IN1000

Uke 9

Plan for dagen

- Liten eksamensoppgave
- Magiske metoder
- Forskjell på return og print

Oppvarming: Hva printes her?

```
a = 5
b = a == 3
print(b)
```

Oppvarming: Hva printes her?

```
a = 5
b = a == 3
print(b)
```

- Når dere ser disse: ==, !=, <, >, tenk True eller False!

Oppvarming: Hva printes her?

```
a = 5
b = (a == 3)
print(b)
```

- Når dere ser disse: ==, !=, <, >, tenk True eller False!

Oppvarming: Hva printes her?

```
a = 5
b = (a == 3)
print(b) # printer False!
```

Oppgave 1b fra eksamen 2022

Hva skrives ut når koden under kjøres?

```
a = 5
b = a > 2
if a == 3 or b:
    a += 10
print(a)
```

Oppgave 1b fra eksamen 2022

Hva skrives ut når koden under kjøres?

```
a = 5
b = a > 2
if False or True:
    a += 10
print(a)
```

Magiske metoder

- **__str__** bestemmer hva som skjer når vi skriver print(objekt)
- **__eq__** bestemmer hva som skjer når vi sammenligner to objekter med ==
- **__init__** bestemmer hva som skjer når vi oppretter et objekt



Hva skjer her?

```
class Sykkel:
    def __init__(self, merke, size):
        self._merke = merke
        self._size = size

sykkel = Sykkel("DNB", "L")
print(sykkel)
```

Hva skjer her?

```
class Sykkel:
    def __init__(self, merke, size):
        self._merke = merke
        self._size = size

sykkel = Sykkel("DNB", "L")
print(sykkel)
```

Printer minneadressen! `<__main__.Sykkel object at 0x10679bc10>`

__str__ bestemmer hva som printes når vi skriver print(objekt)!

```
class Sykkel:
    def __init__(self, merke, size):
        self._merke = merke
        self._size = size

    def __str__(self):
        return f"{self._merke} i størrelse {self._size}"

sykkel = Sykkel("DNB", "L")
print(sykkel)
```

Hva skjer her?

```
class Firkant:
    def __init__(self, lengde, bredde):
        self._lengde = lengde
        self._bredde = bredde

firkant1 = Firkant(10, 20)
firkant2 = Firkant(10, 20)

print(firkant1 == firkant2)
```

Dersom vi ikke har definert en __eq__-metode fungerer ‘==‘ på samme måte som ‘is’

- `firkant1 == firkant2`
- `firkant1 is firkant2`
- Begge disse sjekker om referansene peker til samme objekt i minnet

__eq__ bestemmer hva som printes når vi skriver objekt1 == objekt2!

```
class Firkant:
    def __init__(self, lengde, bredde):
        self._lengde = lengde
        self._bredde = bredde

    def hent_lengde(self):
        return self._lengde

    def hent_bredde(self):
        return self._bredde

    def __eq__(self, annen):
        return (self._lengde == annen.hent_lengde() and self._bredde == annen.hent_bredde())

firkant1 = Firkant(10, 20)
firkant2 = Firkant(10, 20)

print(firkant1 == firkant2)
```

__eq__ bestemmer hva som printes når vi skriver objekt1 == objekt2!

```
class Firkant:
    def __init__(self, lengde, bredde):
        self._lengde = lengde
        self._bredde = bredde

    def hent_lengde(self):
        return self._lengde

    def hent_bredde(self):
        return self._bredde

    def __eq__(self, annen):
        return (self._lengde == annen.hent_lengde() and self._bredde == annen.hent_bredde())

firkant1 = Firkant(23, 20)
firkant2 = Firkant(10, 20)

print(firkant1 == firkant2)
```

Hvilke verdier får instansvariablene her, hvis objektet opprettes slik:
test_person = Person(55, «Kari»)

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

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

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

Hvilke verdier får instansvariablene her, hvis objektet opprettes slik:
test_person = Person(55, «Kari»)

