



Seminartime uke 5 – Gr. 10

IN1000 – HØST 2021

Hvor langt har vi kommet?

| NB: Endringer kan komme | Legges ut ca. | Frist |
|---------------------------|------------------------|------------------------|
| 1.innlevering (3 poeng) | 25. august kl 13.00 | 31. august kl 23.59 |
| 2. innlevering (5 poeng): | 1. september kl 13.00 | 7. september kl 23.59 |
| 3.innlevering (5 poeng): | 08. september kl 13.00 | 14. september kl 23.59 |
| 4.innlevering (5 poeng): | 15. september kl 13.00 | 21. september kl 23.59 |
| 5.innlevering (5 poeng): | 22. september kl 13.00 | 5. oktober kl 23.59 |
| 6. innlevering (6 poeng): | 6. oktober kl 13.00 | 12. oktober kl 23.59 |

| Uke | Tema |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | Løkker <ul style="list-style-type: none">- For-løkker (for loops)- While-løkker (while loops)- Kombinere løkker med samlinger- Funksjoner & prosedyrer m. parametre+returverdi |
| 5 | Filer <ul style="list-style-type: none">- Lese fra fil- Skrive til fil- Parameter-overføring og variabel-skop- Kombinere data for å løse problemer |
| 6 | <ul style="list-style-type: none">- Repetisjon- Repetisjon- Repetisjon! |

Plan for i dag

- Repetisjon:
 - Løkker
 - Parametre og returverdier
 - Skop
- Nytt pensum
 - Filinnlesing
 - Problemløsning med filer, lister, samlinger og funksjoner (oppgaver)

Repetition is the key to getting your message across
Repetition is the key to getting your message across
Repetition is the key to getting your message across
Repetition is the key to getting your message across
Repetition is the key to getting your message across
Repetition is the key to getting your message across
Repetition is the key to getting your message across
Repetition is the key to getting your message across
Repetition is the key to getting your message across
Repetition is the key to getting your message across
Repetition is the key to getting your message across



Repetisjon

Rep: løkker

| While-løkker | For-in-range-løkker | For-each-løkker |
|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <code>while <uttrykk>: <kode som kjøres så lenge uttrykk er true></code> | <code>for <variabel> in <range>: <kode som kjøres> <variabel er en teller></code> | <code>for <element> in <samling>: <kodesnutt> <element kan brukes></code> |

Når vi ikke vet hvor mange ganger den skal kjøre.

Ønsker at while-condition er et uttrykk som evalueres, ikke bare True/False (eller bare en variabel).

Vil at den skal kjøre x antall ganger

Antall kan f.eks være lengde av noe, eller range(x).

Å gå gjennom samlinger

Mister teller-egenskapen, men får direkte tak i elementene.

