

# Workshop ark

Denne tekst er et overblik over hvad der gennemgås i workshoppen "Humanistens Digitale Værktøjskasse". Dette dokument forsøger, at give et overblik over hvordan de forskellige elementer af workshoppen kan bruges også efter workshoppen er slut.

## Hvad skal man bruge til workshoppen?

### Materialer/programmer

 Navn	 Forklaring	 Link
<u>Miniconda</u>	Miniconda er det program, som vi skal bruge til at køre vores kode i. Det består af følgende tre ting: kodesproget python, en pakke-manager og en miljø-manager	<a href="https://docs.conda.io/en/latest/miniconda.html">https://docs.conda.io/en/latest/miniconda.html</a>
<u>Visual Studio Code</u>	Visual Studio Code er det program, som vi bruger til at ændre i vores kode. Det skaber et mere overskueligt overblik, så vi kan følge med i hvad der sker	<a href="https://code.visualstudio.com/download">https://code.visualstudio.com/download</a>

## Hvordan åbner jeg conda?

Conda åbnes forskelligt alt efter om du bruger en windows eller mac.

### Windows

På Windows skal man første gang man skal bruge conda have det sat rigtigt op, hvilket vi gør i løbet af workshoppen, dette gøres ved at åbne programmet "Windows powershell" som ADMINISTRATOR. Her skriver man: `Set-ExecutionPolicy Unrestricted`, trykker ENTER. skriver a og trykker enter igen. Herefter skal man lukke powershell helt.

Derefter skal vi åbne programmet Anaconda powershell prompt (miniconda), også som ADMINISTRATOR. Her skal man skrive: `conda init powershell` . Dette gør, at conda bliver aktivt i powershell.

Nu går man tilbage og åbner powershell som administrator og er klar til at arbejde.

## Mac

På Mac, skal man "bare" åbne terminal efter installationen af miniconda. Dette kan gøres ved at bruge genvejen cmd + mellemrum og så søge "terminal". Det vindue man åbner, ser sådan her ud:

# Opsætning af conda

Første gang man bruger conda, skal man lave det miljø, som man vil arbejde i. Det er her man sikre sig, at forskellige python projekter ikke "ødelægger" hinanden.

For at lave et nyt miljø i conda skriver man følgende command;

```
conda create --name "toolbox" python=3.8
```

Når dette miljø er lavet, skal man gå ind i det. Det gør man ved at aktivere det:

```
conda activate "toolbox"
```

Dette er det grundlæggende conda, som vi bruger i løbet af workshoppen. Hvis man efter at have lavet miljøet "Toolbox" lukker vinduet ned, skal man blot aktivere sit miljø igen.

## Forskellige conda commands


 Command:	 Forklaring
<u><code>conda create - -name "name"</code></u> <u><code>python=3.8</code></u>	Laver et nyt miljø i conda, som kører python 3.8
<u><code>conda activate "name"</code></u>	Aktiverer conda miljøet toolbox
<u><code>conda deactivate</code></u>	deaktiverer det aktive conda miljø
<u><code>conda install</code></u>	installerer pakker til conda

 Command:	 Forklaring
<u>conda uninstall</u>	afinstallerer pakker til conda
<u>conda update</u>	Opdaterer conda
<u>conda update "package"</u>	Opdaterer pakken "package"

## Sådan navigerer man i Shell og Terminal

Når man bruger programmerne Shell (Windows) & Terminal (Mac) interagerer man med computeren på en helt anden måde. For det første, kan man tænke på sig selv som en sej hacker, eller som en computernørd fra start 80'erne, før computere fik det udseende, som vi interagerer med den dag idag. Grunden til at vi skal denne vej ind, er fordi vi denne vej kan få lov til mange andre ting med computeren. Men for at det kan lade sig gøre, skal vi vide hvordan vi bevæger os rundt i systemet, når vi ikke har en mus til at klikke på mapperne.

### Commands til Shell og Terminal

 Command	 Forklaring
<u>cd</u>	Command der skifter mappe, den skal efterfølges af hvor man gerne vil hen
<u>cd ./mappe</u>	Skifter til mappen "mappe" i den mappe, som vi befinder os i
<u>cd ..</u>	Skifter til mappen ovenover den vi er i nu
<u>cd ~/</u>	skifter til computerens hjemmeplacering
<u>cd</u> <u>c:/users/name/billeder</u>	Skifter til mappen billeder under name, under users på c-drevet
<u>dir</u>	viser os indholdet af den nuværende mappe
<u>ls</u>	viser os indholdet af den nuværende mappe

## Det første script i workshopen - rename.py

Dette script kan bruges til at omdøbe en masse filer, der har den samme endelse, på en gang. I workshopen bruger vi scriptet på billeder fra Vikingemuseet i Århus, det kunne også bruges på fx scannede dokumenter, kilder eller noget helt tredje.

I dette script er der to ting, som man skal ændre på, for at få det til at virke. Først åbner man filen rename.py i Visual Studio Code. De to steder, som man skal ændre er i linje 8 og linje 11.

I linje 4, skal man specificere hvor filerne er placeret. Det kunne fx være, at jeg havde mine billeder til at ligge i mappen 1Billeder, i mappen "Humanistens\_digitale\_toolbox" på mit skrivebord. Så ville linje 4 se sådan her ud:

```
path="C:/Users/user/Desktop/Humanistens_digitale_toolbox/1Billeder/"
```

**OBS:** Når man skriver denne path, er det vigtigt at huske at slutte den af med en "/" og for windows-brugere, skal man vende alle "\", så de bliver "/".

