

KODESPORING, BETINGELSER og FUNKSJONER

INF100

VÅR 2026

Crystal Chang Din

Med bidrag fra: Torstein Strømme

Quiz 1

 [INF100](#) > [Quizzes](#) > Quiz 1: velkommen til INF100

Vår 2026

[Hjem](#)
[Kunngjøringer](#)
[Oppgaver](#)
[Vurderinger](#)
[Personer](#)
[Digital litteraturliste](#)
[Panopto Video](#)

Quiz 1: velkommen til INF100

Forfall 27. jan. i 23.59	Poeng 1	Spørsmål 10
Tilgjengelig etter 13. jan. i 0:00	Tidsbegrensning Ingen	
Tillatte forsøk Ubegrenset		

Instruksjoner

Velkommen til INF100! Denne quizen skal kartlegge hvilke forventinger dere har til kurset, samt kartlegge hvilken bakgrunn dere har med programmering fra før.

Selv om quizen er obligatorisk å gjennomføre, vil resultatene ikke ha noen påvirkning på din endelig karakter. Vi vil benytte dine svar til å evaluere og tilrettelegge undervisningstilbudet.

LAB

- Du kan levere så mange ganger du vil.
- Du kan se resultatene av auto-testene umiddelbart.
- Det er mange gruppeledere klare til å hjelpe!
- 5 poeng på lab1 ved å delta i din gruppe denne uken (21. – 27. januar).

DROP-IN GRUPPETIMER

	Man	Tirs	Ons	Tors	Fre
8-10					
10-12				kickststart	
12-14					
14-16		forelesning			

Bruk læringsfellesskapet!

PRESENTASJON 1

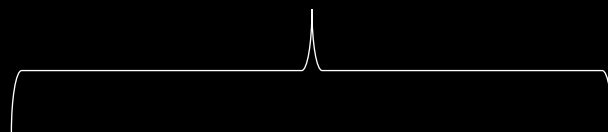
- 28. januar – 10. februar
- I løpet av denne uken: motta tidspunkt fra din gruppeleder på mitt.uib
- Hvis ikke: send melding til din gruppeleder!

VIKTIG!

MELD DEG INN I EN GRUPPE

ORDBOK

setning (engelsk: statement). Ett «steg» i et program, ofte én linje.



```
print('Hello, World!')
```

ORDBOK

setning (engelsk: statement). Ett «steg» i et program, ofte én linje.



```
print('Hello, World!')
```

objekt. En eller annen form for verdi som benyttes i programmet.

Eksempler på objekter:

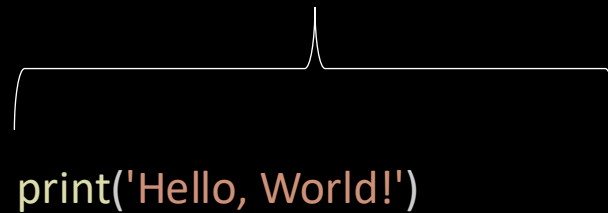
'Hello, World!' '42'

42 3.14 True

[4, 3, 2, 3]

ORDBOK

setning (engelsk: statement). Ett «steg» i et program, ofte én linje.



```
print('Hello, World!')
```

objekt. En eller annen form for verdi som benyttes i programmet.

funksjon. En «kommando» som kan få noe til å skje.

ORDBOK

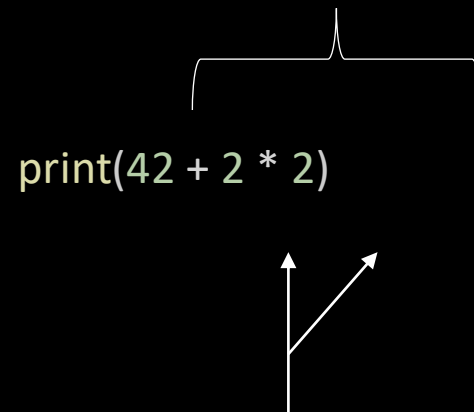
uttrykk (engelsk: expression). Et regnestykke som evaluerer til en verdi (et objekt).



```
print(42 + 2 * 2)
```

ORDBOK

uttrykk (engelsk: expression). Et regnestykke som evaluerer til en verdi (et objekt).



print(42 + 2 * 2)

operasjon. En måte å kombinere to objekter for å produsere et nytt objekt.

Eksempler på operasjoner:

+ - * / ** // %

Gå til
www.menti.com

Skriv inn koden
4116 8726



eller bruker QR kode

KODESPORING

- Koden utføres linje for linje
- Forutsi hva som skjer i neste steg

KODESPORING: EKSEMPEL

```
balance = 1000  
org_balance = balance
```

```
# Første år: 5% rente
```

```
interest_rate = 0.05  
interest_amount = balance * interest_rate  
balance = balance + interest_amount
```

```
# Andre år: 10% rente
```

```
interest_rate = 0.10  
interest_amount = balance * interest_rate  
balance = balance + interest_amount
```

```
difference = balance - org_balance  
print(f'Etter to år har du tjent {difference} kr i renter')
```

KODESPORING: EKSEMPEL

```
balance = 1000
```

```
org_balance = balance
```

Første år: 5% rente

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

Andre år: 10% rente

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

[illegible]

KODESPORING: EKSEMPEL

balance = 1000

```
org_balance = balance
```

Første år: 5% rente

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

Andre år: 10% rente

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

[illegible]

KODESPORING: EKSEMPEL

balance = 1000

```
org_balance = balance
```

Første år: 5% rente

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

Andre år: 10% rente

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

[illegible]

KODESPORING: EKSEMPEL

balance = 1000

```
org_balance = balance
```

Første år: 5% rente

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

Andre år: 10% rente

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

[illegible]

KODESPORING: EKSEMPEL

balance = 1000

```
org_balance = balance
```

Første år: 5% rente

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

Andre år: 10% rente

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

[illegible]

KODESPORING: EKSEMPEL

balance = 1000

```
org_balance = balance
```

Første år: 5% rente

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

Andre år: 10% rente

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

[illegible]

KODESPORING: EKSEMPEL

balance = 1000

```
org_balance = balance
```

Første år: 5% rente

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

Andre år: 10% rente

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

[illegible]

KODESPORING: EKSEMPEL

```
balance = 1000
```

```
org_balance = balance
```

```
# Første år: 5% rente
```

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
# Andre år: 10% rente
```

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

linje	balance	org_balance	interest_rate	interest_amount		
1	1000					
2		1000				
5			0.05			
6				50.0		
7	1050.0					
10			0.10			

KODESPORING: EKSEMPEL

```
balance = 1000
```

```
org_balance = balance
```

```
# Første år: 5% rente
```

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
# Andre år: 10% rente
```

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

linje	balance	org_balance	interest_rate	interest_amount		
1	1000					
2		1000				
5			0.05			
6				50.0		
7	1050.0					
10			0.10			

KODESPORING: EKSEMPEL

```
balance = 1000
```

```
org_balance = balance
```

```
# Første år: 5% rente
```

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
# Andre år: 10% rente
```

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

linje	balance	org_balance	interest_rate	interest_amount		
1	1000					
2		1000				
5			0.05			
6				50.0		
7	1050.0					
10			0.10			
11				105.0		

KODESPORING: EKSEMPEL

```
balance = 1000
```

```
org_balance = balance
```

```
# Første år: 5% rente
```

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
# Andre år: 10% rente
```

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

linje	balance	org_balance	interest_rate	interest_amount		
1	1000					
2		1000				
5			0.05			
6				50.0		
7	1050.0					
10			0.10			
11				105.0		

KODESPORING: EKSEMPEL

```
balance = 1000
```

```
org_balance = balance
```

```
# Første år: 5% rente
```

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
# Andre år: 10% rente
```

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

linje	balance	org_balance	interest_rate	interest_amount		
1	1000					
2		1000				
5			0.05			
6				50.0		
7	1050.0					
10			0.10			
11				105.0		
12	1155.0					

KODESPORING: EKSEMPEL

```
balance = 1000
```

```
org_balance = balance
```

```
# Første år: 5% rente
```

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
# Andre år: 10% rente
```

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

linje	balance	org_balance	interest_rate	interest_amount		
1	1000					
2		1000				
5			0.05			
6				50.0		
7	1050.0					
10			0.10			
11				105.0		
12	1155.0					

KODESPORING: EKSEMPEL

```
balance = 1000
```

```
org_balance = balance
```

```
# Første år: 5% rente
```

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
# Andre år: 10% rente
```

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

linje	balance	org_balance	interest_rate	interest_amount	difference	
1	1000					
2		1000				
5			0.05			
6				50.0		
7	1050.0					
10			0.10			
11				105.0		
12	1155.0					
14					155.0	

KODESPORING: EKSEMPEL

```
balance = 1000
```

```
org_balance = balance
```

```
# Første år: 5% rente
```

```
interest_rate = 0.05
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
# Andre år: 10% rente
```

