

EKSAMENSFORBEREDELSE

INF100

VÅR 2026

Crystal Chang Din

Med bidrag fra: Torstein Strømme

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 H25

KAN MAN KLADDE PÅ EKSAMEN?

JA

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

[https://www4.uib.no/for-studenter/eksamen-
og-innleveringer](https://www4.uib.no/for-studenter/eksamen-og-innleveringer)

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?

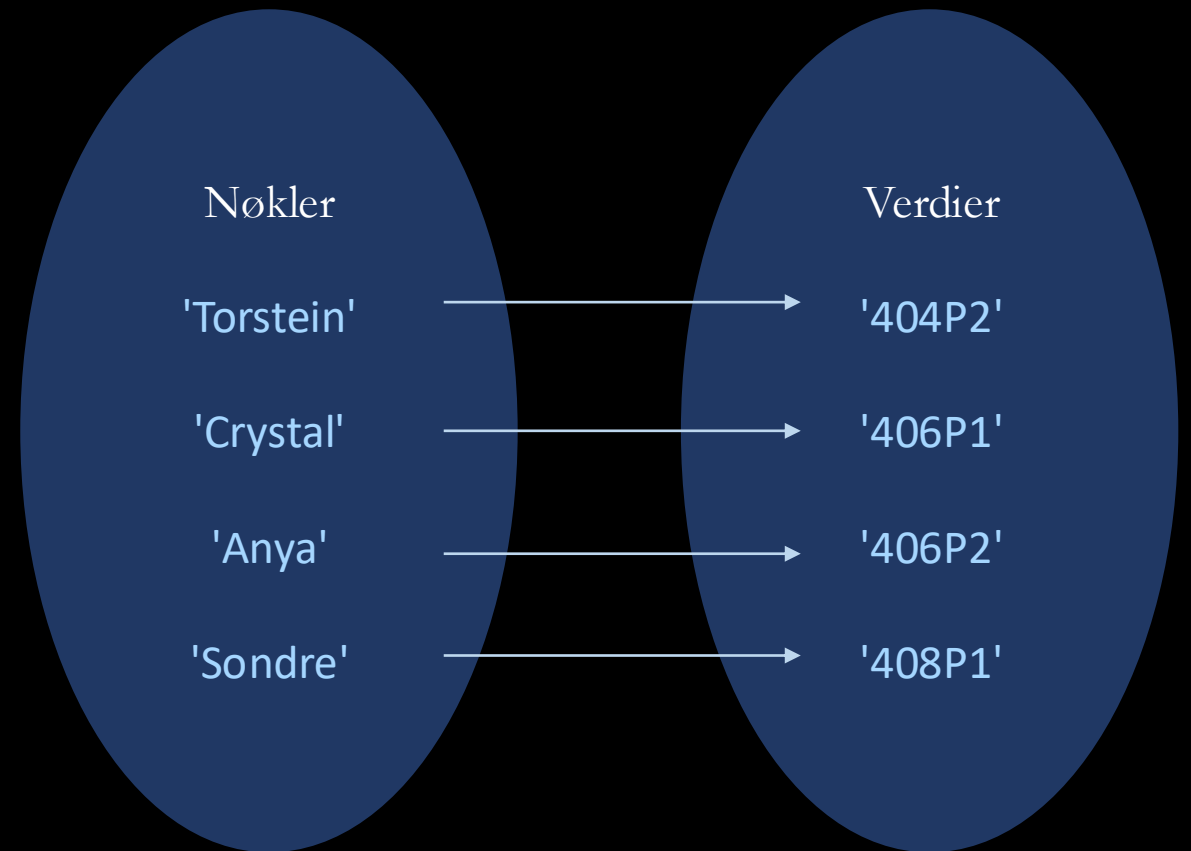
DINE SVAR PÅ QUIZ2
DINE PRESENTASJONER
DINE FAVORITTEKSEMPLER PÅ
CSV, JSON, FILER, LØKKER, ETC
DINE FAVORITTER FRA TIDLIGERE EKSAMENER

HVA ER FORSKJELLEN PÅ LISTER OG
OPPSLAGSVERK?

