

# 1 Funksjoner

## 1.1 Å *definere* en funksjon

Vi har allerede sett mange python funksjoner `print()`, `input()`, `type()`, `str()`, `int()`, `float()`. Men hva er en *funksjon* egentlig?

- En funksjon er en navngitt kodeblokk som tar inn *parametre* og *returnerer* en verdi. Funksjoner brukes for å gjenbruke en kodeblokk med litt variasjoner i følge angitte argumentene.

Vi kan også *definere* våre egne funksjoner. I cellen under definerer vi en funksjon som inneholder akkurat den samme koden som før.

### Syntaks:

```
def funksjonsnavn():  
    # kodeblokken med innrykk her  
    # [...]
```

I eksempelet nedenfor definerer vi en funksjon hvor kodeblokken er vår forrige kode som beregner prisen for et hotell.

**Kommentar om utførelsen:** Ingenting skjer når vi kjører cellen!

```
[1]: def pris_hotell():  
    pris_frokost = 149.90  
    pris_rom = 1000  
    print(f"\n***Priser***\nFrokost: {pris_frokost} \nRom. {pris_rom}\n")  
  
    n_personer = int(input("Hvor mange personer er dere?"))  
    print(f"Dere er {n_personer} personer.")  
  
    pris = pris_rom + n_personer*pris_frokost  
    print(f"Da bør dere betale {pris}Kr.")
```

## 1.2 Å *kalle på* en funksjon og rekkefølgen for kodeutførelse

- For å utføre kodeblokken som er *definert* i en funksjon, må vi *kalle på* funksjonen
- Rekkefølgen for kodeutførelse av funksjoner følger rekkefølgen funksjonene blir kalt på, ikke rekkefølgen de blir definert i!

```
[2]: print("Hallo!")  
  
def pris_hotell():  
    pris_frokost = 149.90  
    pris_rom = 1000  
    print(f"\n***Priser***\nFrokost: {pris_frokost} \nRom. {pris_rom}\n")  
  
    n_personer = int(input("Hvor mange personer er dere?"))  
    print(f"Dere er {n_personer} personer.")  
  
    pris = pris_rom + n_personer*pris_frokost  
    print(f"Da bør dere betale {pris}Kr.")  
  
# "Velkommen!" er skrevet ut før teksten i funksjonen!  
print("Velkommen!")
```

```
# Husk å *kalle på* en funksjon for at koden kjøres:
pris_hotell()

print("Ha det!")
```

```
Hallo!
Velkommen!

***Priser***
Frokost: 149.9
Rom. 1000

Dere er 3 personer.
Da bør dere betale 1449.7Kr.
Ha det!
```

### 1.2.1 Gjenbruke kodeblokker: Hvorfor funksjoner er hjelpsomme!

Det er en god praksis å strukturere koden vår med funksjoner. Dette:

- lar oss bruke kodeblokken flere ganger,
- forbedrer lesbarheten (fordi koden er mer strukturert)
- forenkler fremtidig vedlikehold av koden (Fordi hvis vi må endre noe i kodeblokken, endrer vi koden bare en gang (hvor funksjonen er *definert*), ikke hver gang vi *kaller på* funksjonen)

```
[3]: def pris_hotell():
    pris_frokost = 149.90
    pris_rom = 1000
    print(f"\n***Priser***\nFrokost: {pris_frokost} \nRom. {pris_rom}\n")

    n_personer = int(input("Hvor mange personer er dere?"))
    print(f"Dere er {n_personer} personer.")

    pris = pris_rom + n_personer*pris_frokost
    print(f"Da bør dere betale {pris}Kr.")

# Reise 1
pris_hotell()

# Reise 2
# Nå kan vi bruke kodeblokken en gang til ved å kalle på funksjon igjen!
pris_hotell()
```

```
***Priser***
Frokost: 149.9
Rom. 1000

Dere er 3 personer.
Da bør dere betale 1449.7Kr.
```

```
***Priser***
Frokost: 149.9
Rom. 1000

Dere er 5 personer.
Da bør dere betale 1749.5Kr.
```

### 1.3 Parameter og returverdi

I tillegg, kan vi ha enda mer fleksibilitet med funksjoner ved å:

- bruke parametre som la oss kjøre kodeblokken med forskjellige verdier.
- bruke returverdi som la oss få tak i resultatet.

