



GRUPPETIME

UKE 6

**IN1000 GRUPPE 25 - 4.09.21**

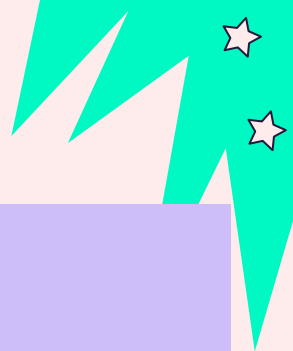


# PLAN FOR GRUPPETIMEN

- Repetisjon betyr mengdetrening!
- Codewars
- Eksamensoppgaver - fra [prøveeksamen V2021](#)



# Oversikt over moduler uke 1-5



## Uke 1

- Et første Python program
- Input fra tastaturet
- Hva er en feilmelding
- Beslutninger (if)
- Variabler

## Uke 2

- Prosedyrer
- Kontrollflyt
- Evaluering av uttrykk
  - *Sjekk antageleser med assert*
- Datatyper
- *Forstå feilmeldinger*
- *Kjør et program i debugger*

## Uke 3

- Samlinger
  - Mengder (set)
  - Ordbøker (dict)
  - Lister (list)
    - *Objekter tilbyr tjenester (metoder)*
- Nøstede samlinger

## Uke 4

- Parametere i prosedyrer
- Funksjoner (retur-verdier)
- Løkker
- Kombinere løkker og samlinger
  - For løkker

## Uke 5

- Lese fra og skrive til filer
- Parameter overføring og variabel skop

*Det er viktig at du føler at du har god oversikt over disse modulene.*

*Er det noe som du føler du ikke har teken på så gjør mengdetrenings oppgaver på Trix eller spør meg på Mattermost @sirisoll*





# CODEWARS OPPGAVER



Consecutive pairs: <https://www.codewars.com/kata/5a3e1319b6486ac96f000049/train/python>

Can we divide it?: <https://www.codewars.com/kata/5a2b703dc5e2845c0900005a/train/python>

Casino chips: <https://www.codewars.com/kata/5e0b72d2d772160011133654/train/python>

Who is the killer?: <https://www.codewars.com/kata/5f709c8fb0d88300292a7a9d/train/python>

Find the capitals: <https://www.codewars.com/kata/539ee3b6757843632d00026b>

Multippel av 3 eller 5: <https://www.codewars.com/kata/514b92a657cdc65150000006/train/python>

Find the odd int: <https://www.codewars.com/kata/54da5a58ea159efa38000836/train/python>

Kontinuerlig sum på liste: <https://www.codewars.com/kata/59b44d00bf10a439dd00006f/train/python>

Split en streng i biter på størrelse x: <https://www.codewars.com/kata/5650ab06d11d675371000003>



# PRØVEKSAMEN V21

## Oppgave 1a

tall1 = 1+2+3

tall2 = 2

tall1 =

Skriv et uttrykk i svarfeltet på høyresiden av siste tilordning som gir **tall1** verdien **11**.  
Uttrykket skal inkludere **tall1**, **tall2** og tallet **1**, men kun én gang hver.



# LØSNING: PRØVEKSAMEN V21



## Oppgave 1a

```
tall1 = 1+2+3
```

```
tall2 = 2
```

```
tall1 = tall1*tall2-1
```



Skriv et uttrykk i svarfeltet på høyresiden av siste tilordning som gir **tall1** verdien **11**.  
Uttrykket skal inkludere **tall1**, **tall2** og tallet **1**, men kun én gang hver.



# PRØVEKSAMEN V21

## Oppgave 1b

```
tall = 4*2
t1 = "a"
if tall>9:
    t1 = t1 + "a"
print(
```

Fyll ut det som mangler for at denne koden skal printe ut strengen **ba**.

