GRUPPE 7

Velkommen til
IN1000 - Uke 4

Plan for dagen

- Kjenne til skrivemåte for while-løkker og for-løkker
- Kunne bruke løkker for å løse problemer
- Kunne bruke løkker sammen med samlinger

Løkker

- Brukes for å kjøre kodelinjer flere ganger
- Brukes ofte på samlinger
- Man kan bestemme hvor mange ganger løkken skal kjøre
  - To måter å gjøre dette på
- En runde i løkken kalles en iterasjon

While-løkker

```python
hemmelig = 7
gjett = None

while gjett != hemmelig:
    gjett = int(input("Gjett et tall: "))

print("Riktig!")
```

Kjører så lenge en betingelse er sann

Betingelsen sjekkes for hver iterasjon

Brukes når man ikke vet hvor mange iterasjoner

Uendelige while-løkker

- En vanlig feil er å lage en while-løkke som aldri terminerer
- Prøv å unngå dette
- Hvis det skulle skje når du kjører programmet ditt, kan du avbryte programmet ved å trykke **ctrl + c**

```
tall = 1

while tall:
    print(tall)
    tall += 1

tall = 1

while tall < 10:
    print(tall)
    #glemmer å øke tall

while True:
    print(tall)
```

Oppgave 1

- Hva skrives ut her?

```
6
9
12
```

```
x = 3

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

Oppgave 2

- Hva blir verdien til a og b?

```
a = -4
b = 8
```

```
a = 10
b = 1

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

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

Oppgave 3

- Hva skrives ut her?

```
tekst = ["Hva", "skrives", "ut", "her", "mon", "tro"]
index = 0

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



For-løkker

```
for i in range(11,1, -1):
    print(i)
```

```
11
10
9
8
7
6
5
4
3
2
```

```
liste = [3, 6, 2, 9, 4, 6, 1]

for i in range(len(liste)):
    print(liste[i])
```

- Brukes når man vet hvor mange iterasjoner
- Fra og med
- Til, ikke inklusiv
- Må ikke være fra et «kjent» tall til et annet

For each

```
liste = [3, 6, 2, 9, 4, 6, 1]

for tall in liste:
    print(tall)
```

- Brukes for å traversere en samling
- Starter på første element, fortsetter til og med siste element
- Kan kalle variabelen hva man vil

For each på ulike samlinger

- Lister, tupler og mengder
- Ordbøker
- Strenger
- Og mer (filer, egne objekter)

```python
liste = [3, 6, 2, 9, 4, 6, 1]

for tall in liste:
    print(tall)
```

```python
ordbok = {"En" : 1, "To" : 2, "Tre" : 3}

for nøkkel in ordbok:
    print(nøkkel) #Skriver ut: En, To, Tre
    print(ordbok[nøkkel]) #Skriver ut: 1, 2, 3
```

```python
streng = "Hei!"

for bokstav in streng:
    print(bokstav) #Skriver ut: H, e, i, !
```

```python
liste = [3, 6, 2, 9, 4, 6, 1]

for tall in liste:
    print(tall)

for i in range(len(liste)):
    print(liste[i])
```

Diskuter

- Hva er forskjellen på disse to for-løkkene?
- Hva kan være fordeler og ulemper med de to forskjellige?

Nøstede løkker

```
for i in range(4):
    for j in range(3):
        print(i, j)
```

- Brukes ofte for å traversere nøstede samlinger

Ytre løkke (i)	Indre løkke (j)
0	0
0	1
0	2
1	0
1	1
1	2
2	0
2	1
2	2
3	0
3	1
3	2

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

- Lag en kommandoløkke som ber brukeren om å skrive inn et tall mellom 1 og 5.
- Løkken skal avsluttes dersom tallet ikke er mellom 1 og 5.
- Løkken skal avsluttes dersom et tall blir skrevet inn to ganger.
- Løkken skal avsluttes dersom alle gyldige tall er skrevet inn

Jobb med trix eller oblig