OPPSLAGSVERK

```
office = {  
  'Torstein': '404P2',  
  'Crystal': '406P1',  
  'Anya': '406P2',  
  'Sondre': '408P1',  
}
```

office

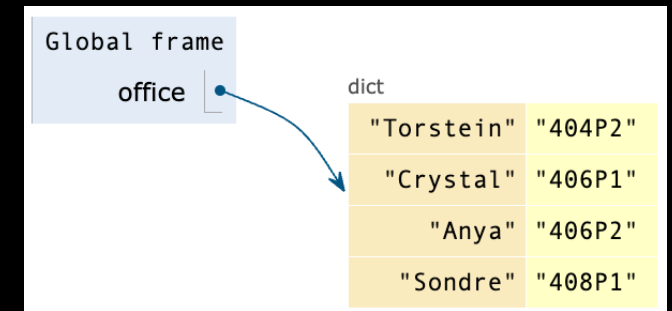


OPPSLAGSVERK

Et oppslagsverk er som en telefonkatalog:
man kan slå opp på en nøkkel, og få vite dens verdi

```
office = {  
    'Torstein': '404P2',  
    'Crystal': '406P1',  
    'Anya': '406P2',  
    'Sondre': '408P1',  
}
```

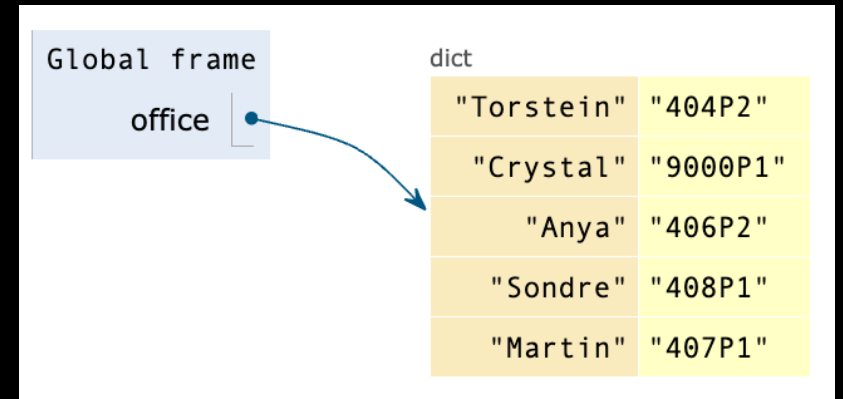
```
print('Torstein sitt kontor er', office['Torstein'])
```



OPPSLAGSVERK

```
office = {  
    'Torstein': '404P2',  
    'Crystal': '406P1',  
    'Anya': '406P2',  
    'Sondre': '408P1',  
}
```

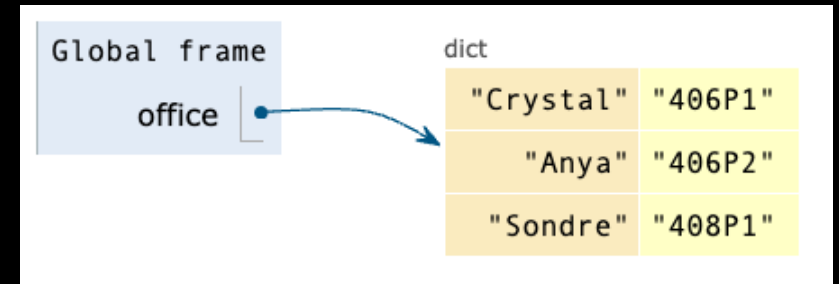
```
office['Martin'] = '407P1' # Legg til ny nøkkel/verdi  
office['Crystal'] = '9000P1' # Crystal bytter kontor
```



OPPSLAGSVERK

```
office = {  
    'Torstein': '404P2',  
    'Crystal': '406P1',  
    'Anya': '406P2',  
    'Sondre': '408P1',  
}
```

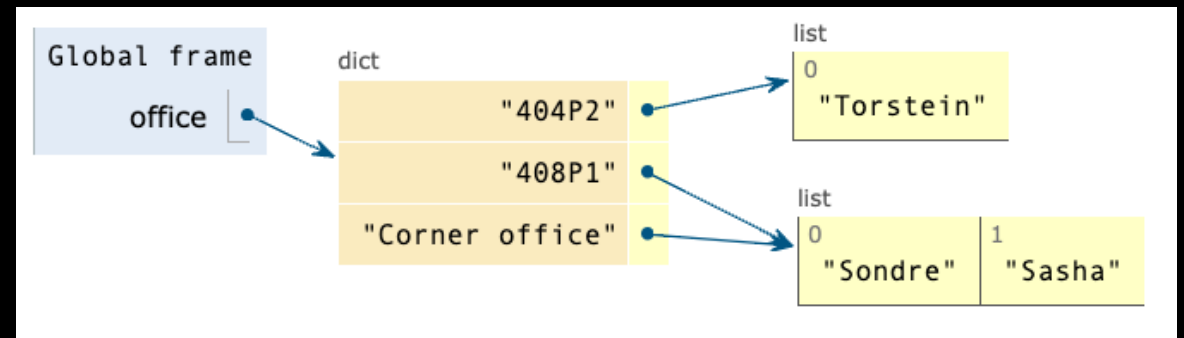
```
office.pop('Torstein') # Fjern nøkkel
```



OPPSLAGSVERK

```
office = {  
    '404P2': ['Torstein'],  
    '408P1': ['Sondre', 'Sasha'],  
}
```

