



# Enkle objekter

## Introduksjon

I denne oppgaven skal vi gi en enkel innføring til klasser og objekter (

## Ordbøker

Tenk deg at du skal lage en sirkel, og ønsker å lagre variabler som kan dette gjøres slik:

```
circle = {  
    "radius" : 3,  
    "color" : "red"  
}
```

Vi kan også hente ut og endre variablene:

```
# endrer radiusen til 5  
circle["radius"] = 5  
  
# skriver ut fargen på sirkelen  
print(circle["color"])
```

Hva om vi ønsker å lage en variabel som regner ut arealet til sirkel

`circle_area()`:

```
import math
```

```
def circle_area(circle):  
    radius = circle["radius"]  
    # formula:  $A = \pi * r * r$   
    area = math.pi * radius * radius  
    return area
```

Så kaller vi funksjonen:

```
print(circle_area(circle))
```

# Klasser og objekter

Vi ønsker å bruke **objekter** i stedet for ordbøker i de neste spillene `radius` og `color`. En **klasse** er en slags 'mal' for et objekt. Klasse kan lage en `Circle`-klasse, også lage et objekt av typen `Circle` som

Dette er enklere å forstå med ett eksempel:

```
# Vi lager Circle-klassen:  
class Circle:  
    radius = 3  
    color = "red"  
  
# Så lager vi circle-objektet  
circle = Circle()
```

Vi lager først `Circle`-klassen, før vi så lager et `Circle`-objekt som datamaskinen at nå lager vi en klasse, du kan sammenlikne det med kommer en funksjon.

Vi kan hente ut og endre variablene til objektet:

```
# Endrer på radiusen
```

```
circle.radius = 5

# Skriver ut fargen
print(circle.color)
```

Nå ønsker vi å lage en funksjon som kan regne ut arealet til sirkele klassen:

```
import math

class Circle:
    radius = 3
    color = "red"

    def area(self):
        area = math.pi * self.radius * self.radius
        return area
```

**Legg merke til innrykket av `area()` i eksempelet over!**

Så kaller vi funksjonen:

```
print(circle.area())
```

Du lurer kanskje på hvorfor vi brukte `self.radius` i funksjonen `area` variabelen som er en del av klassen. Du må alltid bruke `self` når du er inne i klassen.

Som vi ser så er det ikke så stor forskjell mellom bruk av funksjoner

**\*\*Ordbøker\*\***

```
import math

circle = {
    "radius": 3,
    "color": "red"
}

def circle_area(circle):
```

**\*\*Klasse\*\***

```
import math

class Circle:
```

```
area = math.pi * circle["radius"] ** 2
return area
```

```
circle["radius"] = 5
print(circle["color"])
print(circle_area(circle))
```

Vi kommer dermed til å bruke klasser i de neste oppgavene - det er mer avanserte ting med klasser.

Nå kommer det et program som er skrevet ved bruk av ordbøker. Du skal bruke klasser.

```
rectangle = {
    "width": 3,
    "length": 5,
    "color": "blue"
}

def rectangle_area(rectangle):
    area = rectangle["width"] * rectangle["length"]
    return area

rectangle["width"] = 10
print(rectangle["color"])
print(rectangle_area(rectangle))
```

---

# Test programmet ditt

Programmet over skriver ut det følgende:

**blue**

50

Pass på at din "oversettelse" gjør det samme.

**Lisens:** [CC BY-SA 4.0](#) **Forfatter:** Ole Kristian Pedersen, Kodeklu