Gruppetime uke 2
IN1000

KODEFLYT, UTSKRIFT, VARIABLER, TYPER, KOMMENTARER, INPUT, BESLUTNINGER

Plan for timen

- Hvordan bruke Discourse
- Kodestil og indentering
- Input fra keyboard og typekonvertering
- Sammenligning
- Feilmeldinger
- Prosedyrer uten returverdi eller parametere

Discourse

- Hvordan bruke anonym modus
- Kanaler
- Hvordan lage nytt innlegg

Kodestil og indentering

Hva er forskjellen?

```
navn = "Nora"
alder = 16
if alder >= 18:
        print(navn, "er myndig.")
else:
      print(navn, "er ikke myndig")
```

```
NAVN = "Nora"
ALDER = 16
if ALDER >= 18:
      print(NAVN, "er myndig.")
else:
      print(NAVN, "er ikke myndig.")
```

```
n = "Nora"
a = 16
if a >= 18:
      print(n, "er myndig.")
else:
      print(n, "er ikke myndig.")
```

```
navn="Nora"
alder=16
if alder>=18:
      print(navn,"er myndig.")
else:
      print(navn,"er ikke myndig")
```

Input fra keyboard og typekonvertering

Input

```
print("Istedenfor å først printe også ta input ")
svar = input()
print(svar)

svar = input("Kan du skrive alt på en linje slik som dette ")
print(svar)

# Bruk \n for å legge ved et linjeskift før eller etter tekst
svar = input("Her gjør vi det samme bare med et linjeskift\n")
print(svar)
```

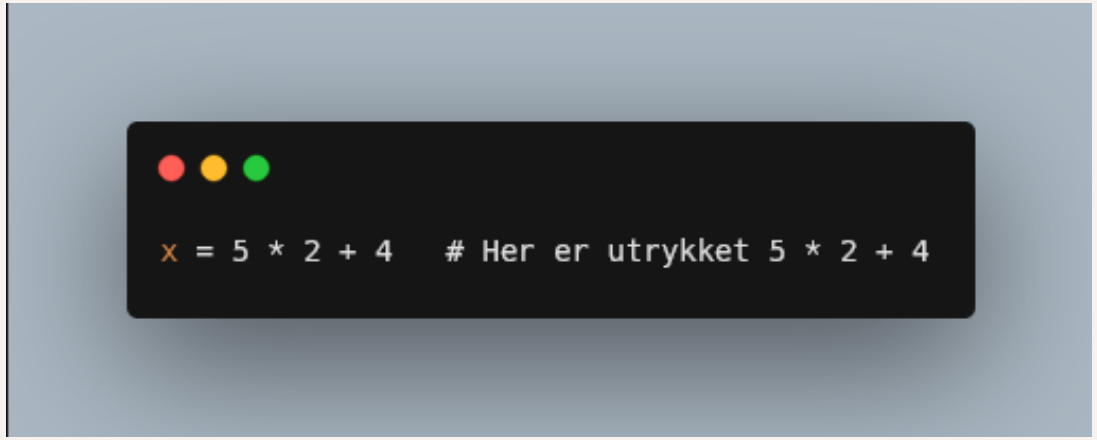
- Skriv ut melding og få input på samme linje
- Bruk \n for linjeskift

Typekonvertering

- str(4) # konvertere til string -> “4”
- int(“5”) # konvertere string til int -> 5
- int(“ 6 “) # konvertere string til int, mellomrom vil bli ignorert -> 6
- float(“17.25”) # konvertere string til float -> 17.25
- float(“3x6”) # gir Error! Siden x ikke er et tall.

Uttrykk og evaluering

Hva er et uttrykk i Python



- I Python refererer begrepet "uttrykk" til en kombinasjon av verdier og operatorer som beregner en verdi når koden utføres. Et uttrykk kan for eksempel være en enkelt verdi, en variabel eller en matematisk beregning. Uttrykk brukes til å evaluere og produsere verdier, som deretter kan brukes videre i koden.

Boolske operatorer: and, or, not

- Boolske operatorer: and, or, not
- Finnes to boolske verdier: True og False
  - Som en lysbryter, kan være slått av eller på, ikke en mellomting eller begge samtidig, ingen andre verdier er mulige
- **and** gir hvis begge utrykk er sanne
- **or** gir sant dersom en, eller begge av utrykkene er sanne
- **not** gir utrykkene motsatt verdi av hva de hadde

Oppgave

Oppgave 1
Hva evalueres de ulike uttrykkene til?

```
True and True                #
True and False               #

True or True                 #
True or False                #
False or False               #

not(True)                    #
not(False)                   #
```

Ekstra (litt mer utfordrende):

```
True and not(True)           #

True and not(False) #

not(True) or not(False)      #
```

Løsning

Løsning

```
True and True                # True
True and False               # False

True or True                 # True
True or False                # True
False or False               # False

not(True)                    # False
not(False)                   # True

Ekstra (litt mer utfordrende):
True and not(True)           # False

True and not(False)          # True

not(True) or not(False)      # True
```

Sammenligning

Hvordan sammenligne utrykk i Python

a == b	a er lik b
a != b	a er ikke lik b
a > b	a er større enn b
a >= b	a er større eller lik b
a < b	a er mindre enn b
a <= b	a er mindre eller lik b

Oppgave

Oppgave 2

Evaluer følgende uttrykk (True eller False):

```
6>6     #
6>=6    #
6<6     #
6<=6    #
6==6    #
6!=6    #
```

