- Als eggs\\bacon.txt al bestaat wordt deze file overschreven
- source bestand wordt verwijderd

VERPLAATSEN FILES EN FOLDERS

```
>>> shutil.move('C:\\bacon.txt', 'C:\\eggs\\new_bacon.txt')  
'C:\\eggs\\new_bacon.txt'
```

- File → filename
- De file wordt hernoemd

```
>>> shutil.move('C:\\bacon.txt', 'C:\\eggs')  
'C:\\eggs'
```

- Als de eindfolder niet bestaat wordt het bestand genoemd naar de naam van die folder (file zonder extensie)

VERPLAATSEN FILES EN FOLDERS

```
>>> shutil.move('spam.txt', 'c:\\does_not_exist\\eggs\\ham')
```

- Als er een folder (buiten de eindfolder) niet bestaat wordt er een `FileNotFoundError` exception gegeven

PERMANENT VERWIJDEREN VAN FILES EN FOLDERS

- `os.unlink(path)` : file verwijderen
- `os.rmdir(path)` : lege folder verwijderen
- `shutil.rmtree(path)` : folder verwijderen

VOORKOMEN VAN ONGELUKKEN BIJ MANIPULATIE VAN FILES EN FOLDERS

```
import os
for filename in os.listdir():
    if filename.endswith('.rxt'):
        #os.unlink(filename)
        print(filename)
```

- Voor ongelukken te voorkomen gebruik je print om te verifiëren welke file je manipuleert

VEILIG VERWIJDEREN MET SEND2TRASH MODULE

- Om files en folders naar de vuilbak te sturen i.p.v. permanent te verwijderen
- Instaleren via pip (pip install send2trash)
- Import send2trash
- `send2trash.send2trash(path)`

DOORLOPEN VAN FOLDERS

- for folderName, subfolders, filenames in `os.walk(path)`:
- folderName : string van het path folder
- Subfolders : lijst van alle sub folders
- Filenames : lijst van alle files

DOOROPEN VAN FOLDERS

```
import os

for folderName, subfolders, filenames in os.walk('C:\\\\delicious'):
    print('The current folder is ' + folderName)

    for subfolder in subfolders:
        print('SUBFOLDER OF ' + folderName + ': ' + subfolder)

    for filename in filenames:
        print('FILE INSIDE ' + folderName + ': ' + filename)

print('')
```

COMPRESSING FILES MET ZIPFILE MODULE

- Om deze functies te kunnen gebruiken moet zipfile en os geïmporteerd worden
- Import zipfile, os

LEZEN VAN ZIP FILES

- `Object = zipfile.ZipFile(filename)` : Objecten aanmaken
- `Object.namelist()` : lijst van alles in de zip file
- `Object.getinfo(file in namelist)` : maakt object met info over file aan
- `.file_size`, `.compress_size`, ...
- Op einde `Object.close()`

LEZEN VAN ZIP FILES

```
>>> import zipfile, os
>>> os.chdir('C:\\')      # move to the folder with example.zip
>>> exampleZip = zipfile.ZipFile('example.zip')
>>> exampleZip.namelist()
['spam.txt', 'cats/', 'cats/catnames.txt', 'cats/zophie.jpg']
>>> spamInfo = exampleZip.getinfo('spam.txt')
>>> spamInfo.file_size
13908
>>> spamInfo.compress_size
3828
>>> 'Compressed file is %sx smaller!' % (round(spamInfo.file_size / spamInfo
.compress_size, 2))
'Compressed file is 3.63x smaller!'
>>> exampleZip.close()
```


EXTRACTING VAN ZIP FILES

- `Object.extractall(optioneel:destination)` : de inhoud van de zip file in de huidige directory zetten of een directory meegeven
- `Object.extract(file, optioneel:destination)` : gespecificeerde file in de huidige directory zetten of een directory meegeven

MAKEN VAN / TOEVOEGEN AAN ZIP FILES

```
>>> import zipfile  
>>> newZip = zipfile.ZipFile('new.zip', 'w')  
>>> newZip.write('spam.txt', compress_type=zipfile.ZIP_DEFLATED)  
>>> newZip.close()
```

- maken van een nieuwe ZIP file
- Naar deze ZIP file schrijven
- ‘w’ → overschrijft de huidige contents in de ZIP file
- ‘a’ → voegt de nieuwe contents toe

QUIZ (1/2) DIRECTORY

- Wat doet : `shutil.copy(path1, path2)`?
- Wat is de filename als het gekopieerd is : `shutil.copy('dog.txt', 'C:\\animals\\cat')`
- Wat als de cat folder niet bestaat?
- Wat gebeurt er met de originele file bij `shutil.move()`?
- Wat is het verschil tussen `os.rmdir()` en `shutil.rmtree()`?
- Met welke 2 manieren kan je ongelukken voorkomen?
- Hoe doorloopt `os.walk()` een directory?

QUIZ (2/2) ZIP

- Wat voor value krijg je als je `.namelist()` gebruikt?
- Hoe krijg je de info van een ZIP file?
- Wat is het verschil als je een ZIP file met 'w' of 'a' maakt?
- Wat mag je niet vergeten na dat je met een ZIP file hebt gewerkt?