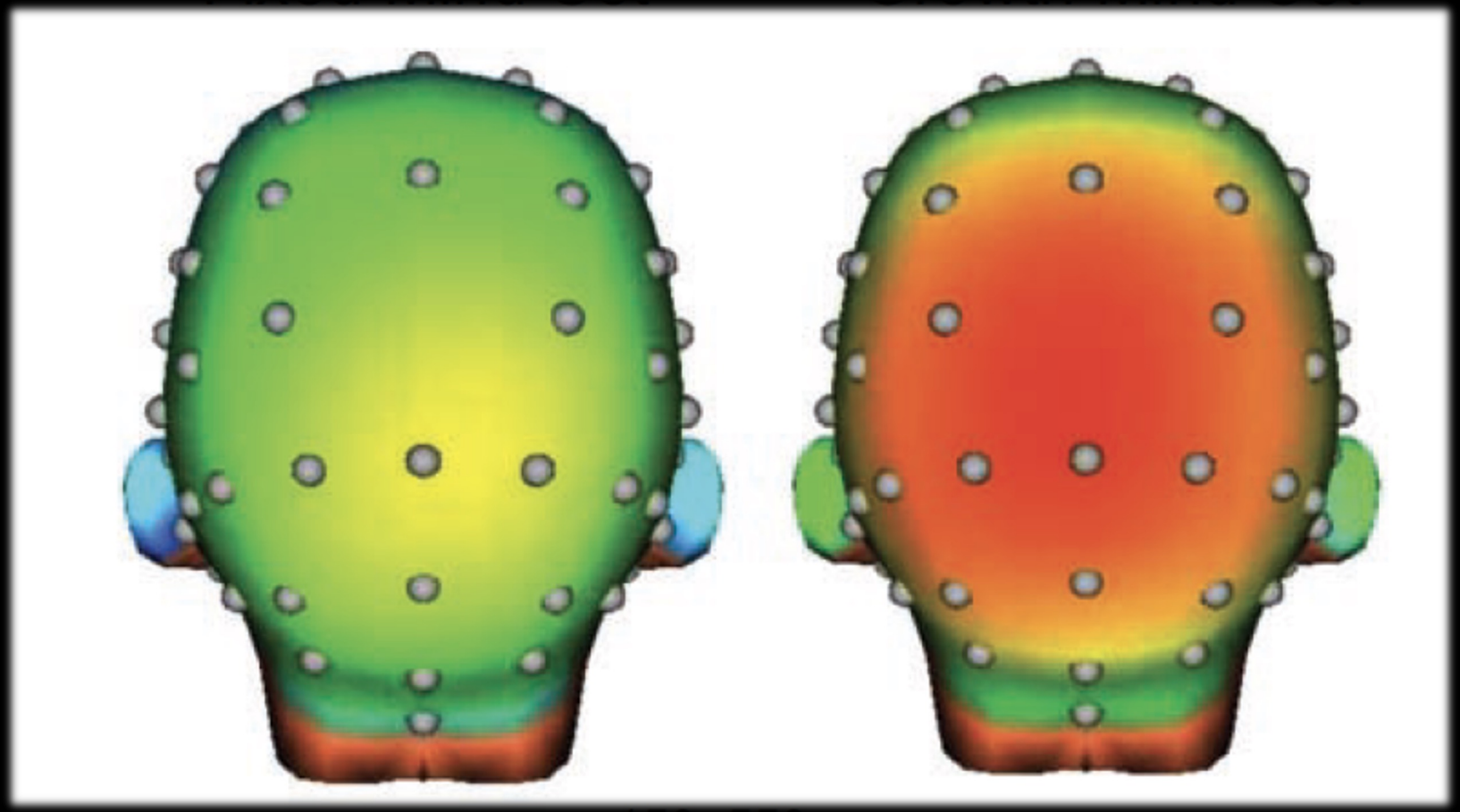
MINNET LØKKER

INF100

HØST 2026

Torstein Strømme



mindre tro på egen innsats ⟷ større tro på egen innsats

Moser, Schroder, Heeter, Moran, Lee 2011

MINNET

HP Laptop 14-em0924no 14" FHD

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

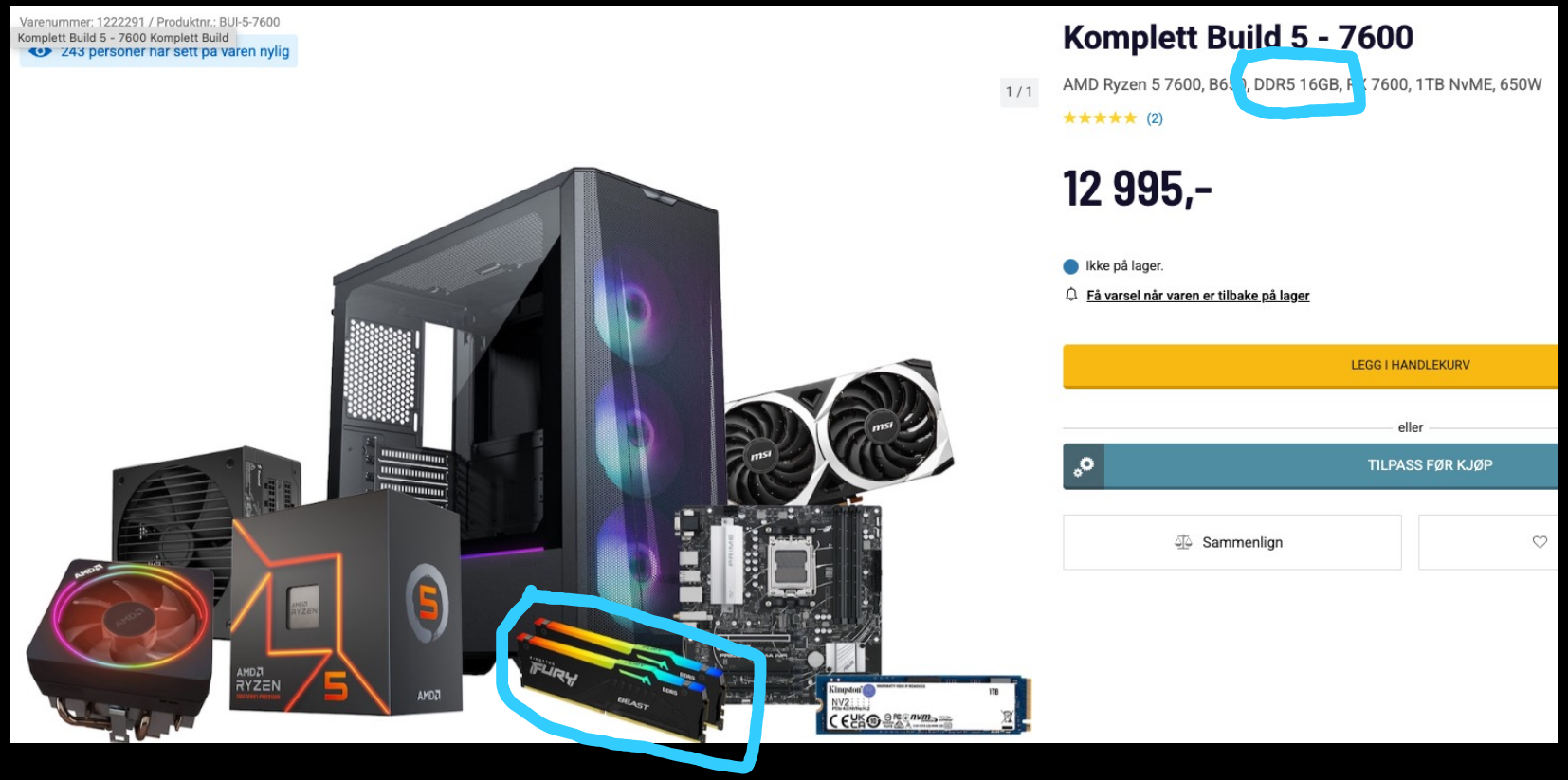
★★★☆☆ (3 anmeldelser)

SPAR 3 000,-

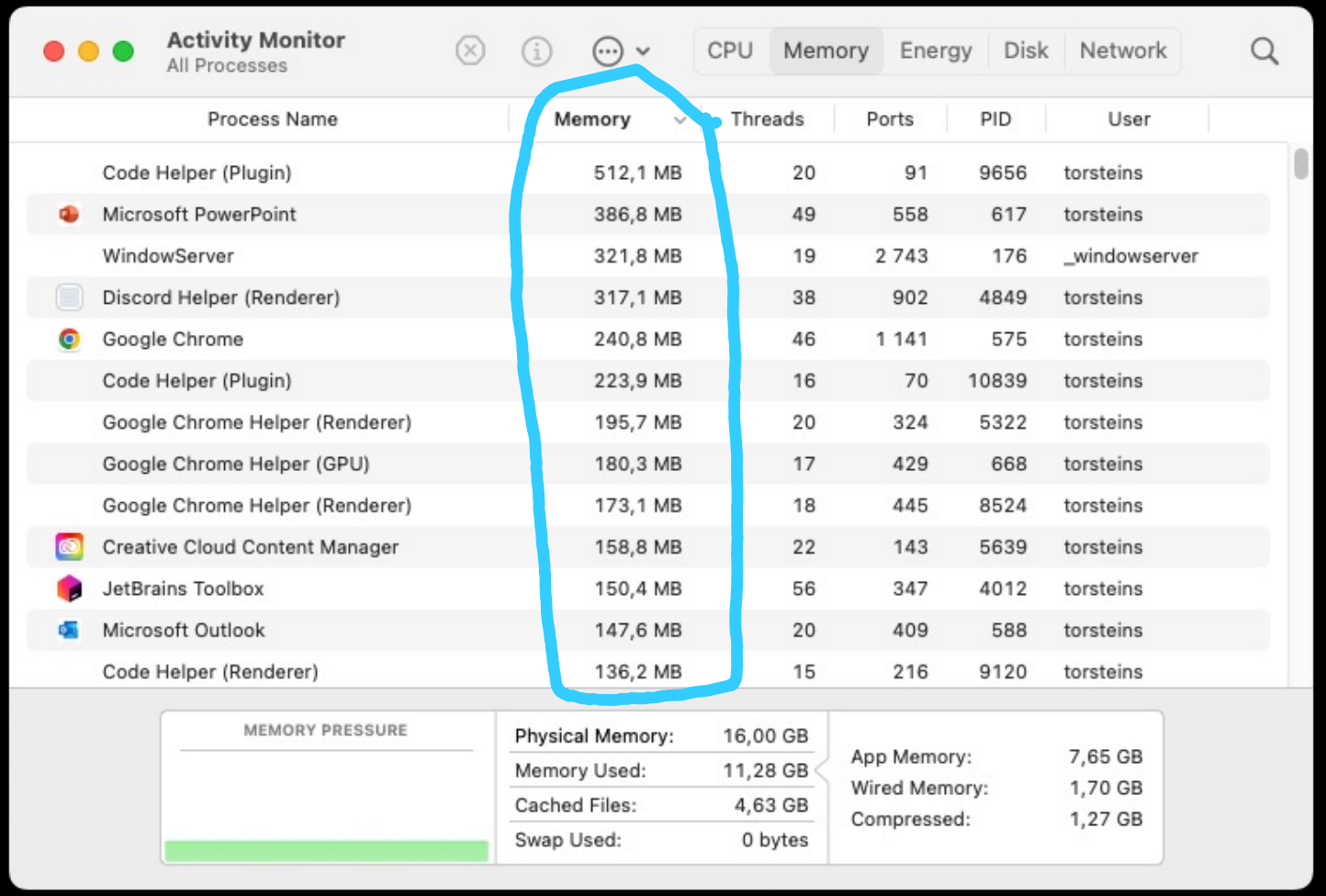
Lenovo Ideapad Slim 3 14" FHD (arctic grey)

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

8 990,-



Random Access Memory (RAM)


