



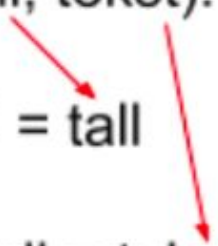
# IN1000 – SEMINAR 9

Klasser og objekter 3



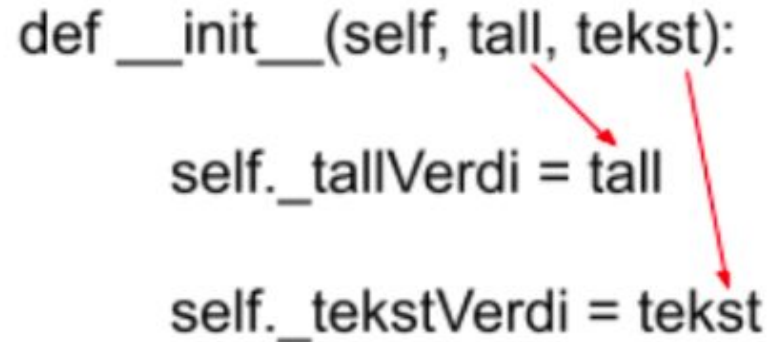
# Rep: `__init__`

```
def __init__(self, tall, tekst):  
    self._tallVerdi = tall  
    self._tekstVerdi = tekst
```



# Rep: `__init__`

```
def __init__(self, tall, tekst):  
    self._tallVerdi = tall  
    self._tekstVerdi = tekst
```

A diagram illustrating the parameter passing in the `__init__` method. Two red arrows point from the parameters `tall` and `tekst` in the function signature to the corresponding values `tall` and `tekst` in the assignment statements `self._tallVerdi = tall` and `self._tekstVerdi = tekst`.

```
def __init__(self, tall, tekst):  
    self._tallVerdi = tall  
    self._tekstVerdi = tekst  
    self._startVerdi = 0
```

- Hvis man har en variabel som alltid skal starte på samme verdi kan man sette den direkte. Trenger da ikke å ta den inn som et parameter

# Oppgave 1

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

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

1)

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

2)

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

3)

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

4)

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

5)

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

# Retur og print

Når man printer og ikke returnerer får man ikke brukt verdien senere.

print og return har ingenting med hverandre å gjøre

PS: Ofte hvis oppgaven lyder noe lignende lag en metode som henter x eller lag en metode som returnere x så betyr det at du skal returnere x og ikke printe x

# Løkker og return

Hva skjer i følgende kode:

```
def listeHarNegativtTall(liste):  
    for tall in liste:  
        if tall < 0:  
            return True  
        else:  
            return False  
  
liste = [-10, 4, 3, -9, 100]  
print("listeHarNegativtTall på listen:" ,liste, "returnerer", listeHarNegativtTall(liste))  
  
liste = [10, 4, 3, -9, 100]  
print("listeHarNegativtTall på listen:" ,liste, "returnerer", listeHarNegativtTall(liste))  
  
liste = [10, 4, 3, 9, 100]  
print("listeHarNegativtTall på listen:" ,liste, "returnerer", listeHarNegativtTall(liste))
```

# Magiske metoder

- `__str__` : for brukeren, gir en bedre forklaring av objekter.
  - *kalles på når du bruker `str(<objekt>)`*
  - *`print(<objekt>)`*
- `__eq__` : definerer hva som gjør et objekt av likt et annet
  - *`<objekt1> == <objekt2>`*

Finnes mange fler magiske metoder i python men disse to er de viktigste i 1000

# Diskusjon

Hva er forskjellen på `__str__` og `hentStreng` her ?

```
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
```



# Oppgave 3

Vi skal nå legge til en magiske metode i klassen blomst. Vi skal erstatte skrivUtInfo metoden vår med en `__str__` slik at når vi printer et blomst objekt vil infoen om blomsten printes ut, på akkurat samme måte som skrivUtInfo gjorde

Vi skal lage en ny klasse **Blomsterkasse**, en blomsterkasse tar inn en instansvariabler antall blomster det er plass til i bredden (kalt bredde) , disse to skal bli tatt inn i konstruktøren. Videre skal konstrukten lage en liste som representerer alle plassene i blomsterkassen. Konstruktøren skal lage listen og fylle den opp med None (None betyr at det ikke er noe der)

Lag en metode plantBlomst (med en blomst som parameter), som planter blomsten på den første ledige plassen, hvis det er en ledig plass skal blomsten plasseres og det skal returneres True, hvis det ikke er flere ledige plasser skal det returneres False

Lag en metode vannAlleBlomster, som vanner alle blomstene i blomster i blomsterkassen (her kan du bruke vann metoden til blomst klassen)

Lag metoden nesteDag som øker dagen med en hos alle blomsten

Lag metoden ryddIBedd som fjerner alle blomstene som har dårlig status (når en blomst fjenes byttes den ut med None)

Lag en metode skrivUtInfoOmBlomster som skriver ut infoen om alle blomstene i blomster kassen

Lag en metode antallBlomster som returnerer antall blomster som er i blomsterkassen

Lag en `__str__` metode til blomsterkassen som returnere en streng som sier noe om bredden, antall blomster og info om hver blomst

# Oppgave 4

Utvid Blomsterkasse klassen med en metode lesInnBlomsterFraFil.

Metoden skal ta inn et parameter filnavn.

Hver linje i file inneholder informasjonen om en blomst på formen <hoyde>:<art>

2.11:Amaranthus

2.7:Aster

2.58:Holly

Metoden skal returnerer True hvis alle blomstene i filen fikk plass ellers False.

Videre skal metoden lese inn linje for linje å opprette en blomst og plassere dem inn på en ledig plass.