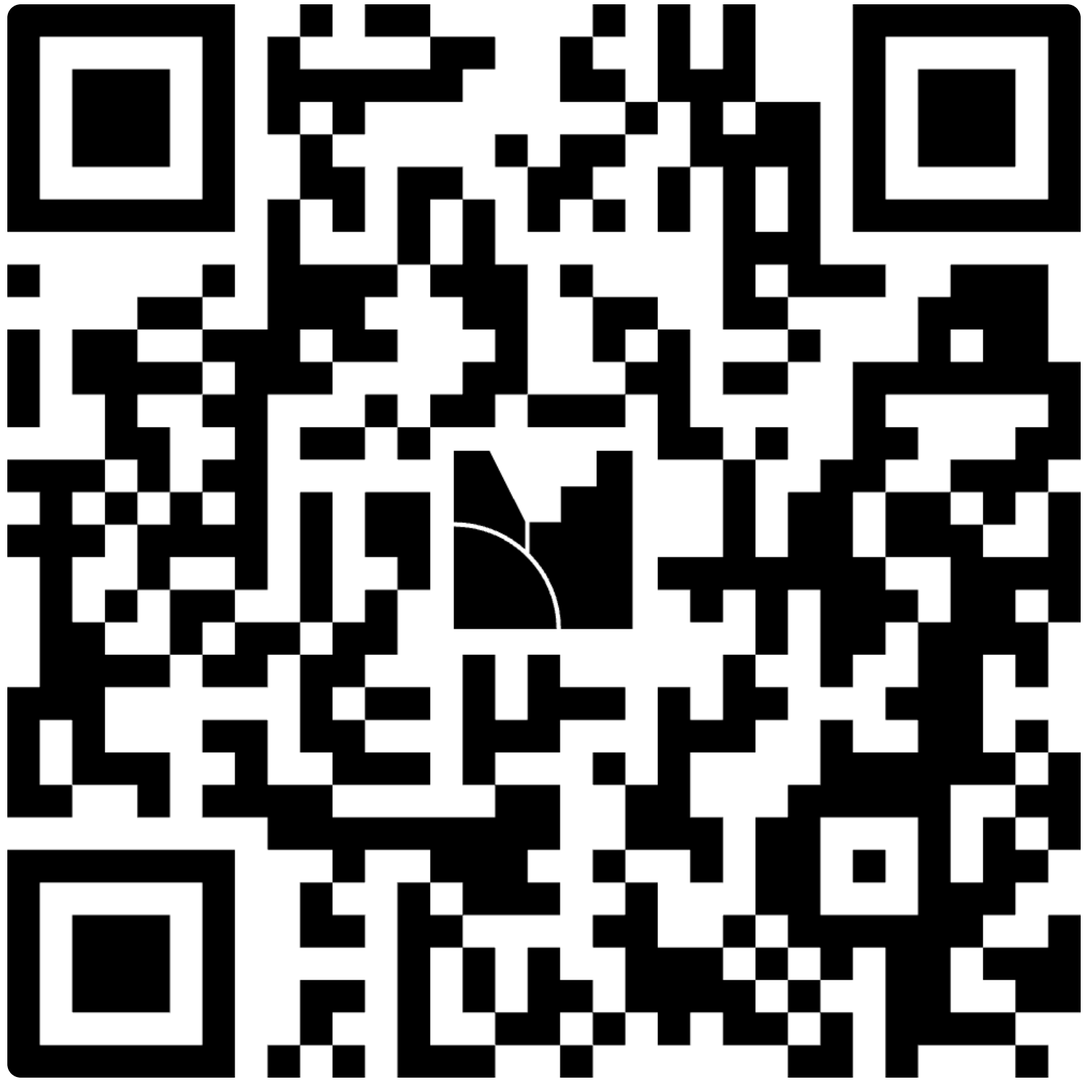
Mentimeter

Uke 4 - konsept

Gruppe 14

Kristin - kristlgj@ifi.uio.no



Mentimeter

Har dere fått sett på opptak av forelesning?



Mentimeter

Plan for gruppetimen

- Fagordsquiz
- Recap av pensum
  - while-løkker
  - for-løkker

Mentimeter

Læringsmål denne uken

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

Mentimeter

Fagordsquiz: looper i Python

- Kort quiz om sentrale begreper og syntaks
- Ingen poengscore
- Ingen tidspress

Hva vil det si å *iterere*?