```
interest_rate = 0.10
```

```
interest_amount = balance * interest_rate
```

```
balance = balance + interest_amount
```

```
difference = balance - org_balance
```

```
print(f'Etter to år har du tjent {difference} kr i renter')
```

linje	balance	org_balance	interest_rate	interest_amount	difference	print
1	1000					
2		1000				
5			0.05			
6				50.0		
7	1050.0					
10			0.10			
11				105.0		
12	1155.0					
14					155.0	
15						



Etter to år har du tjent 155.0 kr i renter

EKSAMENSOPPGAVE HØST 24

1(b)

```
a = 1
b = 2
a = a + b
b = a * b
a -= 1
print(b - a)
```

Kva skriv dette programmet ut? (hvis programmet krasjar, skriv berre Error)

EKSAMENSOPPGAVE HØST 24

1(b)

```
a = 1
b = 2
a = a + b
b = a * b
a -= 1
print(b - a)
```

Kva skriv dette programmet ut? (hvis programmet krasjar, skriv berre Error)

<https://inf100.iib.no/bin/saving/>

```
balance = 1000
org_balance = balance

# Første år: 5% rente
interest_rate = 0.05
interest_amount = balance * interest_rate
balance = balance + interest_amount

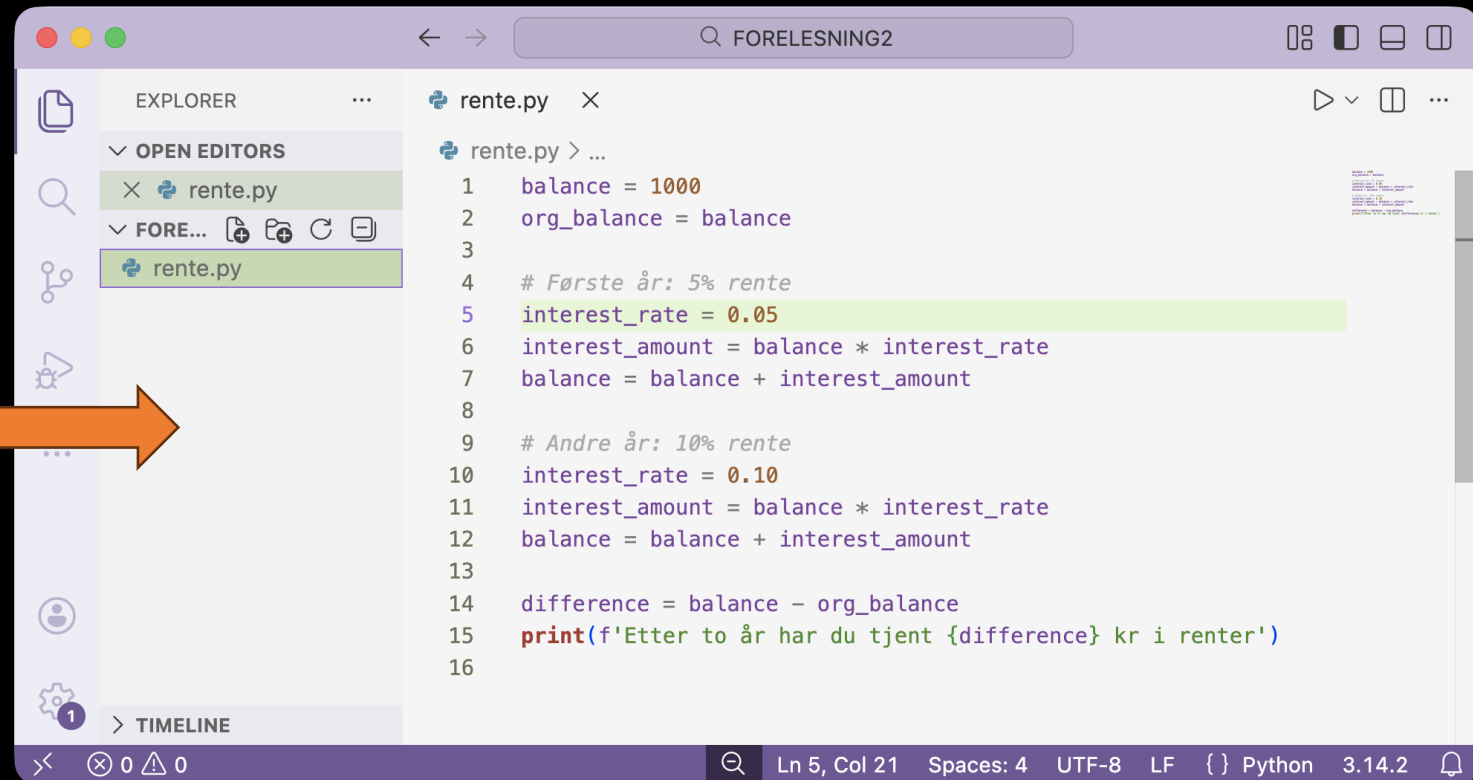
# Andre år: 10% rente
interest_rate = 0.10
interest_amount = balance * interest_rate
balance = balance + interest_amount

difference = balance - org_balance
print(f'Etter to år har du tjent {difference} kr i renter')
```

Kopier

Se steg

Kjør



FORELESNING2

EXPLORER

OPEN EDITORS

rente.py

FORE...

rente.py

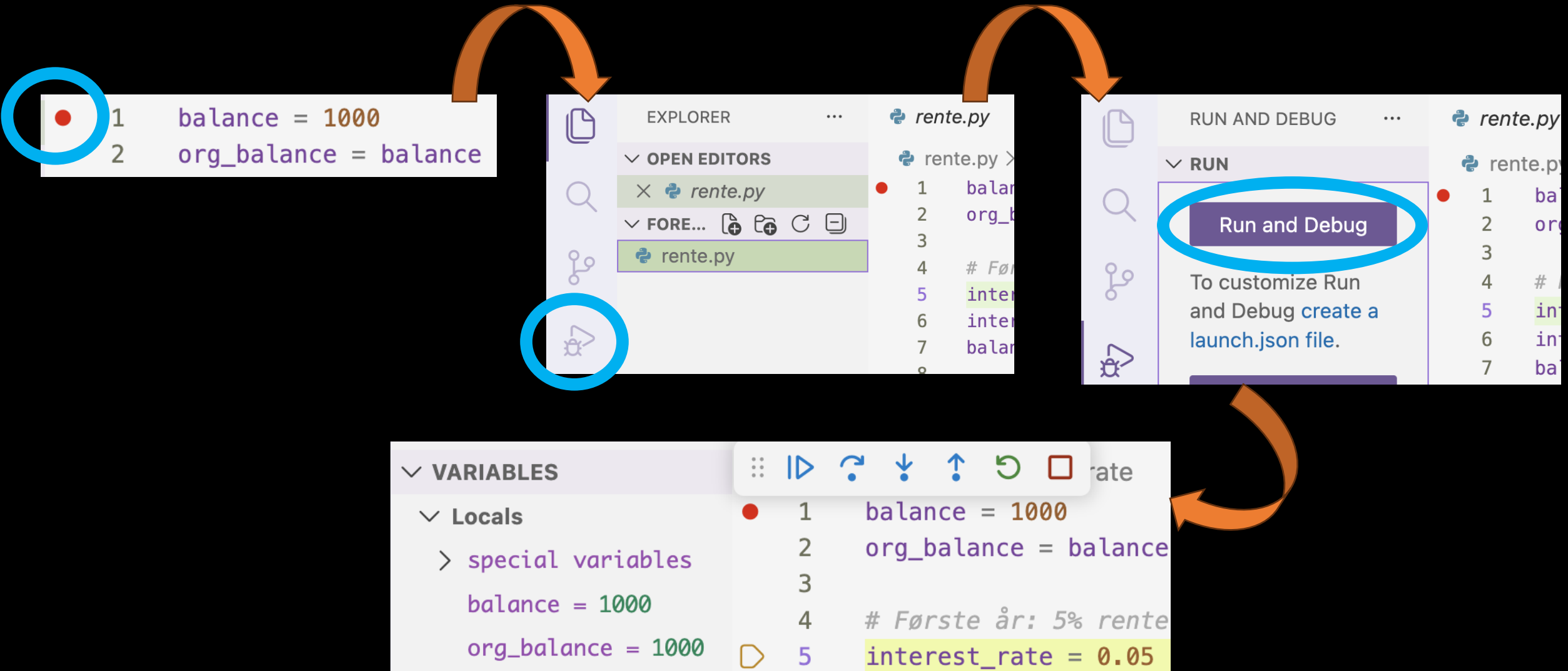
rente.py > ...