Activity Monitor
All Processes
CPU
Memory
Energy
Disk
Network
Process Name
Memory
Threads
Ports
PID
User
Code Helper (Plugin) 512,1 MB 20 91 9656 torsteins
Microsoft PowerPoint 386,8 MB 49 558 617 torsteins
WindowServer 321,8 MB 19 2 743 176 _windowserver
Discord Helper (Renderer) 317,1 MB 38 902 4849 torsteins
Google Chrome 240,8 MB 46 1 141 575 torsteins
Code Helper (Plugin) 223,9 MB 16 70 10839 torsteins
Google Chrome Helper (Renderer) 195,7 MB 20 324 5322 torsteins
Google Chrome Helper (GPU) 180,3 MB 17 429 668 torsteins
Google Chrome Helper (Renderer) 173,1 MB 18 445 8524 torsteins
Creative Cloud Content Manager 158,8 MB 22 143 5639 torsteins
JetBrains Toolbox 150,4 MB 56 347 4012 torsteins
Microsoft Outlook 147,6 MB 20 409 588 torsteins
Code Helper (Renderer) 136,2 MB 15 216 9120 torsteins
MEMORY PRESSURE
Physical Memory: 16,00 GB
Memory Used: 11,28 GB
Cached Files: 4,63 GB
Swap Used: 0 bytes
App Memory: 7,65 GB
Wired Memory: 1,70 GB
Compressed: 1,27 GB

MINNET

- En lagringsplass for 0'er 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*

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

MINNET

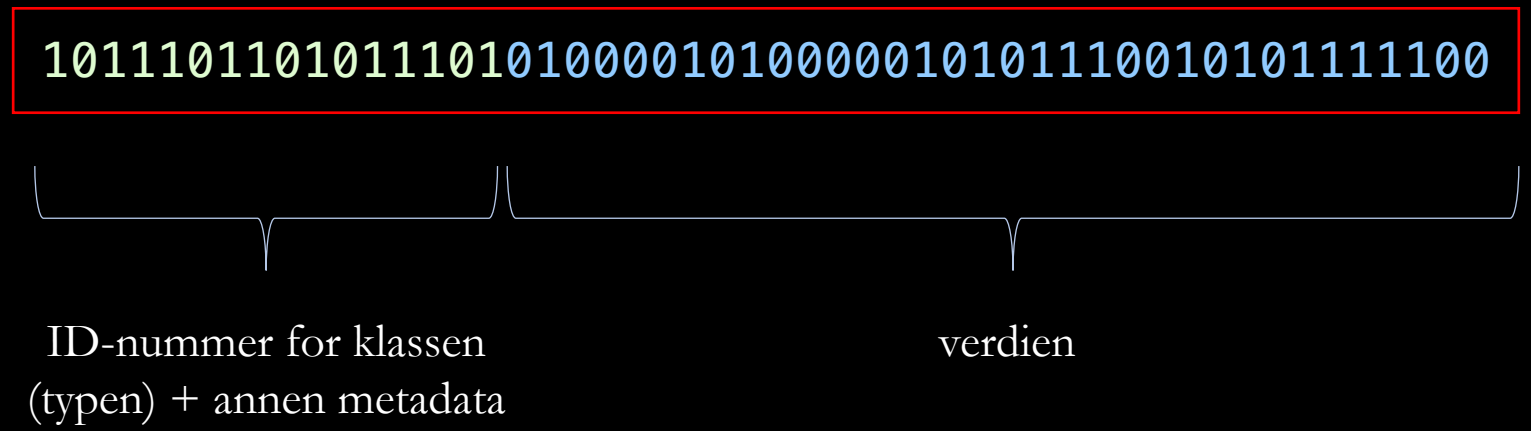
et objekt

…01110110101110101000010100000101011100101011111000111010101010101000…

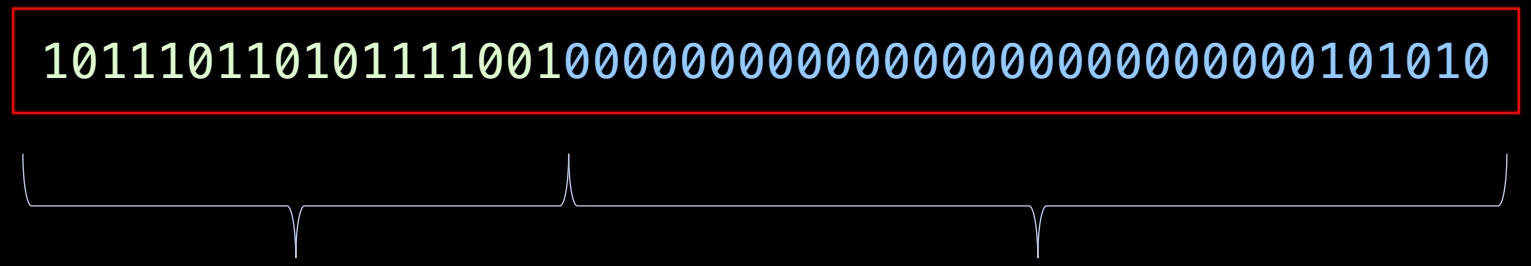
x

en variabel

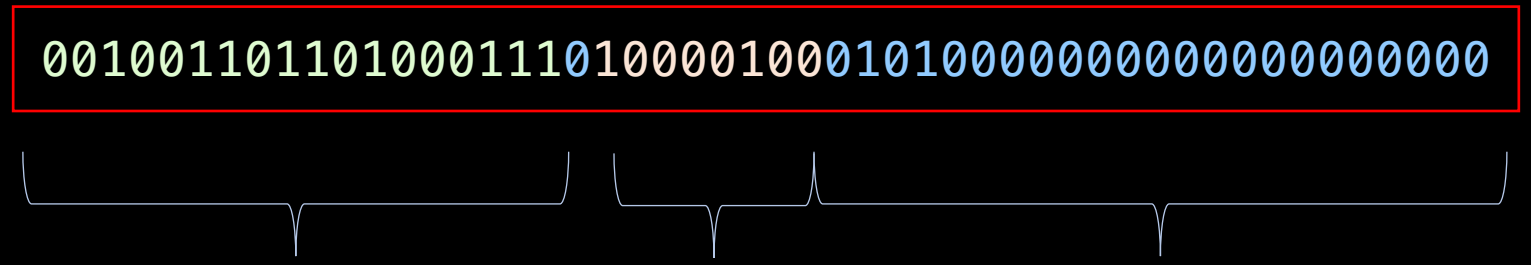
OBJEKT


10111011010111010100001010000010101110010101111100
ID-nummer for klassen
(typen) + annen metadata
verdien

42

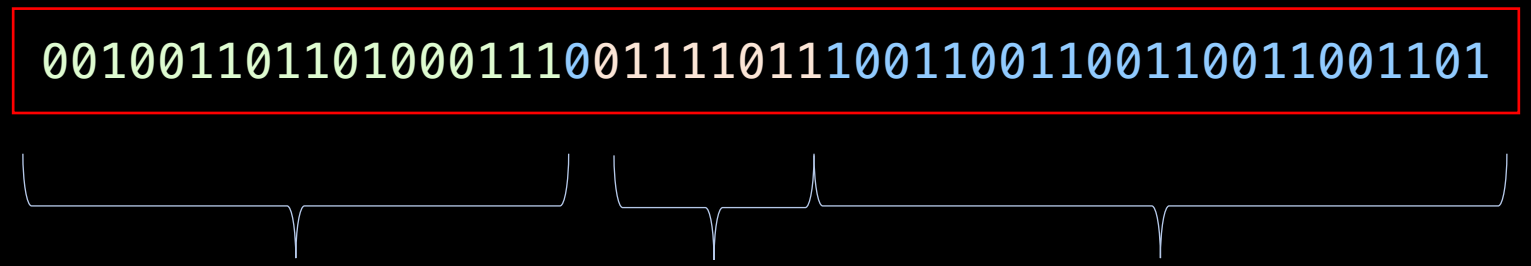

10111011010111100100000000000000000000000101010
«dette objektet er en int»
42 i totallsystemet

42.0



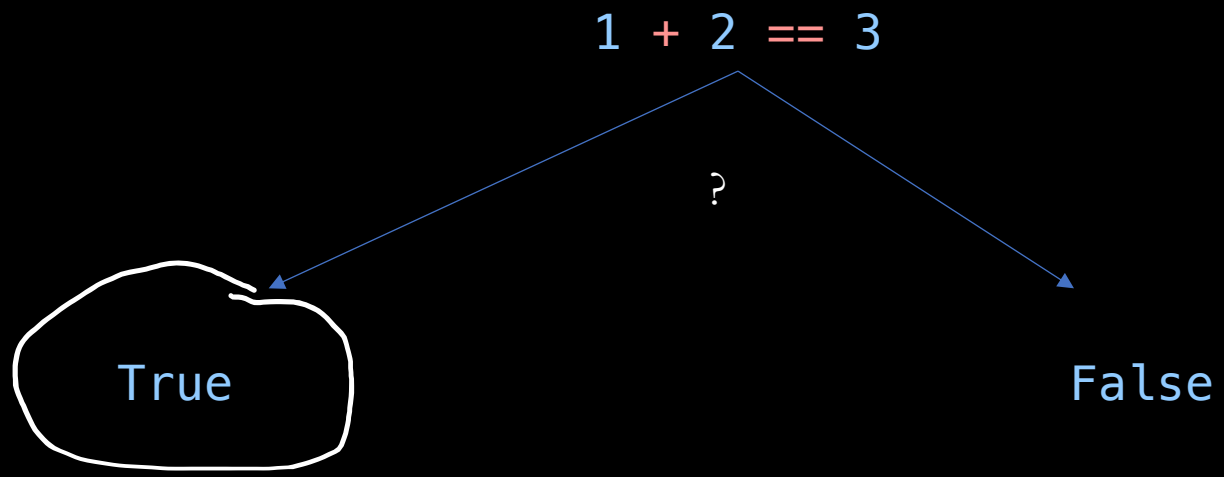
$$2^{132\ -127} \cdot (1\ +\ 0.3125) \qquad \Rightarrow 42.0$$