Gjenta noe for hvert element i en samling

Lage en funksjon

Summere opp alle verdier i en samling

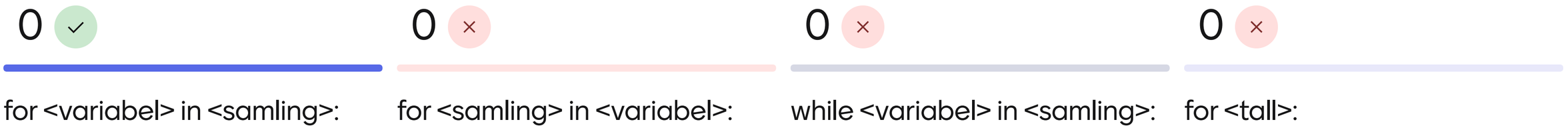
Lage en ny samling som er en iterasjon av den forrige

Mentimeter

Hvordan lager man en while-løkke?

0 ✓	0 ✗	0 ✗	0 ✗
while <uttrykk>:	for <uttrykk> in <samling>:	while(<uttrykk>)	loop while <uttrykk>:

Hvordan lager man en for-løkke?



Mentimeter

Hva brukes range(<tall>) ofte til?

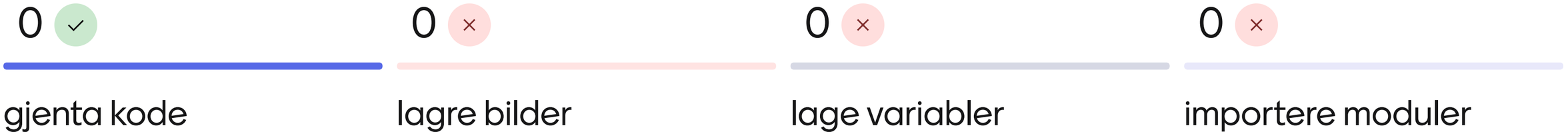
0 ✓ Lage en sekvens av tall

0 × Sjekke datatype

0 × Sortere en liste

0 × Stoppe en løkke

Hvilket ord passer best: Løkker er brukt når vi vil...



Mentimeter

Hva ble gjennomgått på forelesningen? Tenk selv i 2 min, så diskuter

- Hva lærte du, hva skjønte du ikke?
- Se gjerne over slides fra forelesningen om dere ikke husker!

Mentimeter

Hva er løkker?

- kjører innholdet flere ganger
- slipper å skrive gjentagende kode
- stor funksjonalitet i sammenheng med samlinger!
- hver “kjøring” av en løkke, altså hver runde, kaller vi en iterasjon

To typer:

- while-løkker - kjører så lenge en betingelse er sann
- for-løkker - kjører et gitt antall ganger

Mentimeter

While-løkke

Syntaks:

```
while *condition*:
    *statement*
    *statement*
    ...
```

- Kjører innholdet i løkka om og om igjen helt til condition (betingelsen) ikke lenger er True
- Nesten som en if - bare at den kjøres mange ganger
- For hver kjøring av løkka så sjekker den om betingelsen fortsatt er sann - hvis den er det så kjøres løkka, hvis ikke går programmet videre i koden
- Viktig: pass på at betingelsen vil bli False til slutt!

Eksempler:

- en løkke som kjører så lenge en variabel har en verdi mindre enn 10
- en løkke som kjører helt til brukerinput er “stopp”
- en løkke som kjører for alltid/evig (while True) – ikke gjør dette!

Mentimeter

Kodeflyt

- Hvordan eksekveres programmet?
- Når slutter løkken?
- Hva er verdien av x for hver iterasjon?
- Hva blir output?

