UKE 4 SEMINAR

LØKKER
PROSEDYRER MED PARAMETRE
FUNKSJONER

EKSEMPEL PÅ WHILE-LOOP

```
tall = 0
while tall < 5:
    print(tall)
    tall += 1
```

```
userInput = ""
while userInput != "exit" and userInput != "quit" and userInput != "ja":
    print("Programmet fortsett her kan vi gjøre skøy ting, f.eks. kalle på metoder")
    userInput = input("Vil du avslutte programmet? ").lower()
```

OPPGAVE

Hva skrives ut?

```
x = 3

while x < 10:
    x += 1

print(x)
```

OPPGAVE

Hva skrives ut?

```
tekst = ["hadet", "på", "badet", "din", "gamle", "sjokolade"]
indeks = 0

while indeks < len(tekst):
    print(tekst[indeks])
    indeks += 2
```

OPPGAVE

Hva skrives ut?

```
tekst = ["hadet", "på", "badet", "din", "gamle", "sjokolade"]
indeks = 0

while indeks < len(tekst):
    indeks += 2
    print(tekst[indeks])
```

EKSEMPEL PÅ “FOR IN LIST”-LOOP

```
numberList = [1, 2, 3, 4, 5]
nameList = ["Kari", "Mari", "Alfred"]

for number in numberList:
    print(number)

# Bruker variabelnavn name
for name in nameList:
    print(name)

# Bruker variabelnavn banan, får samme resultat
for banana in nameList:
    print(banana)
```

EKSEMPEL PÅ “FOR IN RANGE()”-LOOP

```
print("Print 1 til 6")
for tall in range(1, 7):
    print(tall)

print("Print 0 til 6, med step 2")
for tall in range(0, 7, 2):
    print(tall)

print("Print 6 til 1")
for tall in range(6, -1, -1):
    print(tall)
```

LEK

EKSEMPEL PÅ «FOR IN DICTIONARY»-LOOP

```
ordbok = {"nøkkel1":"verdi1", "nøkkel2":"verdi2"}

# Itererer over nøkler og verdier
print("\nFor-loop over ordbok")
for nøkkel in ordbok:
    print("Nøkkelen er:", nøkkel)
    print("Verdien er:", ordbok[nøkkel])
```

EKSEMPEL PÅ «FOR IN DICTIONARY»-LOOP

```
ordbok = {"nøkkel1":"verdi1", "nøkkel2":"verdi2"}

# Itererer over nøklene
for nøkkel in ordbok.keys():
    print("Nøkkelen er: ", nøkkel)

# Itererer over verdiene
for verdi in ordbok.values():
    print("Verdien er: ", verdi)

# Itererer over nøkler og verdier
for nøkkel, verdi in ordbok.items():
    print("Nøkkelen er: ", nøkkel)
    print("Verdien er: ", verdi)
```

FOR VS. WHILE

For brukes gjerne når man vet hvor mange ganger noe skal skjer.

While brukes ofte når man ikke vet hvor mange ganger noe skal skje.
For eksempel hvis man vil lagre et ukjent antall verdier en bruker taster helt til brukeren taster 0, da er det fordelaktig å bruke while.

PROSEDYRE OG FUNKSJON

Prosedyre

- Defineres med def
- Har ikke returverdi

Eks:

```
def sum(a, b):
        c = a + b
        print(c)
```

Funksjon

- Defineres med def
- Har alltid en returverdi

Eks:

```
def sum(a, b):
        c = a + b
        return c
```

PROSEDYRE MED PARAMETER

Fordeler:Vi kan gjenbruke kode!

```
# Syntaks
def prosedyrenavn(parameter1, parameter2):
    # Du kan ha så mange parametere man vil
    # Parametrene kan være hvilke verdier som helst
    # <kode>


# Eksempel
def summer(tall1, tall1):
    print(“sum: “, tall1 + tall2)
```

LØS I PAR

Skriv en prosedyre "storst_av_to" som tar to tall som parametre og skriver ut verdien til det største tallet.

FUNKSJON

```
# Syntaks
def funksjonsnavn(parameter1, parameter2):
    # <kode>
    return #<verdi>


# Eksempel
def summer(tall1, tall2):
    return tall1 + tall2
```

LØS I PAR

Skriv en funksjon "storst_av_to" som tar imot to tall som parametre og **returnerer** det største tallet.
Deretter skrives tallet ut etter funksjonskallet.

OPPGAVE

Skriv et program med en prosedyre print_hei()
Prosedyren skal skrive ut teksten "Hei" på skjermen.

Definer en variabel "tall" med en verdi 5 og lag en while-løkke.
Løkken skal kalle prosedyren print_hei().
Bruk variabelen tall slik at løkken er ferdig etter du har kalt print_hei 5 ganger.

Skriv prosedyren print_hei.

Løs oppgaven to ganger - ved hjelp av while-løkke og for-løkke.

REPETERE?

Variabler
Input
Typer
Konvertering av typer
Forståelse av feilmeldinger
Lister
Ordbøker
For-løkker
While-løkker
Prosedyrer
Funksjoner

TRIX OG OBLIG

