



UNIVERSITETET I BERGEN

# FLERE SPØRSMÅL

INF100

HØST 2024

Torstein Strømme

FÅR VI VITE RESULTATENE PÅ  
PRØVEEKSAMEN?

SENSORVEILEDNING LIGGER UTE

KAN MAN KLADDE PÅ EKSAMEN?

JA

HVA LIKER DU Å SPILLE?

SNAKE

# HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
a = ['foo', 'bar', 'baz']
```

```
d = {  
    0: 'foo',  
    1: 'bar',  
    2: 'baz',  
}
```

```
a[2] == d[2]
```

# HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
a = ['foo', 'bar', 'baz']
```

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

```
d = {  
    0: 'foo',  
    1: 'bar',  
    2: 'baz',  
}
```

```
for i in d:  
    print(i, d[i])
```

# HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
a = ['foo', 'bar', 'baz']
```

```
for value in a:  
    print(value)
```

```
d = {  
    0: 'foo',  
    1: 'bar',  
    2: 'baz',  
}
```

```
for value in d.values():  
    print(value)
```

# HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
a = ['foo', 'bar', 'baz']
```

```
a.append('qux')
```

```
d = {  
    0: 'foo',  
    1: 'bar',  
    2: 'baz',  
}
```

```
d[len(d)] = 'qux'
```



# HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
a = ['foo', 'bar', 'baz']
```

```
d = {  
    0: 'foo',  
    1: 'bar',  
    2: 'baz',  
}
```

# HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
a = ['foo', 'bar', 'baz']
```

```
a.pop(1)
```

```
d = {  
    0: 'foo',  
    1: 'bar',  
    2: 'baz',  
}
```

```
?? d.pop(1) ??
```

# HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
a = ['foo', 'baz']
```

```
a.pop(1)
```

```
d = {  
    0: 'foo',  
    2: 'baz',  
}
```

```
?? d.pop(1) ??
```

# HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
a = ['foo', 'baz']
```

```
d = {  
    0: 'foo',  
    2: 'baz',  
}
```

Både lister og oppslagsverk

- Er samlinger med (referanser til) annen data
- Kan være store eller små
- Kan muteres
- Oppslag på gitt nøkkel/indeks er svært raskt

# HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
a = ['foo', 'baz']
```

```
d = {  
    0: 'foo',  
    2: 'baz',  
}
```

## Lister

- Verdiene har alltid en rekkefølge
- -> Kan beskjæres

## Oppslagsverk

- Kan ha nøkler («indekser») som ikke har rekkefølge
- Kan ha nøkler som ikke er heltall (f. eks. strenger)

HVA ER REGLENNE RUNDT BRUK AV  
EKSTERNE PAKKER PÅ EKSAMEN?

PAKKER VI HAR BRUKT I EMNET ER OK

VIL NIVÅET PÅ EKSAMEN VÆRE OMTRENT  
LIKT SOM PRØVEEKSAMEN, ELLER KAN  
DEN VÆRE VANSKELIGERE?

DEN VIL IKKE VÆRE VANSKELIGERE

BLIR DET NOEN SPØRSMÅL PÅ EKSAMEN  
DER MAN MÅ KODE EN HEL KODE SELV?

JA



# HVA ER FORSKJELLEN PÅ FOR- LØKKER OG WHILE-LØKKER?