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

Mentimeter

Kodeflyt

- Hvordan eksekveres programmet?
- Når slutter løkken?
- Hva er verdiene av a og b for hver iterasjon?
- Hva blir output?

```
a = 10
b = 1

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

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

Mentimeter

```
tall = 0
while tall < 10:
    print(tall)
    tall += 1

    if tall == 10:
        print("Tall er nå 10") # Skjer denne kodelinja?
```

Mentimeter

Evige while-løkker

```
while True:
    print("Hjelp!")
```

- unngå evige while-løkker!
- ctrl+c i terminalen for å terminere et program

- kodeord “break” for å gå ut av løkker (ikke bruk det med mindre dere absolutt MÅ - aka ikke bruk det)

```
while 3 < 5:
    print("Eeeeevig")
```

```
tall = 0
while tall < 10:
    print(tall)
```

Mentimeter

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


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

Er det noen forskjell på disse while-løkkene?

Mentimeter

For-løkke

Syntaks:

```
for *variable* in *collection*:
    *statement*
    *statement*
    ...
```

- flere måter å bruker for-løkker på
  - kjøre et bestemt antall ganger
  - kjøre gjennom elementene i en samling

Eksempler:

- en løkke som kjører 10 ganger
- en løkke som itererer over en liste (altså kjører så mange ganger som det er elementer i lista)

For-løkke og range()

```python
for i in range(5):
    print(f"Teller er nå {i}")
```

- kjøre en løkke et bestemt antall ganger
- kan kalle “i” hva du vil (dette er en variabel)
- tenk på “i” som en teller som teller iterasjoner (hver gang løkka kjøres)
- range(5) vil kjøre 5 ganger, der teller vil telle fra og med 0 opp til 5
    - (altså 0, 1, 2, 3, 4 og ikke 1, 2, 3, 4, 5) - husk indekser i lister!

```python
range(5)        # Fra og med 0, opp til (men ikke til og med) 5
range(3, 5)     # Fra og med 3, opp til (men ikke til og med) 5
range(0, 5, 2)  # Fra og med 0, opp til (men ikke til og med) 5, "hopp"/økning med 2

range(5)        # [0, 1, 2, 3, 4]
range(3, 5)     # [3, 4]
range(0, 5, 2)  # [0, 2, 4]
```

Mentimeter

For-løkke og samlinger

```
for element in liste:
```

- range() lager egt bare kjapt en liste
- kan altså bruke for-løkker for å “gå”/iterere gjennom en samling (for eksempel en liste)
- for hver iterering/runde ser man på et og et element i samlingen
- løkka vil kjøre hele lengden av samlingen
- kall variabelen noe forståelig!

```
navn_liste = ["Bob", "Bib", "Bab", "Beb"]
for navn in navn_liste:
    print(navn)
```

```
for i in range(5):
    print(f"Teller er {i}")


liste = [0, 1, 2, 3, 4]
for tall in liste:
    print(f"Teller er {tall}")
```

Mentimeter

Strenger er en type liste!

```
tekst = "Kjempegøy"
for bokstav in tekst:
    print(bokstav)
```

```
K
j
e
m
p
e
g
ø
y
```

Mentimeter

```
liste = ["hvem", "hva", "hvor"]
for element in liste:
    element = "supert"
```

```
['hvem', 'hva', 'hvor']
```

Variabelen (her “element”) vi ser på for hver iterasjon er en kopi av elementene i lista vi går igjennom. Når vi endrer variabelen endrer ikke selve elementet i lista seg, vi endrer bare kopien!

Mentimeter

```
liste = ["hvem", "hva", "hvor"]
for element in liste:
    element = "supert"
```

```
['hvem', 'hva', 'hvor']
```

```
liste = ["hvem", "hva", "hvor"]
for indeks in range(len(liste)):
    liste[indeks] = "supert"

print(liste)
```

```
['supert', 'supert', 'supert']
```

```
liste = ["hvem", "hva", "hvor"]
indeks = 0
for element in liste:
    liste[indeks] = "supert"
    indeks += 1

print(liste)
```

Mentimeter

Når bruker man hvilken løkke?

- for-løkker når man vet hvor mange ganger noe skal skje
  - feks: du skal kalle på en prosedyre et gitt antall ganger
  - feks: du skal endre alle elementene i en liste
- while-løkke når antall ganger avhenger av noe (du vet ikke akkurat tallet)
  - feks: du skal kalle på en prosedyre så mange ganger bruker vil
  - feks: du skal be bruker om ny input helt til inputen er gyldig
  - feks: du skal gjøre noe helt til en variabel får en bestemt verdi
- flere ganger kan man velge!

Mentimeter

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 1

Mentimeter

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.

Oppgave 2

Mentimeter

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 3

Skriv ned en ting som ble klarere og en ting du fortsatt sliter med:)