

KRASJ OG FEIL MINNET

INF100

HØST 2025

Torstein Strømme

RECAP

```
1  def incremented_thrice(x):  
2      x = x + 1  
3      x += 1  
4      return x + 1  
5  
6  a = 5  
7  b = incremented_thrice(a)  
8  c = incremented_thrice(b)  
9  
10 print(f"a: {a}, b: {b}, c: {c}")
```

HVA MED DETTE?

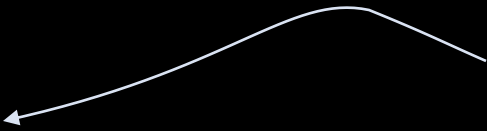
ikke
samme
variabel!

```
1  def incremented_thrice(x):
2      x = x + 1
3      x += 1
4      return x + 1
5
6  x = 5
7  y = incremented_thrice(x)
8  y = incremented_thrice(y)
9
10 print(f"x: {x}, y: {y}")
```

HVA MED DETTE?

```
1  def incremented_thrice(x):  
2      x = x + 1  
3      x += 1  
4      print(x + 1)  
5  
6  x = 5  
7  y = incremented_thrice(x)  
8  y = incremented_thrice(y)  
9  
10 print(f"x: {x}, y: {y}")
```

uten retursetning vil None returneres



KRASJ OG FEIL

TRE FORMER FOR FEIL

- Syntaks
 - Programmet krasjer før det begynner å kjøre

```
def volume_of_box(x, y, z)  
    print(x * y + z)
```

```
print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen
```

line 4

```
    print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen
```

SyntaxError: unterminated string literal (detected at line 4)

TRE FORMER FOR FEIL

- Syntaks
 - Programmet krasjer før det begynner å kjøre

```
def volume_of_box(x, y, z)  
    print(x * y + z)
```

```
print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 1  
    def volume_of_box(x, y, z)  
                                ^
```

```
SyntaxError: expected ':'
```

TRE FORMER FOR FEIL

- Syntaks
 - Programmet krasjer før det begynner å kjøre

```
def volume_of_box(x, y, z):  
    print(x * y + z)
```

```
print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 4  
    print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen"  
      ^  
SyntaxError: '(' was never closed
```

TRE FORMER FOR FEIL

- Krasj (engelsk: runtime error)
 - Programmet krasjer når det kjører

```
def volume_of_box(x, y, z):  
    print(x * y + z)  
  
print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 4, in <module>  
    print("Det er plass til " + volum_of_box(1, 2, 3) + " m3 i boksen")  
NameError: name 'volum_of_box' is not defined. Did you mean:  
'volume_of_box'?
```

TRE FORMER FOR FEIL

- Krasj (engelsk: runtime error)
 - Programmet krasjer når det kjører

```
def volume_of_box(x, y, z):  
    print(x * y + z)
```

```
print("Det er plass til " + volume_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 4, in <module>  
    print("Det er plass til " + volume_of_box(1, 2, 3) + " m3 i boksen")  
TypeError: can only concatenate str (not "NoneType") to str
```

TRE FORMER FOR FEIL

- Krasj (engelsk: runtime error)
 - Programmet krasjer når det kjører

```
def volume_of_box(x, y, z):  
    return x * y + z
```

```
print("Det er plass til " + volume_of_box(1, 2, 3) + " m3 i boksen")
```

```
line 4, in <module>  
    print("Det er plass til " + volume_of_box(1, 2, 3) + " m3 i boksen")  
TypeError: can only concatenate str (not "int") to str
```

TRE FORMER FOR FEIL

- Logisk feil
 - Programmet gir feil svar

```
def volume_of_box(x, y, z):  
    return x * y + z
```

```
print("Det er plass til " + str(volume_of_box(1, 2, 3)) + " m3 i boksen")
```

```
Det er plass til 5 m3 i boksen
```

TRE FORMER FOR FEIL

- Syntaks
 - Programmet krasjer før det begynner å kjøre
 - Feilmelding gir visuell indikasjon på hva som er feil
- Krasj
 - Programmet krasjer underveis i kjøring
- Logiske feil
 - Programmet gir galt svar

IndentationError
SyntaxError

AttributeError
IndexError
KeyError
NameError
TypeError
ZeroDivisionError
...

