



# GRUPPETIME

## UKE 9

**IN1000 GRUPPE 25 - 25.10.21**



# PLAN FOR GRUPPETIMEN

- Jobbe sammen på [MetroRetro](#)
- Repetisjon
- Magiske metoder

# LÆRINGSMÅL UKE 9

- Spesielle metoder for sammenligning og utskrift i egendefinerte klasser.
- Samlinger av objekter i beholdere (containers) som liste og ordbok.
- Strukturer med objekter av flere egendefinerte klasser.
- Repetisjon og (enda) mer detaljer om grensesnitt, innkapsling, hva skjer med referanser og objekter under kjøring



# METRORETRO

# OPPGAVE 1

person.py



Hvilke verdier får variablene til instansene av objektene av type Person her, gitt opprettelsen av person slik:

```
test_person = Person(13, "Kari")
```

```
def __init__(self, alder, navn):  
    self._alder = 0  
    self._navn = " "
```

```
_alder = 0  
_navn = " "
```

```
def __init__(self, alder, navn):  
    self._alder = navn  
    self._navn = alder
```

```
_alder = "Kari"  
_navn = 13
```

```
def __init__(self, alder, navn):  
    self._alder = alder  
    self._navn = navn
```

```
_alder = 13  
_navn = "Kari"
```

```
def __init__(self, a, b):  
    self._alder = a
```

```
_alder = 13
```

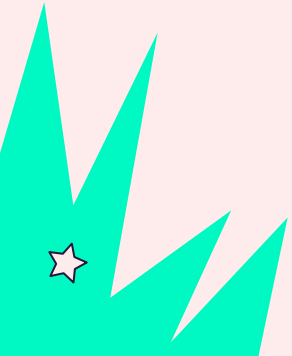
```
def __init__(self, alder, navn):  
    self._a = alder  
    self._n = navn
```

```
_a = 13  
_n = "Kari"
```



# **METRORETRO**

## **OPPGAVE 2**



# LØSNING: OPPGAVE 2

```
class Bil:
```

```
    def __init__(self, regnr, type, aarsModell):  
        self._regnr = regnr  
        self._type = type  
        self._aarsModell = aarsModell
```

```
bil1 = Bil("AB123", "Toyota", 1997)  
bil2 = Bil("EL456", "BMW", 2016)  
bil3 = Bil("DB5435", "Citroen", 2004)
```

# LØSNING: OPPGAVE 2

```
class Koffert:  
    def __init__(self, toalettsakerListe, klesListe):  
        self._toalettsaker = toalettsakerListe  
        self._klaer = klesListe
```

```
bag1 = Koffert(["tannborste", "tannkrem"], ["jakke", "genser", "bukse"])
```

```
toalettS = ["haarborste", "neglsaks", "tannborste"]  
klaer = ["sokker", "tskjorte", "shorts"]
```

```
bag2 = Koffert(toalettS, klaer)
```

```
bag3 = Koffert(toalettS, ["badetoy", "skjorte", "jeans"])
```



# LØSNING: OPPGAVE 2

```
class Hund:
    def __init__(self, a, b, c):
        self._alder = a
        self._rase = b
        self._bjefferMye = c
        self._metthet = 10
#dersom metthet skal alltid settes til 10 ved start trenger
ikke metthet aa vaere med som parameter i
konstruktoeren.
```

```
labrador = Hund(10, "Labrador", False)
shiba = Hund(4, "Shiba Inu", True)
dachs = Hund(7, "Dachs", True)
```

#boolske verdier kan ogsaa brukes og  
vaere nyttige variabler i objekter! Her  
burde True/False brukes i stede for feks.  
"Ja" eller "Nei" som verdi for variabelen  
.\_bjefferMye

# FINN FEILEN

```
def listeHarNegativtTall (liste):  
    for tall in liste:  
        if tall < 0:  
            return True  
    else:  
        return False
```

```
liste = [-10, 4, 3, -9, 100]  
print(f"listeHarNegativtTall på listen: {liste} returnerer {listeHarNegativtTall (liste)}")
```

```
liste = [10, 4, 3, -9, 100]  
print(f"listeHarNegativtTall på listen: {liste} returnerer {listeHarNegativtTall (liste)}")
```

```
liste = [10, 4, 3, 9, 100]  
print(f"listeHarNegativtTall på listen: {liste} returnerer {listeHarNegativtTall (liste)}")
```



# MAGISKE METODER: `__eq__` & `__str__`

## `__eq__`

- Kalles når vi bruker referansevar == referansevar
- Hvordan kan vi vite at to objekter er like?
  - Hvis de har samme verdi for instansvariabelen navn?  
Men hva om navn ikke er en passende instansvariabel?
    - Vi kan definere hva som skal til for at to objekter regner som like med `__eq__`

## `__str__`

- Kalles når vi bruker
  - `print(referansevar)`
  - `str(referansevar)`
- Lager brukervennlig utskrift, lag den slik du ønsker
  - Det er opp til programmereren å lage gode print-setninger

hund.py

```
class Hund:
    def __init__(self, kjonn, navn, alder):
        self._navn = navn
        self._alder = alder

    def __str__(self):
        return self._navn

    def hentStreng(self):
        return self._navn

nyHund = Hund("Ole", 9)
print(nyHund.hentStreng())
print(nyhund)

if nyHund.hentStreng() == "Ola":
    print("Ola Normann?")

if str(nyHund) == "Ola":
    print("Ola Normann?")
```

# KONTAKT



...

sirisoll@uio.no  
@sirisoll på Matteredmost