Merk at en funksjon alltid returnerer noe. Men hvis det er ingen `return`-setning, returnerer det `None`.

#### Syntaks:

```
def funksjonsnavn(param1, param2, param3):
    # kodeblokken med innrykk her
    # [...]
    return returverdi
```

Kommentar om utførelsen: Det oppstår en feil!

```
[4]: # Her legger vi to parametre til funksjonen
def pris_hotell(pris_frokost, pris_rom):

    print(f"\n***Priser***\nFrokost: {pris_frokost} \nRom. {pris_rom}\n")
    n_personer = int(input("Hvor mange personer er dere?"))

    pris = pris_rom + n_personer*pris_frokost
    print(f"Da bør dere betale {pris}Kr.")

    # Og vi returnerer prisen
    return pris

# Pengene før vi reiser
pengene_mine = 5000

# /\ Det utløser en feilmelding /\:
# argumentet mangler! Vi må gi et argument for hver parameter
# i funksjonsdefinisjon
pris_hotell()
# /\ Det utløser en feilmelding /\:
# pris er ikke definert utenfor funksjonen!
print(f"Etter din første reise har du {pengene_mine - pris}")
```

-----

**TypeError**

Traceback (most recent call last)

Cell In[4], line 20

```
14 pengene_mine = 5000
```

```
17 # /\ Det utløser en feilmelding /\:
```

```

18 # argumentet mangler! Vi må gi et argument for hver parameter
19 # i funksjonsdefinisjon
----> 20 pris_hotell()
21 # /\ Det utløser en feilmelding /\:
22 # pris er ikke definert utenfor funksjonen!
23 print(f"Etter din første reise har du {pengene_mine - pris}")

TypeError: pris_hotell() missing 2 required positional arguments: 'pris_frokost' and
↳ 'pris_rom'

```

### 1.3.1 Parameter og returverdi: uten feil

Hvis funksjonen var definert med parametre og en returverdi:

- **må vi** spesifisere verdien til parametre når vi kaller på funksjonen
- **kan vi** få tak i returverdien når vi kaller på funksjonen

I tillegg:

- Variablene (og funksjonene) er definerte i en spesifikk kontekst (skopet deres) og de eksisterer ikke utenfor skopet deres.
- En variabel som var opprettet i en funksjon eksisterer ikke utenfor funksjonen. Hvis man vil få tak i en variabel som ble definert i en funksjon utenfor funksjonen, **må** man returnere variabelen i funksjonen

```

[6]: # Her legger vi en parameter til funksjonen
def pris_hotell(pris_frokost, pris_rom):

    print(f"\n***Priser***\nFrokost: {pris_frokost} \nRom. {pris_rom}\n")
    n_personer = int(input("Hvor mange personer er dere?"))

    pris = pris_rom + n_personer*pris_frokost
    print(f"Da bør dere betale {pris}Kr.")

    # Og vi returnerer prisen
    return pris

pengene_mine = 5000

# Nå **må vi** gi to argumenter når vi kaller på funksjonen
# Og **kan vi** få tak i resultatet
# Reise 1
pris_reise1 = pris_hotell(149.9, 1000)
pengene_mine = pengene_mine - pris_reise1
print(f"Etter din første reise har du {pengene_mine}")

# Reise 2 (med rabatt på romprisen)
pris_reise2 = pris_hotell(149.9, 800)
pengene_mine = pengene_mine - pris_reise2
print(f"Etter din andre reise har du {pengene_mine}")

```

```

***Priser***
Frokost: 149.9

```

Rom. 1000

Da bør dere betale 1449.7Kr.

Etter din første reise har du 3550.3

\*\*\*Priser\*\*\*

Frokost: 149.9

Rom. 800

Da bør dere betale 1249.7Kr.

Etter din andre reise har du 2300.6000000000004