

FUNKSJON KRASJ OG FEIL MINNET

INF100

HØST 2025

Crystal Chang Din

Med bidrag fra: Torstein Strømme

Å DEFINERE EN FUNKSJON

Mellom parentesene er *parametre*.
Det kan være valgfritt antall.

Vi *definerer* en funksjon som heter `incremented_thrice`

```
def incremented_thrice(x):  
    x = x + 1  
    x += 1  
    return x + 1
```

Kolon

x er en *parameter* (en variabel)
som refererer til en verdi bestemt
av den som kaller funksjonen

Kropp. Kode etter kolon
som har *innrykk* og blir
utført når funksjonen
kalles

Retursetning. Avslutter funksjonen.
Returverdien bestemmes her.

hvis ingen retursetning,
vil None returneres

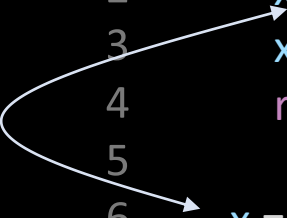
LIVE MED DEBUGGER

```
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}")
```



KODESPORING

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}")

```

[illegible]

KODESPORING

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}")

```

[illegible]

KODESPORING

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}")

```

[illegible]

KODESPORING

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}")

```

[illegible]

KODESPORING

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}")

```

[illegible]

KODESPORING

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}")

```

[illegible]

KODESPORING

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}")
```

linje	x	x_it	return	y	x_it ₂		
6	5						
7, 1		5					
7, 2		6					
7, 3		7					
7, 4			8				
7				8			
8, 1					8		
8, 2					9		

KODESPORING

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}")
```

linje	x	x_it	return	y	x_it ₂		
6	5						
7, 1		5					
7, 2		6					
7, 3		7					
7, 4			8				
7				8			
8, 1					8		
8, 2					9		
8, 3					10		

KODESPORING

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}")
```

linje	x	x_it	return	y	x_it ₂	return	
6	5						
7, 1		5					
7, 2		6					
7, 3		7					
7, 4			8				
7				8			
8, 1					8		
8, 2					9		
8, 3					10		
8, 4						11	

KODESPORING

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}")
```

linje	x	x_it	return	y	x_it ₂	return	
6	5						
7, 1		5					
7, 2		6					
7, 3		7					
7, 4			8				
7				8			
8, 1					8		
8, 2					9		
8, 3					10		
8, 4						11	
8				11			

KODESPORING

ikke
samme
variabel!

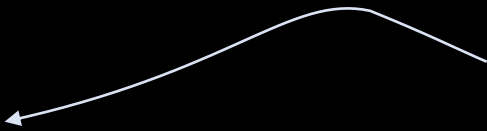
```
1  def incremented_thrice(x):  
2      x = x + 1  
3      x += 1  
4      return x + 1  
5  
6  x = 5  
7  y = incremented_thrice(x)  
8  y = incremented_thrice(y)  
9  
10 print(f"x: {x}, y: {y}")
```

linje	x	x_it	return	y	x_it ₂	return	print
6	5						
7, 1		5					
7, 2		6					
7, 3		7					
7, 4			8				
7				8			
8, 1					8		
8, 2					9		
8, 3					10		
8, 4						11	
8				11			
10							x: 5, y: 11

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



KODESPORING

```
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}")
```

linje	x	x_it	print	y	x_it ₂
6	5				
7, 1		5			
7, 2		6			
7, 3		7			
7, 4			8		
7				None	

KODESPORING

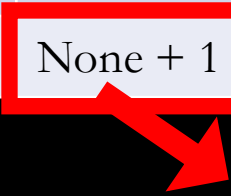
```
1  def incremented_thrice(x):
2      x = x + 1
3      x += 1
4      print(x + 1)
5
6  x = 5
7  y = incremented_thrice(x)
8  y = incremented_thrice(y)
9
10 print(f"x: {x}, y: {y}")
```

linje	x	x_it	print	y	x_it ₂
6	5				
7, 1		5			
7, 2		6			
7, 3		7			
7, 4			8		
7				None	
8, 1					None