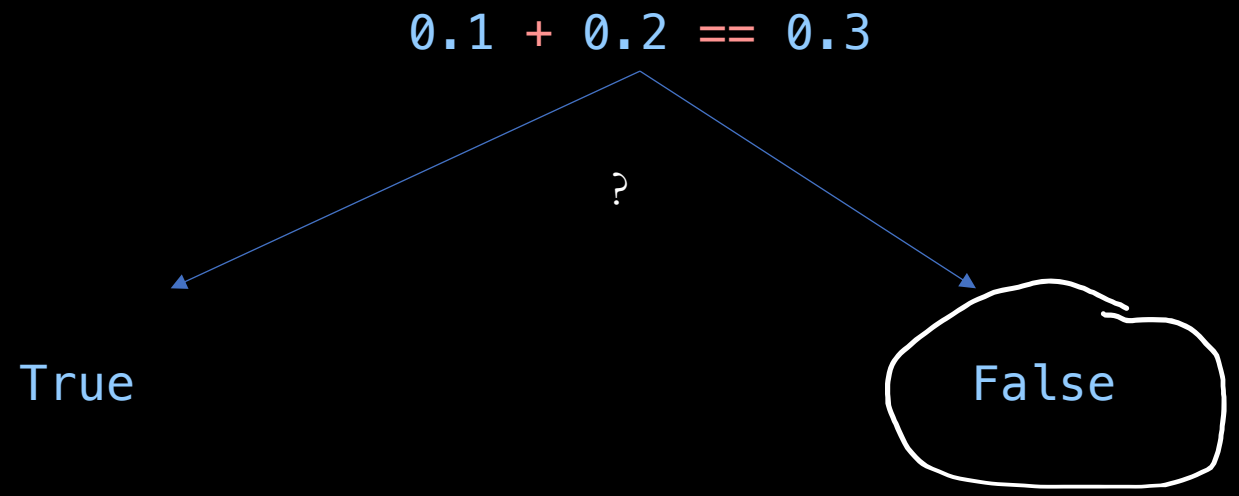
0.1

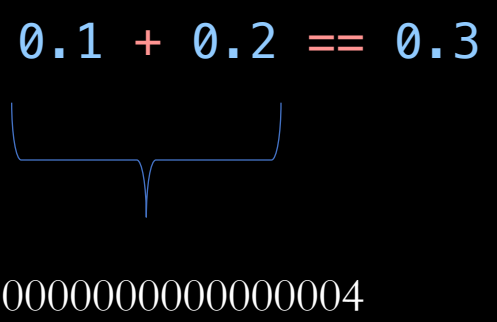


$$2^{123-127} \cdot (1 + 0.60000002384185791)$$

$$\Rightarrow 0.100000001490116119384765625$$


1 + 2 == 3
?
True
False


0.1 + 0.2 == 0.3
?
True
False


0.1 + 0.2 == 0.3
0.30000000000000004

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

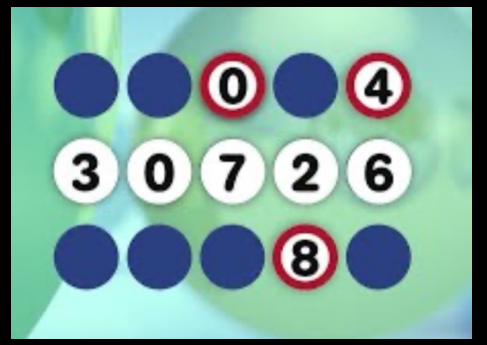
print vs return

assert

LØKKER

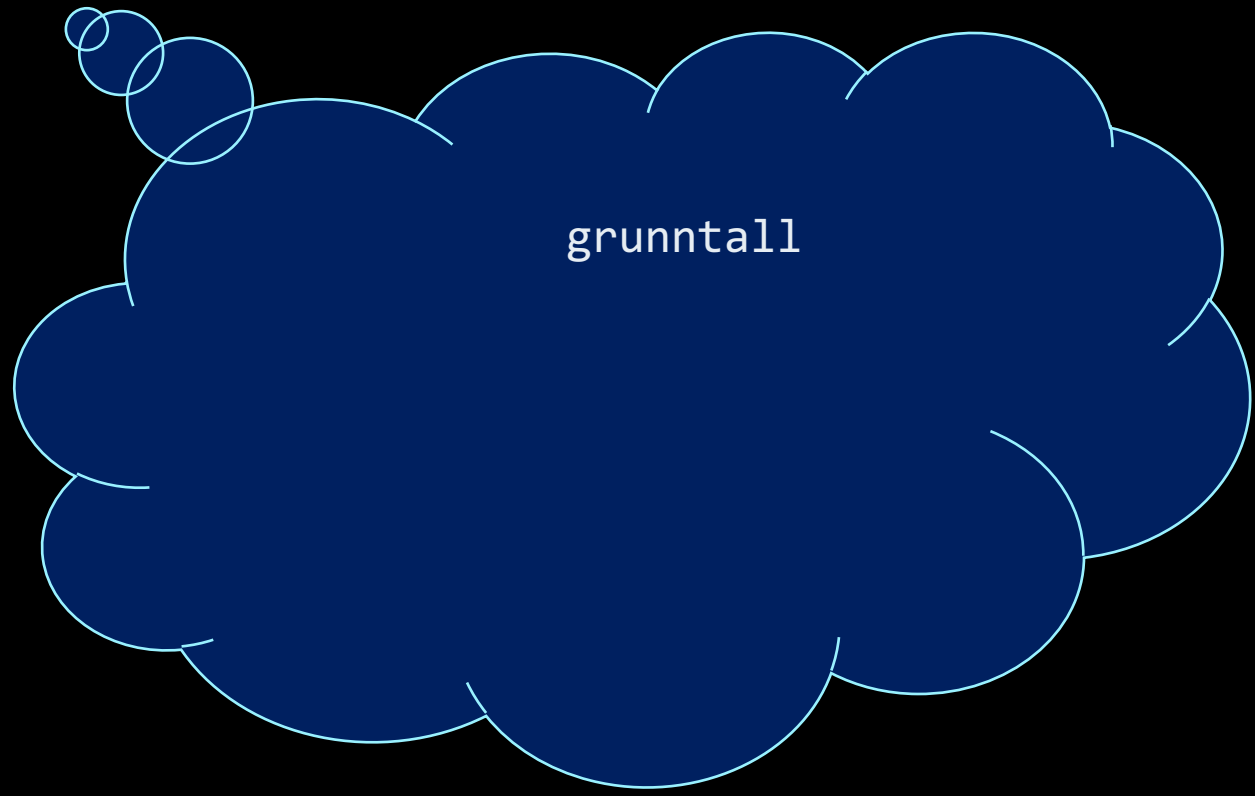

2 260 000
1 627 000
994 000
746 000
565 000
452 000
0
4
3
0
7
2
6
8

OPPGAVE

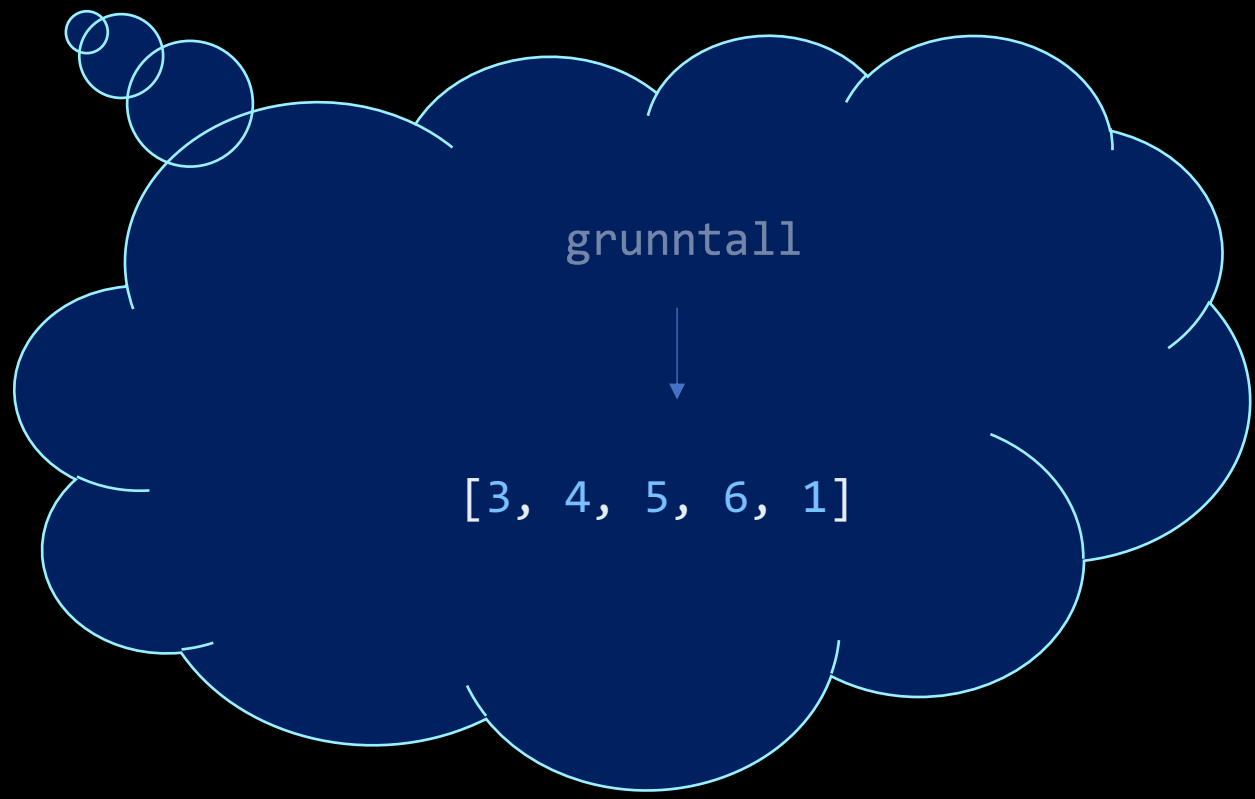


- Anta at du har en liste med fem «grunntall»
  - for eksempel `[3, 0, 7, 2, 6]`
- Skriv et program som fungerer som en «kunstig intelligens» som hjelper til med å spille Joker
- For hvert grunntall, fra først i listen til sist i listen: skriv ut «opp» eller «ned» avhengig av hva som er lurest å gjøre.

```
for num in grunntall:
    if num <= 4:
        print('opp')
    else:
        print('ned')
```



```
for num in grunntall:
    if num <= 4:
        print('opp')
    else:
        print('ned')
```



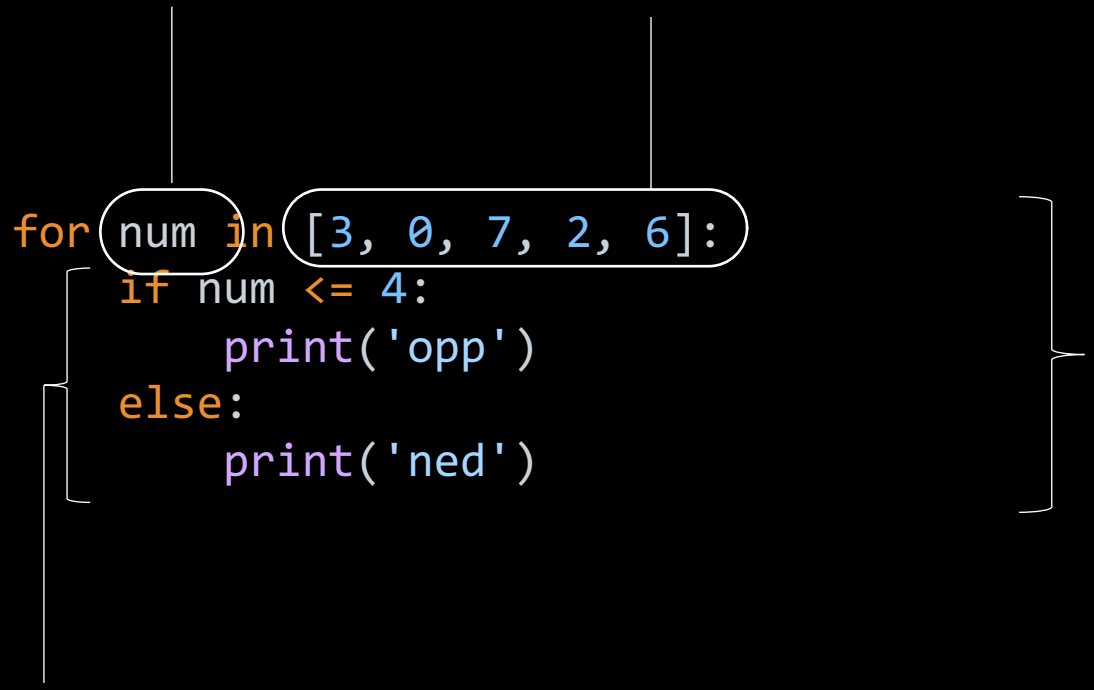
```
for num in grunntall:
    if num <= 4:
        print('opp')
    else:
        print('ned')
```



```
for num in [3, 0, 7, 2, 6]:
    if num <= 4:
        print('opp')
    else:
        print('ned')
```

iterand. variabel som viser til et nytt element i den iterable for hver iterasjon av løkken

