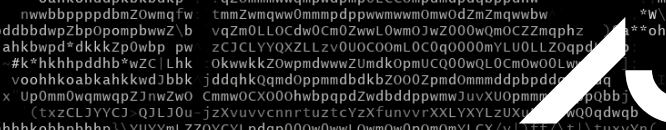


Tældin tekst

Introduktion til digital tekstanalyse > Session 2

- > Ulf Dalvad Berthelsen
- > udb@cc.au.dk
- > <https://github.com/ulfberthelsen>



Computational thinking

‘Computational tænkning’ dækker kort sagt over tænkemåder, der gør os i stand til at handle i verden gennem computere med henblik på at gøre ting vi ellers ikke ville have været i stand til at gøre.

<https://videnomlaesning.dk/viden-og-vaerktoejer/forskerklummen/2019/hvilket-sprog-taler-din-computer/>

Hvad bruges computere til?

- **Løse praktiske opgaver** (streaming, tekstbehandling, GPS, netbank, e-boks, SoMe, mail, mm.)
- **Skabe ny viden** løse nye problemer (forskning, datascience, engineering, mm.)
- **Være kreative** (kunst, musik, design, mm.)

Computational thinking

Aspekter ved computational tænkning (nogle blandt mange)

- **Abstraktion** > fx pre-processing af tekst (fx fjernelse af tegn, store bogstaver og bøjnings endelse)
- **Dekomposition** > nedbryde komplekse processer i enkeltdele (fx fra rå tekst til preprocessed ordlister)
- **Mønstergenkendelse** > genkende strukturer i et datasæt (fx syntaktiske strukturer, RegEx, NER)
- Konstruktion af algoritmer > konstruere et script, der giver det ønskede output
- **Debugging** > systematisk fejlfinding (det handler ikke om ikke at lave fejl, men om at vide, hvordan man finder dem)

Hvaderen computer?

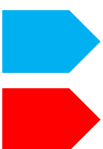
Hvader en computer?

En computer er en mekanisk eller elektronisk maskine, der gennem fysisk manipulation (fx ved håndkraft eller elektriske impulser) kan programmeres til automatisk at gennemføre logiske og aritmetiske operationer.

Hvaderen computer?

Input

Processor



Output

Hukommelse



Hvaderen computer?

Input

Processor

Output

Hukommelse

Hvaderen computer?



Input

Processor

Output

Hukommelse

Hvaderen computer?