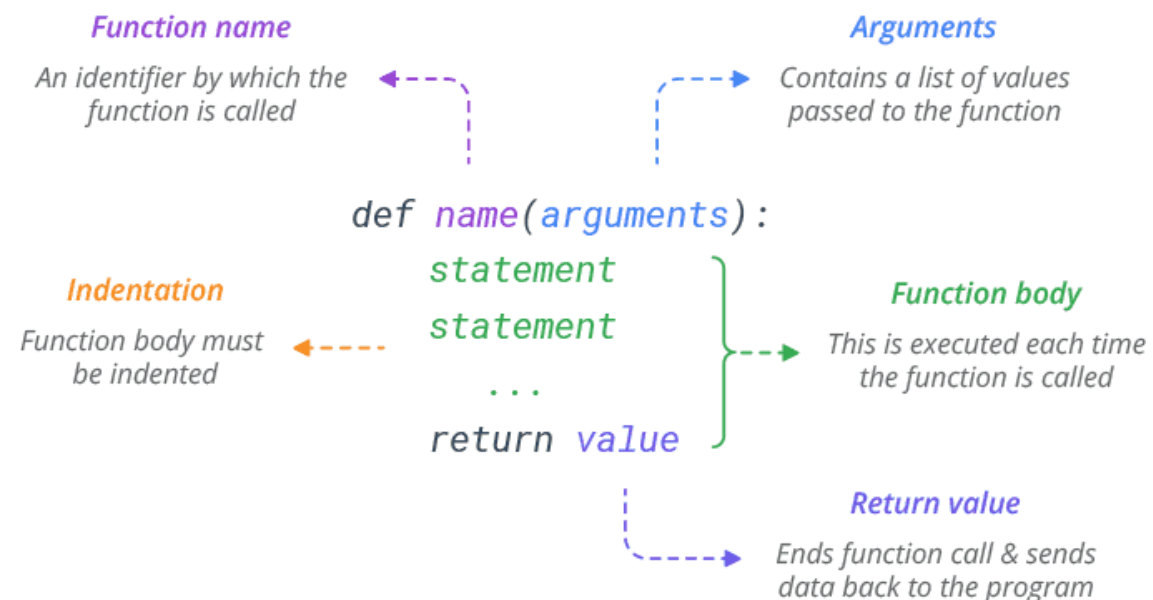
Rep: parametre og returverdier

Parameter/argumenter

- Parametre/argumenter brukes for å sende inn data som skal brukes i funksjonen
- Gjør det mulig å sende LOKALE variabler (husk! Vi bør unngå globale variabler)
- Etterspør man en parameter som ikke brukes i koden, bør den fjernes

Returverdier

- Returverdi i en funksjon er det man får ut ved å kjøre funksjonen
- Retur \neq print. Du får ikke ut noe, men må aktivt printe resultatet
- Retur *avslutter* den kjørende koden, tilsvarende som «break»



Oppgave (3min)

- Lag en funksjon «power» som tar inn en liste (består av ints) og et tall.
- Funksjonen skal opphøye alle elementer i listen med det innsendte tallet og legge disse til i en ny liste (bruk operatoren **).
- Når alle tallene er ferdig, returneres den nye listen.

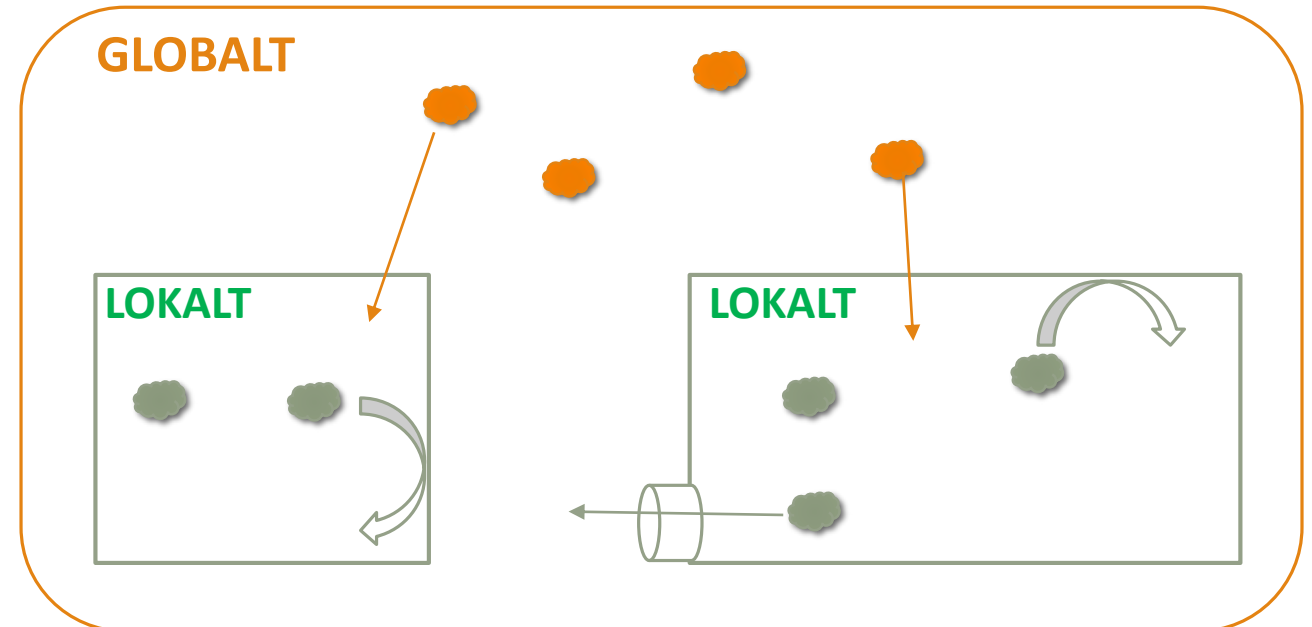
Oversatt:

- Antall parametre: 2
- Innsendte parametre: liste, int
- Returverdi: liste
- Innhold: for-each-løkke, ny liste



Skop

- Globale skop: alt i det globale skopet er tilgjengelig for alle
- Lokale skop: alt i det lokale skoper er *kun* tilgjengelig inne i den aktuelle kodeblokken
- Bruke lokale variabler ved å sende som parametre, og bruke returvedier
- Utenfra (globalt) og inn (lokalt) er OK!
- For å komme ut må man ha en utgang (return)





Filinnlesing

Filinnlesing

- Å lese fil betyr å lese hva som står i fila
- Må huske å åpne og lukke fila (som med bøker)
- Hvilken filer kan man lese? → ALT! Pdf, excel, textfil, bildefil, osv...
- Hva kan man gjøre med filer?
 - Lese fra fil: `read``r```
 - Skrive til fil: `write``w```
 - Skrive til fil slutten av fila: `append``a```

Syntaks: filinnlesing (åpne/lukke fila)

ÅPNE FIL:

Lese en fil:

```
innfil = open("filnavn.filtype", "r")
```

"r" står for "read"

Eks:

Gitt at det finnes en fil som heter "input.txt", hvis vi skal lese den skriver vi:

```
innfil = open("input.txt", "r")
```

Skrive til en fil:

```
utfil = open("filnavn.fitype", "w")
```

w står for "write"

Skrive til ved å legge til:

```
utfil = open("filnavn.filtype", "a")
```

a står for "append"

LUKKE FIL:

```
innfil.close()
```

lukker når man er helt ferdig med filen.

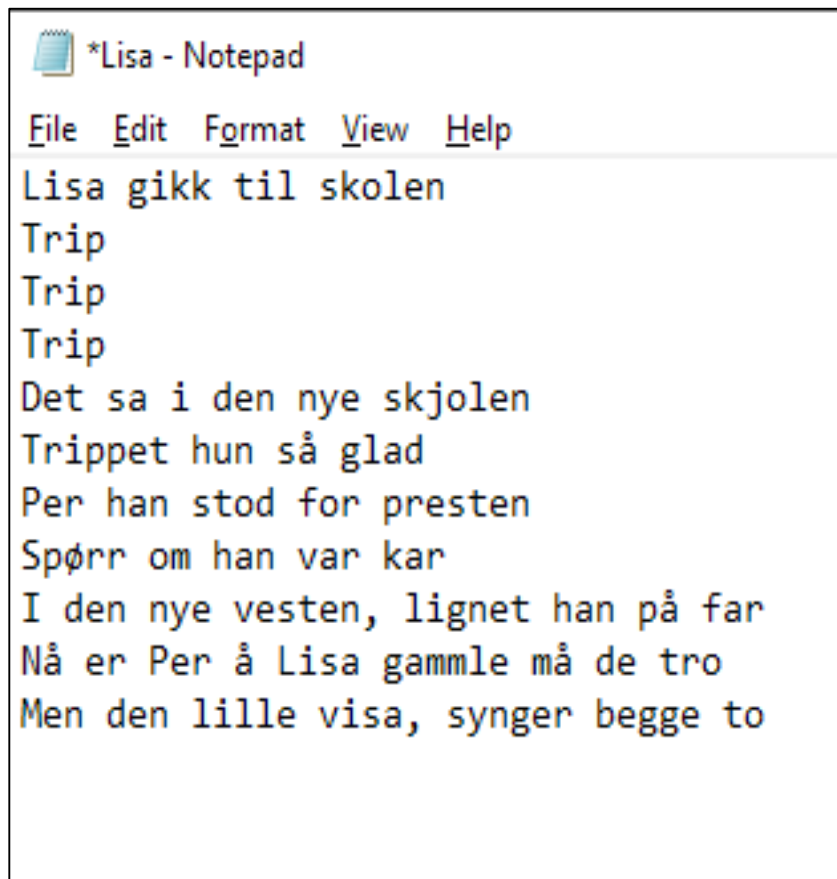
```
utfil.close()
```

LESE FRA FIL:

```
linje = innfil.readline()
```

.readline() vil lese en linje fra oppgitt fil som en String, her lagres denne string-verdien i variabelen "linje"

