



# GRUPPETIME

## UKE 2

**IN1000 GRUPPE 25 - 6.09.21**



# PLAN FOR GRUPPETIMEN

- Oblig 1 - hvordan gikk det?
- Jobbe sammen på MetroRetro
- Evaluering av uttrykk
- Kodeflyt
- Datatyper
- Prosedyrer



# **JOBB SAMMEN PÅ METRORETRO**

<https://metroretro.io/board/LBU8GO8O4FJH>



# PASS PÅ KODEFLYTEN

```
pizza = input("Vil du ha en pizza? ")

if pizza == "ja":
    print("Her har du en pizza!")
if pizza == "nei":
    print("Den er grei.")
else:
    print("Det forstod jeg ikke helt.")
```

elif fikser  
problemet

Utskrift:

Her har du en pizza  
Det forstod jeg ikke helt.

*Feil: Her er det to if-sjekker når det burde bare vært en if-elif-else*

# PASS PÅ KODEFLYTEN

```
if 5 < 10:  
    print("Nr. 1")  
elif 5 < 10:  
    print("Nr. 2")
```

Utskrift:  
Nr. 1

```
if 5 < 10:  
    print("Nr. 1")  
if 5 < 10:  
    print("Nr. 2")
```

Utskrift:  
Nr. 1  
Nr.2

# FINN 3 FEIL

```
tall = int(input("Skriv inn et tall: "))  
  
if tall < 5:  
    print(Tallet er større enn 5)  
else:  
    print(Tallet er ikke større enn 5)
```

Kan gi ValueError hvis man skriver inn en float

Logisk feil

SyntaxError

```
navn = "Nora"
alder = 16
if alder >= 18:
    print(navn, "er myndig.")
else:
    print(navn, "er ikke myndig")
```

```
n = "Nora"
a = 16
if a >= 18:
    print(n, "er myndig.")
else:
    print(n, "er ikke myndig.")
```

```
NAVN = "Nora"
ALDER = 16
if ALDER >= 18:
    print(NAVN, "er myndig.")
else:
    print(NAVN, "er ikke myndig.")
```

## DISKUTER:

- hva er forskjeller og likheter  
i kodeeksemplene?

# TRUE/FALSE?

1. `True and True`
2. `True and False`
  
3. `True or True`
4. `True or False`
5. `False or False`
  
6. `not(True)`
7. `not(False)`
  
- #Ekstra (utfordring):*
8. `True and not(True)`
9. `True and not(False)`
10. `not(True) or not(False)`

Et boolsk uttrykk evaluerer til enten `True` eller `False`



## AND

True	True	True
True	False	False
False	True	False
False	False	False

## OR

True	True	True
True	False	True
False	True	True
False	False	False

## NOT

True	False
False	True

→ Vi kan sette sammen boolske uttrykk med **and** og **or**

# SAMMENLIGNINGSOPERATØRER

$a == b$	a er lik b
$a != b$	a er ikke lik b
$a > b$	a er større enn b
$a >= b$	a er større eller lik b
$a < b$	a er mindre enn b
$a <= b$	a er mindre eller lik b



# LÆRDOMMER FRA OBLIG 1

## **formelle krav**

Svar på spørsmål som en kommentar i Devilry

Lever alle programmene med riktig filnavn og filtype

## **kommentarer**

Oppsummer kort hva programmet skal gjøre øverst i filen

Kommenter underveis ovenfor relevant kodeblokk

# Slik skrives en komm.



# LÆRDOMMER FRA OBLIG 1

Husk å konvertere til riktig datatype

Programmet skal kjøres slik som oppgaven beskrev - vær nøye

Unngå logiske feil - ta en pause og gjør tester

Ha med brukervennlighet

## **kodestil**

Vi følger PEP 8

Spaces > TABS:  
mellomrom ser likt ut uansett editor/program man bruker

Blank lines for å separere relevante kodeblokker

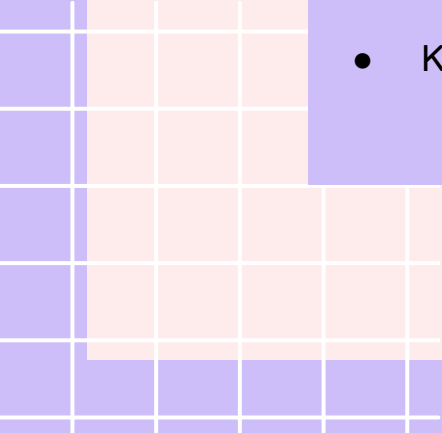


# Gå gjennom gamle obliker sammen

Hvordan løste dere oppgavene forskjellig?  
Kunne dere ha forbedret koden deres?  
Hva funket ikke - hvorfor ikke?



