




# IN1000 UKE 5

Gruppe 8





# Plan for dagen



- **Rask oppsummering av pensum (gjøre opp for fredagen)**
- **Prosedyrer med parametre**
- **Funksjoner**
- **Lese fra og skrive til filer**

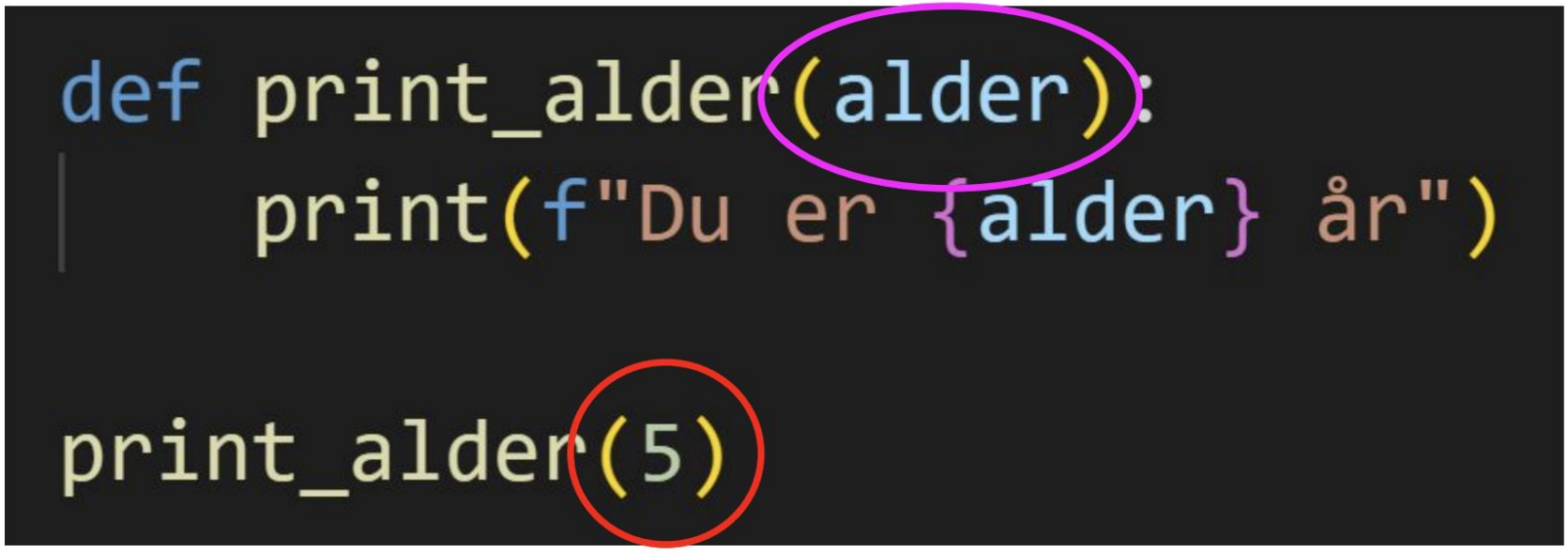


# Prosedyrer med parametre

- Mange ganger ønsker vi å utføre samme funksjonalitet (f.eks. doble et tall), men med variasjon (forskjellige tall)
- Det som varierer er parametrene
- Parametre er noe man tar inn til prosedyren så man kan bruke dem inne i prosedyren