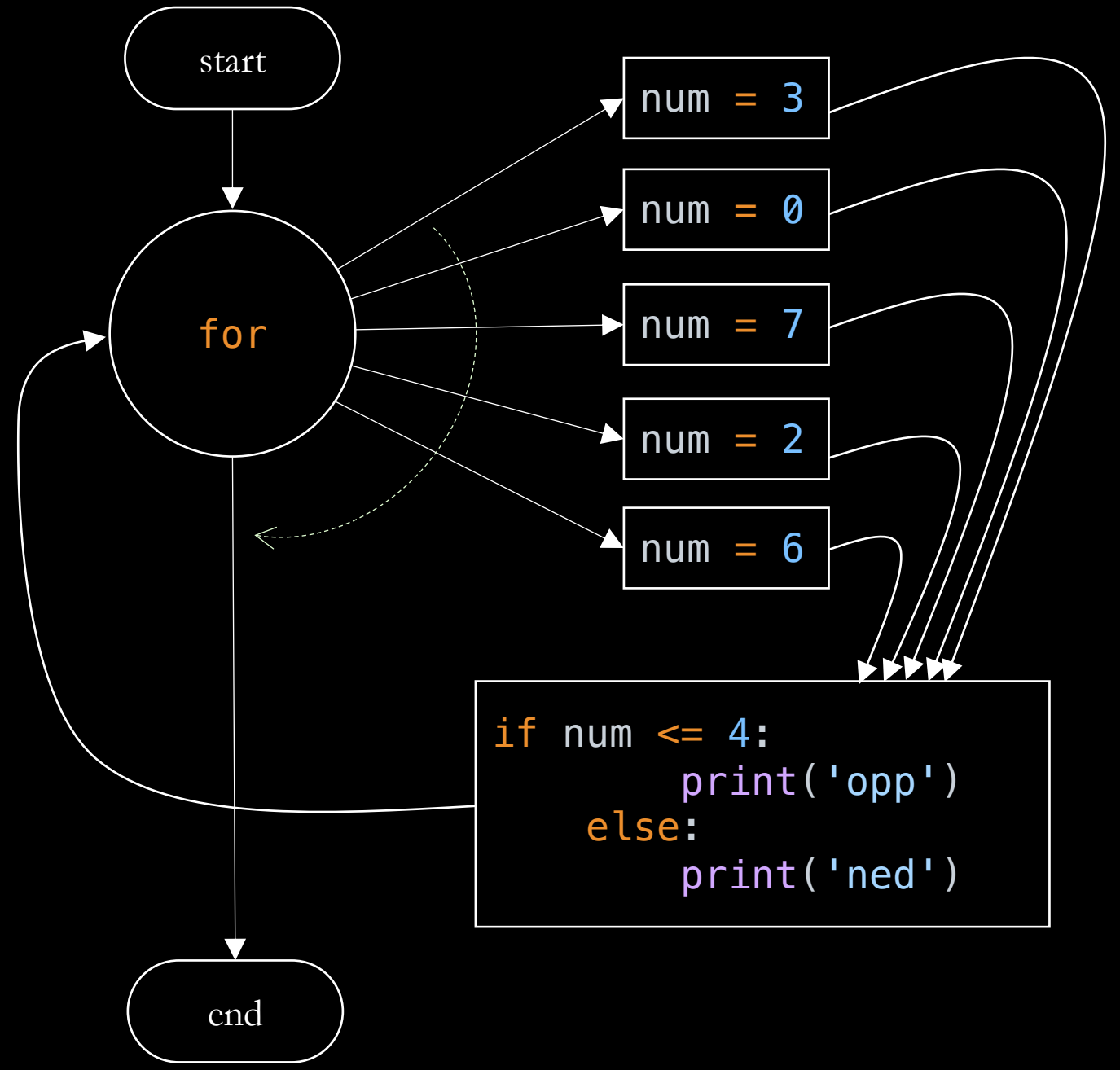
iterabel. en samling av elementer (f. eks. en liste) som skal itererers over



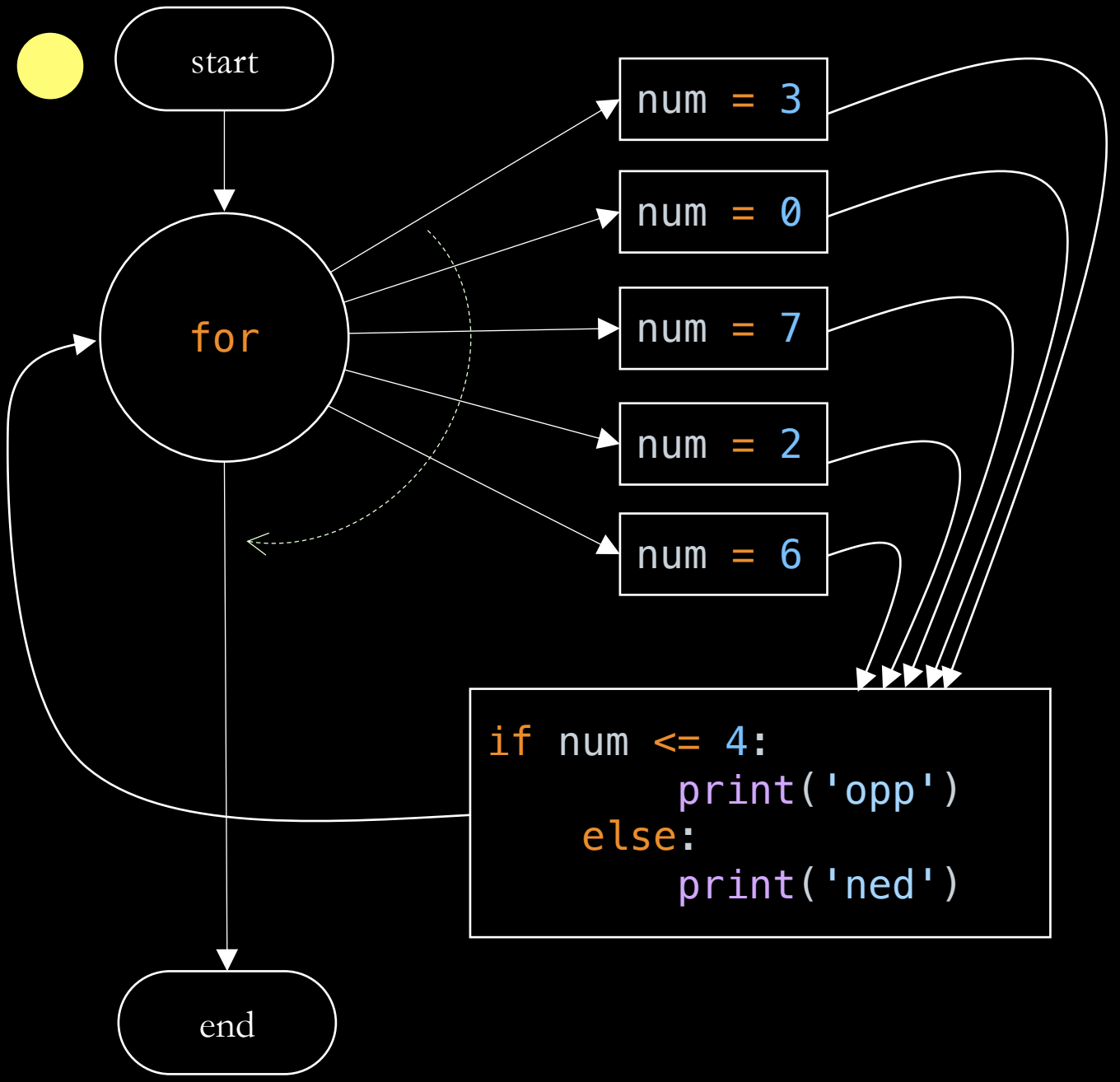
for-løkke. en setning med en løkkekropp som utføres én gang for hvert element i en iterabel.

løkkekropp. innrykket blokk av setninger som utføres én gang for hvert element i iterabelen.

```
for num in [3, 0, 7, 2, 6]:
    if num <= 4:
        print('opp')
    else:
        print('ned')
```



```
for num in [3, 0, 7, 2, 6]:
    if num <= 4:
        print('opp')
    else:
        print('ned')
```



HVA PRINTER DETTE PROGRAMMET?

```
print('start')
for x in [0, 1, 2, 3]:
    print('loop', end=' ')
    print(x)
print('end')
```

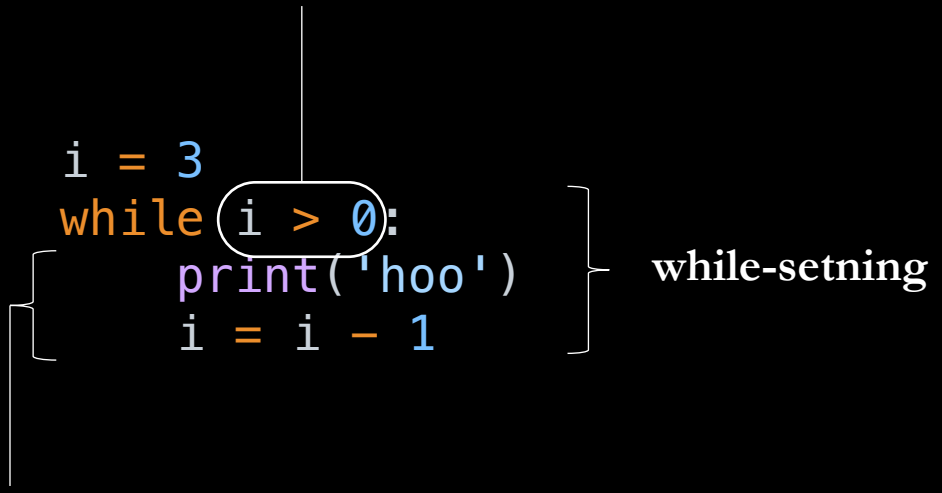
HVA PRINTER DETTE PROGRAMMET?

```
x = 99
print(x)
for x in [3, 3, 2]:
    print('loop', end=' ')
    print(x)
print('huzzah')
print(x)
```

hoo
hoo
hoo

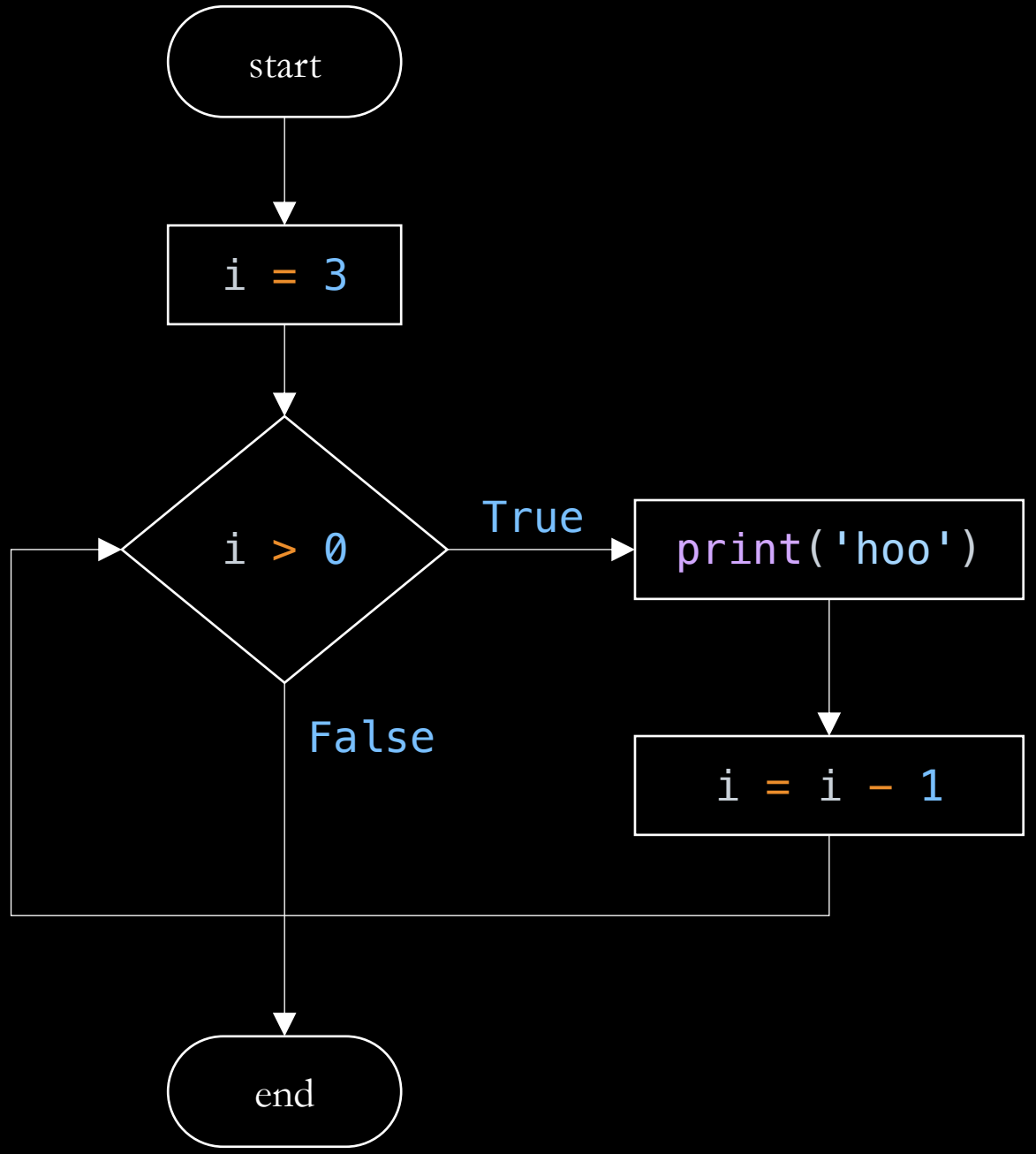
```
print('hoo')
print('hoo')
print('hoo')
```

betingelse. et uttrykk som evaluerer til en boolsk verdi (True eller False)