ASSERT

- Krasj programmet med vilje når noe ikke er som det skal
- Tester koden, og beskytter mot logiske feil

```
assert True # Gjør ingenting  
assert False # Krasjer
```

- Vi bruker prinsippet om assert når vi retter kode automatisk (CodeGrade)
- Det er mulig å slå av assert for å optimisere kjøretid (men: ikke gjør det)
- Sjekk at assert er aktivt: legg inn `assert False` og se at det krasjer

ASSERT

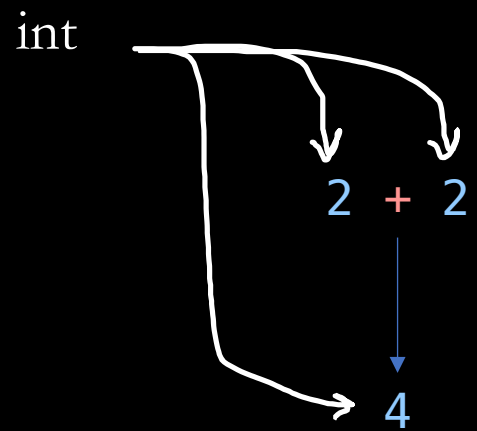
- Krasj programmet med vilje når noe ikke er som det skal
- Tester koden, og beskytter mot logiske feil

```
def volume_of_box(x, y, z):  
    return (x * y + z)
```

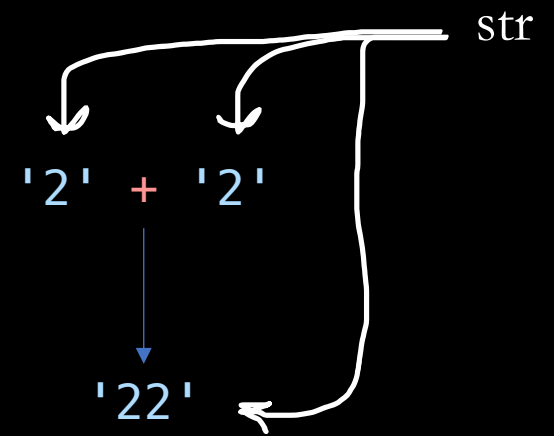
```
assert 6 == volume_of_box(1, 2, 3)  
print("Det er plass til " + str(volume_of_box(1, 2, 3)) + " m3 i boksen")
```

```
line 4, in <module>  
    assert 6 == volume_of_box(1, 2, 3)  
AssertionError
```