Input

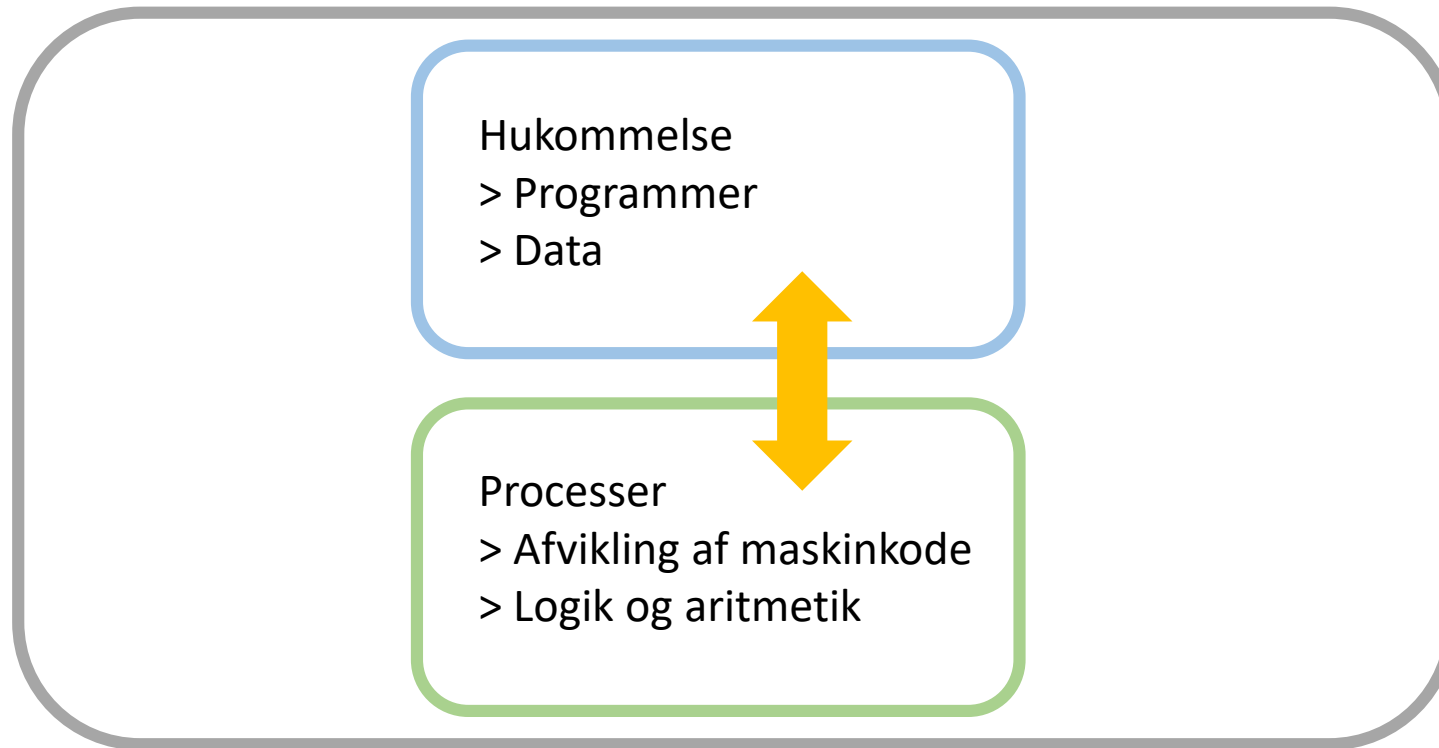
Processor

Output

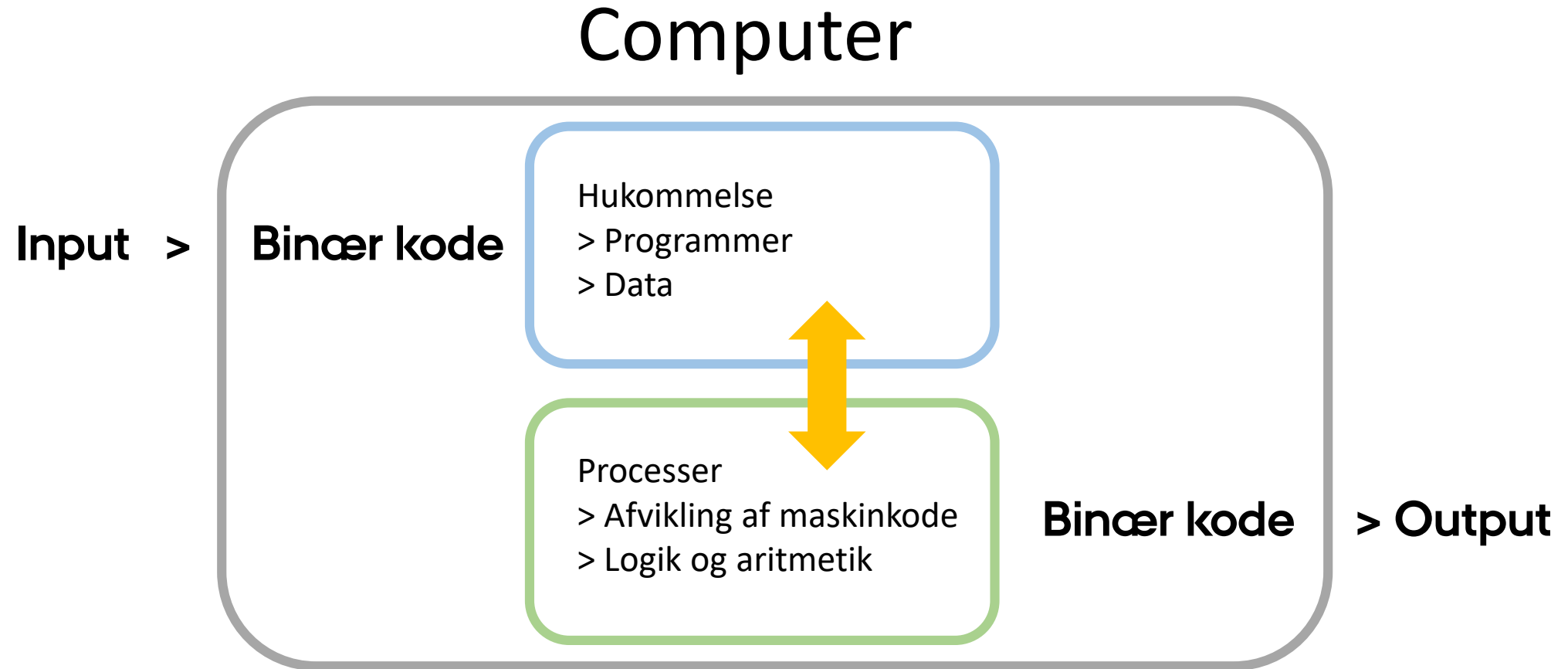
Hukommelse

Hvad er en computer?

Computer



Hvad er en computer?



Hvad er en computer?

Stort set alle fænomener (lyd, billeder, tekst, afstand, tal, mm.) kan konverteres til binær kode og dermed behandles computationelt

Computer

Input

>
Lyd
Billede
Tekst
Tal
etc.

Binær kode

Hukommelse
> Programmer
> Data

Processer
> Afvikling af maskinkode
> Logik og aritmetik

Binær kode

> Output

Lyd
Billede
Tekst
Tal
etc.

Hvad er et programmeringssprog?

High level



- > GUI Grafisk brugergrænseflade, fx Windows eller FB-app'en
- > **Python**, Ruby, Processing (intuitivt, let at læse og skrive)
- >
- > C++, Java og mange flere ...
- >
- > Assembly languages (maskinspecifikke)
- > Maskinkode, executable (0 og 1, direkte instruktion til maskinen)

Low level

Hvad er et programmeringssprog?

High level



- > GUI Grafisk brugergrænseflade, i
- > **Python**, Ruby, Processing (intu
- >
- > C++, Java og mange flere ...
- >
- > Assembly languages (maskinspec
- > Maskinkode, executable (0 og 1,

Low level

Implementering af programmet i maskinen:

Compiler vs interpreter

- En compiler oversætter hele programmet til maskinkode, hvorefter det afvikles
- En interpreter oversætter og afvikler løbende
- Python bliver først *compiled* til bytekode og derefter *interpreted* til maskinkode