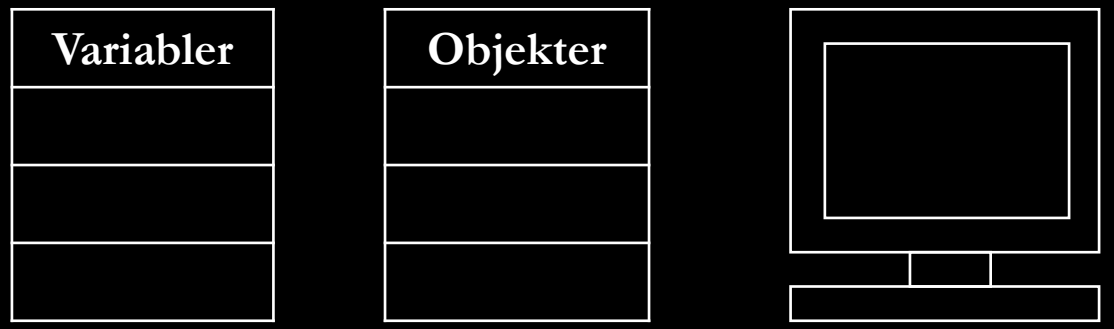


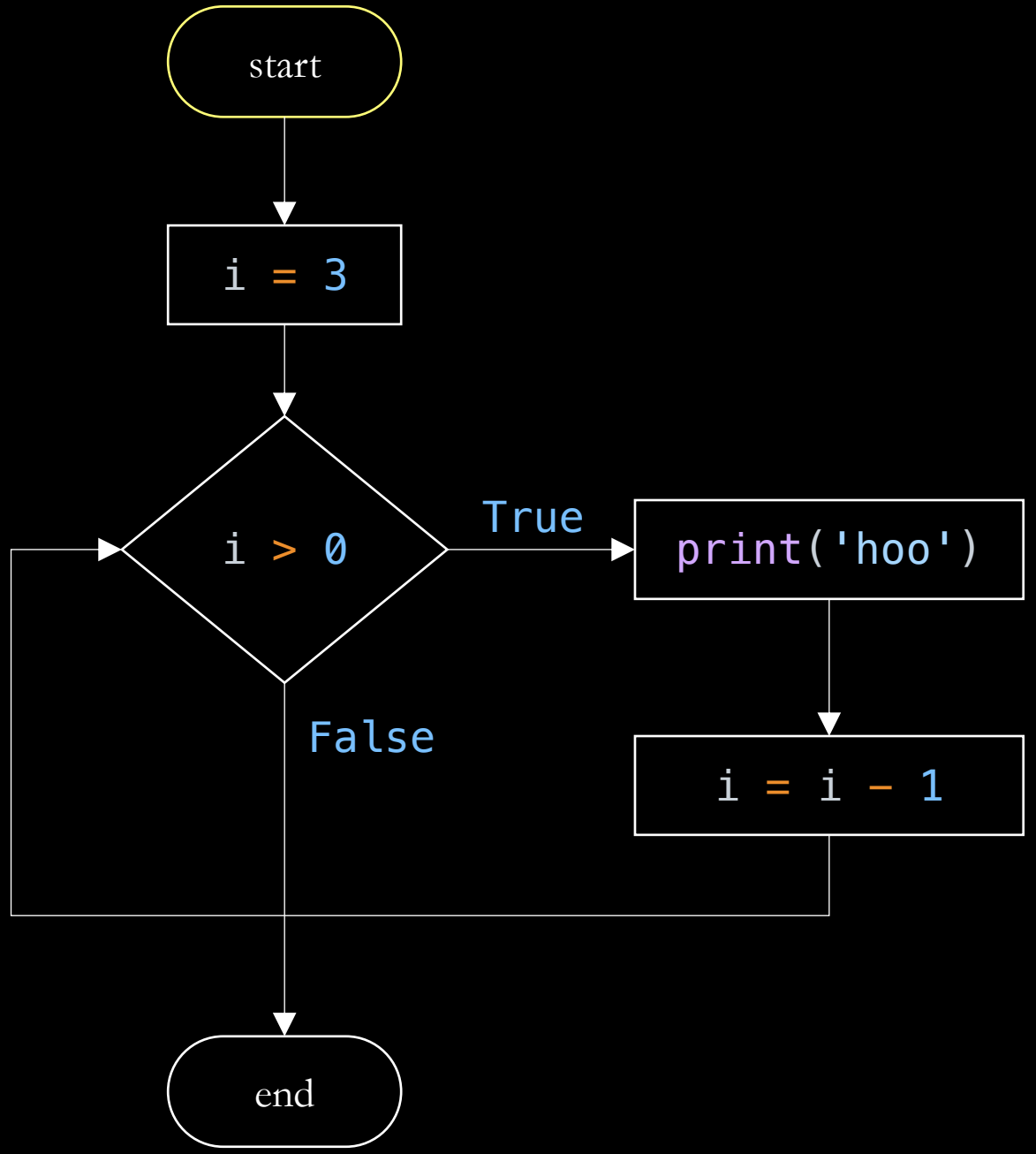
løkkekropp. innrykket blokk av setninger som utføres så lenge betingelsen er True.

```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```

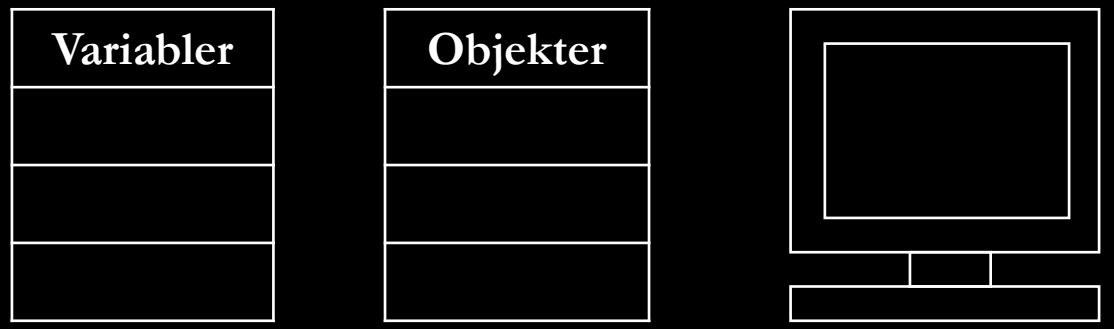


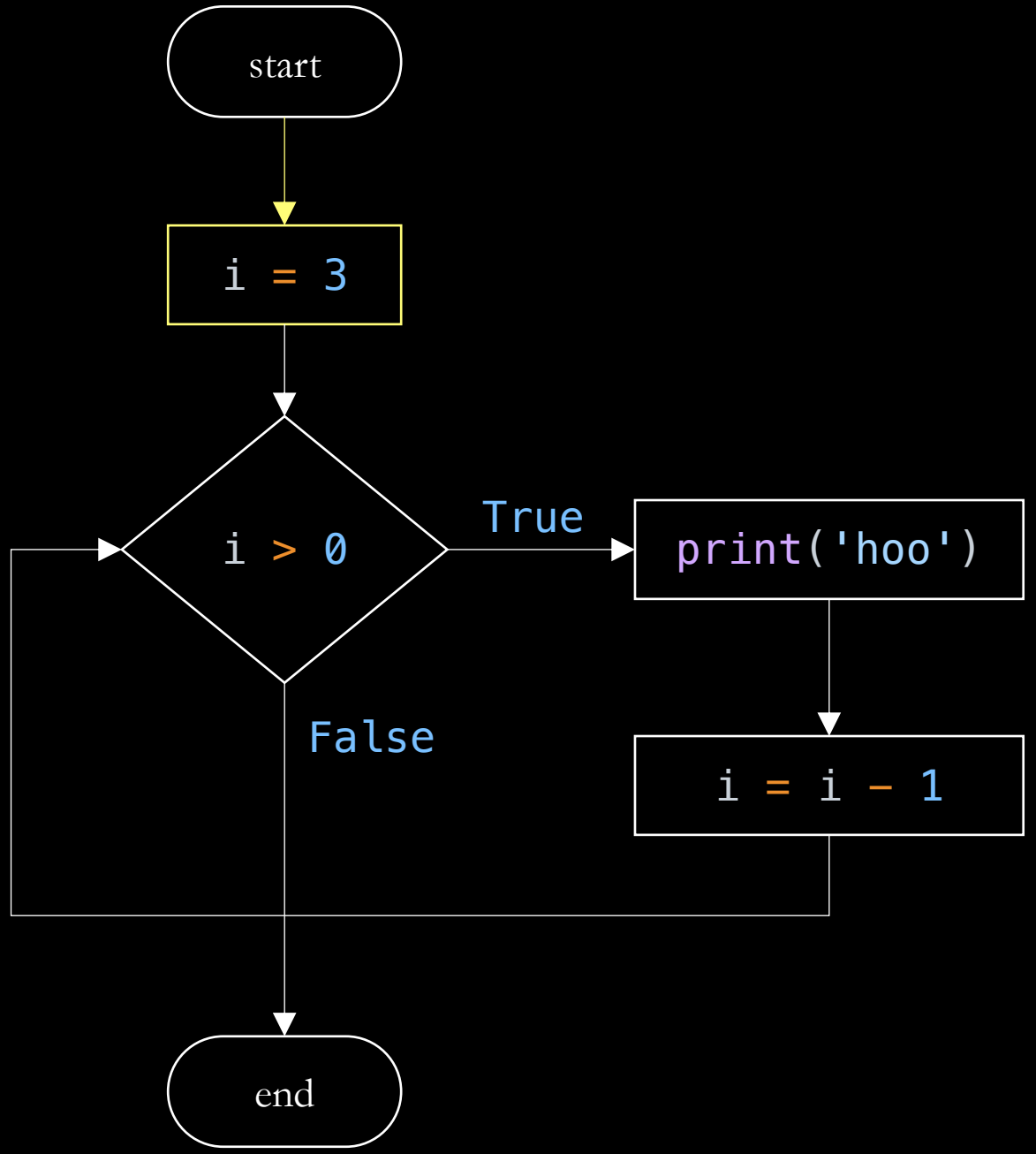
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



