

SPØRSMÅL

INF100

HØST 2025

Torstein Strømme

HVOR MANGE POENG MÅ MAN HA FOR Å BESTÅ?

```
def get_letter_grade(point_score):  
    if point_score >= 90: return 'A'  
    elif point_score >= 80: return 'B'  
    elif point_score >= 65: return 'C'  
    elif point_score >= 55: return 'D'  
    elif point_score >= 40: return 'E'  
    else: return 'F'
```

ER DET SÅNN AT LABENE TELLER SOM
EKSTRA POENG PÅ EKSAMEN, ELLER ER
LABENE EN EGEN PROSENTDEL AV DEN
ENDELIGE KARAKTEREN I INF100?

JA

ER DET SÅNN AT LABENE TELLER SOM
EKSTRA POENG PÅ EKSAMEN, ELLER ER
LABENE EN EGEN PROSENTDEL AV DEN
ENDELIGE KARAKTEREN I INF100?

0 - 80
0 - 200

```
def get_point_score(exam_score, total_lab_score):  
    return exam_score + total_lab_score / 10
```

KAN LABENE TELLE NOK TIL Å
BESTÅ?

NEI

MÅ MAN HA 40% RIKTIG PÅ EKSAMEN FOR
AT DE PROSENTENE FRA LAB TELLES,
ELLER ER DET PROSENTEN MAN FÅR
TILSAMMEN SOM UTGJØR KARAKTEREN?

BÅDE INNLEVERINGAR OG EKSAMEN
MÅ VERE BESTÅTT

ER DET SANNSYNLIG AT KLOKKESLETTET
FOR EKSAMEN BLIR FLYTTET? STÅR AT
MAN MÅ VÆRE FORBREDT PÅ AT DET KAN
BLI FLYTTET FREM TIL 14 DAGER FØR
EKSAMEN

TJA
(DET HAR SKJEDD FØR)

HVORDAN BLIR EKSAMEN?

SAMME FORMAT SOM V25

KAN MAN KLADDE PÅ EKSAMEN?

JA

HVORDAN SETTE OPP PCEN KLAR FOR
EKSAMEN? MÅ MAN LASTE NED NOE?

[https://www.uib.no/student/114302/gj%C3%](https://www.uib.no/student/114302/gj%C3%B8r-deg-klar-til-eksamen)
[B8r-deg-klar-til-eksamen](https://www.uib.no/student/114302/gj%C3%B8r-deg-klar-til-eksamen)

HVILKE HJELPEMIDLER ER TILLATT
PÅ EKSAMEN?

BØKER FRÅ LITTERATURLISTA OG OPP TIL
6 TOSIDIGE A4 ARK MED EIGNE NOTAT.

HVA VIL DU ANBEFALE Å SKRIVE PÅ DE 6
TOSIDIGE A4 ARKENE MED NOTATER VI
FÅR HA MED PÅ EKSAMEN? HVA SLAGS
INFORMASJON TROR DU VI KOMMER TIL Å
FÅ MEST BRUK FOR?

DINE SVAR PÅ QUIZ2
DINE PRESENTASJONER
DINE FAVORITTEKSEMPLE PÅ
CSV, JSON, PLOTTING, FILER, LØKKER, ETC
DINE FAVORITTER FRA TIDLIGERE EKSAMENER

HVA LIKER DU Å SPILLE?

SNAKE

HVA ER FORSKJELLEN PÅ LISTER OG
OPPSLAGSVERK?

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)

HVILKE GRUNNLEGGENDE KODEFERDIGHETER BURDE VI ABSOLUTT KUNNE FOR Å FÅ EN GJENNOMSNIITTLIG KARAKTER ELLER BEDRE?

A: Kan løse problemer som krever at man gjør egne valg i forhold til hvilke datastruktur som er egnet og hvordan løsningsrommet representeres. (for eksempel: finn lengste subsekvens av en liste slik at differansen mellom etterfølgende elementer i subsekvensen er lik)

C: Kan kombinere ulike konsepter for å løse klassiske problemstillinger (for eksempel: gitt en CSV med kolonner for student-id og poengscore på ulike laber: skriv ut alle studenter som har total lab-score på minst 100 poeng)

E: Kjenner til begreper og kan spore enkle kodesnutter. Kan skrive kode med løkker og betingelser under isolerte omstendigheter (f. eks. gitt en liste a med tall, skriv ut alle tallene i listen som er større enn 42).

Hvordan blir prøveeksamen?

Er det noe det ikke er lov til å ha med i notatene? Feks bilder?

Vil det bli trukket mye poeng om koden din er skrevet på en vanskeligere måte ...

Skjønte ikke helt poenggivningen, hvis man får 100% på labene og 20% på eksa...

Hvordan vil du anbefale å jobbe med programmering for å bli bedre og få mer f...

Det er slik at hvis man besto snarveisprøven så gjeler eksamen 100% av karakte...

kan man få halve poeng på eksamen? Og gjøres poengsummen fra lab om til helta...

Hva slags tema bruker du i VScode?

Blir presentasjonen fra dagens forelesning lagt ut?

dude må jeg printe ut kursnotatene eller har jeg tilgang digitalt under eksamen

HVORDAN ØVE TIL EKSAMEN?

PROGRAMMERE
FORKLARE
LØKKE-ØVELSER
LISTE-ØVELSER
TIDLIGERE EKSAMENER

HVORDAN FUNGERER ENUMERATE?

```
students = ['Ola', 'Kari', 'Kine']
```

```
enumerated_students = enumerate(students)
```

```
enumerated_students = [  
    (0, 'Ola'),  
    (1, 'Kari'),  
    (2, 'Kine'),  
]
```

HVORDAN FUNGERER ENUMERATE?

Løkke med både
indeks og element

```
students = ['Ola', 'Kari', 'Kine']  
for i, student in enumerate(students):  
    print(student, 'is on index', i)
```

Ola is on index 0
Kari is on index 1
Kine is on index 2

NÅR SKAL VI BRUKE DESTRUKTIVE FUNKSJONER, OG NÅR SKAL VI BRUKE IKKE-DESTRUKTIVE FUNKSJONER?

egnet for «store» data

DESTRUKTIVE FUNKSJONER

Mer effektivt for kjøretid og minnebruk

Ikke nødvendig med retur-verdi

egnet for «små» data

IKKE-DESTRUKTIVE FUNKSJONER

Når du vil ta vare på originalen

Når du vil unngå bugs

Funksjonskallet «kommuniserer» kun ved returverdi

HVA ER REGLENNE RUNDT BRUK AV
EKSTERNE PAKKER PÅ EKSAMEN?

PAKKER VI HAR BRUKT I EMNET ER OK

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

JA

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

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

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='"')
```