parameter




argument

**Parameter:** variabel i prosedyren som tar imot en verdi

**Argument:** Verdi sendt inn når prosedyren kalles

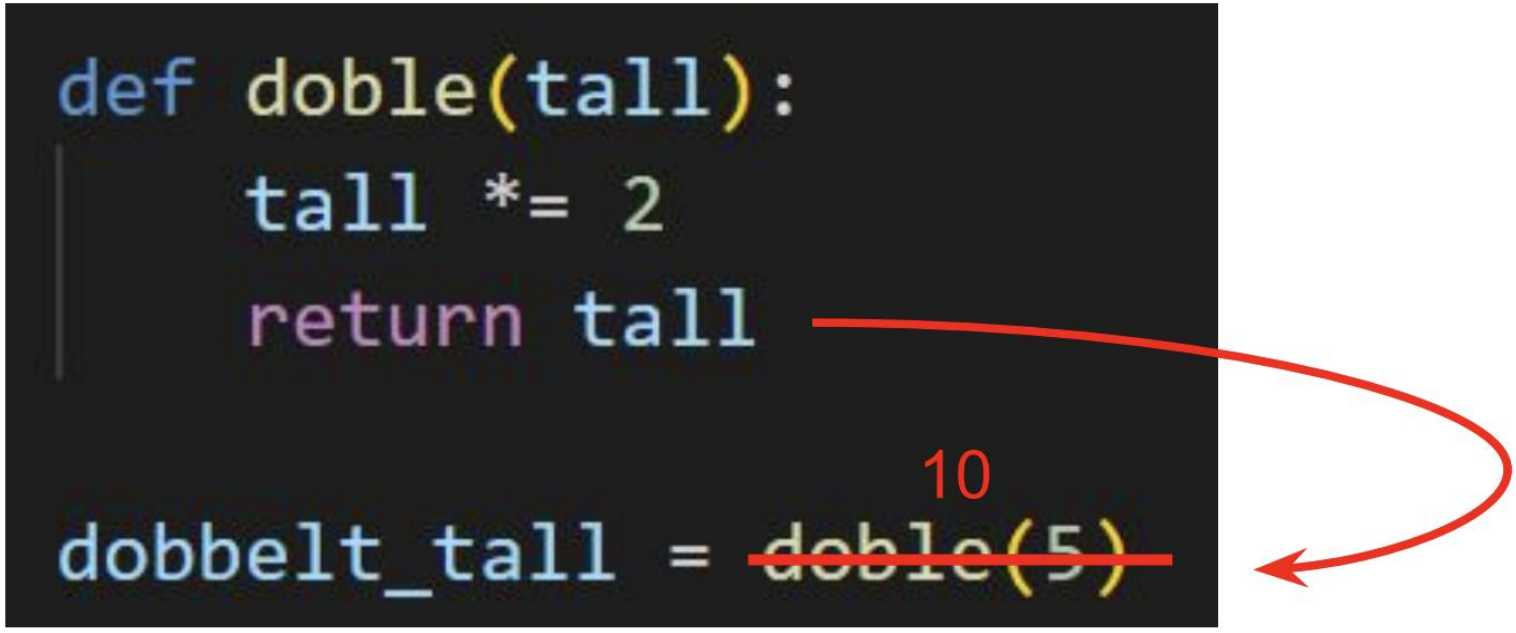


# Funksjoner



- Funksjoner er prosedyrer med returverdi, dvs. vi får tilbake en verdi etter funksjonskallet
- Dersom man ønsker å returnere mer enn 1 ting må man lage en samling, legge tingene i samlingen og returnere den
- Som regel når man kaller en funksjon tar man vare på verdien den returnerer med en variabel




def doble(tall):
tall *= 2
return tall
10
dobbelt_tall = doble(5)

# Funksjoner

- Defineres med *def*
- Funksjoner har alltid en **returverdi**

**Syntaks:**

```
def funksjon(parameter1, par2, ...):
    <din kode>
    return <returverdi>
```



# Prosedyrer

- Tilsvarende funksjon, men **uten returverdi**

**Syntaks:**

```
def prosedyre(parameter1, par2, ...):
    <din kode>
```







# Innlesing fra fil

- Tenk på en fil som en samling med linjer (dvs. vi kan iterere over linjene)
- Linjene har linjeskift (\n) etter seg

  **.strip()** fjerner linjeskift/whitespace
- Ofte har vi forskjellig data på hver linje og ønsker å dele dem

  **.split()** splitter en linje på blanke tegn

  **.split(“,”)** splitter en linje på komma

  split() returnerer en *liste*






# Åpne filer



## Lese en fil:

```
innfil = open("filnavn.filtype", "r")
# "r" står for "read" ("r" er default)
```

## Skrive til en fil:

```
utfil = open("filnavn.filtype", "w")
 # "w" står for "write"
```

## Skrive til en fil som eksisterer fra før:

```
nyUtfil = open("filnavn.filtype", "a")
# a står for "append"
```

## Lukke fil:

```
innfil.close()
utfil.close()
```

# Når filen er åpnet:

```
fil = open("filnavn", "r")
for linje in fil:
    <gjør noe med linje>
```

## FOR Å FEKS. SKRIVE TIL FIL:

```
utfil.write("Skriv dette

til fil\n")
```



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

for linje in fil:
    biter = linje.split()
    linjenr = biter[0]
    ord = biter[1]

    print(f"På {linjenr} står det: {ord}")
```

```
1   linje1 hei
2   linje2 paa
3   linje3 deg
```

```
På linje1 står det: hei
På linje2 står det: paa
På linje3 står det: deg
```

# Oppgaver



# Oppgave 1

Skriv en funksjon som tar imot to tall og returnerer det største av de to tallene.

a) Lagre tallet funksjonen returnerer i en variabel. Skriv ut variabelen.

b) Hva hadde stått i terminalen hvis du i a) hadde glemt return og så printet variabelen? Begrunn resultatet.

# Oppgave 2

Diskuter følgende spørsmål:

a) Hva skiller prosedyrer med parametre fra funksjoner med parametre?

b) Kan det fortsatt være en funksjon hvis den ikke har parametre?

c) Begrunn svaret i b). Hvorfor eller hvorfor ikke? Tenk tilbake på forelesningen og hva som skiller prosedyrer fra funksjoner.

# Oppgave 3

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 4

Skriv et program med en liste [6, 3, 2, 8, 11].

a) Bruk en for-løkke til å skrive ut alle verdiene i listen.

b) Skriv en prosedyre med en parameter *liste*. Skriv ut alle verdiene i lista ved å bruke prosedyren. Besvar deretter følgende spørsmål:
   - Hva er prosedyrens argument i dette tilfellet?
   - Hva er verdien til prosedyrens parameter i dette tilfellet?

c) 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 5

Anta at du har filen “historiske_personer.txt”.

a) Les inn alle linjene og lagre dem i en liste. Første historiske person (linje) i filen skal ligge først i lista, osv. **Tips:** husk å åpne og lukke filen.

b) 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.

c) Test at programmet fra a) fungerer ved å bruke filen du opprettet i b).

# Oppgave 6

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 verdier.

**Tips:** husk å åpne og lukke filen ved behov.

a) Opprett en ordbok med fem historiske personer og deres fødselsår.

b) 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".

c) 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*.