```
1 balance = 1000
2 org_balance = balance
3
4 # Første år: 5% rente
5 interest_rate = 0.05
6 interest_amount = balance * interest_rate
7 balance = balance + interest_amount
8
9 # Andre år: 10% rente
10 interest_rate = 0.10
11 interest_amount = balance * interest_rate
12 balance = balance + interest_amount
13
14 difference = balance - org_balance
15 print(f'Etter to år har du tjent {difference} kr i renter')
16
```

1 > TIMELINE

Ln 5, Col 21 Spaces: 4 UTF-8 LF {} Python 3.14.2

KODESPORING MED DEBUGGER



TYPED

int 42

float 1.5

str "....."

dict { ...: ..., ...: ... }

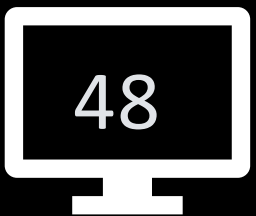
list [..., ..., ...]

bool True eller False

NoneType None

*typen bestemmer
hva operator betyr*

```
x = 42  
y = 6  
total = x + y  
print(total)
```



```
x = "42"  
y = "6"  
total = x + y  
print(total)
```



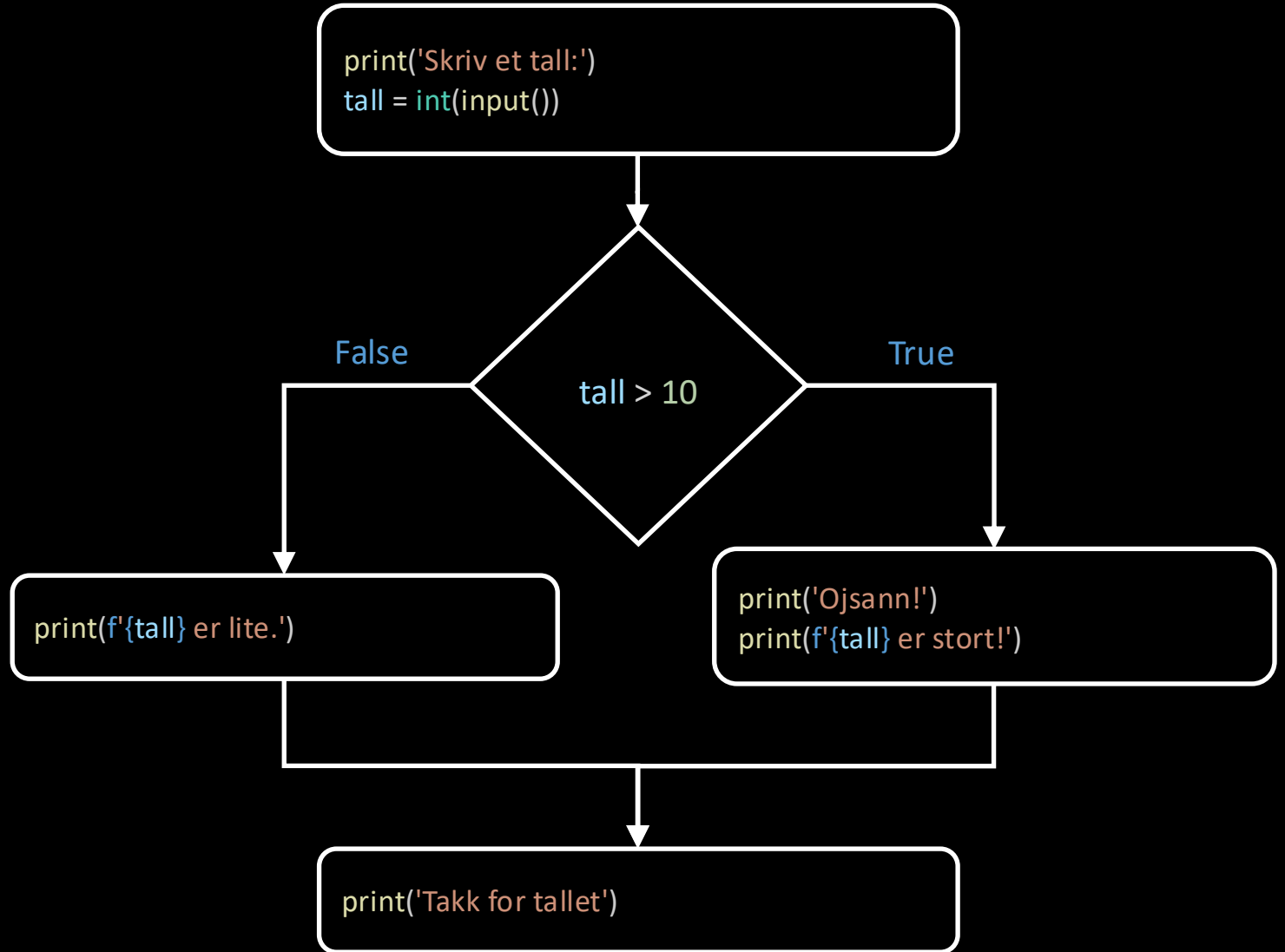
BETINGELSER

BETINGELSER

```
print('Skriv et tall:')  
tall = int(input())
```

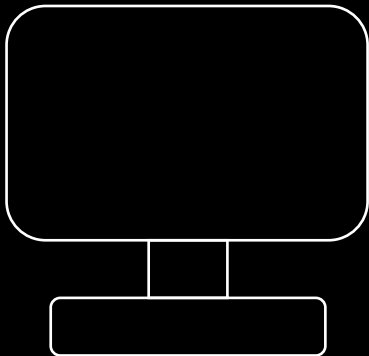
```
if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')
```

```
print('Takk for tallet')
```



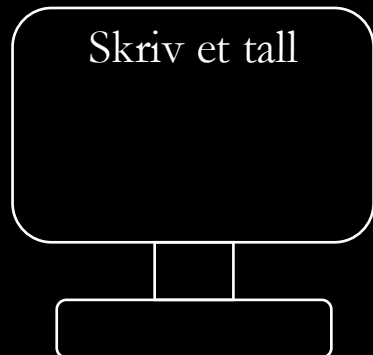
BETINGELSER

```
→ print('Skriv et tall:')  
tall = int(input())  
  
if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')  
  
print('Takk for tallet')
```



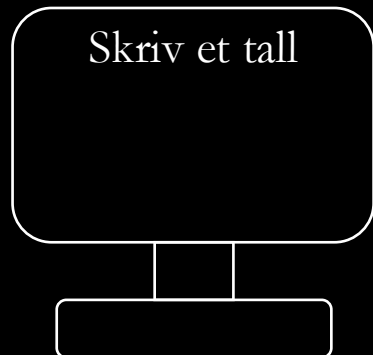
BETINGELSER

```
→ print('Skriv et tall:')  
tall = int(input())  
  
if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')  
  
print('Takk for tallet')
```



BETINGELSER

```
→ print('Skriv et tall:')  
   tall = int(input())  
  
   if tall > 10:  
       print('Ojsann!')  
       print(f'{tall} er stort!')  
   else:  
       print(f'{tall} er lite.')  
  