```
i = 1
while i <= 10:
    print(i)
    i += 1
```

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

# HVA ER FORSKJELLEN PÅ FOR- LØKKER OG WHILE-LØKKER?

*betingelse (uttrykk utføres hver iterasjon)*

```
i = 1  
while i <= 10:  
    print(i)  
    i += 1
```

*samling (uttrykk utføres én gang)*

*iterand*

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

# HVA ER FORSKJELLEN PÅ FOR- LØKKER OG WHILE-LØKKER?

WHILE

FOR



Løkken kan vare «for alltid»



Egnet når man vet hvor mange iterasjoner  
som behøves før løkken starter



Egnet når man ikke vet hvor mange iterasjoner  
som behøves på forhånd



KAN MAN SKRIVE MED ORD HVORDAN  
MAN VILLE LØST OPPGAVEN HVIS MAN  
IKKE HUSKER SYNTAKS?

GIR DELVIS UTTELLING

ER KURSNOTATENE VI FÅR UNDER  
EKSAMEN LIK DEN VI HAR BRUKT DETTE  
SEMESTERET, ELLER ER DEN LIK DE SOM  
ER UTE I PRØVEEKSAMENENE?

SOM DETTE SEMESTERET

# NÅR SKAL MAN BRUKE TRY/EXCEPT OG NÅR SKAL MAN BRUKE IF/ELSE?

```
num = input('Din alder: ')

if num.isdigit():
    num = int(num)
    num = num + 1
    print('Snart er du', num, 'år.')
else:
    print('Du må skrive et tall!')
```

```
num = input('Din alder: ')

try:
    num = int(num)
    num = num + 1
    print('Snart er du', num, 'år.')
except ValueError:
    print('Du må skrive et tall!')
```

# NÅR SKAL MAN BRUKE TRY/EXCEPT OG NÅR SKAL MAN BRUKE IF/ELSE?

RETURN

Avslutter ett funksjonskall

KRASJ/EXCEPTION

Avbryter alle funksjonskall helt til 'except'

# NÅR SKAL MAN BRUKE TRY/EXCEPT OG NÅR SKAL MAN BRUKE IF/ELSE?

## IF/ELSE

Bruk ved «normal» programflyt

## TRY/EXCEPT

Når håndtering med if/else gir veldig kompleks kode

Brukes ofte for å håndtere feilformattert data

Når kode for å håndtere feilen er «langt unna»  
der feilen skjer

Må aldri brukes for å skjule egne feil!



HVORDAN TAR SENSOR HENSYN TIL AT  
MAN IKKE KAN KJØRE KODE/SE SYNTAX  
FEIL?

LETER ETTER STRUKTURER SOM  
BEHØVES I KORREKT LØSNING

KAN MAN TA MED DELER AV ET  
LØSNINGSFORSLAG PÅ EN GAMMEL  
EKSAMEN I NOTATENE SINE?

JÅ, MEN IKKE KOPIER FRA ANDRE NÅR  
DU BESVARER EKSAMEN

# HVORDAN ÅPNE, LESE, SKRIVE OG BRUKE FILER?

## LESE FRA FIL

```
from pathlib import Path  
content = Path('my_file.txt').read_text(encoding='utf-8')
```

## SKRIVE TIL FIL

```
content = 'My content'  
Path('my_file.txt').write_text(content, encoding='utf-8')
```

# HVA ER EGENTLIG QUOTECHAR NÅR VI SKAL BRUKE CSV-BIBLIOTEKET?

*sample.csv*

```
lat,long,place_name
-33.8567844,151.213108,"Opera House, Sydney"
-35.3075,149.124417,"Parliament House, Canberra"
```

*foo.py*

```
from pathlib import Path
import csv
import io

content = Path('sample.csv').read_text(encoding='utf-8')
reader = csv.DictReader(io.StringIO(content), delimiter=',', quotechar='')
```

# HVA ER EGENTLIG QUOTECHAR NÅR VI SKAL BRUKE CSV-BIBLIOTEKET?

*sample.csv*

```
lat, long, place_name
-33.8567844, 151.213108, "Opera House, Sydney"
-35.3075, 149.124417, "Parliament House, Canberra"
```



*foo.py*

```
from pathlib import Path
import csv
import io

content = Path('sample.csv').read_text(encoding='utf-8')
reader = csv.DictReader(io.StringIO(content), delimiter=',', quotechar='"')
```

# HVORDAN LAGER MAN ASTEROIDS?