# Hva dere bør kunne fra uke1

- `print()`
  - `input()`
    - `input("Hva heter du?: ")` eller `input("Hva heter du:\n")`
  - if-tester
    - if-elif-else → vær obs på logisk feil
  - Kunne lese en feilmelding
    - To hovedtyper: [SyntaxError](#) og [exceptions](#)
- 



# LÆRINGSMÅL UKE 2

- Forstå hvordan én enkelt linje utføres
  - Datatyper, evaluering av uttrykk og funksjoner
- Ha god forståelse av variabler
- Forstå og kunne bruke enkle prosedyrer uten parametre
- Forstå hvordan et helt program utføres
  - kodeflyt fra linje til linje, inkludert for beslutninger og prosedyrer



# KONVERTERING MELLOM TYPER

```
str(4)           # konvertere til string > "4"  
int("5")        # konvertere string til int > 5  
int(" 6 ")      # konvertere string til int, mellomrom vil bli ignorert  
float("17.25")  # konvertere string til float > 17.25  
float("3x6")    # gir Error! Siden x ikke er et tall.
```

→ Man kan sjekke datatypen med `type()`





# PROSEDYRER uten returverdi eller parametere

- Prosedyrer uten returverdi eller parametere er nyttig dersom man skal gjøre det samme flere ganger → unngår å repetere samme kode
  - `def funksjonsNavn():`  
*<det prosedyren skal gjøre>*
- NB! Prosedyren må være definert før man kaller på den

```
def siHallo() :  
    print("Hallo!")  
  
siHallo()
```



# PROSEDYRER og veien videre

- Prosedyre - uten parametre og returverdi
- Prosedyre - med parametre
- Funksjon - med returverdi
- Instans-metode (OO)



# PROSEDYRE ØVELSE I METRORETRO

1. Skriv to enkle prosedyrer som dere kan kalle hva som helst
2. Bytt ark i MetroRetro med neste gruppe til høyre  
1 → 2      2 → 3      3 → 4      4 → 1
  - a. Skriv et hovedprogram hvor dere kan legge til hva dere vil, men dere må ha minst ett kall på hver av prosedyrene skrevet på arket
3. Bytt ark i MetroRetro med neste gruppe til høyre
  - a. Nummerer kodeflyten (i hvilken rekkefølge skjer ting når programmet kjøres) og skriv utskriften i riktig rekkefølge
4. Bytt ark i MetroRetro med neste gruppe til høyre
  - a. Rett forrige gruppe sin løsning hvis det er noe å rette

<https://metroretro.io/board/LB0J6GFYJZQ4>



**Vil du strekke deg til flere utfordringer?**

<https://www.codewars.com/>

<https://projecteuler.net/>

<https://open.kattis.com/>

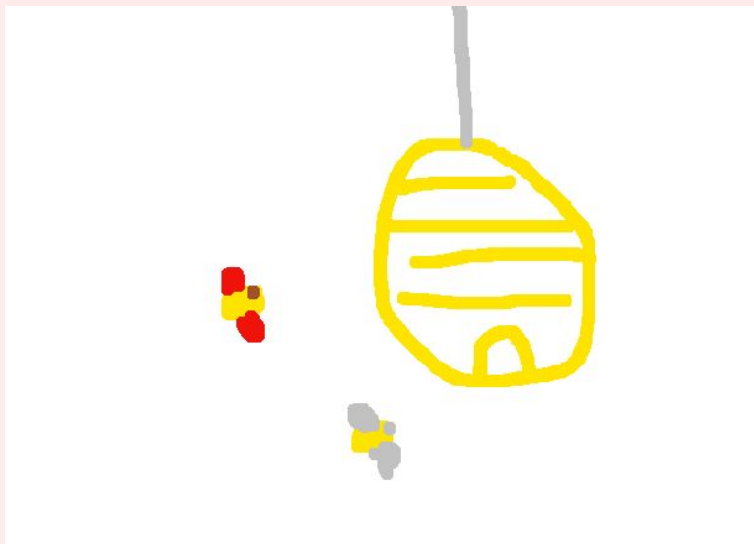


# KONTAKT



...

sirisoll@uio.no  
sirisoll på Mattermost



**UKENS**

**MESTERVERK: "hive"**

**- *Artsy bees***