

Velkommen :)

Seminartime - IN1000
srhellst på Matteredmost
Epost: srhellst@ifi.uio.no



UiO : Universitetet i Oslo

Oppgave 1

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

Oppgave 2

Gitt følgende program, opprett 3 instanser av hver med fornuftige verdier.

```
class Bil:
```

```
    def __init__(self, regnr, type, aarsModell):
```

```
        self._regnr = regnr
```

```
        self._type = type
```

```
        self._aarsModell = aarsModell
```

```
class Hund:
```

```
    def __init__(self, a, b, c):
```

```
        self._alder = a
```

```
        self._rase = b
```

```
        self._bjefferMye = c
```

```
        self._metthet = 10
```

```
class Koffert:
```

```
    def __init__(self, toalettsakerListe, klesListe):
```

```
        self._toalettsaker = toalettsakerListe
```

```
        self._klaer = klesListe
```

Oppgave 3

Anta at du har en *dictionary* (ordbok) som består av studentobjekter, den er på dette formatet:

```
studenter = {navn : student, navn: student, ... }
```

Altså at nøkkelen er studentens navn, og at verdien er et studentobjekt.

Du kan anta at *student* har en metode som heter *hentAlder()*.

Bruk en for-løkke til å skrive ut navn og alder på alle studentene i ordboken:

Oppgave 4

Hva printes på de to siste linjene?

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

    def hentStreng(self):
        return self._navn

nyHund = Hund("Ole", 19)
print(nyHund.hentStreng())
print(nyhund)
```

Oppgave 5

Skriv av koden, og legg til følgende:

1. En `__str__()` metode som returnerer 'navn' til et objekt av klassen `Hund`
2. To If-tester:
 - a. Dersom `nyHund.hentStreng()` er lik 'Ola', så skal programmet printe 'Ola Nordmann?'
 - b. Dersom `str(nyHund)` er lik 'Ola', så skal programmet printe 'Ola Nordmann?'

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

    def hentStreng(self):
        return self._navn

nyHund = Hund("Ole", 19)
print(nyHund.hentStreng())
print(nyhund)
```

Oppgave 6

Hva printes på de to siste linjene?

```
class Person:
    def __init__(self, first_name, last_name, age):
        self.first_name = first_name
        self.last_name = last_name
        self.age = age

john = Person('John', 'Doe', 25)
jane = Person('Jane', 'Doe', 25)

print(john is jane)
print(john == jane)
```

Oppgave 7

- Skriv en `__eq__()` metode som returnerer *True* dersom 'age' er likt på objektene som sammenlignes
- Hva printes på de to siste linjene etter man har lagt til `__eq__()` metoden?

```
class Person:
    def __init__(self, first_name, last_name, age):
        self.first_name = first_name
        self.last_name = last_name
        self.age = age

john = Person('John', 'Doe', 25)
jane = Person('Jane', 'Doe', 25)

print(john is jane)
print(john == jane)
```