- Det er ikke noget vi får brug for at bekymre os om

Hvad er programmering?

Programmering/coding/scripting er det at skrive instruktioner i et programmeringssprog med henblik på at få computeren (den fysiske maskine) til at producere et bestemt output > tænk på en bage- eller strikkeopskrift

Programmer kan være meget korte:

Fx programmet `print("Hello world!")` der består af en linje

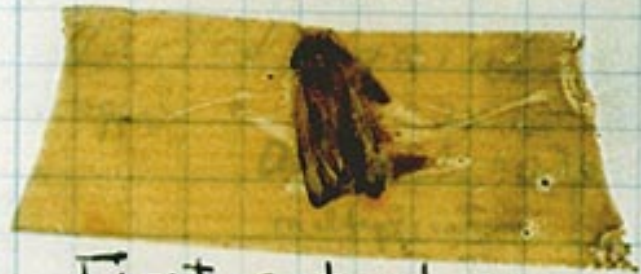
Eller meget lange:

Fx styresystemet Microsoft Windows, der består af millioner af kodelinjer

Hvad er programmering?

- **Programmering vs computer science**
 - Programmering er det praktiske arbejde med at give computeren instruktioner
 - Computer science/datalogi er det videnskabelige og teoretiske studie af, hvad en computer er og hvordan den virker
- **Problemløsning vs creative computing**
 - Svar vs udtryk
- **Det er omstændeligt fordi:**
 - Computeren har ikke intuition og baggrundsviden
 - Kræver entydig syntaks
- **God idé at begynde sit projekt med pseudo-kode**
 - Dvs. lave en skitse på almindeligt dansk + pile, streger, bokse mm.
- **Hvad er bugs?**

9/9

0800 Antan started
1000 " stopped - antan ✓
1300 (032) MP - MC 2.130
(033) PRO 2 2.130
conv 2.130
Relays 6-2 in 033 for
in Relay
Relays changed
1100 Started Cosine Tape (S)
1525 Started Mult + Adder T
1545  Relays (moth)
First actual case of b
1630 Antan started.
1700 closed down.

Hvad er objektorienteret programmering?