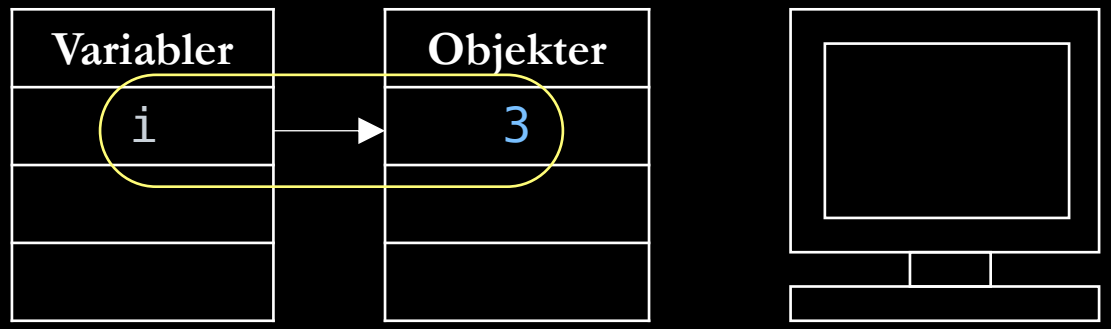


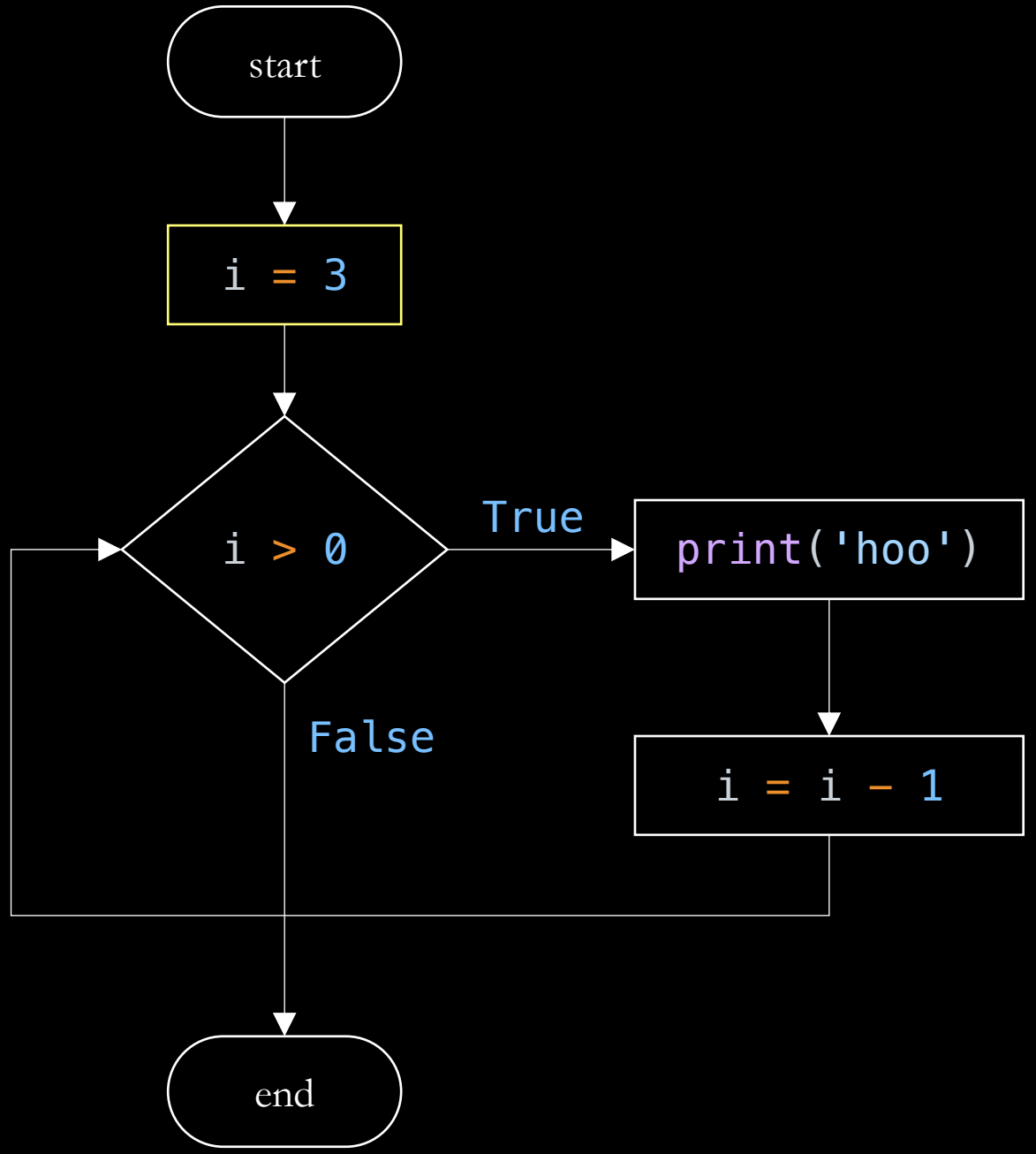
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



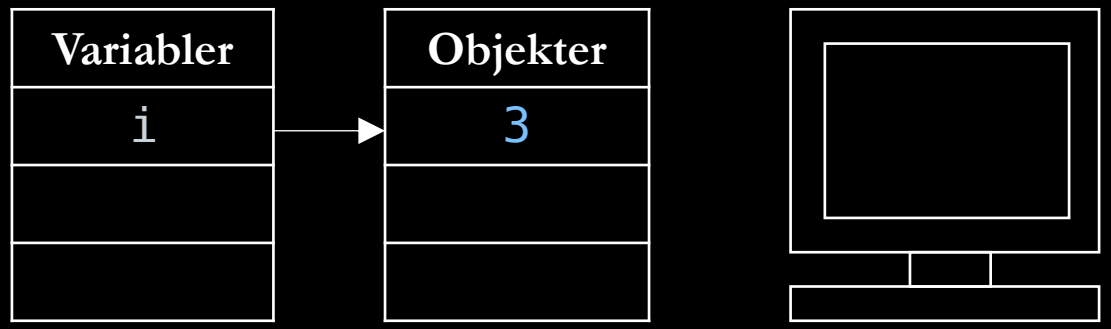


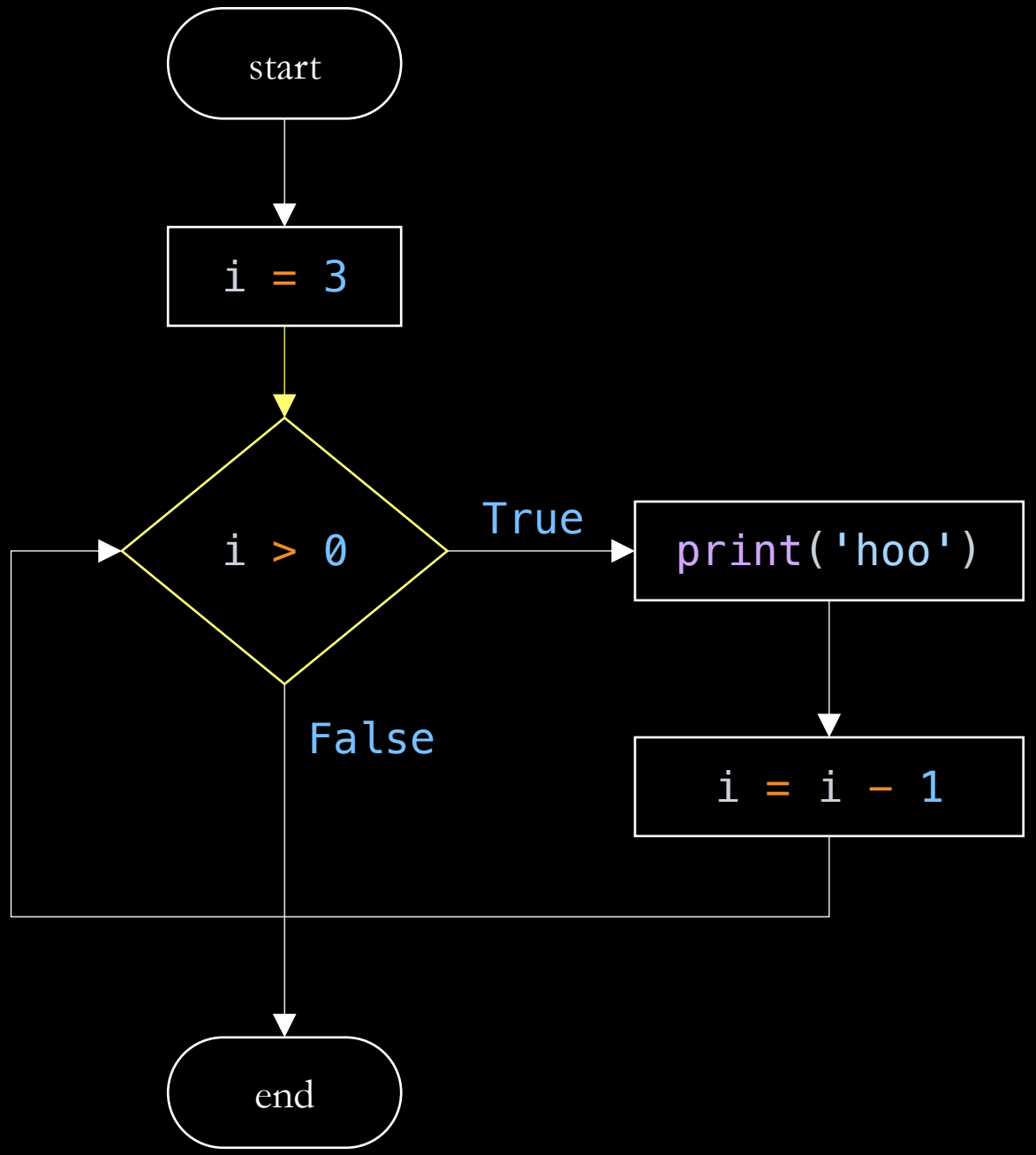
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```

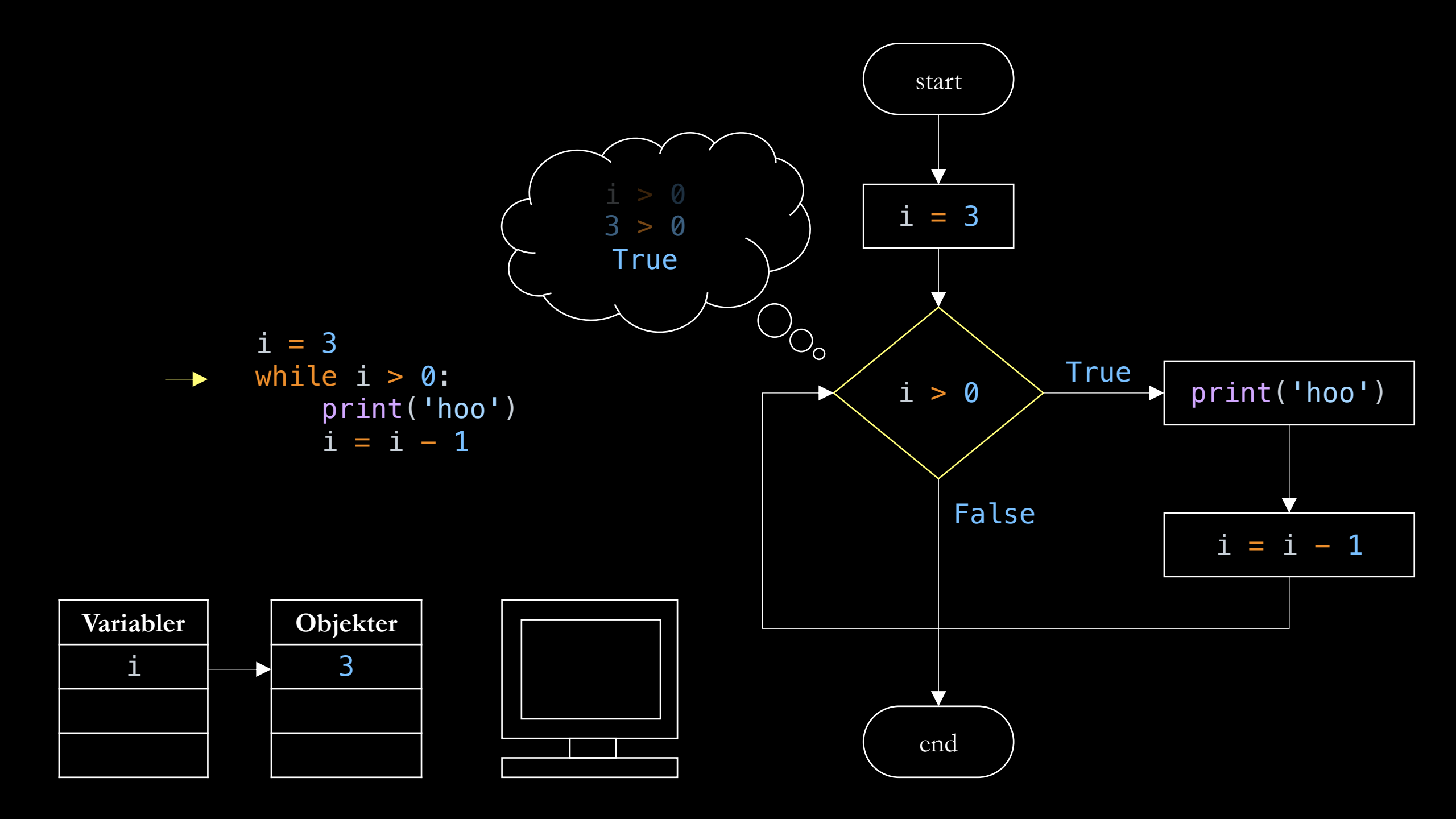


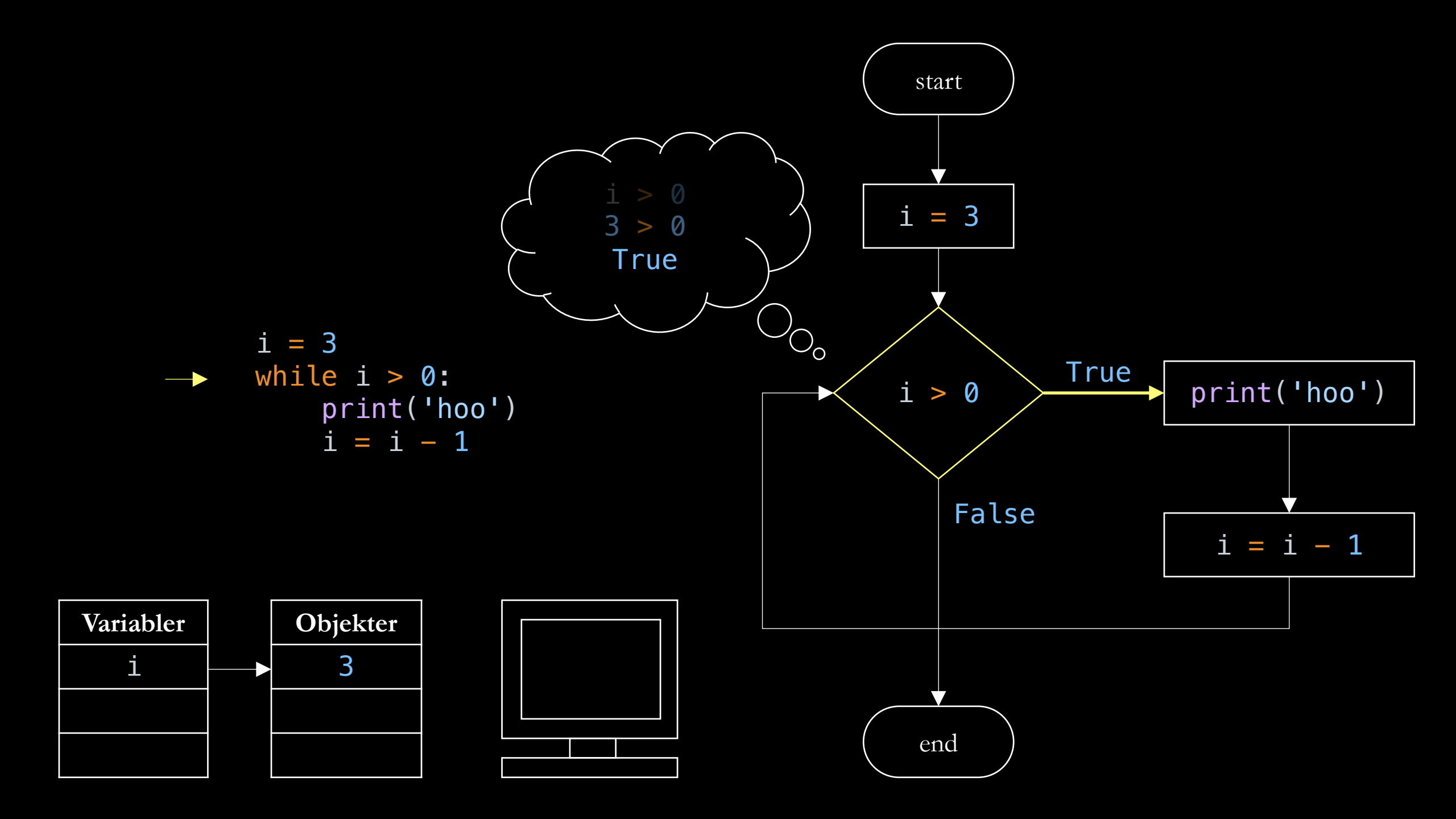