   print('Takk for tallet')
```



BETINGELSER

→ `print('Skriv et tall:')`
`tall = int(input())`

`if tall > 1`

`print`

`else`

`int(input())`



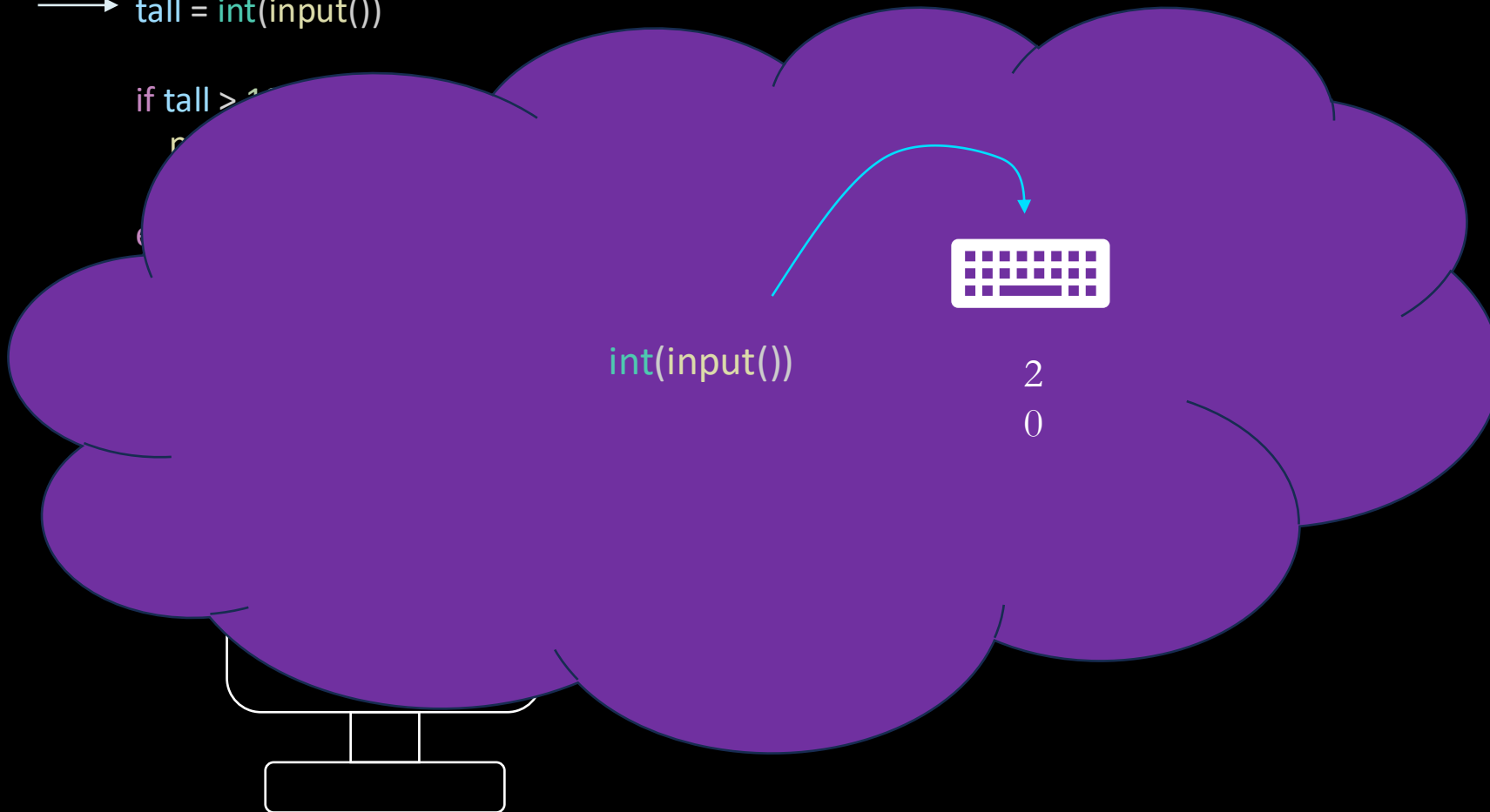
BETINGELSER

→ `print('Skriv et tall:')`
`tall = int(input())`

`if tall > 1`

`print`

`else`



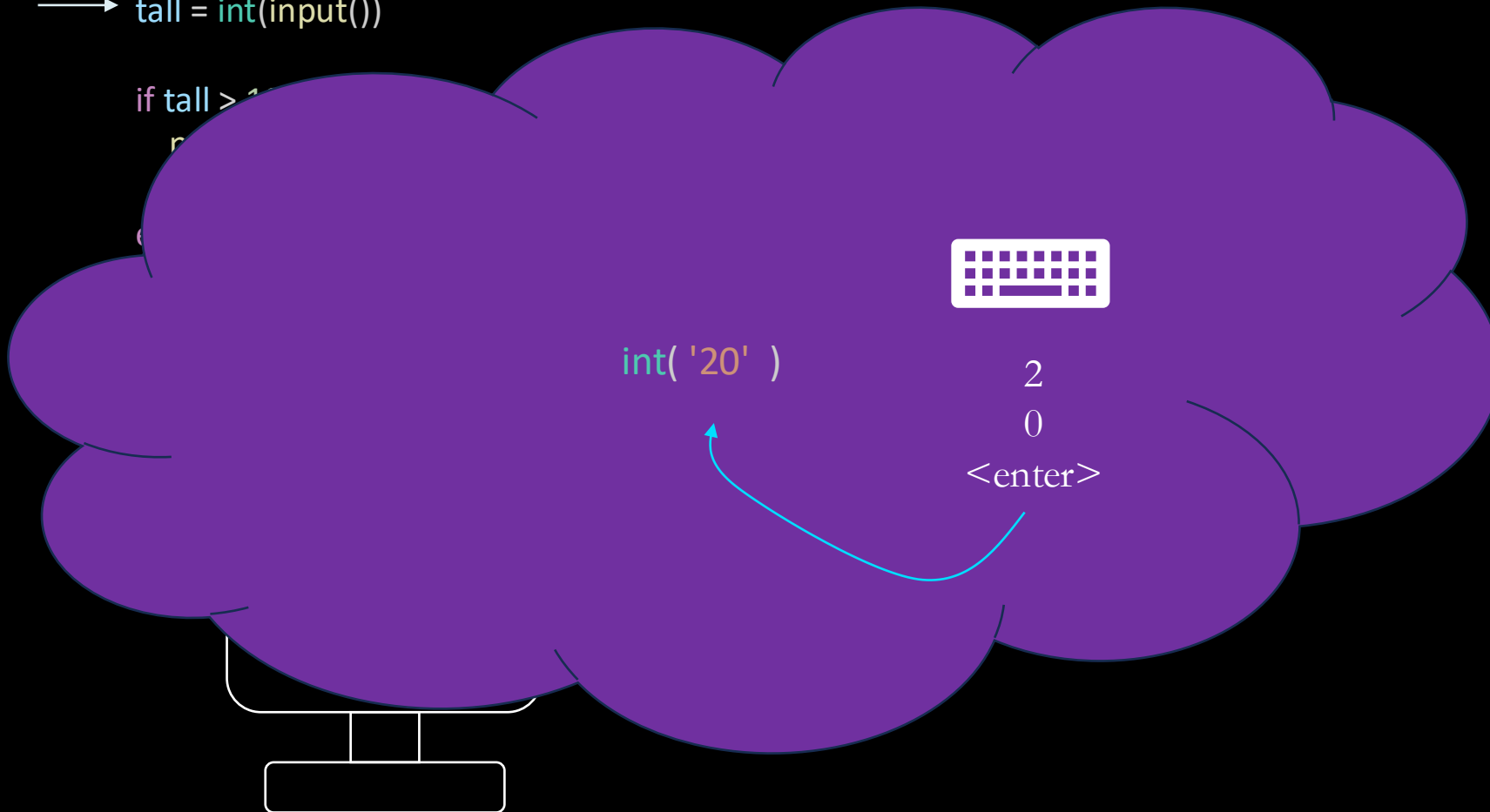
BETINGELSER

→ `print('Skriv et tall:')`
`tall = int(input())`

`if tall > 1`

`print`

`else`



BETINGELSER

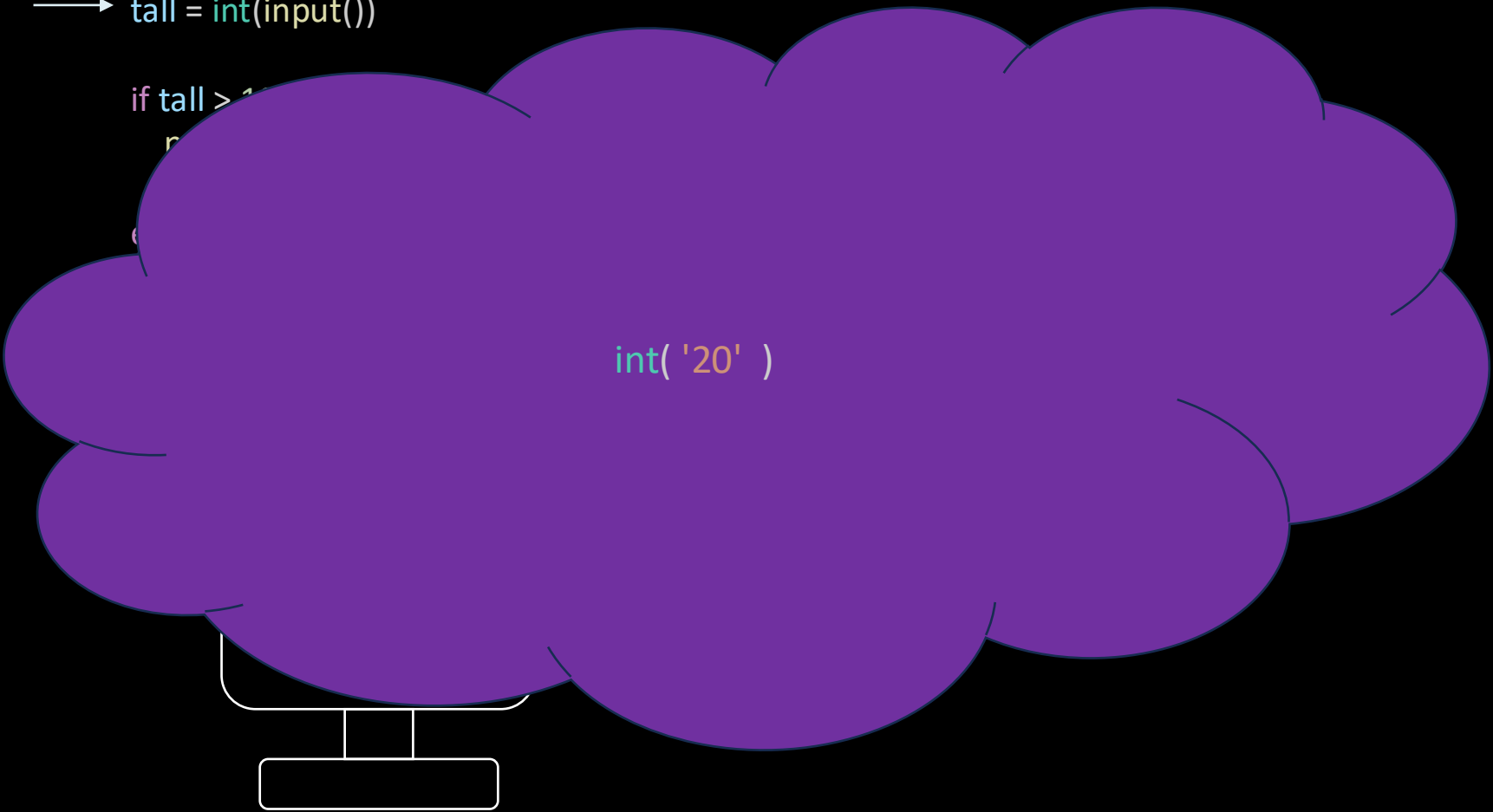
→ `print('Skriv et tall:')`
`tall = int(input())`

`if tall > 1`

`print`

`else`

`int('20')`



BETINGELSER

→ `print('Skriv et tall:')`
`tall = int(input())`

`if tall > 1`

`print`

`else`

20

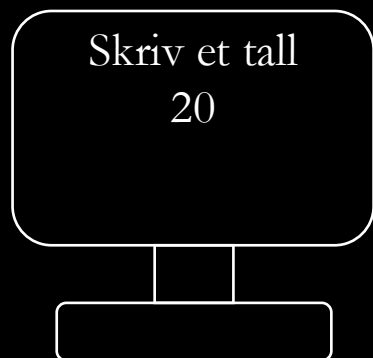


BETINGELSER

→ `print('Skriv et tall:')
tall = int(input())` 20