Det du fyller ut skal inkludere bruk av variabelen **t1**



# LØSNING: PRØVEKSAMEN V21



## Oppgave 1b

```
tall = 4*2
t1 = "a"
if tall>9:
    t1 = t1 + "a"
print("b"+t1)
```

Fyll ut det som mangler for at denne koden skal printe ut strengen **ba**.  
Det du fyller ut skal inkludere bruk av variabelen **t1**.





# PRØVEKSAMEN V21

## Oppgave 1c

Fyll ut en logisk (boolsk) operator i svarfeltet som gjør at denne koden skriver ut strengen **ja**.

```
tall1 = 2*3
```

```
tall2 = 2**3
```

```
if (tall1>7) 
```

```
    print("ja")
```

```
else:
```

```
    print("nei")
```



# LØSNING: PRØVEKSAMEN V21



## Oppgave 1c

Fyll ut en logisk (boolsk) operator i svarfeltet som gjør at denne koden skriver ut strengen **ja**.

```
tall1 = 2*3
```



```
tall2 = 2**3
```

```
if (tall1>7) or (tall2>7) :
```

```
    print("ja")
```

```
else:
```

```
    print("nei")
```



# PRØVEKSAMEN V21

## Oppgave 1f

Noe mangler i uttrykket i linje 7 for at koden vil skrive ut tallet **2** til terminalen. Skriv det komplette uttrykket i svarfeltet.

```
tall = 0
operasjoner = "idd.d..did"
for operasjon in operasjoner:
    if operasjon == "d":
        tall = tall*2
    elif operasjon == "i":
        tall = tall           #linje 7
    else:
        tall = 0
print(tall)
```





# LØSNING: PRØVEKSAMEN V21



## Oppgave 1f

Noe mangler i uttrykket i linje 7 for at koden vil skrive ut tallet **2** til terminalen. Skriv det komplette uttrykket i svarfeltet.

```
tall = 0
operasjoner = "idd.d..did"
for operasjon in operasjoner:
    if operasjon == "d":
        tall = tall*2
    elif operasjon == "i":
        tall = tall+1
    else:
        tall = 0
print(tall)
```



# EKSAMEN H20

```
1 bok = {"a": [3, 4, 5], "b": [6, 7]}
2 |
3 print(bok["a"])
```

## Oppgave 1g

Det mangler en programsetning i linje 2 i vedlagte kode for at følgende skrives ut på terminalen: [3, 4, 5, 8]

Programsetningen skal inneholde et metodekall på bok["a"].

Det skal altså ikke gjøres en tilordning til bok["a"].

Skriv hele setningen som mangler i linje 2:

# LØSNING: PRØVEKSAMEN V21

```
1 bok = {"a": [3, 4, 5], "b": [6, 7]}
2 |
3 print(bok["a"])
```

## Oppgave 1g

Det mangler en programsetning i linje 2 i vedlagte kode for at følgende skrives ut på terminalen: [3, 4, 5, 8]

Programsetningen skal inneholde et metodekall på bok["a"].

Det skal altså ikke gjøres en tilordning til bok["a"].

Skriv hele setningen som mangler i linje 2:

```
bok["a"].append(8)
```

# PRØVEKSAMEN V21

## Oppgave 3b

Per vil lage seg en handleliste og raskt kunne fjerne utsolgte varer fra denne lista. Skriv en funksjon **fjern\_utsolgte(handleliste, utsolgte)**. Funksjonen tar som argumenter en liste med navn på varer (handleliste: liste av streng-verdier) og en liste over utsolgte varer (utsolgte: liste av streng-verdier). Funksjonen skal returnere en ny liste som inneholde de ønskede varene som ikke er utsolgte, dvs en ny liste som inneholder alle streng-verdier som finnes i listen handleliste, men ikke finnes i listen utsolgte. I tillegg skal funksjonen skrive til terminalen navnet på hver vare fra handlelisten som er utsolgt.