Hvordan ser en fil ut?



```
*Lisa - Notepad
File Edit Format View Help
Lisa gikk til skolen
Trip
Trip
Trip
Det sa i den nye skjolen
Trippet hun så glad
Per han stod for presten
Spørr om han var kar
I den nye vesten, lignet han på far
Nå er Per å Lisa gamle må de tro
Men den lille visa, synger begge to
```

Lisa.txt

| | |
|---------|---------------------------------------|
| Linje 1 | Lisa gikk til skolen\n |
| Linje 2 | Trip\n |
| Linje 3 | Trip\n |
| . | Trip\n |
| . | Det sa i den nye skjolen\n |
| . | Trippet hun så glad\n |
| | Per han stod for presten\n |
| | Spørr om han var kar\n |
| | I den nye vesten, lignet han på far\n |
| | Nå er Per å Lisa gamle må de tro\n |
| | Men den lille visa, synger begge to\n |
| | \n |
| | |

Syntaks: filinnlesing (endre fila)

Ønsker å lese innholdet i fila «Hei.txt»:

```
innfil = open("test.txt", "r")    #Åpner fila
linje = innfil.readline()        #Leser første linje: Hei!\n
linje2 = innfil.readline()       #Leser andre linje: Dette er en setning\n
```

Hei.txt

```
Hei!
Dette er en setning
```

- Slutten av fila defineres som en tom streng (`` ``).
- Hver linje kan lagres i en variabel, og håndteres som en string
- Kan fjerne ``\n`` med funksjonen strip()
- Det betyr at vi kan gjøre videre operasjoner med strengen, som f.eks split()
- Man bruker gjerne en while-løkke til å lese hele fila

Hvordan ser en fil ut?

Lisa.txt

| | |
|---------|-------------------------------------|
| Linje 1 | Lisa gikk til skolen |
| Linje 2 | Trip |
| Linje 3 | Trip |
| . | Trip |
| . | Det sa i den nye skjolen |
| . | Trippet hun så glad |
| | Per han stod for presten |
| | Spørr om han var kar |
| | I den nye vesten, lignet han på far |
| | Nå er Per å Lisa gamle må de tro |
| | Men den lille visa, synger begge to |
| | \n |
| | |

```
innfil = open("Lisa.txt", "r")  
linje = innfil.readline()  
linje2 = innfil.readline()
```



```
linje = Lisa gikk til skolen
```

```
linje2 = Trip
```

```
linje.split()
```



```
linje = ["Lisa", "gikk", "til", "skolen"]
```

```
linje2 = "Trip"
```



OPPGAVER

Oppgave 1 (oppvarming)

- Hent filen «historie.txt» fra gruppesiden vår.
- Åpne og les denne filen inn, og lagre dataen på hver linje i en liste
- Print ut lista

Oppgave 2 (litt varmere)

Dette er en videreføring fra forrige oppgave

- Endre slik at hver linje blir lagret i en ordbok istedenfor liste.
- Nøklene i ordboka er årstallet (tallverdien), og verdiene er tekstbeskrivelsen av hva som skjedde det året.
- Til slutt skriver du ut alle *nøklene* i ordboka.
- Bonus (liten utfordring): skriv ut hendelsene som skjedde det tidligste, og seneste året i oversikten.

Oppgave 3 (highest level shiiiiit)

Problemløsning: dere får et problem og skal velge litt hvordan dere vil løse det selv (les hele oppgaven før du starter).

Grunnet korona har man i flere gruppetimer ha oppmøteregistrering. Gitt filen gruppeXNavn.txt spør brukerne (i terminalen) om brukeren er til stedet, hvis brukeren taster inn x er studenten der, ellers er studenten ikke tilstede.

Oppmøtet skal registreres i en annen fil som heter uke<ukenummer>.txt (du må spørre brukeren om hvilken uke det er). Her skal alle navnene skrives til fil, men hvis brukeren er til stede skal det stå en x bak navnet.

Filen som man skriver ut skal f.eks. Se slik ut:

Tiril Torgren: x

Sandra Frogn:

Emil Orgensbø: x