```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```

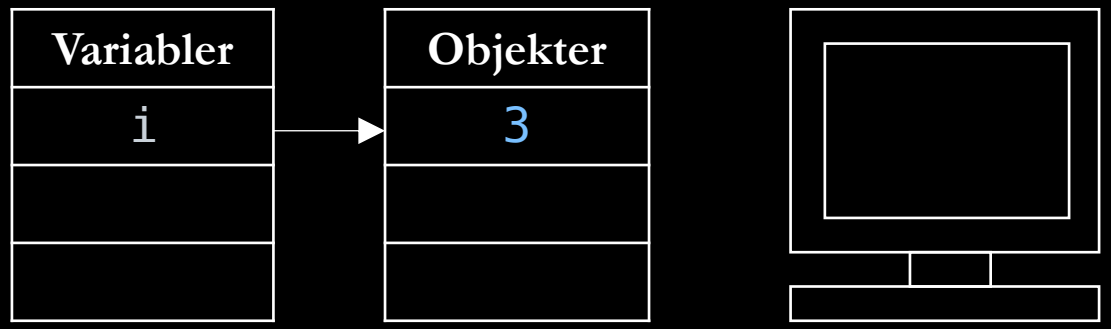


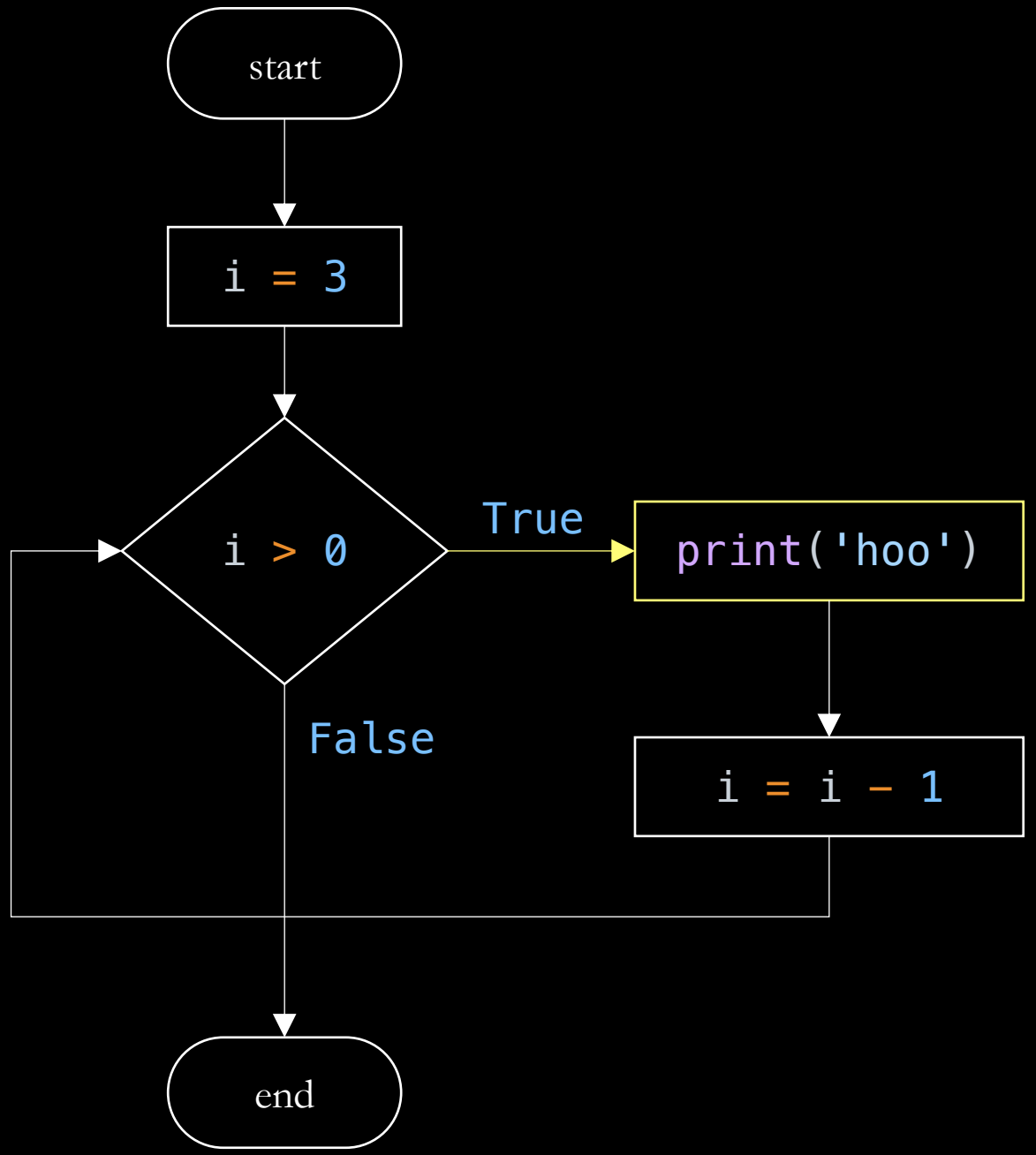



start
i = 3
i > 0
True
False
print('hoo')
i = i - 1
end
i > 0
3 > 0
True
i = 3
while i > 0:
print('hoo')
i = i - 1
Variabler
i
Objekter
3

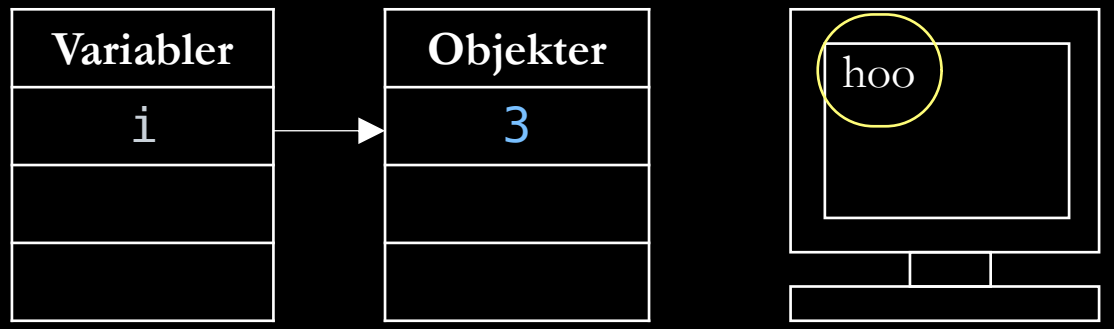

start
i = 3
i > 0
True
False
print('hoo')
i = i - 1
end
i > 0
3 > 0
True
i = 3
while i > 0:
print('hoo')
i = i - 1
Variabler
i
Objekter
3

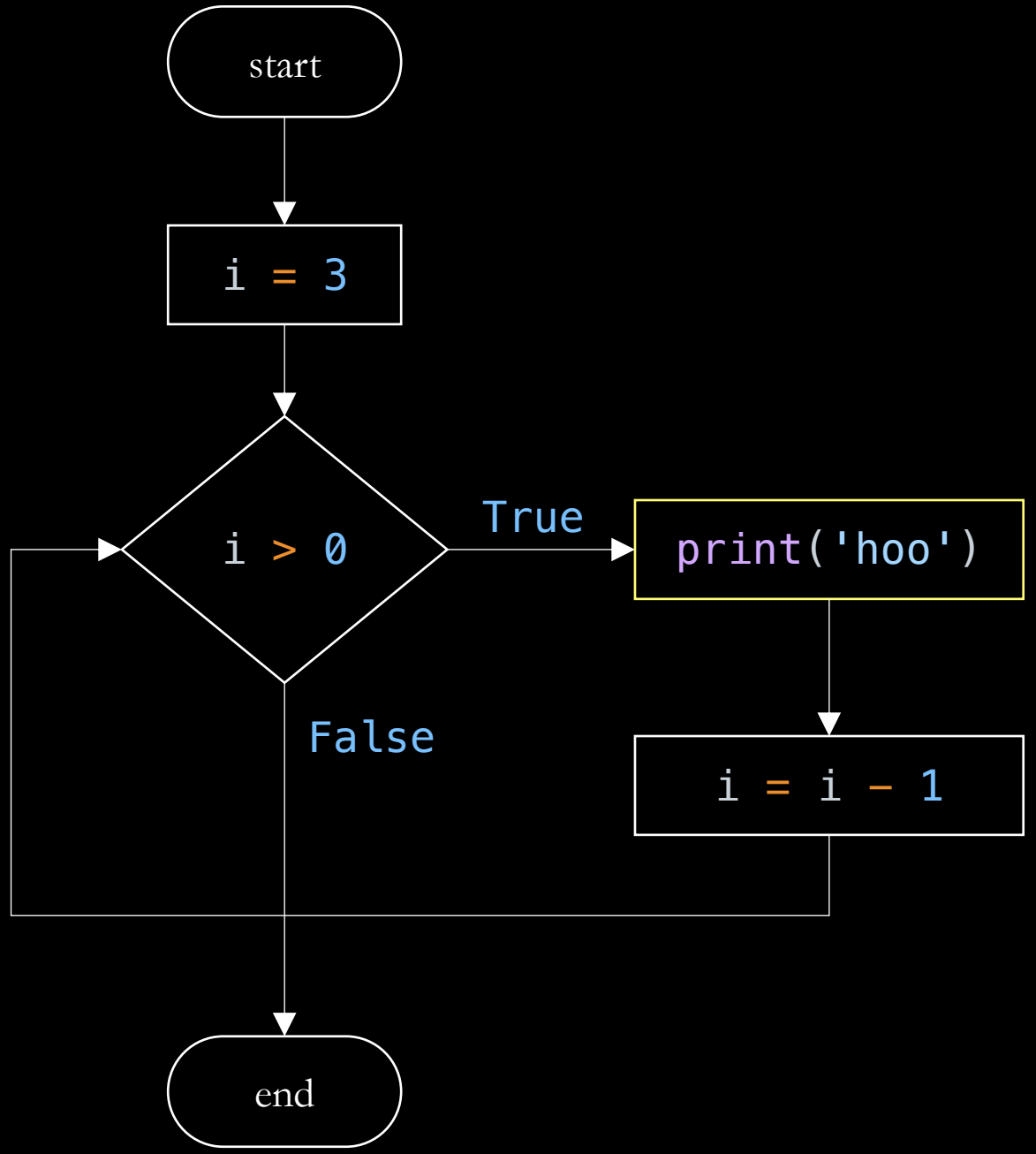
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