- Objektorienteret programmering (OOP) er et programmeringsparadigme, der opstår i løbet af 60'erne og 70'erne
- OOP er en tilgang til programmering, hvor man skaber funktionalitet ved at organisere data og funktioner i meningsfulde enheder – tænk på en brugerprofil på de sociale medier
- I Python er kan man selv lave **klasser** (skabeloner) og **objekter**, men der findes også mange prædefinerede objekt-klasser
- En `string` er et eksempel på et objekt.
 - Den kan indeholde data (tegn)
 - Den har egenskaber (fx længde)
 - Der er tilknyttet funktionalitet (vi kan splitte, tælle osv)
- Funktioner der er specifikke for en bestemt klasse af objekter kaldes **methods** og skrives normalt `variabelnavn.funktionsnavn()` - fx `x.split()`
læses 'anvend funktionen `split` på indholdet af variabel `x`'

Gennemgang af øvelserne

- > Start Spyder
- > Hent scripts på https://github.com/ulfberthelsen/TDT_1
- > Download som zip-fil > gem > udpak alle

The screenshot shows the GitHub interface for the repository 'ulfberthelsen / TDT_1'. The top navigation bar includes a search bar, 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository name 'ulfberthelsen / TDT_1' is displayed as 'Public'. Below this, there are buttons for 'Pin', 'Unwatch' (1), 'Fork' (0), and 'Star' (0). A secondary navigation bar contains links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area shows the 'main' branch with 1 branch and 0 tags. It includes a table of files for upload:

File Name	Action	Time
TDT_1.py	Add files via upload	1 minute ago
TDT_2.py	Add files via upload	1 minute ago
TDT_Basic_Python_1.ipynb	Add files via upload	1 minute ago
TDT_Jupyter_markdown.ipynb	Add files via upload	1 minute ago

Below the table is a blue box with the text: 'Help people interested in this repository understand your project by adding a README.' and a green 'Add a README' button. On the right side, the 'About' section states 'No description, website, or topics provided.' and shows 0 stars, 1 watching, and 0 forks. The 'Releases' section states 'No releases published' with a link to 'Create a new release'. The 'Packages' section states 'No packages published' with a link to 'Publish your first package'.

Gennemgang af øvelserne

The screenshot displays the Spyder Python IDE interface. The main editor on the left shows a Python script named `Tæl_Python_1.py` with the following content:

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Fri Feb  5 16:08:37 2021
4
5  @author: UDB
6  """
7
8  """
9  Lang kommentar
10 """
11
12 # Kort kommentar
13
14 """
15 Topics: 1) Expressions 2) Data types 3) Variables
16
17 Brug SHIFT + ENTER for at afvikle scriptet
18
19 Hjælp: https://docs.python.org/3/tutorial/
20
21 """
22 # Expressions - evalueres til en enkelt værdi
23 ## Simpleste instruktionstype
24 ## Værdier + operatorer
25 ## Resultatet vises kun i konsollen, hvis man anvender
26
27 # Statements - gør noget, fx print() eller =
28
29 2+2 #Expression, evalueres, men vises ikke
30 print(2+2) # Statement, printer noget, printer i k
31
```

The right-hand pane contains the Variable explorer and the IPython console. The Variable explorer shows the following variables:

Var	Type	Size	Value
a	int	1	1
b	float	1	2.5
c	float	1	3.5
d	str	1	Hello
e	str	1	
f	str	1	world!

The IPython console shows the execution of the script:

```
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]: runcell(0, 'C:/Users/au100440/Desktop/Tæl din tekst/Python_scripts/Tæl_Python_1.py')
4
Hello world!
Helloworld!
[1, 2.5, 3.5, 'Hello', ' ', 'world!', 'Hello world!', 'Helloworld!']