Løsning

Løsning

```
6>6  # False
6>=6 # True
6<6  # False
6<=6 # True
6==6 # True
6!=6 # False
```

Feilmeldinger

Typer feilmeldinger

- To hovedtyper:
  - Syntaksfeil (syntax error)

```
  File "HelloWorld.py", line 43
    print(Tallet er større enn 5)
                  ^
SyntaxError: invalid syntax
```

  - Unntak (exceptions)

```
Skriv inn et tall: w
Traceback (most recent call last):
  File "HelloWorld.py", line 37, in <module>
    tall = float(input("Skriv inn et tall: "))
ValueError: could not convert string to float: 'w'e
```

Logiske feil

- Gir ingen feilmelding
- Eksempel:

```
total = 30000
antallMnd = 13
snitt = total / antallMnd
```

  - Dette blir feil fordi det er 12 mnd i et år - ikke noe galt med koden i seg selv

Oppgave – finn tre feil

Oppgave 3
Finn 3 feil i dette programmet:

```
tall = int(input("Skriv inn et tall: "))

if tall < 5:
   print(Tallet er større enn 5)
else:
   print(Tallet er ikke større enn 5)
```

Løsning

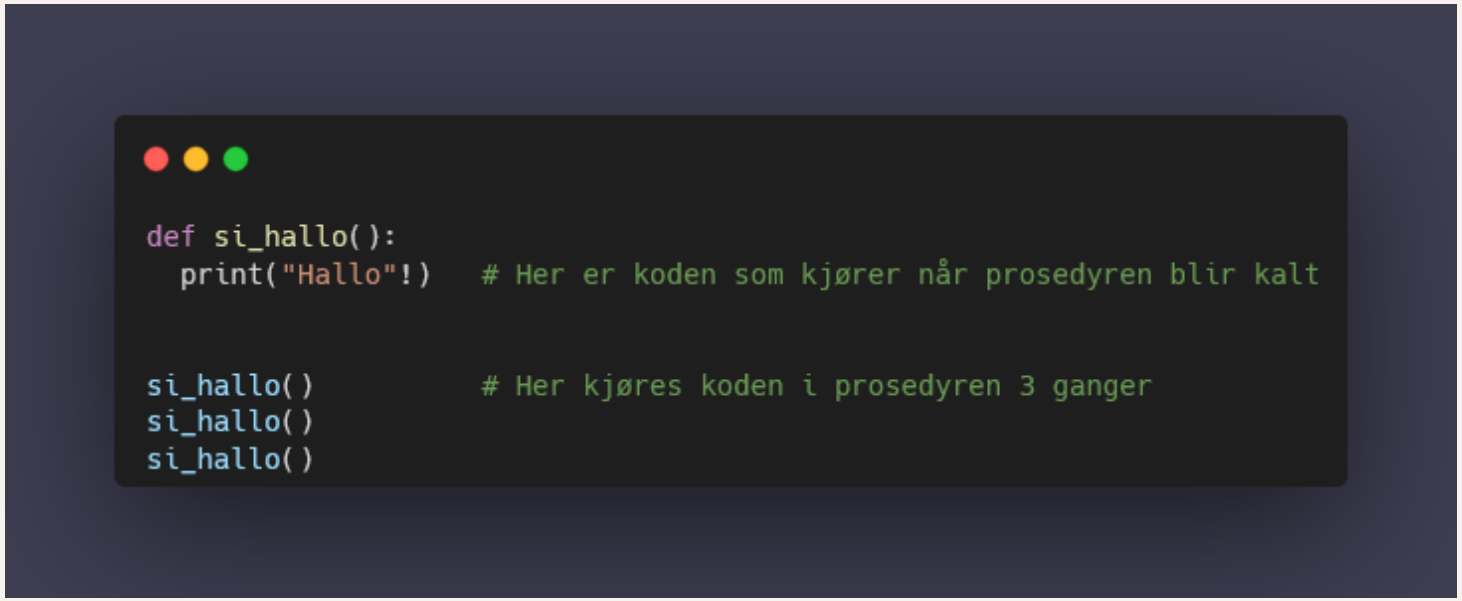
Løsning

1. Det inne i print( ) skal stå med " " rundt, ellers gir det syntaksfeil.
2. tall < 5 evalueres til true/sant dersom tallet som skrives inn er mindre enn 5 (da printes det ut at tallet er større enn 5). Dette er en logisk feil.
3. Konverterer til int - tillater ikke annet enn heltall. Feks. gir man input 5.5 kræsjer programmet, altså vil det gi en runtime error.

Prosedyrer uten returverdi eller parametere

Hva er en prosedyre

- En "blokk" med gjenbrukbar kode
- Prosedyren kan kalles på hvor som helst i programmet **etter den er definert**



```
def si_hallo():
  print("Hallo"!)   # Her er koden som kjører når prosedyren blir kalt


si_hallo()          # Her kjøres koden i prosedyren 3 ganger
si_hallo()
si_hallo()
```

Flere prosedyrer

- En prosedyre kan kalle på en annen prosedyre:

```python
def p2():
    print("Hei jeg er p2")

def p1():
    print("Hei jeg er p1 og nå kaller jeg på p2")
    p2()
    print("Det var alt")

p1()
```

Output i terminalen:

```
Hei jeg er p1 og nå kaller jeg på p2
Hei jeg er p2
Det var alt
```

Jobbe med innlevering/Trix

- Ukens pensum:
  - Trix oppgave 02.01 - 02.20
  - Innlevering 2

