

Gruppetime uke 9

In1000 gruppe 7, høst 19

Agenda for dagen

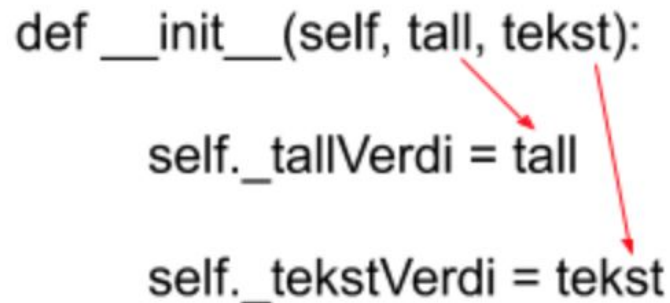
- Litt repetisjon
- ★ Magiske metoder ★
- Mer komplekse datastrukturer
- Tegning av datastrukturer

Repetisjon

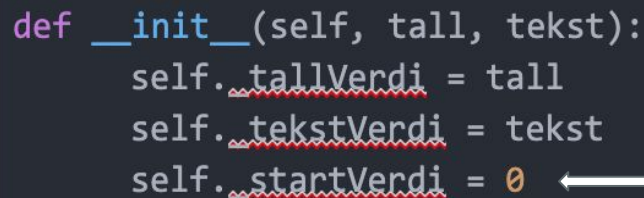
Konstruktør

- Parametre sendes til konstruktør, setter instansvariabler
- Dersom en verdi skal gjelde for alle instanser av klassen, trenger den ikke sendes som parameter
- Variabelnavnet har ikke noe med navnet på parametret å gjøre

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



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



Repetisjon

Return vs print

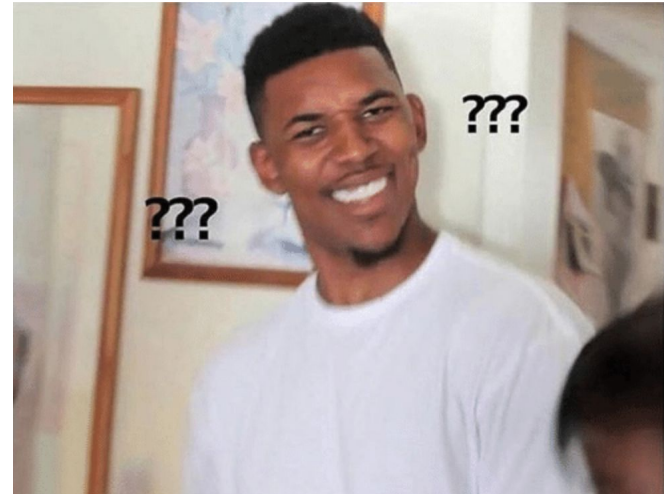
- Det er ikke alltid man har lyst til å printe ut en verdi
- Ved print får man ikke tak i verdien utenfor metoden
- Dersom en oppgave etterspør metoder som skal **hente ut** eller **returnere** en verdi, skal alltid **return** brukes
- Eksempel med bruk av returverdi i if-tester
 - Er en person myndig?
 - `Person.py / testPerson.py`

Magiske metoder

- `__str__` : returnerer en string-representasjon av objektet
 - `print(person)` uten `__str__`-metode: `<person.Person object at 0x10fc730f0>` → viser egentlig bare hva slags type objektet er, samt hvilken minneadresse det er lagret på
 - Eksempel: implementering av `__str__` i klassen `Person`
- `__eq__` : brukes for å sammenligne to objekter på egenskaper vi selv velger
 - Eksempel: implementering av `__eq__` i klassen `Person`

'is' vs '=='

- Om vi implementerer `__eq__` metoden sammenligner en gitt egenskap
 - `pers == pers2`
- Om vi skal sjekke om to referanser refererer til samme objekt: `is`
 - Eksempel: `pers is pers2`
 - Tegning på tavlen med referanser



Mer komplekse datastrukturer

- Matriser med objekter - nøstede lister
 - Magiske metoder
 - Tilfeldig generering av egenskaper
 - Skrive ut matrisen
 - Finne naboelementer til hvert objekt i matrisen
- hund.py → testHund.py
- kennel.py → testKennel.py

<code>[i - 1][j - 1]</code>	<code>[i - 1][j]</code>	<code>[i - 1][j + 1]</code>
<code>[i][j - 1]</code>	<code>[i][j]</code>	<code>[i][j + 1]</code>
<code>[i + 1][j - 1]</code>	<code>[i + 1][j]</code>	<code>[i + 1][j + 1]</code>

Tegning av datastruktur

Hvordan ser datastrukturen ut når vi kjører dette programmet?

```
def main():  
    kennelen = Kennel(2,2)  
  
main()
```


Hvis tid - oppgaver

Oppgaver ligger ute på emnesiden/grupper/gruppe7/uke9 :)