TYPED



addisjon

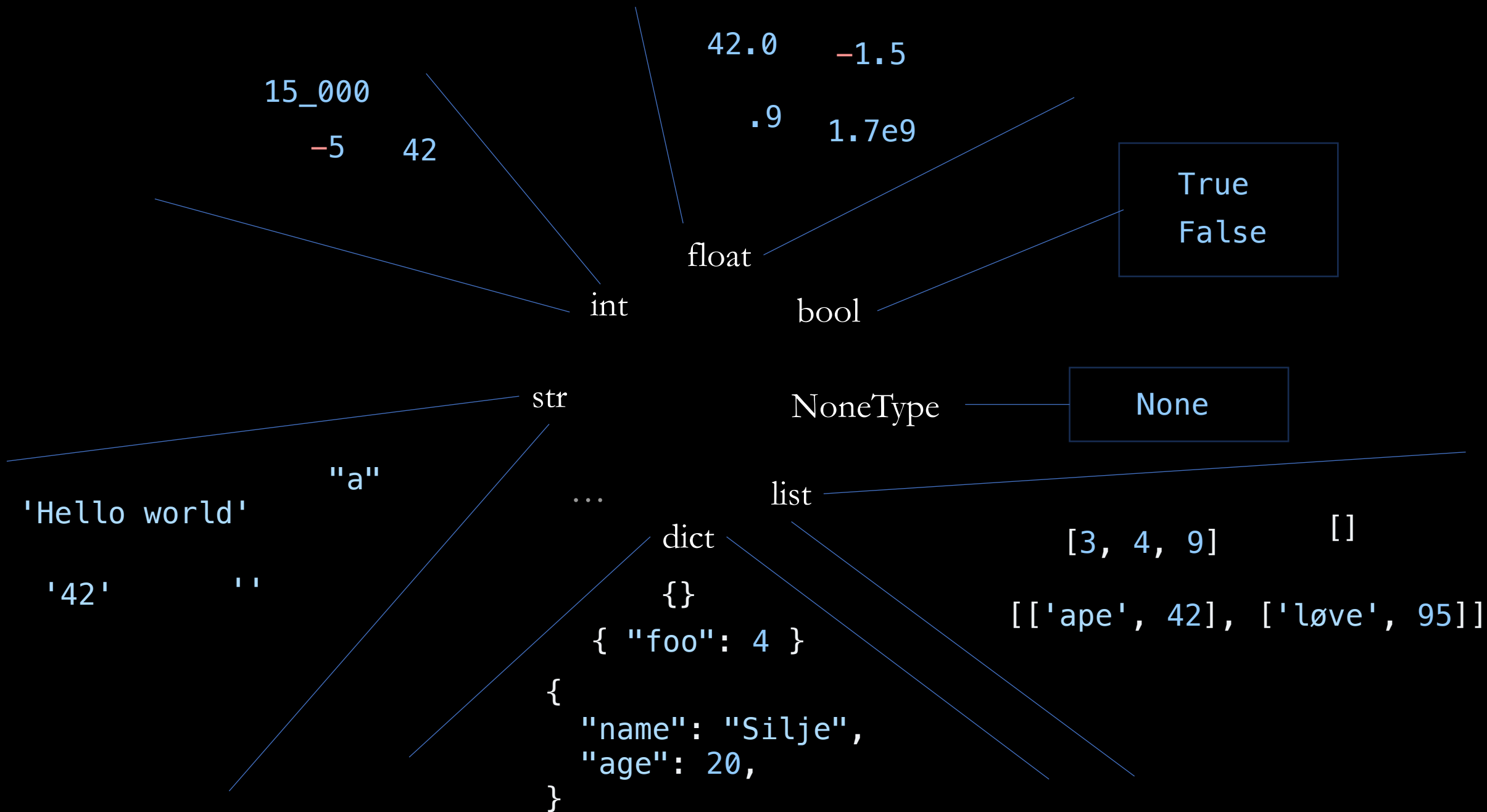


kontatinasjon



sammenligner tallverdi

sammenligner «alfabetisk»



MINNET



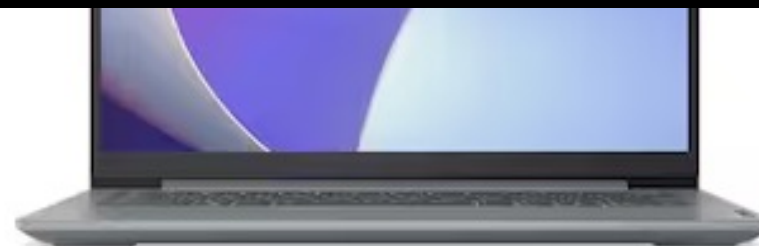
HP Laptop 14-em0924no 14" FHD

Ryzen 5 7520U, 8 GB RAM, 512 GB SSD, Windows 11 Home

★★★★☆ (3 anmeldelser)

7 000,-

SPAR 3 000,-



Lenovo Ideapad Slim 3 14" FHD (arctic grey)

Intel Graphics, Core i5-12450H, 16 GB RAM, 1 TB SSD, Windows 11 Home

8 000,-

Varenummer: 1222291 / Produktnr.: BUI-5-7600
Komplett Build 5 - 7600 Komplett Build

243 personer har sett på varen nylig



Komplett Build 5 - 7600

1 / 1

AMD Ryzen 5 7600, B650, DDR5 16GB, RTX 7600, 1TB NVME, 650W

★★★★★ (2)

12 995,-

Ikke på lager.