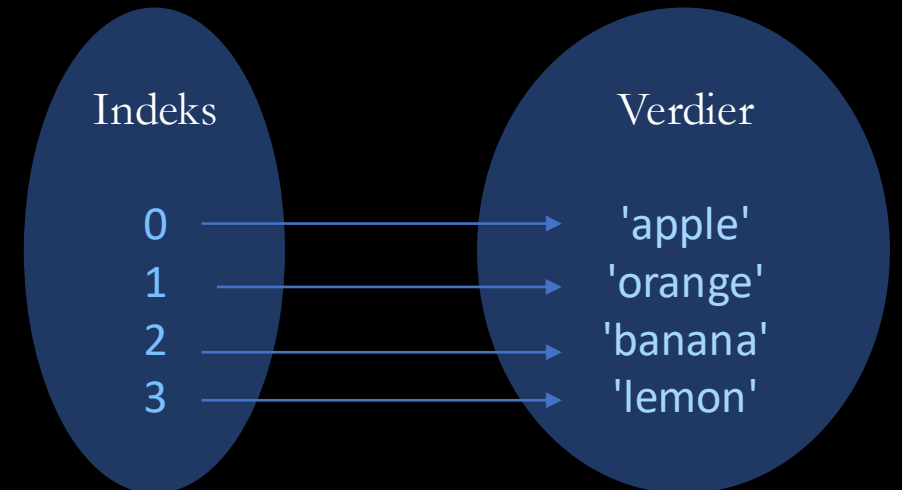
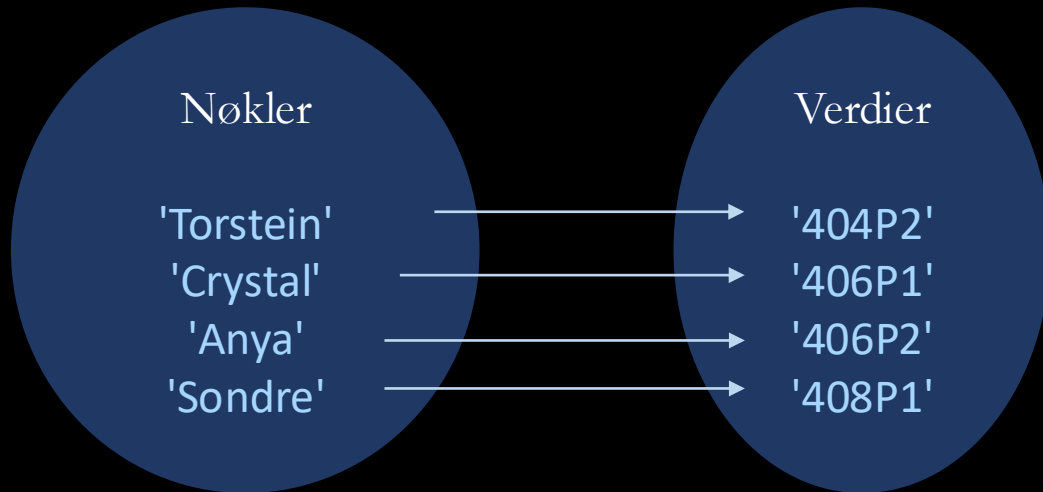
```
office['Corner office'] = office['408P1'] # alias
```



HVA ER FORSKJELLEN PÅ LISTER OG OPPSLAGSVERK?

```
office = {  
  'Torstein': '404P2',  
  'Crystal': '406P1',  
  'Anya': '406P2',  
  'Sondre': '408P1',  
}
```

```
fruits = [  
  'apple',  
  'orange',  
  'banana',  
  'lemon',  
]
```



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 («indeksen») 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 ØVE TIL EKSAMEN?

PROGRAMMERE
FORKLARE
LØKKE-ØVELSER
LISTE-ØVELSER
TIDLIGERE EKSAMENER

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 REGLENENE 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?

J^A, MEN IKKE KOPIER FRA ANDRE NÅR
DU BESVARER EKSAMEN

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"
```

quotechar



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='"')
```

delimiter



HJELP, DET ER EKSAMEN!

Bruk kursnotatene, ikke KI

Gjør tidligere eksamensoppgaver

Forklare løsninger for hverandre

Gjør tidligere laber på nytt uten hjelp

Begynn tidlig med repetisjon

Øv på å skrive pseudokode/skrive med «grove steg»

Fokuser på grunnleggende konsepter

CodingBat og lignende

Lag egne notater til å ta med på eksamen

Delta på gruppetimer, drop-inn timer

Skriv ned underveis det som er vanskelig

Øv på manuell kodesporing

Bruker Python tutor

VANLIGE FEIL PÅ EKSAMEN

- Å ikke prøve
- Manglende identifikasjon av delproblemer
 - For mange ting i samme funksjon (manglende bruk av hjelpefunksjoner)
- Dårlige variabelnavn
 - `for i in a:` vs `for i in range(len(a)):`

VANLIGE FEIL

- Presedens

- `x and y in z`
- `x == 3 or 4`

- Funksjoner

- `print` vs `return`
- `return` i destruktive funksjoner

- Lister

- indeks vs. element
- løkker over indeks vs elementer

- Løkker

- For tidlig `return`
- Hva skal gjøres én gang, hva skal gjentas flere ganger?

