GRUPPE 7

Velkommen til
IN1000 - Uke 5

Plan for dagen

- Ha kjennskap til parametere i prosedyrer og parameteroverføring
- Kjenne til funksjoner og kunne bruke de for å få unngå redundans og få mer strukturerte programmer
- Kjenner til innlesing fra og utskrift til fil, inkludert organisering av informasjon innad på linjer
- Kunne bruke filer, løkker, samlinger og funksjoner for å løse mer sammensatte problemer

Parameter og argument

- Parameter

```
def print_sum(a,b):
    print(a+b)
```

- Argument

```
print_sum(5,7)
```

Hva skjer når vi tar inn et argument?

```python
def dobble(tall):
    tall *= 2

x = 1
dobble(x)
print(x) #skriver ut 1
```

```python
def legg_til(liste):
    liste.append(10)

l = [1, 2, 3]
legg_til(l)
print(l) # [1, 2, 3, 10]
```

Prosedyre vs funksjon

- Prosedyre

```
def prosedyre(argument):
    #ting skjer
```

- Funksjon

```
def funksjon(argument):
    #ting skjer
    return #verdi
```

```python
def summer(liste):
    summen = 0

    for tall in liste:
        summen += tall

    return summen


liste_sum = summer([1,2,3,4,5])

print(liste_sum) # printer 15
```

Eksempel på funksjon

- Variabelen får verdien funksjonen returnerer

Når programmet når returen, avsluttes funksjonen

```
def summer(liste):
    summen = 0

    for tall in liste:
        summen += tall

    return summen
    print("Dette skrives ikke ut")
```

Oppgaver

- **Oppgave 1:**
  - Lag en funksjon som tar inn to heltall og returnerer det største.
  - Lagre returverdien i en variabel, og skriv ut

- **Oppgave 2:**
  - Lag en funksjon som tar inn et heltall og returnerer True dersom tallet er større enn 0, ellers False.
  - Prøv å gjør dette med så lite kode som mulig.

Filer

- Noen ganger ønsker vi å ta inn en fil og bruke det som star i filen i programet
- Andre ganger ønsker vi å skrive ut til filen
- En fil kan se slik ut:



Hvordan åpne og lese inn en fil

- Pass på at filen ligger i samme mappe som python filen!



- Syntaks:

```
fil = open("text.txt", "r") #r står for read

linje1 = fil.readline() #Dette er en fil
linje2 = fil.readline() #som har tre
linje3 = fil.readline() #linjer
```

Lese inn en fil med while-løkke

```
fil = open("text.txt") #kan droppe å ha med "r"

linje = fil.readline()

while linje != "":
    linje = fil.readline()
    print(linje)
```



Lese inn en fil med for-løkke

```
fil = open("text.txt")

for linje in fil:
    print(linje)
```



.split() og .strip()

- .split() deler opp linjen ut ifra hvilket tegn som gis som argument i funksjonen. Den returnerer en liste med bitene. Dersom ingen argument blir gitt, deles den opp på mellomrom.

- .strip() fjerner tegnet du gir som argument fra starten og slutten av linjen/strengen. Dersom det ikke gis noe argument fjernes mellomrom og linjeskift.

Eksempel

```python
fil = open("text.txt")

linje = fil.readline()

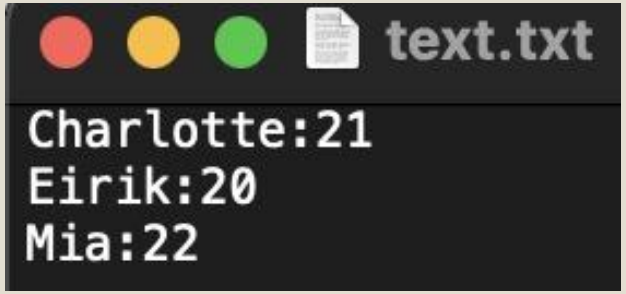
linjer = linje.strip().split()

print(linjer) # ["Dette", "er", "en", "fil"]
print(linjer[0]) # Dette
```



Oppgave

- Opprett en .txt fil som inneholder disse linjene



- Opprett en tom ordbok i python
- Les .txt filen linje for linje. For hver linje lagre navnet i en variabel, og alderen i en annen. Legg så navnet inn i ordboken som nøkkel, og alder inn som verdi.
- Skriv ut ordboken.

Hvordan skrive til fil

```
fil = open("text.txt", "w") # w står for write

fil.write("Dette er en linje")
```



- Når du åpner en fil i skrive modus (med «w»), blir innholdet i filen slettet
- Bruk .close() til å lukke filen

Oppgave 1

- Skriv en funksjon som tar imot to sannhetsverdier (True/False) og returnerer 1 hvis begge verdier er sanne, 0 hvis begge verdier ikke er sanne. *Hva tar funksjonen ikke hensyn til (hva kan gå "galt")?*

Oppgave 2

- Skriv et program med en liste [6, 3, 2, 8, 11]
  - Bruk en for-løkke til å skrive ut alle verdiene i listen
  - Skriv en prosedyre med en parameter liste. Skriv ut alle verdiene i lista ved å bruke prosedyre. Besvar deretter følgende spørsmål:
    - Hva er prosedyrens argument i dette tilfellet?
    - Hva er verdien til prosedyrens parameter i dette tilfellet?
  - Lag en funksjon som finner og returnerer den laveste verdien i lista. **Hint:** her kan det være lurt å ha en variabel som mellomlagrer den til enhver tid minste verdien

Oppgave 3

- Anta at du har filen «historiske_personer.txt»
  - Les inn alle linjene og lagre dem i en liste. Første historiske person (linje) i filen skal ligge først i lista, o.s.v. **Tips:** husk å åpne og lukke filen
  - Opprett filen «historiske_personer.txt». Fyll den med noen historiske personer ved å skrive direkte i fila. Legg fila i samme mappe som .py-filen din.
  - Test at programmet fra a) fungerer ved å bruke filen du opprettet i b).

Oppgave 4

- Lag et program som skriver historiske personers navn og fødselsår til fil. Navnene og fødselsårene skal være lagret i en ordbok, med navn som nøkler og fødselsår som verider. **Tips:** husk å åpne og lukke filen ved behov.
  - Opprett en ordbok med fem historiske personer og deres fødselsår.
  - Skriv navn og fødselsår til fil. Hver linje skal ha en historisk persons navn og fødselsår, atskilt med «;» (semikolon). Filen skal hete «historiske_personer_fodt.txt».
  - Lag et nytt program, som bruker en funksjon til å lese inn filen du laget i b) og som returnerer en ordbok tilsvarende den i a). Funksjonen skal ha parameter filnavn.