```
if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')
```

```
print('Takk for tallet')
```



VARIABLER

OBJEKTER

tall



20

BETINGELSER

```
print('Skriv et tall:')  
tall = int(input())
```

```
→ if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')  
  
print('Takk for tallet')
```



VARIABLER

OBJEKTER

tall

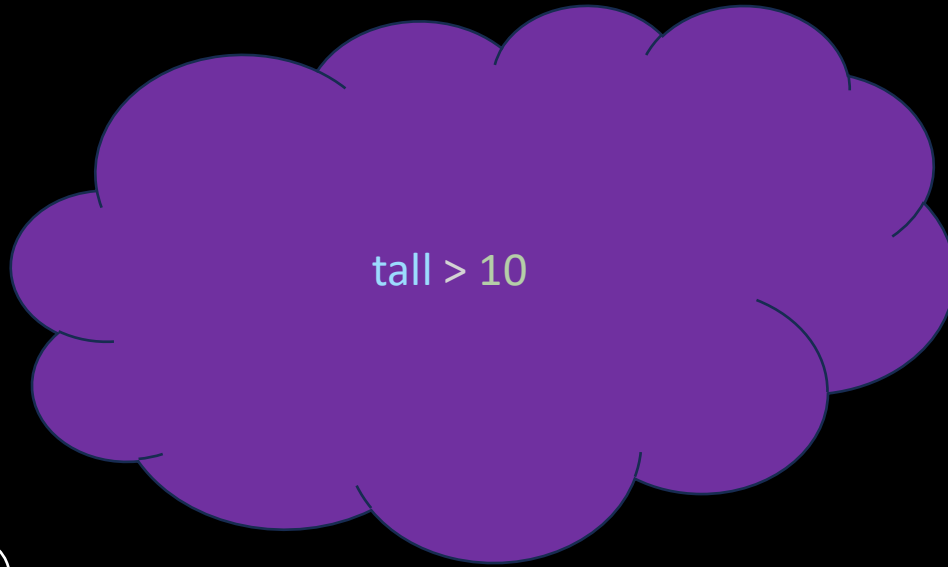


20

BETINGELSER

```
print('Skriv et tall:')  
tall = int(input())
```

```
→ if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')  
  
print('Takk for tallet')
```



VARIABLER

OBJEKTER

tall

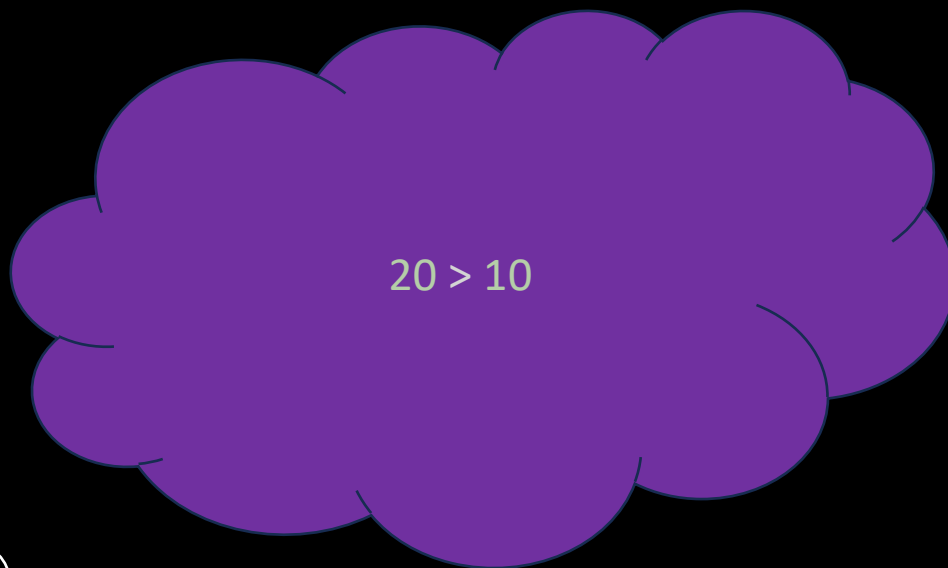
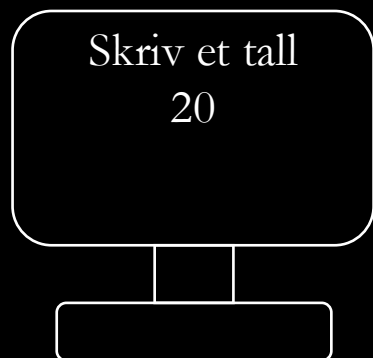


20

BETINGELSER

```
print('Skriv et tall:')  
tall = int(input())
```

```
→ if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')  
  
print('Takk for tallet')
```



VARIABLER

OBJEKTER

tall

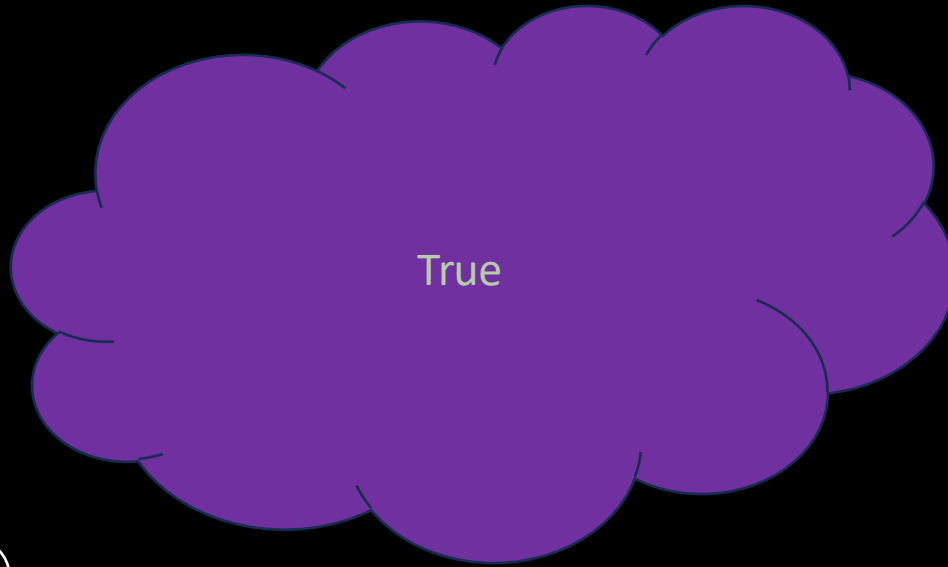


20

BETINGELSER

```
print('Skriv et tall:')  
tall = int(input())
```

```
→ if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')  
  
print('Takk for tallet')
```



VARIABLER

OBJEKTER

tall



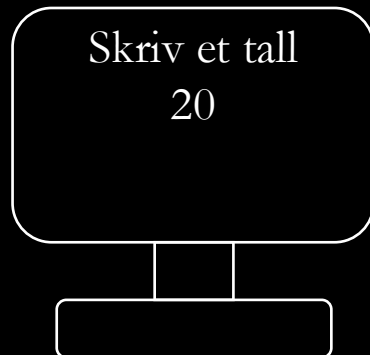
20

BETINGELSER

```
print('Skriv et tall:')  
tall = int(input())
```

```
→ if tall > 10:   
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')
```

```
print('Takk for tallet')
```



VARIABLER

OBJEKTER

tall



20

BETINGELSER

```
print('Skriv et tall:')  
tall = int(input())
```

```
→ if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')
```

```
print('Takk for tallet')
```



VARIABLER

OBJEKTER

tall



20

BETINGELSER

```
print('Skriv et tall:')  
tall = int(input())
```

```
if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')
```

```
print('Takk for tallet')
```

VARIABLER

OBJEKTER

tall



20



BETINGELSER

```
print('Skriv et tall:')  
tall = int(input())
```

```
if tall > 10:  
    print('Ojsann!')  
    print(f'{tall} er stort!')  
else:  
    print(f'{tall} er lite.')
```

