Uke 4, time 1

Onsdag 10. september

Introduksjon til Python

Konseptforståelse

Løkker

- While
- For in

Løkker med samlinger

Nøstede løkker



Løkker

Vi bruker løkker når samme kode skal kjøres flere ganger

I Python har vi to hovedgrupper av løkker, en basert på vilkår – `while`, og en basert på sekvens – `for`

While-løkker

Hvis vi skal kjøre en kode om igjen så lenge vi har en betingelse som er sann, så bruker vi `while`

Strukturen er lik den som vi bruker med `if`, og koden som skal kjøres rykkes inn på samme måte

```
fortsett = True
tallene = []
print("Avslutt med [Enter]")

while fortsett:
    svar = input("Skriv et heltall: ")
    if svar == "":
        fortsett = False
    else:
        tallene.append(int(svar))

print("Tallene du skrev:")
print(*tallene)
```

While-løkker

```
teller = 10
while teller > 0:
    print(f"Teller {teller}")
    teller -= 2
```

```
x = 0
while x < 20:
    y = 0
    while y < 20:
        print(f"({x:2},{y:2}) ", end="")
        y += 3
    print()
    x += 3
```

While-løkker

```
teller = 0
while teller < 16:
    if teller % 2 == 0:
        print("Høyre")
    else:
        print("Venstre")
    teller += 1
```

```
setning = "Hva sier reven?"
teller = len(setning)
while teller > 0:
    teller -= 1
    print(setning[teller], end="")
print()
```

While-løkker

```
print("Skriv et navn")
print("Bare [Enter] avslutter")

while (svar := input()) != "":
    svar = svar.capitalize()
    print(f"{svar} er kul")
```

```
import turtle

screen = turtle.Screen()

skilpadde = turtle.Turtle()
skilpadde.shape("turtle")
skilpadde.shapesize(2,2,2)

teller = 6

while teller > 0:
    skilpadde.forward(50)  # Gå 50 skritt
    skilpadde.right(60)    # Snu 60°
    teller -= 1

turtle.done()
```

For-løkker

Når noe skal gjøres et bestemt antall ganger så kan vi bruke `for`

I Python så brukes `for` sammen med det vi kaller sekvenser, og blant andre samlinger (lister, mengder, ordbøker) er en type sekvenser

En vanlig måte er ved å bruke `range()`

I dette eksempelet kjøres løkken 10 ganger:

```
for _ in range(10):
    print("Hei!")
```

For-løkker

Vi kan også hente ut tallene som `range()` gir ut

Viktig her er at `range()` gir oss tall fra 0 til tallet vi oppgir minus 1

I eksempelet her får vi tallene: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Det kan virke litt rart, men det passer fint med f.eks indekser i lister

```
for n in range(10):
    print(n)
```

For-løkker med samlinger

Veldig ofte så ønsker vi å gå gjennom alle elementene i f.eks en liste

Det fungerer også med mengder og ordbøker

```
dyrene = ["hest", "sau", "ku", "esel"]

for dyr in dyrene:
    if dyr == "esel":
        print("Se mamma, et esel!")
    else:
        print("Se mamma, en " + dyr)
```

For-løkker med samlinger

Med ordbøker så vil få få ut nøkkelen, men så kan vi hente ut verdien ut fra nøkkelen:

Viktig å være obs på, ikke gjør endringer på listen inne i løkken eller mengden, eller legg til eller fjern nøkler i ordbøker

```
dyrene = {
    "hest": "en",
    "sau": "en",
    "ku": "en",
    "esel": "et"
}

for dyr in dyrene:
    print("Se", dyrene[dyr], dyr)
```

For-løkker med samlinger

```
dyrene = {
    "hest": 3,
    "sau": 12,
    "ku": 7,
    "esel": 1
}

for dyr in dyrene:
    if dyrene[dyr] == 1:
        print(f"1 {dyr}")
    else:
        print(f"{dyrene[dyr]} {dyr}er")
```

```
# tekst-strenger er en type lister
# hvor hvert element tilsvarer en bokstav

setning = "Hest er best"
for bokstav in setning:
    if bokstav == "b":
        print("f")
    else:
        print(bokstav)
```

Nøstede løkker

Vi kan ha løkker inne i løkker, og vi kan mikse og matche både `while` og `for`

Men det kan fort bli rotete kode om man går veldig dypt med løkker i løkker i løkker osv.