EKSAMENSSTRATEGI

- Les gjennom alle oppgavene
 - Ikke kast bort tid på et problem du ikke har en klar plan for før du har lest nøye gjennom alle oppgavene.
- Første gang du leser problemet: identifiser delproblemer
 - Skriv ned kommentarer/idéer. Hva skal gjentas flere ganger? Hjelpesfunksjoner gjør hva?
 - Selv om du ikke klarer løse hele problemet, kjenner du kanskje igjen en del av det du klarer å løse.
- Prioriter oppgaver du kan løse raskt og oppgaver med mye poeng.

EKSAMENSSTRATEGI

- Bruke selvbeskrivende variabelnavn
 - Lettere for deg selv å forstå hva du driver med
 - Lettere for sensor å forstå hva du driver med
- Spis godt før du starter
- Sov godt før du starter
- Gå en lang tur dagen før (men ikke så sent på kvelden at du ikke får sove)

TIPS FRA GRUPPELEDERNE

Hvordan kan INF100 studenter forberede seg til eksamen?

- Gjennomføre en eksamen med samme oppsett som eksamen, altså 4 timer med kun kursnotat som hjelpemiddel
- Studentene har tilgang til kursnotatene på eksamen så det er lurt å bli vandt til å bruke kursnotatene
- Gjøre oppgaver og ikke minst gjøre seg vant til spørsmål som kommer på eksamen
- Bruke debugger i VSCode
- Øve på kodesporing manuelt
- Diskutere og samarbeide med andre medstudenter: diskutere ulike måter å løse oppgaver på, hvorfor noe fungerer eller ikke, forklare konsepter til hverandre
- Øve på å kode i tekst-program, uten støtte fra VSCode(auto-indent, auto-complete)
- Vær obs på at man hadde tilgang til internett på gamle eksamener, men ikke på den virkelige eksamen
- Ta notater på ting som er vanskelig
- Send inn spørsmål og ønsker til gruppeleder før gruppetime
- Slurvefeil – viktig å lese oppgavene nøye
- Folgefeil – på del 3 Kodeskriving er det ofte lett å gi opp hvis man ikke får til hele koden
- Viktig å dele opp problemet og man kan anta at noe funker for å sikre seg poeng
- Gjøre ekstra oppgavene fra labene uten hjelpemidler
- Bruk Python Tutor for forståelse
- Hvis du har brukt mye KI, pass på at du klarer å løse oppgaver helt selvstendig
- Ikkje drive med siste liten lesing dagen før
- Få frisk luft; Gå en kveldstur dagen før; Ta pauser og vær sosial; Gå ut hvis det er sol

Hva er vanlige feil på eksamen?

- Misforstå oppgaven
- Undervurderer hvor viktig kodeforsåelse er for eksamen
- Ikke lese gjennom eksamen før man begynner
- Fordeling av tid; Sitte for lenge på en (enkel) oppgave. Gå heller videre og kom tilbake
- Sitte for lenge på små slurvefeil i syntax som er logist korrekt
- Slurving på kodesporingsoppgaver; Gjør lesefeil på disse oppgavene; Gå for fort gjennom, ikke tenke seg om
- Ikke prøve om oppgaven virker for vanskelig. Tips: forklar hva ville gjort med ord om man ikke klare å kode det. Å skrive ut løsningen sin gir poeng selv om du ikke koder den.
- Overser hermetegn og anførselstegn
- Ikke bruke kursnotatene
- Ikke manuelt kodespore
- Ikke lese oppgaveteksten godt not, det peier å være flere lure oppgaver der man må lese koden godt og lage seg en oversikt over hvordan variabler endres underveis
- Glemme å bryte opp koden opp i hjelpemetoder
- Dårlige variabel og funksjonsnavn

Kan du gi noen eksamensstrategier?

- Se rask over hele eksamen for å se hvor lang hver oppgave er. Ofte tar oppgavene på slutten lengst tid
- Lage et tidsbudsjett: dele eksamenstiden på poeng og skrive hvor mye tid på oppgaver
- Lese oppgaven to ganger, løse, og så lese oppgaven på nytt
- Noter ned hvilke oppgaver man har hoppet over underveis eller er usikre på så man kan gå tilbake senere
- Skrive ned tankemåten for flervalg så det er lettere å vite hva man har tenkt når man går igjennom oppgavene på slutten av eksamen