```
print('Takk for tallet')
```

VARIABLER

OBJEKTER

tall



20



BETINGELSER

```
print('Skriv et tall:')
```

```
tall = int(input())
```

```
if tall > 10:
```

```
    print('Ojsann!')
```

```
    print(f'{tall} er stort!')
```

```
else:
```

```
    print(f'{tall} er lite.')
```

```
→ print('Takk for tallet')
```



VARIABLER

OBJEKTER

tall



20

BETINGELSER

```
print('Skriv et tall:')
```

```
tall = int(input())
```

```
if tall > 10:
```

```
    print('Ojsann!')
```

```
    print(f'{tall} er stort!')
```

```
else:
```

```
    print(f'{tall} er lite.')
```

```
print('Takk for tallet')
```



VARIABLER

OBJEKTER

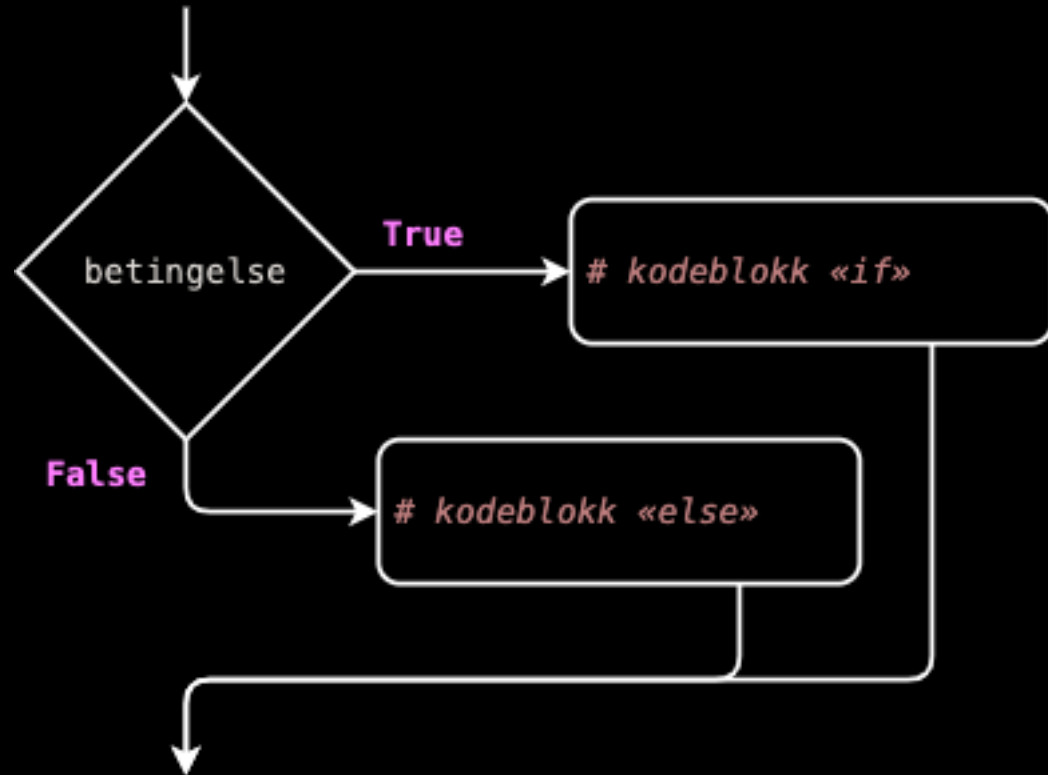
tall



20

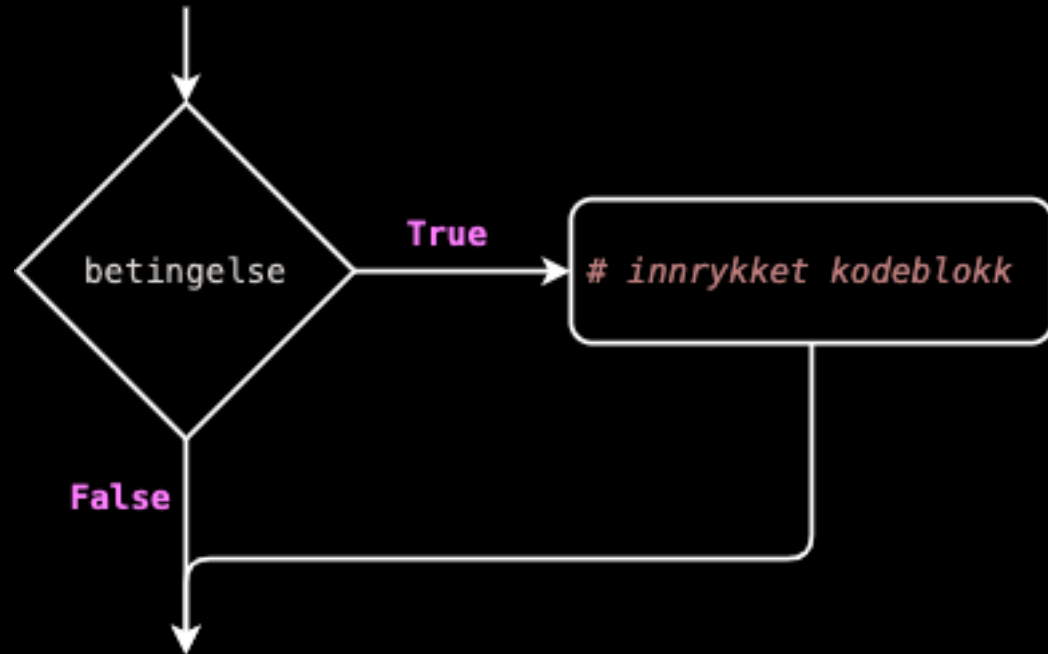
IF-ELSE

```
if (betingelse):  
    #  
    # kodeblokk «if»  
    #  
else:  
    #  
    # kodeblokk «else»  
    #  
# ...
```



IF

```
if (betingelse):  
    #  
    # kodeblokk «if»  
    #  
# ...
```

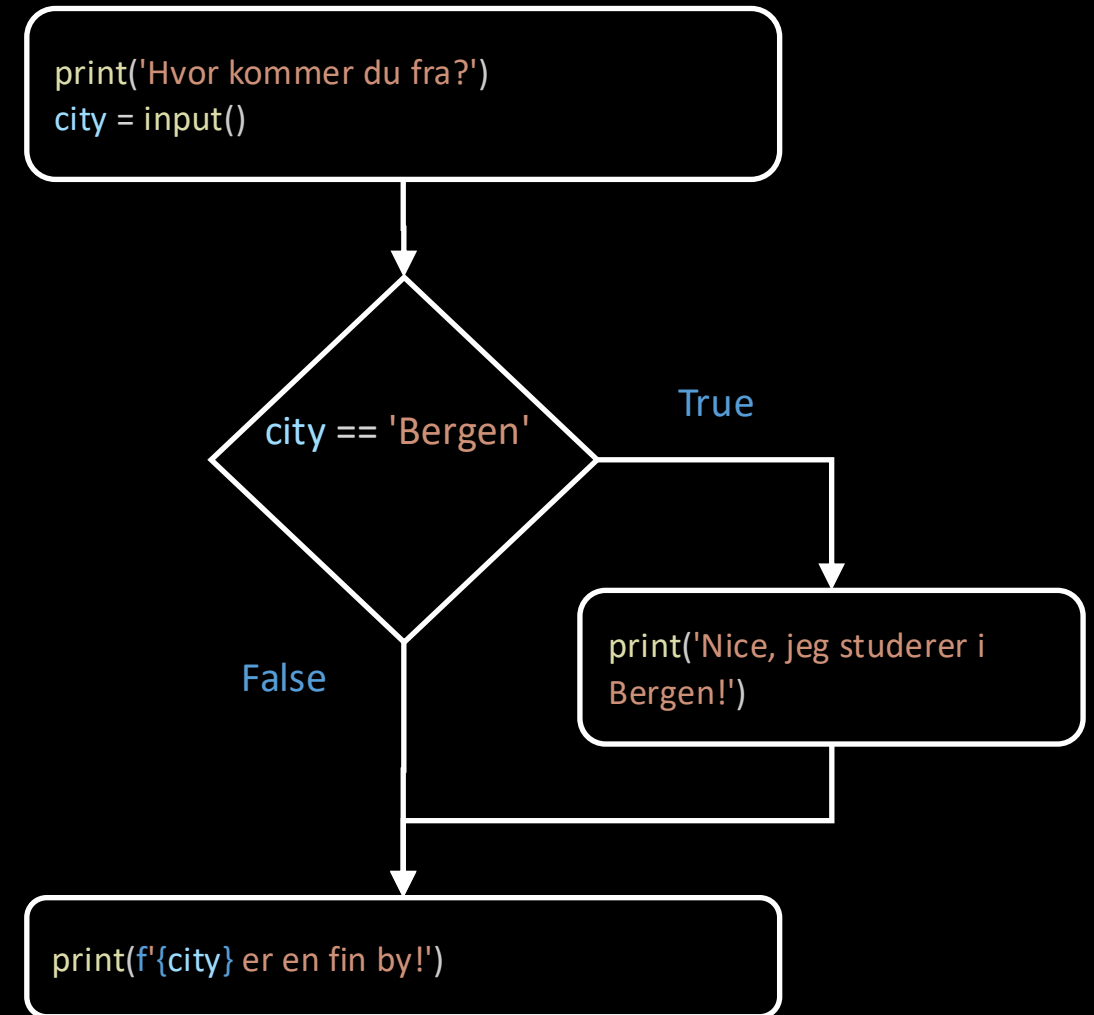


IF