Altså skal for eksempel kallet `fjern_utsolgte(["melk", "brus", "pasta"], ["kanel","brus"])` evaluere til listen `["melk", "pasta"]`, samtidig som teksten `brus` skrives til terminalen.

# LØSNING: PRØVEKSAMEN V21

## Oppgave 3b

```
def fjern_utsolgte(handleliste, utsolgte):  
    nyliste = []  
    for vare in handleliste:  
        if not vare in utsolgte:  
            nyliste.append(vare)  
        else:  
            print(vare)  
    return nyliste
```

```
assert fjern_utsolgte(["melk", "brus", "pasta"], ["kanel", "brus"]) == ["melk",  
"pasta"]
```







# PRØVEKSAMEN V21

## Oppgave 3c

Lise skal reise som backpacker gjennom flere land i asia og vil finne ut hvilke vaksiner hun må ta før reisen. Skriv en funksjon **samlet\_vaksinasjon(krav\_hvert\_land)**.

Funksjonen tar som argument en liste av lister hvor hver indre liste er hvilke vaksiner som trengs for et av landene hun skal reise til. Funksjonen skal returnere en samlet liste over hvilke vaksiner som trengs, dvs en liste av streng-verdier som inkluderer alle vaksinene funnet i et av landene. Dersom samme vaksine finnes i lista for mange ulike land, skal den bare være med én gang i den returnerte lista.

Altså skal for eksempel kallet `samlet_vaksinasjon([["difteri","tyfoid"], ["hepatit","difteri"]])` evaluere til en liste `['difteri', 'tyfoid', 'hepatit']` (rekkefølgen av vaksinene i den returnerte lista er uviktig).





# LØSNING: PRØVEKSAMEN V21



## Oppgave 3c

```
def samlet_vaksinasjon(krav_hvert_land):  
    vaksiner = []  
    for krav in krav_hvert_land:  
        for vaksine in krav:  
            if not vaksine in vaksiner:  
                vaksiner.append(vaksine)  
    return vaksiner
```

```
assert samlet_vaksinasjon(["difteri","tyfoid"], ["hepatit","difteri"]) == ['difteri', 'tyfoid',  
'hepatit']
```



# PRØVEKSAMEN V21

## Oppgave 5

Karl vil kunne sikre seg at han er forskånt for å måtte lese et bestemt fy-ord og dets synoner. Skriv en funksjon sjekk\_om\_fyord(setning, fyord, synonym\_liste). Funksjonen tar som argument en setning (en streng bestående av flere ord med mellomrom mellom) , et fyord (en streng bestående av ett enkelt ord uten noen mellomrom) og en samling av ulike synonymer (en liste av lister, hvor hver indre liste består av streng-verdier som er enkeltord).

Funksjonen skal sjekke om minst ett av ordene i setningen er et synonym med det oppgitte fyordet (ifølge den oppgitte samlingen av synonymer) eller om fyordet i seg selv finnes i setningen. Dersom fyordet eller et synonym av fyordet finnes i setningen skal funksjonen returnere True. Hvis ikke skal funksjonen returnere False.

Altså skal for eksempel kallet sjekk\_om\_fyord("spis masse godsaker", "snop", [{"saft","lemonade"}, {"snacks","snop","godsaker"}, {"mye","masse"}]) returnere True, mens kallet sjekk\_om\_fyord("spis masse godsaker", "lemonade", [{"saft","lemonade"}, {"snacks","snop","godsaker"}, {"mye","masse"}]) skal returnere False.





# LØSNING: PRØVEKSAMEN V21



## Oppgave 5

```
def sjekk_om_fyord(setning, fyord, synonym_liste):
    biter = setning.split()
    for bit in biter:
        for synonymer in synonym_liste:
            if bit in synonymer and fyord in synonymer:
                return True
    return fyord in biter
```



# KONTAKT



...

[sirisoll@uio.no](mailto:sirisoll@uio.no)

@sirisoll på Mattermost



**UKENS  
MESTERVERK: HipDog**