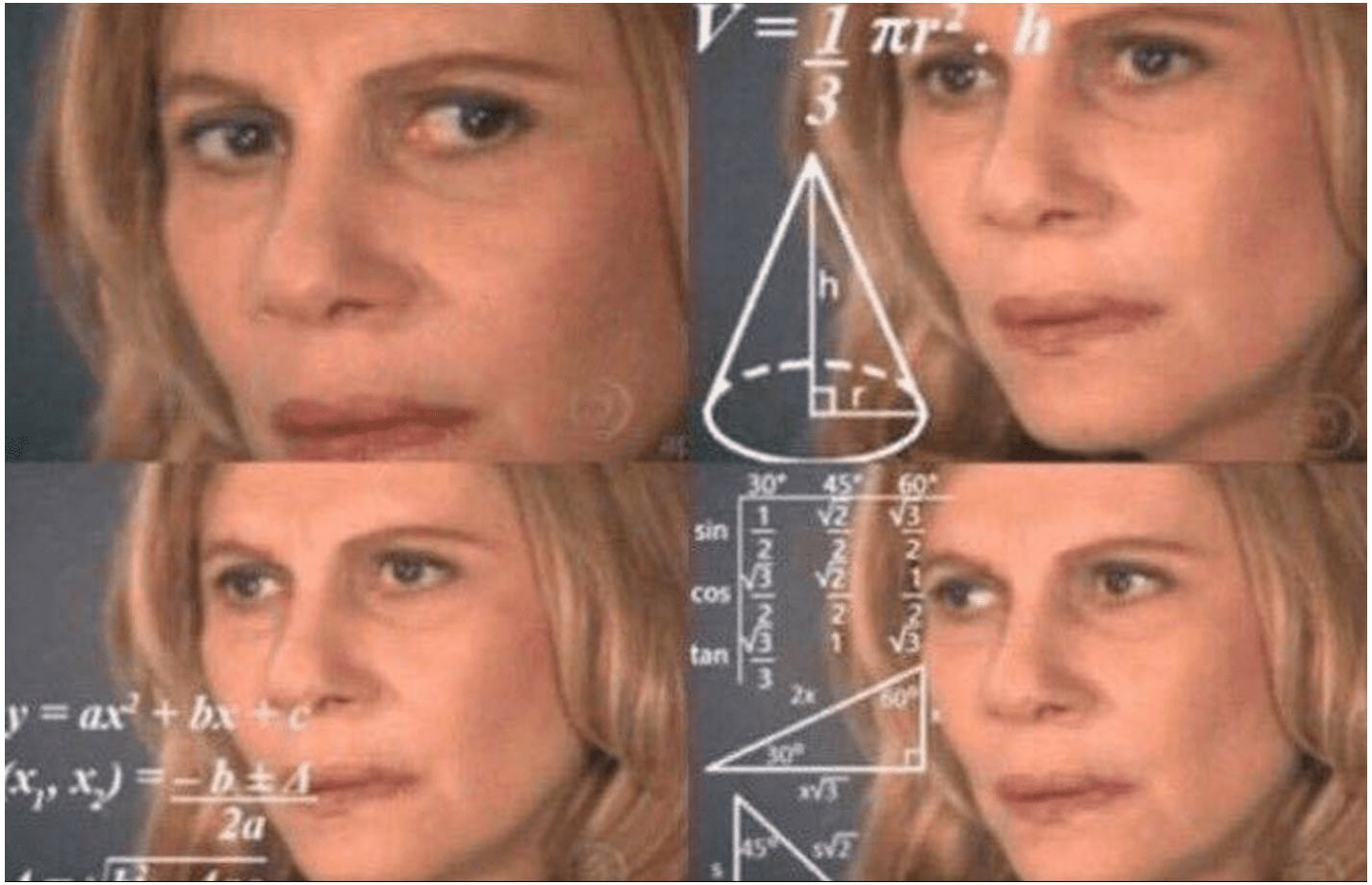
```
class Person:
    def __init__(self, a, b):
        self._alder = a
        self._navn = b
```

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

Man kaller aldri direkte på magiske metoder!

- Man skriver aldri f.eks. person.__str__(), den kalles automatisk når man skriver print(person)

Hva er forskjellen på return og print?



```
total = 0
s = "kokos og solkrem"
for i in range(1,len(s)):
    if s[i]=="o" and s[i-1] in ["k", "s"]:
        total += 1
print(total)
```