```
print('Hvor kommer du fra?')  
city = input()
```

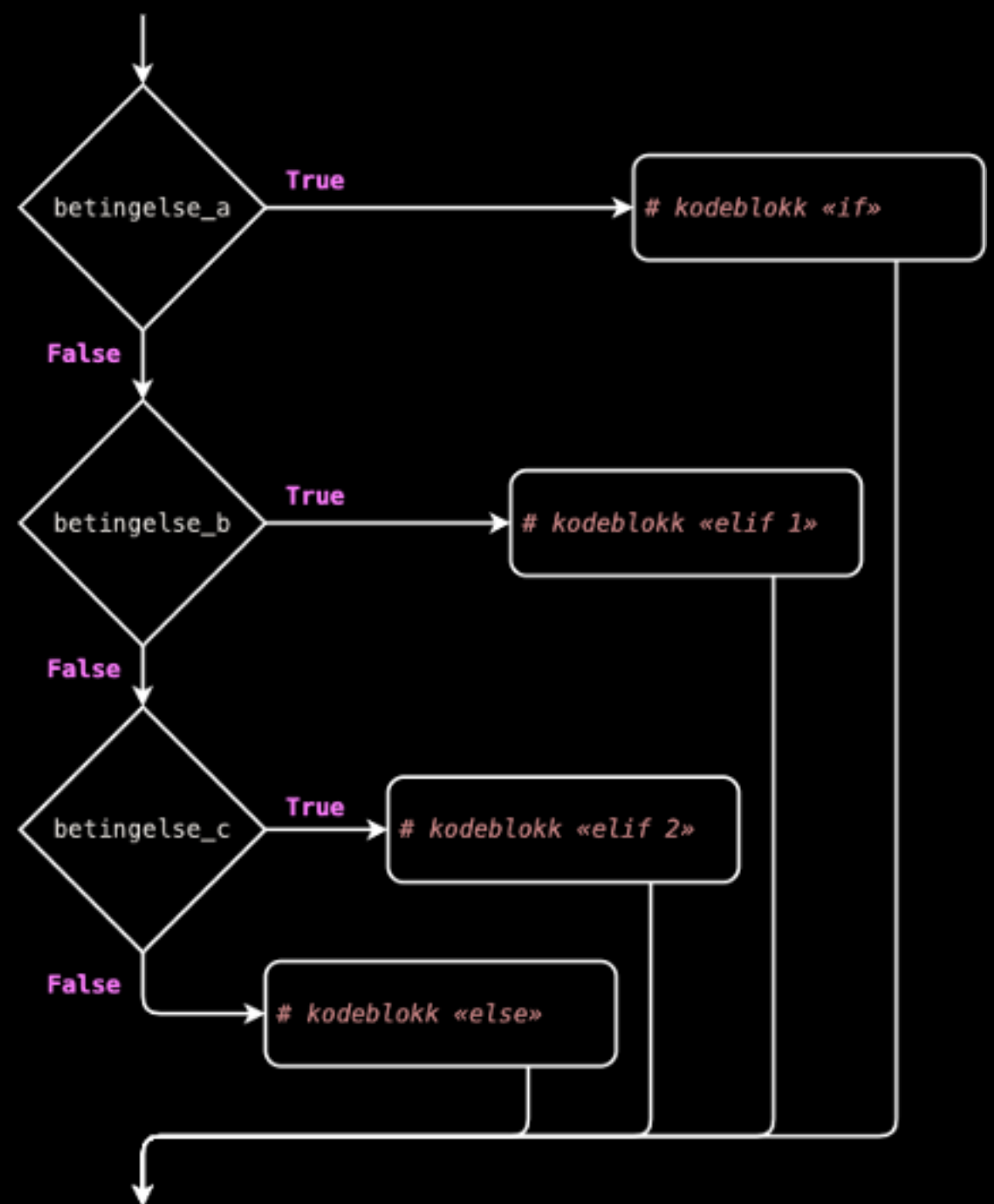
```
if (city == 'Bergen'):  
    print('Nice, jeg studerer i Bergen!')
```

```
print(f'{city} er en fin by!')
```



IF-ELIF-ELSE

```
if betingelse_a:
    #
    # kodeblokk «if»
    #
elif betingelse_b:
    #
    # kodeblokk «elif 1»
    #
elif betingelse_c:
    #
    # kodeblokk «elif 2»
    #
else:
    #
    # kodeblokk «else»
    #
# ...
```

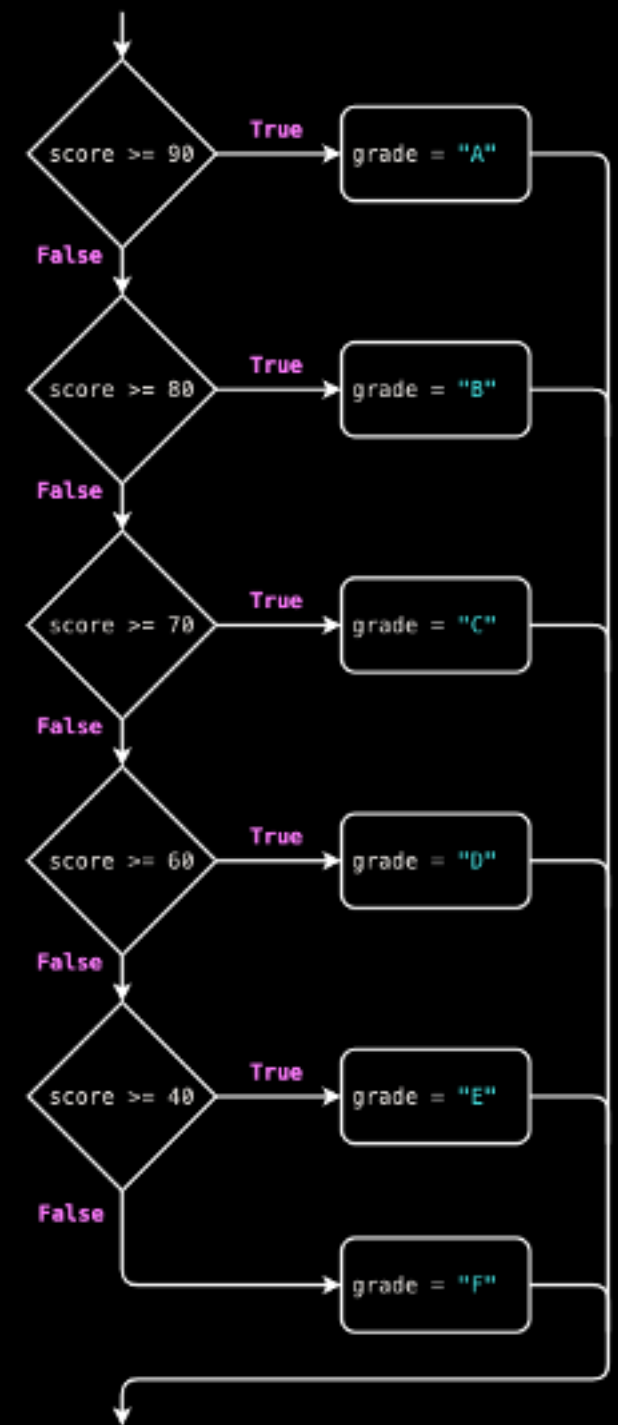


IF-ELIF-ELSE

```
print("Hvor mange poeng fikk du?")  
score = int(input())
```

```
if score >= 90:  
    grade = "A"  
elif score >= 80:  
    grade = "B"  
elif score >= 70:  
    grade = "C"  
elif score >= 60:  
    grade = "D"  
elif score >= 40:  
    grade = "E"  
else:  
    grade = "F"
```

```
print(f"Du fikk en {grade}.")
```



True

False

BOOLSKE UTTRYKK

Uttrykk	Evaluerer til
True	True
False	False
4 == 5	False
66 == 66	True

BOOLSKE UTTRYKK

Uttrykk	Evaluerer til
True	True
False	False
4 == 5	False
66 == 66	True
45 > 33	True
2.9 < 2.7	False

> < >= <= == in

BOOLSKE UTTRYKK

Uttrykk	Evaluerer til
True	True
False	False
4 == 5	False
66 == 66	True
45 > 33	True
2.9 < 2.7	False
'Abcd' < 'Abcz'	True
'ask' in 'oppvask'	True
0.2 + 0.3 == 0.5	True

ORDBOK

Boolsk verdi. En verdi som er enten True eller False.