KODESPORING

```
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}")
```

linje	x	x_it	print	y	x_it ₂
6	5				
7, 1		5			
7, 2		6			
7, 3		7			
7, 4			8		
7				None	
8, 1					None
8, 2					None + 1



krasj

KRASJ OG FEIL



Bilde: Canada Science and Technology Museum



Bilde: Jason Scott, CC-BY 2.0

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(4, 5, 6)) + " m3 i boksen")
```

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


MINNET



HP Laptop 14-em0924no 14" FHD

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

★★★★☆ (3 anmeldelser)

7.000,-

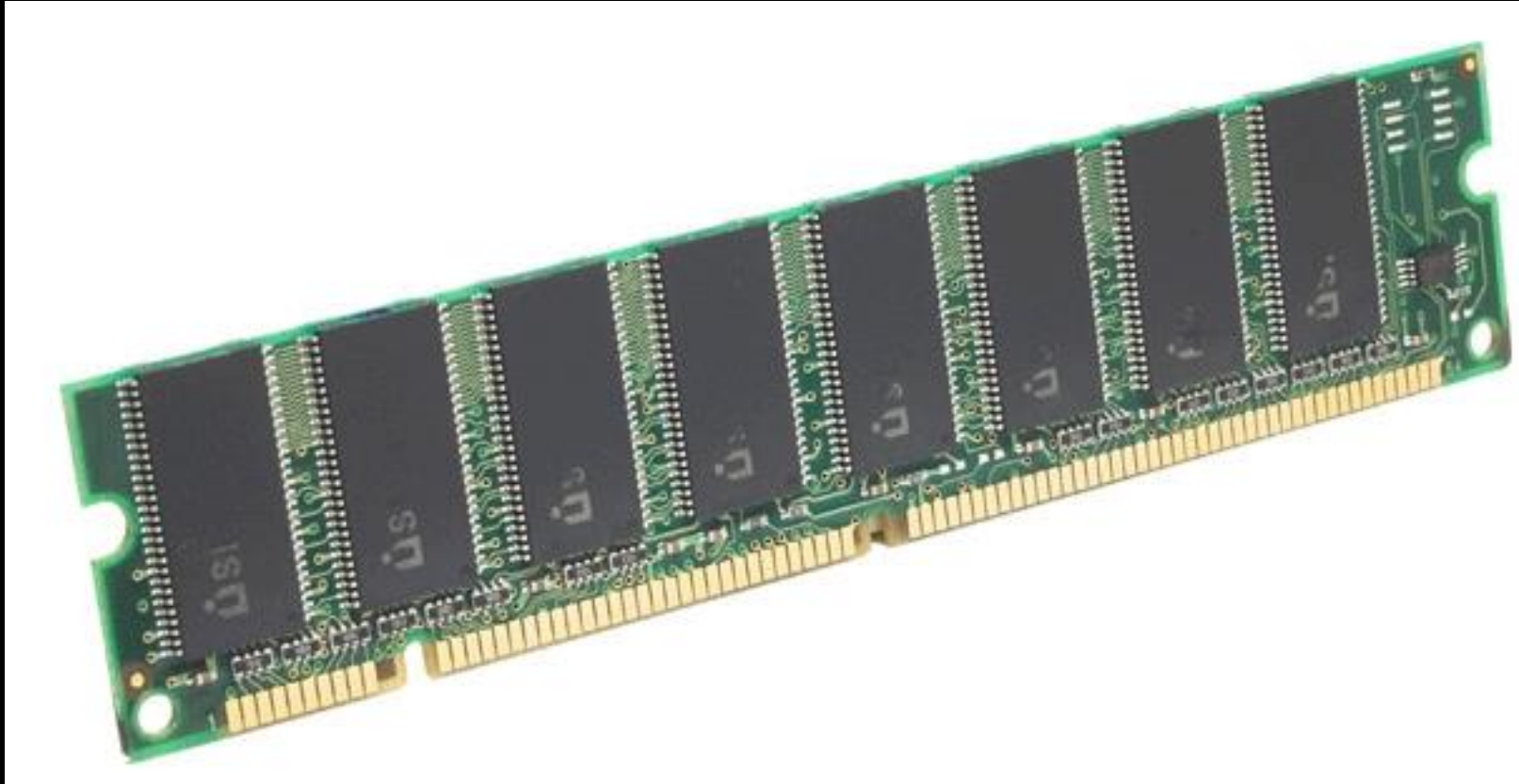
SPAR 3.000,-



Lenovo Ideapad Slim 3 14" FHD (arctic grey)