In [2]:
```

The status bar at the bottom indicates the current state: LSP Python: ready, conda: base (Python 3.8.3), Line 10, Col 4, UTF-8, CRLF, RW, Mem 41%.

Demo - Jupyter Notebook

- **Åbn** Jupyter Notebook – vær opmærksom på at den åbner i din browser
- Vælg den rigtige mappe
- Åben filen
- Hvis du **vil lave en ny fil** skal du vælge **New > Python 3** i rullegardinet til højre over tekstboksen
- Husk at tjekke at du er i den rigtige mappe. Jupyter opretter filen i den mappe, du er i.
- Vælg **markdown** i rullegardinet midtfor
- Tryk **SHIFT + ENTER** for at afvikle markdown eller kode

The screenshot shows the Jupyter Notebook web interface. At the top, there's a header with the Jupyter logo and 'Quit' and 'Logout' buttons. Below the header, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser. The breadcrumb path is '/ Desktop / Tæl din tekst / Jupyter_scripts'. There are two files listed: 'Untitled.ipynb' and 'Untitled1.ipynb'. A dropdown menu is open from the 'New' button, showing options for 'Notebook: Python 3' and 'Other: Text File, Folder, Terminal'.

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕ ↻

0 ▾ / Desktop / Tæl din tekst / Jupyter_scripts

Name ▾

..

Untitled.ipynb Running 5 B

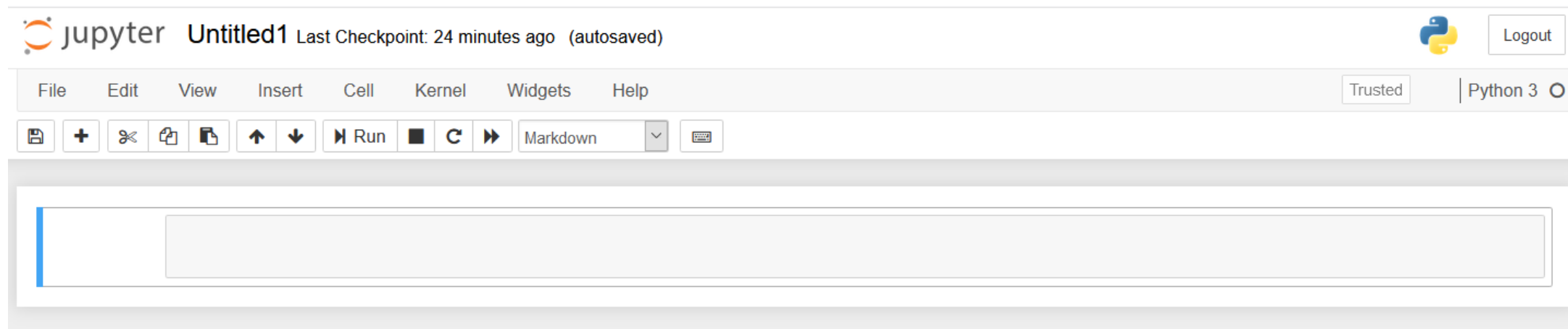
Untitled1.ipynb Running 0 B

Notebook:
Python 3

Other:
Text File
Folder
Terminal

Demo - Jupyter Notebook

- **Åbn** Jupyter Notebook – vær opmærksom på at den åbner i din browser
- Vælg den rigtige mappe
- Åben filen
- Hvis du **vil lave en ny fil** skal du vælge `New > Python 3` i rullegardinet til højre over tekstboksen
- Husk at tjekke at du er i den rigtige mappe. Jupyter opretter filen i den mappe, du er i.
- Vælg `markdown` i rullegardinet midtfor
- Tryk `SHIFT + ENTER` for at afvikle markdown eller kode



Gruppearbejde og pause

Først **15 minutters** pause! Derefter **45 minutters** gruppearbejde i break out groups

- Filerne ligger i den mappe, I downloadede fra github
- Åben filerne i hver sin Jupyter Notebook
 - `TDT_Jupyter_markdown.ipynb` er en kort intro til tekstdelen i Jupyter
 - `TDT_Basic_Python_1.ipynb` opgaven til gruppearbejdet

Jeg cirkulerer rundt i grupperne og forsøger efter bedste evne at svare på spørgsmål

[illegible]

Fælles opsamling

- Strings og string-methods
- Lister
- Vores første forsøg på konkret manipulation af tekst
- Resultatet af at splitte en string er en liste
- Hvad er forskellen på strings og lister?

Forberedelse til session 3

- Sig hej til din arbejdsgruppe

- Gruppe 1: XXX

- Gruppe 2: XXX

- Gruppe 3: XXX

- Genlæs scripts og slides

- Læs teksterne

- Lav øvelserne

Tak for i dag!

Introduktion til digital tekstanalyse > Session 2

- > Ulf Dalvad Berthelsen
- > udb@cc.au.dk
- > <https://github.com/ulfberthelsen>