```
x = int(input())
```

Betingelse. Uttrykk som evaluerer til en boolsk verdi

```
if x < 0:
    print('flip it')
    x = -x
else:
    print('leave it')
```

Innrykk. Antall mellomrom foran en kodeblokk.

```
print('absolute value of x is ', x)
```

If-setning. Et avsnitt av kildekoden hvor kontrollflyten kan gå én av to (eller flere) veier avhengig av betingelse.

Kodeblokk. Et avsnitt av kildekoden gruppert sammen med samme innrykk. Starter alltid med kolon (:).

ORDBOK

Boolsk verdi. En verdi som er enten True eller False.

```
x = int(input())
```

Betingelse. Uttrykk som evaluerer til en boolsk verdi

```
if x < 0:
    print('flip it')
    x = -x
else:
    print('leave it')
```

Innrykk. Antall mellomrom foran en kodeblokk.

```
print('absolute value of x is ', x)
```

If-setning. Et avsnitt av kildekoden hvor kontrollflyten kan gå én av to (eller flere) veier avhengig av betingelse.

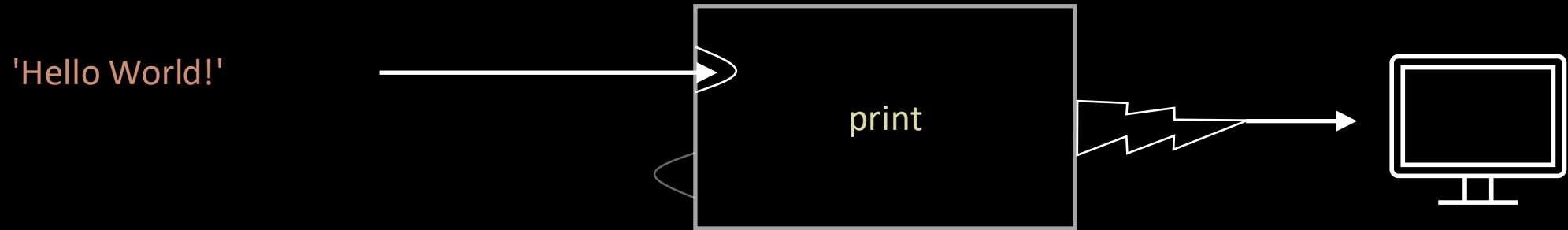
Kodeblokk. Et avsnitt av kildekoden gruppert sammen med samme innrykk. Starter alltid med kolon (:).

FUNKSJONER



FUNKSJONER MED EFFEKT

```
print('Hello World!')
```



eksempel på en «effekt:»

- noe skjer på skjermen
- noe vises i terminalen
- noe blir lagret i en fil

FUNKSJONER MED RETURVERDI

`max(1, 2, 3)`

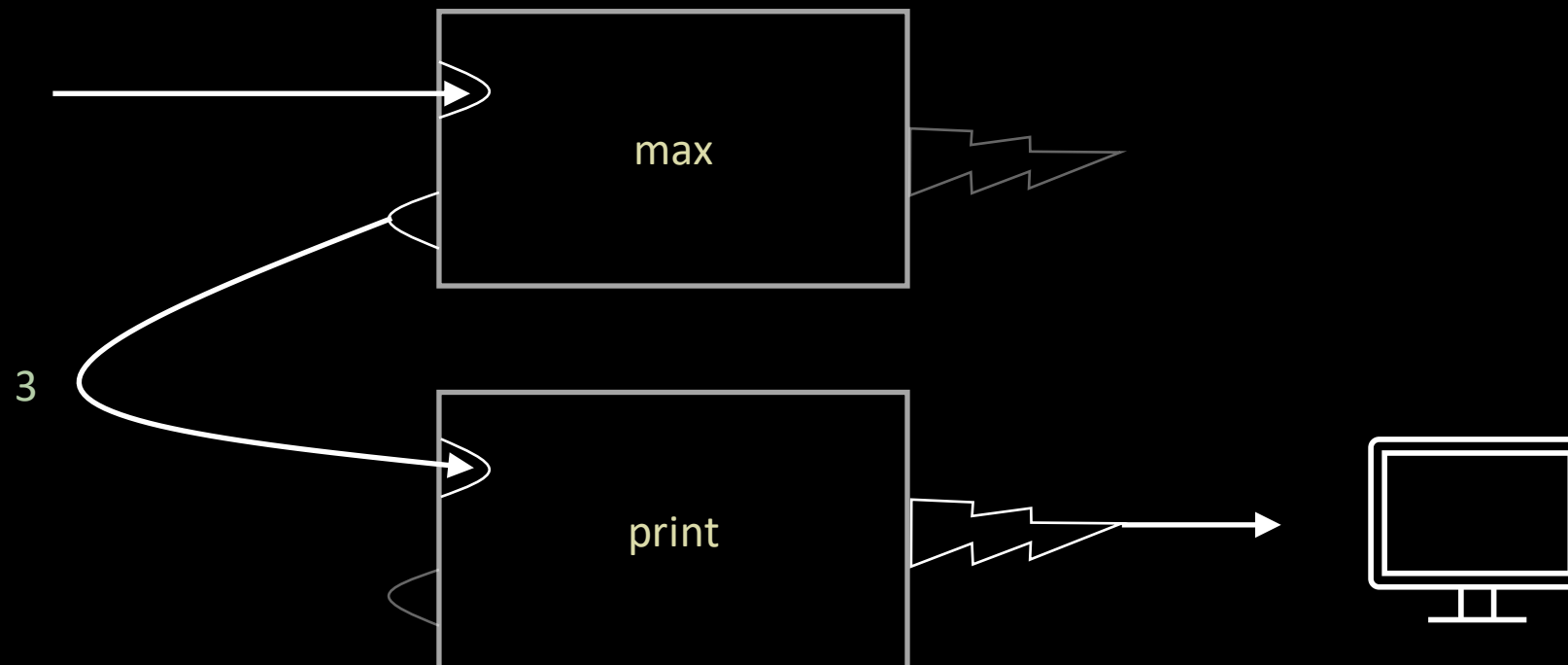


KOMPONERING AV FUNKSJONSKALL

```
print(max(1, 2, 3))
```

evaluerer til sin returverdi

1, 2, 3



INNEBYGDE FUNKSJONER

```
print("Skriv ut noe til terminalen")
```

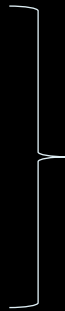
```
x = len("Lengden av en streng")
```

```
x = sum(1, 2, 3)
```

```
x = min(1, 2, 3)
```

```
x = max(1, 2, 3)
```

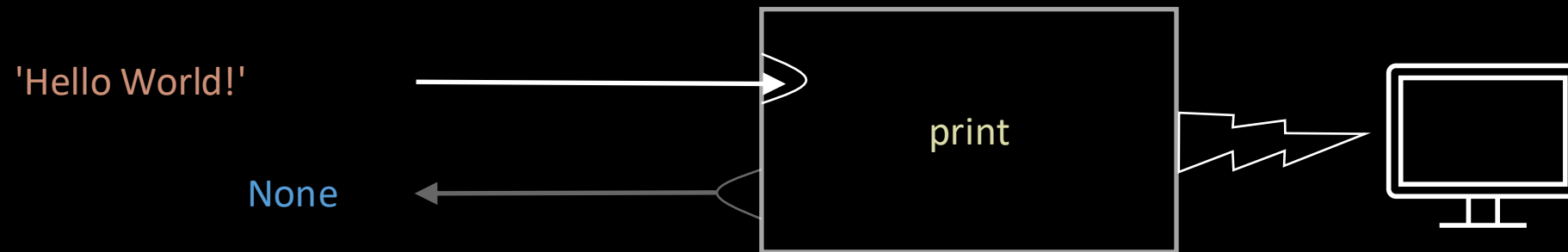
```
x = abs(-3)
```



Har returverdi

ALLE FUNKSJONER HAR RETURVERDI

```
print('Hello World!')
```



...selv om noen funksjoner alltid returnerer verdien `None`

SANT ELLER USANT?

- Alle meningsfulle funksjoner må ha input  f. eks. `print()`
- Alle meningsfulle funksjoner har en effekt  f. eks. `max` -funksjonen
- Alle funksjoner har en returverdi  kan være `None`

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 *innrykke* blir utført
når funksjonen kalles

Retursetning. Avslutter funksjonen.
Returverdien bestemmes her.

hvis ingen retursetning,
vil None returneres

LIVE MED DEBUGGER

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

```
a = 5  
b = incremented_thrice(a)  
c = incremented_thrice(b)  
  
print(f"a: {a}, b: {b}, c: {c}")
```