Store løkker som igjen er i løkker øker også hvor mye som kjøres av kode

```
setning = input("Skriv en setning: ")
ordene = setning.split()

for ord in ordene:
    for bokstav in ord:
        print(bokstav)
    print("---")


i = 1_000_000_000
n = 0
while i > 0:
    i -= 1
    if i % 1_000_000 == 0:
        print("Venter", i)
    for y in range(100):
        n += 1
print(n)
```

Oppgaver eller Trix uke 4

Oppgave 1

a) Forklar hva som skjer i programmet under:

```
x = 3
while x < 10:
   x += 1
   print(x)
```

b) Når slutter while-løkka å kjøre og hvorfor?

Oppgave 2

Hva skrives ut på skjermen i programmet under? Kom frem til et svar før dere kjører programmet. Kjør også programmet i Python Tutor. Her kan det være nyttig å skrive ned verdiene av a og b for hver runde løkka kjører (det vil si for hver iterasjon).

```
a = 10
b = 1

while a > 0:
   b = b * 2
   a = a - b

print("a =", a)
print("b =", b)
```

Oppgave 3

Skriv et program som har en variabel a med verdien 1 og en while-løkke. For hver runde (hver iterasjon) i while-løkka skal variabelen a adderes med 2. Løkka skal være ferdig når verdien av a er større enn 100. Skriv så ut verdien av a. Når sjekkes betingelsen i while-løkka for siste gang, og hvorfor slutter while-løkka da å kjøre?

Oppgave 4

Skriv et program som inneholder en while-løkke. Inne i løkka skal bruker bes om å taste et tall. Dersom tallet ikke er 10, skal løkka fortsette å be om tall. Når bruker taster inn tallet 10, skal vedkommende få beskjeden “Du har tastet inn tallet 10. Programmet avsluttes …”, før programmet så avslutter.

Utfordring: summer alle tallene brukeren skriver inn frem til 10 blir tastet. Når 10 tastes inn, skrives summen av tallene (ikke inkludert 10) ut. Med andre ord; hvis bruker taster sekvensen 1, 3, 4 og 10, skrives “Sum er: 8” før programmet avslutter.

Oppgave 5

a) Hva skrives ut i terminalen ved kjøring av programmet under? Kom frem til et svar før dere kjører programmet. Kjør programmet også i Python Tutor.

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

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

b) Hvorfor oppstår det feil hvis dere bytter ut < med <= i løkkebetingelsen over?

c) Hva slags løkke ville dere valgt hvis dere skulle skrevet ut alle ordene i lista? Diskutér, og lag deretter et program som skriver ut alle ordene.

Oppgaver eller Trix uke 4

Oppgave 6

Dere har lista

```
artister = ["Taylor Swift", "Astrid S", "Sam Smith"]
```

Print ut alle elementene i lista med en for-løkke ved:

a) Å løpe (iterere) gjennom lista uten å bruke listeeindekser. Beskriv all koden og hva som endrer seg underveis i kjøringen.

b) Å løpe (iterere) gjennom lista ved å bruke listeeindekser. Beskriv all koden og hva som endrer seg underveis i kjøringen.

c) Sammenlign de to variantene av for-løkke i a og b. Hva oppdager dere?

Oppgave 7

Gitt følgende liste:

```
tegnliste = ["h", "e", "i", ",", " ", "d", "u", "!"]
```

a) Bruk en for-løkke og lag en setning med tegnene. Dere kan anta at tegnene står i riktig rekkefølge i tegnliste. Skriv ut setningen i terminalen til slutt.

b) Lag en mengde av tegnliste. Gjør så det samme som dere gjorde i a), men ved å bruke mengden i stedet for. Hva oppdager dere, og hvorfor er det slik?

Oppgave 8

Skriv et program med en prosedyre, print_hei. Prosedyren skal skrive ut teksten “Hei” til terminalen. Bruk så en løkke, som kaller prosedyren print_hei. Når løkken er ferdig, skal “Hei” ha blitt skrevet ut fem ganger.

a) Skriv prosedyren print_hei. Løs så oppgaven ved hjelp av en while-løkke, deretter en for-løkke.

b) Hvilken av de to løkkene anser dere som best egnet til å løse oppgaven? Diskutér fordeler og ulemper.

Trix

Oppgaver 04.xx