[Få varsel når varen er tilbake på lager](#)

LEGG I HANDLEKURV

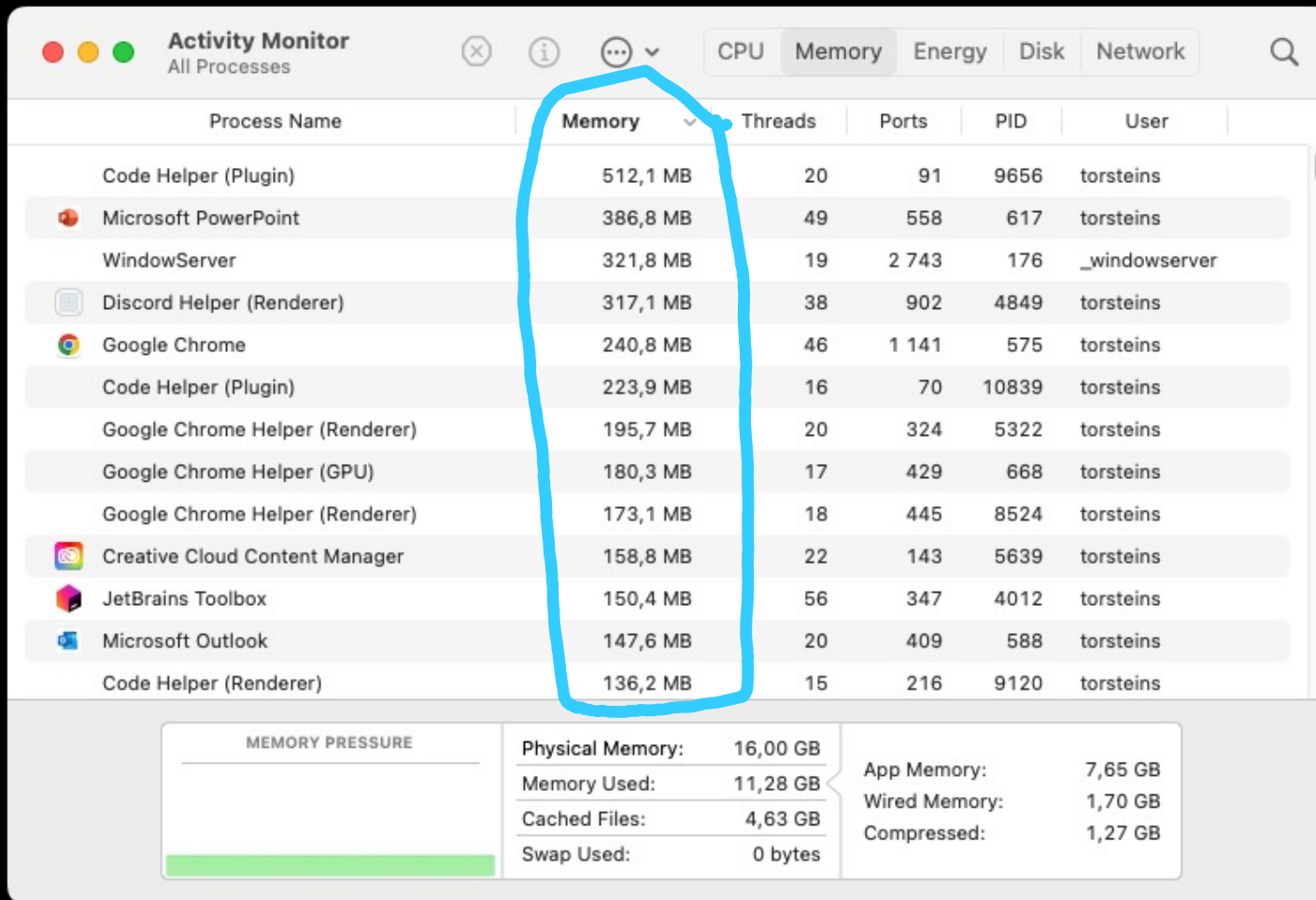
eller

TILPASS FØR KJØP

Sammenlign



Random Access Memory (RAM)



MINNET

- En lagringsplass for 0'ere og 1'ere

én **bit** med informasjon



8 bit = 1 Byte

4 GB RAM tilsvarer 32 000 000 000 bits

MINNET

- En lagringsplass for 0'er og 1'ere

én **bit** med informasjon



0

1

2

3

4

...