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

8.000,-



Random Access Memory (RAM)

MINNET

- En lagringsplass for 0'ere og 1'ere

én **bit** med informasjon



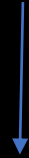
8 bit = 1 Byte

4 GB RAM tilsvarer 32 000 000 000 bits

MINNET

- En lagringsplass for 0'er og 1'ere

én **bit** med informasjon



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

...01110110101110101000010100000101011100101011111000111010101010101000...



x

en variabel

OBJEKT

10111011010111010100001010000010101110010101111100



ID-nummer for klassen
(typen) + annen metadata

verdien

42

101110110101111001000000000000000000000000000000101010

«dette objektet er en int»

42 i totallsystemet

0.1

00100110110100011100111101110011001100110011001101

«dette objektet er en float»

123
i totallsystemet

0.6000000238418579
i totallsystemet

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

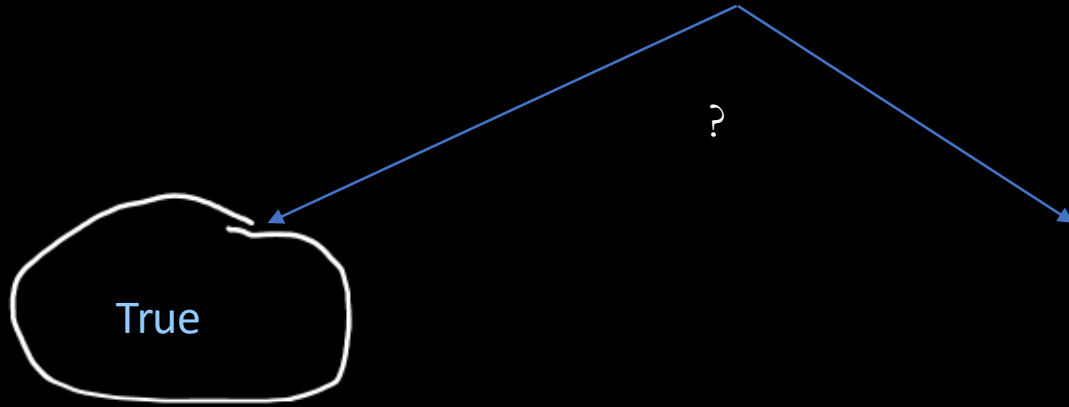
$\Rightarrow$ 0.100000001490116119384765625

1 + 2 == 3

?

True

False

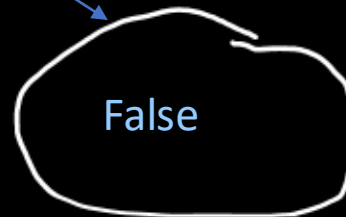


$0.1 + 0.2 == 0.3$

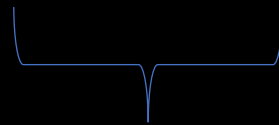
?

True

False

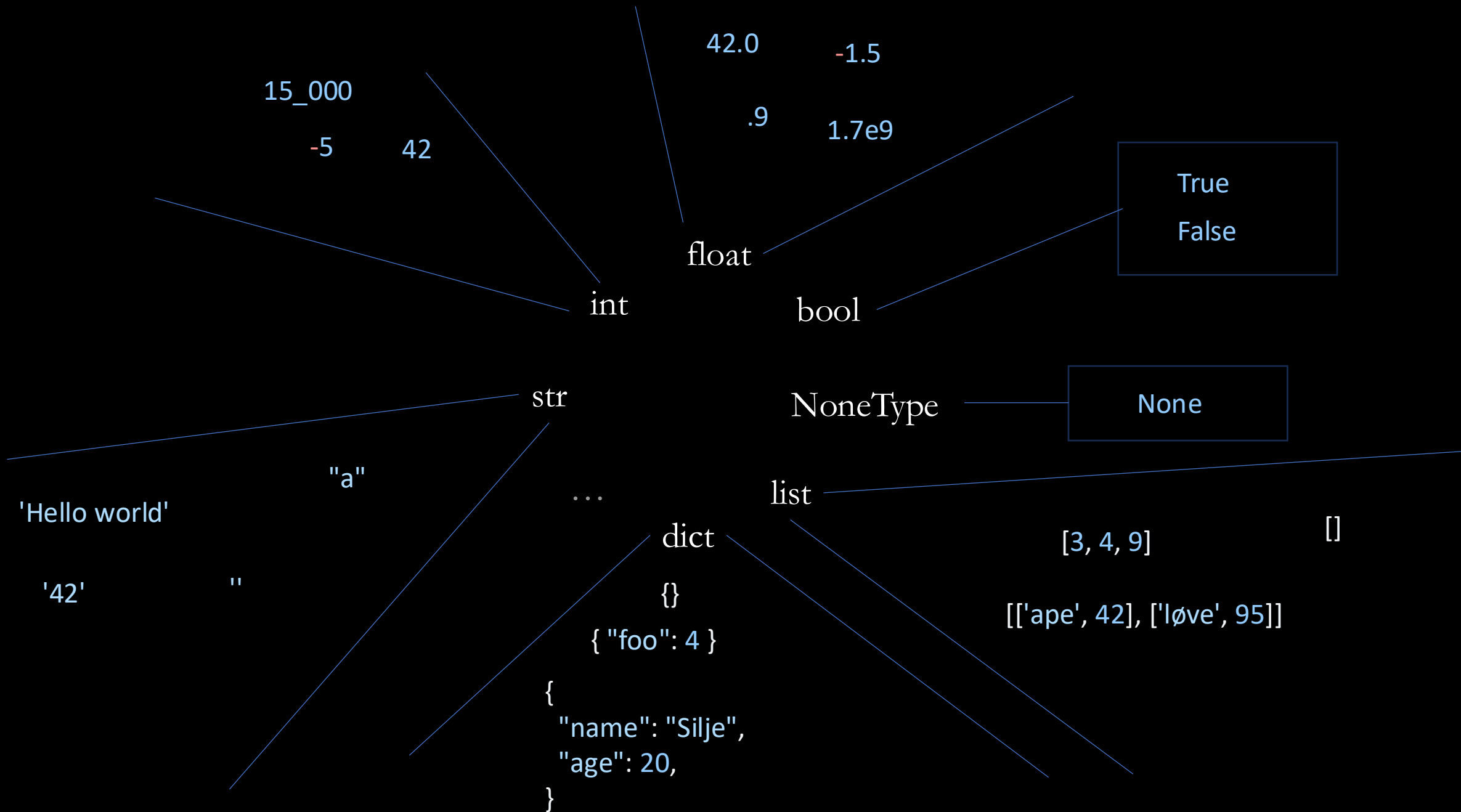


$$0.1 + 0.2 == 0.3$$



0.30000000000000004

TYPED



Gå til
www.menti.com

Skriv inn koden
3347 7691



eller bruker QR kode