I linje 8, skal man skrive det navn, som filerne skal have + den endelse, som alle filerne ALLEREDE har. Navnet skrives istedet for name\_of\_files og filendelsen istedet for .endning. Under workshoppene ville vi gerne have vores .jpg-billeder til at hedde: vikingemuseet:

```
os.rename(os.path.join(path, file), os.path.join(path, 'vikingemuseet' + file + '.jpg'))
```

For at køre scriptet, skal man i powershell (windows) / Terminal (mac) først finde hen til det. Dette gøres med commanden cd efterfulgt af pathen til filen. Det kan fx se sådan her ud:

```
cd C:\Users\user\Desktop\Humanistens_digitale_toolbox\Scripts
```

**OBS:** På windows bruger man \. På Mac skal man bruge /.

Når man tror man har fundet den rigtige placering, kan man tjekke ved at bruge enten commanden "dir" eller "ls". Hvis scriptets navn kommer frem på skærmen er man klar til at køre filen, dette gøres ved at skrive:

```
python rename.py
```

Hvis man ingen fejlkode får, er filerne omdøbt. Hvis man modtager en fejlkode, så er det højst sandsynligt fordi man enten har glemt et " eller lavet et mellemrum et sted, så kig koden igennem igen i Visual Studio Code.

# Det andet script i workshoppen - Selectivecopy.py

Dette script kan kopiere mange filer på en gang. I workshoppen bruges det til at kopiere forskellige rapporter fra flere mapper til en samlet mappe. Scriptet virker på alle filer der har samme endelse. Jeg bruger det fx mod slutningen af et semester, til at samle alle tekster fra et kursus til en samlet mappe.

I dette script skal man lave ændringer i linje 9,10, 24 & 25. I linje 9 skal man indsætte pathen til der hvor filerne befinder sig og i linje 10 skal man indsætte pathen til der filerne skal kopieres til.

Linje 9 kommer til at se nogenlunde sådan her ud:

```
source = "C:/Users/user/Desktop/Humanistens_digitale_toolbox/2Rapporter/"
```

**OBS:** Husk / tilsidst og for windowsbrugere at vende \ til /.

Linje 10 kommer til at se sådan her ud:

```
source = "C:/Users/user/Desktop/Humanistens_digitale_toolbox/2Rapporter/tekster/"
```

Hvis den nye placering er en ny mappe, på samme placering som filerne befinder sig, er det vigtigt at ændre linjerne 24 og 25. Her skal man ændre navnet på mappen til det navn, som man kalder den nye mappe

```
if 'tekster' in subfolders:  
    subfolders.remove('tekster')
```

Dette gøres for at scriptet ikke skal kopiere alle filerne to gange.

## OCRmyPDF - tredje element i workshoppen

Den næste del af workshoppen bruges lidt anderledes. Her skal vi have installeret et par pakker i conda, så vi kan lære vores computer at OCR scanne filer. Det vil sige, at vi gør PDF-filer søgbare og man kan markere i dem.

## Windows opsætning

Dette script kræver lidt forskellige installationer, der skal løbes igennem første gang, før vi kan begynde at bruge programmet.

Det første vi skal gøre er at installere Chocolatey, som vi skal bruge til at installere vores ocr-program:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object  
System.Net.WebClient).DownloadString(' https://chocolatey.org/install.ps1 '))
```

Kopier ovenstående tekst ind i powershell, tryk enter, bekræft ved at skrive a og trykke enter.

Når denne installation er kørt, skal følgende køres på samme måde ved at copy-paste, trykke enter, bekræfte med a og enter.

```
choco install python3
```

```
choco install --pre tesseract
```

```
choco install ghostscript
```

```
choco install pngquant
```

```
pip install ocrmypdf
```

Disse linjer skal køres hver for sig og skal alle bekræftes med "a" og enter.

Vi skal nu lære vores computer at læse dansk. Det gør man ved at tage filerne fra workshoppens mappe "tesseract\_languagedata" og lægge dem i den mappe hvor tesseract ligger på ens computer. Denne mappe ligger typisk på denne placering: C:\Program Files\Tesseract-OCR\tessdata\

## Mac opsætning

Dette script kræver tre installationer. Først skal vi installere pakken "ghostscript" i vores conda-miljø, dette gøres ved at skrive:

```
conda install -c conda-forge ghostscript
```

Her er det okay at føle sig som en hacker, da der sker en del på skærmen. På et tidspunkt skal man bekærfte installationen og det gøres ved at skrive "y" og trykke enter.

Den næste pakke der skal installeres er tesseract. Her er fremgangsmåden den samme. For at installere tesseract, skriver man følgende:

```
conda install -c conda-forge tesseract
```

Når tesseract er installeret mangler vi kun at installere selve OCRmyPDF. Dette gøres med denne command i conda:

```
pip install ocrmypdf
```

## Brug af OCRmyPDF

Når man vil køre OCRmyPDF på sine filer skal man først navigere hen til filerne i powershell (windows) / terminal (mac). Dette gøres med cd commanden:

```
cd C:\Users\user\Desktop\Humanistens_digitale_toolbox\3Scan
```

Herfra skriver man blot ocrmypdf efterfulgt af filens navn og så det navn, som den nye ocr scannede fil skal have:

```
ocrmypdf straffeloven1930.pdf straffeloven1930ocr.pdf
```

Når man kører ovenstående command begynder computeren at arbejde på højtryk. Når man begynder at føle sig mere tryk med dette script, kan man eventuelt give output-filen samme navn som input-filen og på den måde lægge ocr laget ned over den eksisterende fil. Hvis man vil udforske scriptet mere, kan man med fordel åbne help menuen ved at køre koden:

```
ocrmypdf -h
```