hver bit har en **adresse**

MINNET

- En del av minnet er forbehold **objekter**
 - Et **objekt** er et område i minnet som «hører sammen»
 - Objekter har en minneadresse (id), en type (klasse) og en verdi



MINNET

- En variabel er en navngitt referanse til et objekt
 - «referanse til et objekt» = egentlig bare en minneadresse*

X



*en hvit logn du kan leve fint med helt frem til du skal konstruere dine egne programmeringsspråk eller operativsystemer

MINNET

et objekt

...01110110101110101000010100000101011110010101111100011101010101010101000...



X

en variabel

OBJEKT

10111011010111010100001010000010101110010101111100



ID-nummer for klassen
(typen) + annen metadata

verdien

42.0

001001101101000111010000100010100000000000000000

«dette objektet er en float»

132
i totallsystemet

.3125
i totallsystemet

$$2^{132-127} \cdot (1 + 0.3125) \Rightarrow 42.0$$

0.1

00100110110100011100111101110011001100110011001101

«dette objektet er en float»

123
i totallsystemet

0.6000000238418579
i totallsystemet

$$2^{123-127} \cdot (1 + 0.6000000238418579)$$

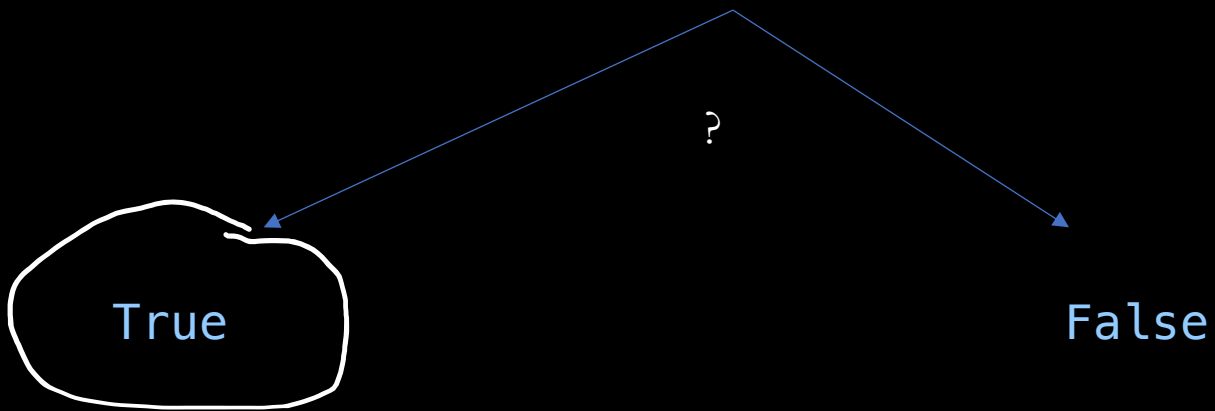
$\Rightarrow$ 0.100000001490116119384765625

1 + 2 == 3

?

True

False

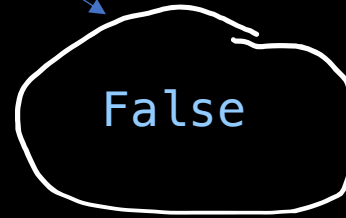


0.1 + 0.2 == 0.3

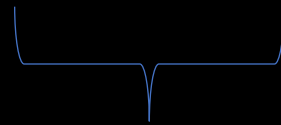
?

True

False



$$0.1 + 0.2 == 0.3$$



0.30000000000000004

FUNKSJONER

Oppgave: skriv en funksjon som sjekker om to tall er nesten like

- Hvordan er input representert?
- Hva er det matematiske grunnlaget?
- Skal funksjonen ha en effekt eller en returverdi?
 - Effekt: f. eks. noe som vises på skjermen
 - Returverdi: hvis vi vil bruke verdien til noe mer enn å se på den
- Hvordan tester vi om funksjonen fungerer?

bestemmer parametrene til funksjonen

ellers får vi logiske feil

print vs return

assert