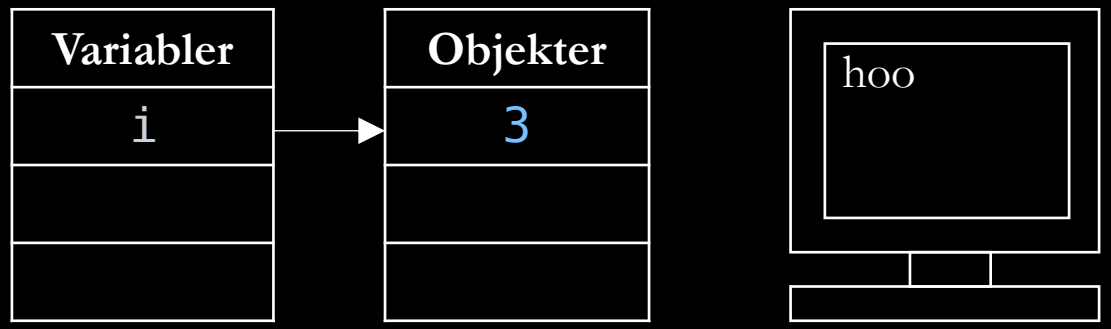


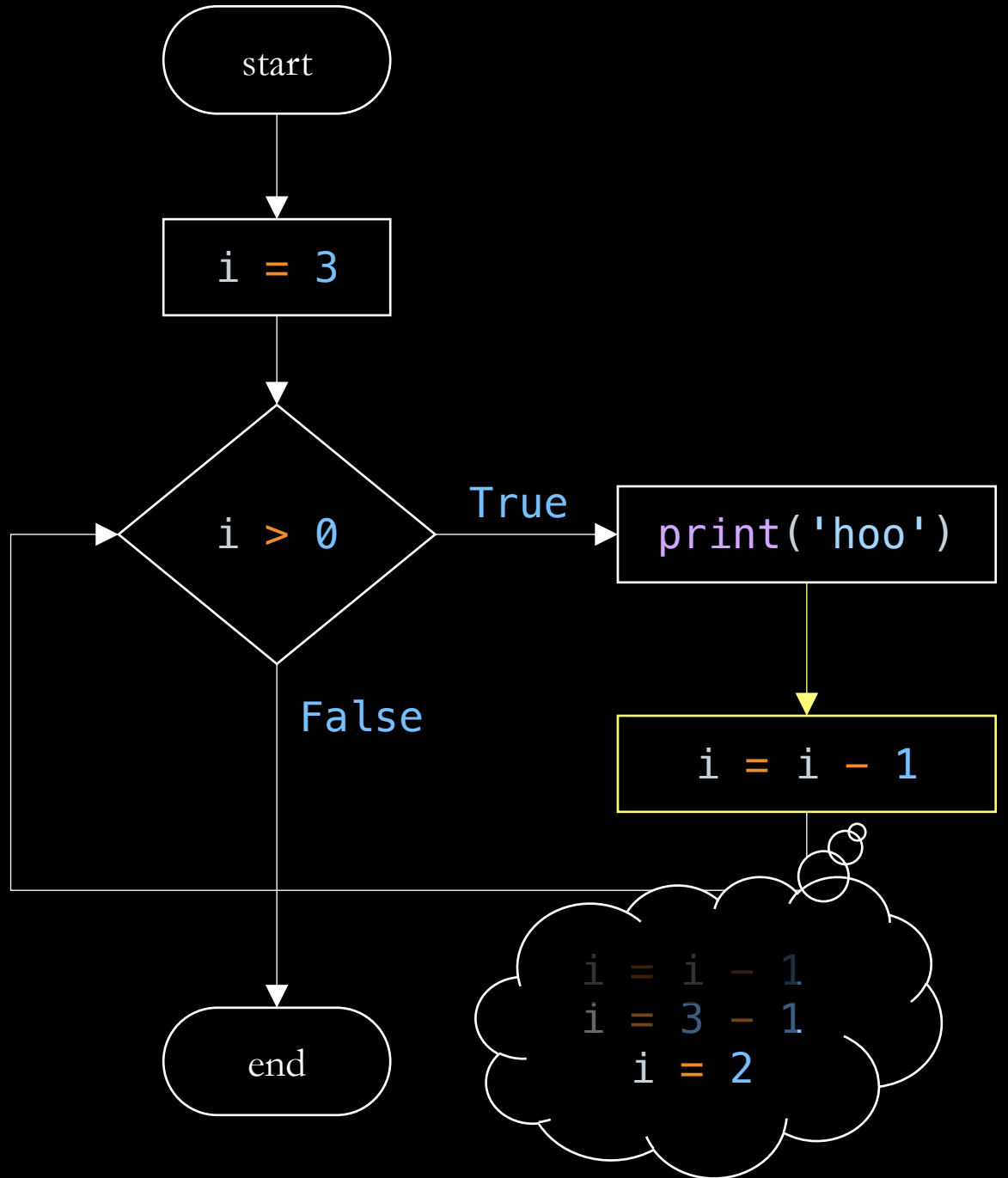
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



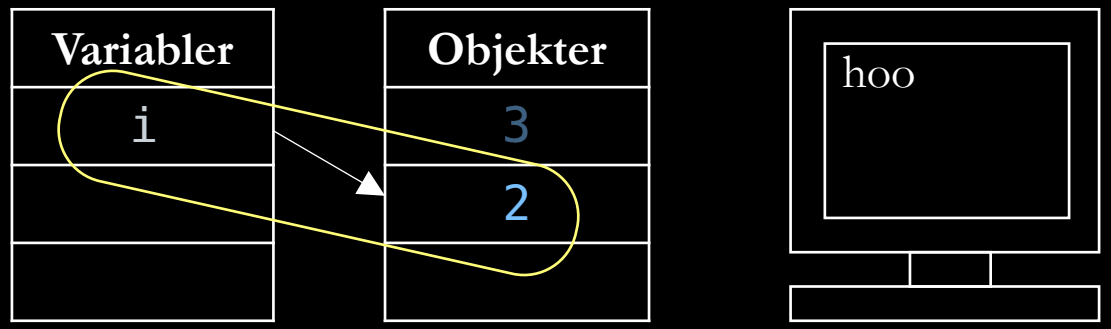


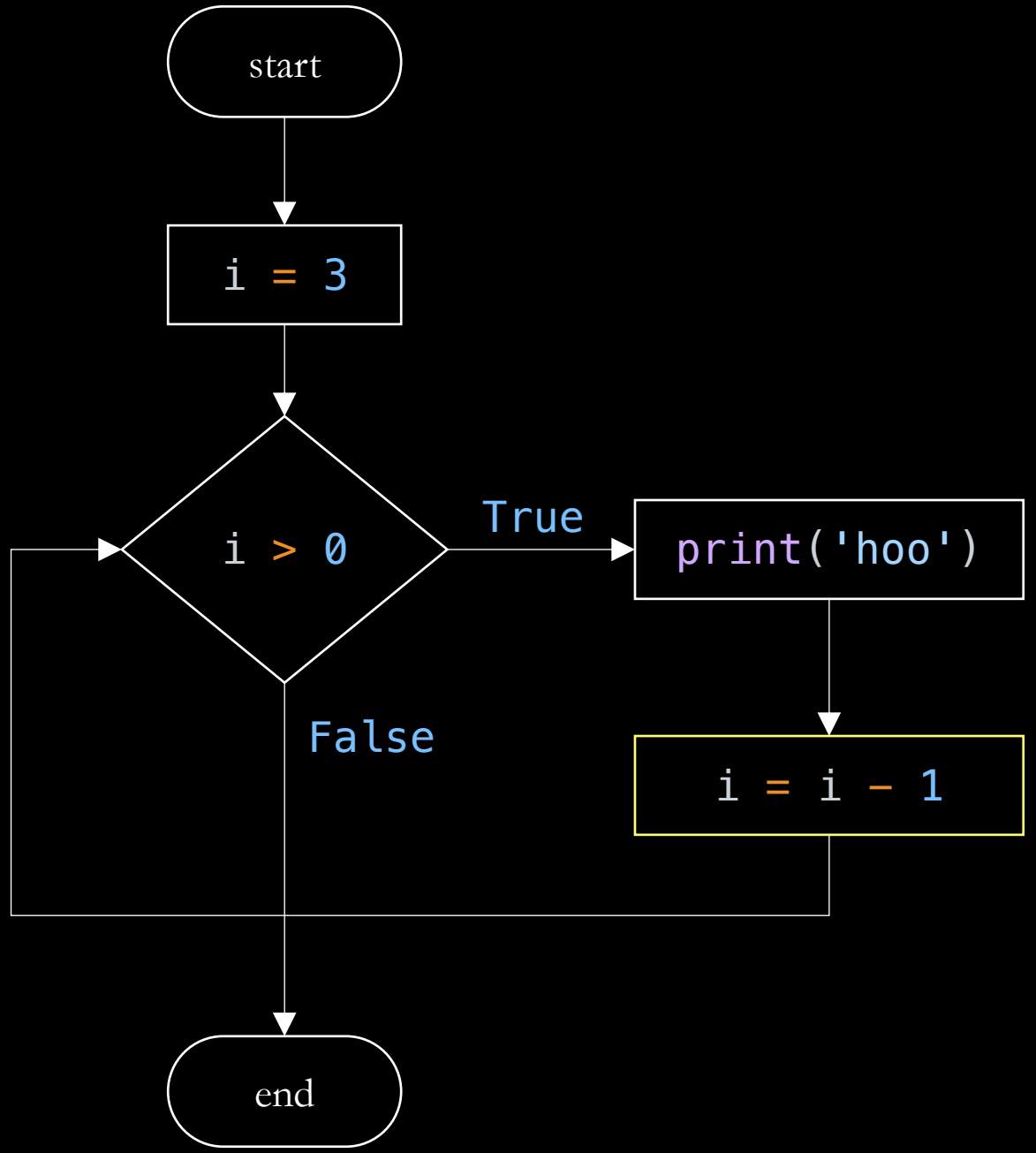
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



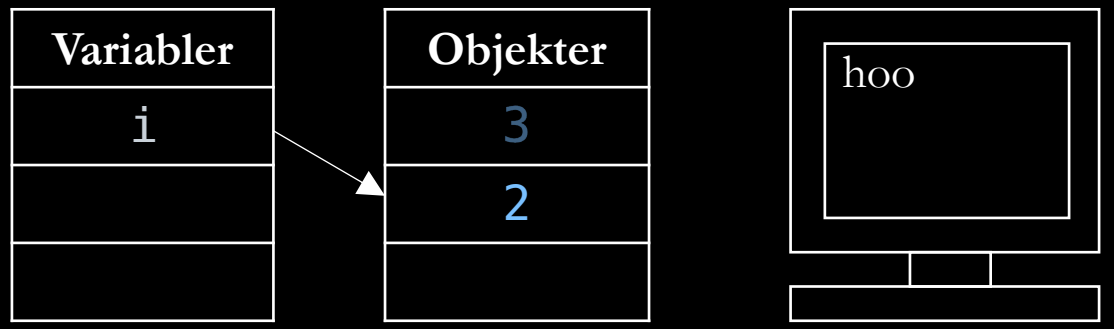


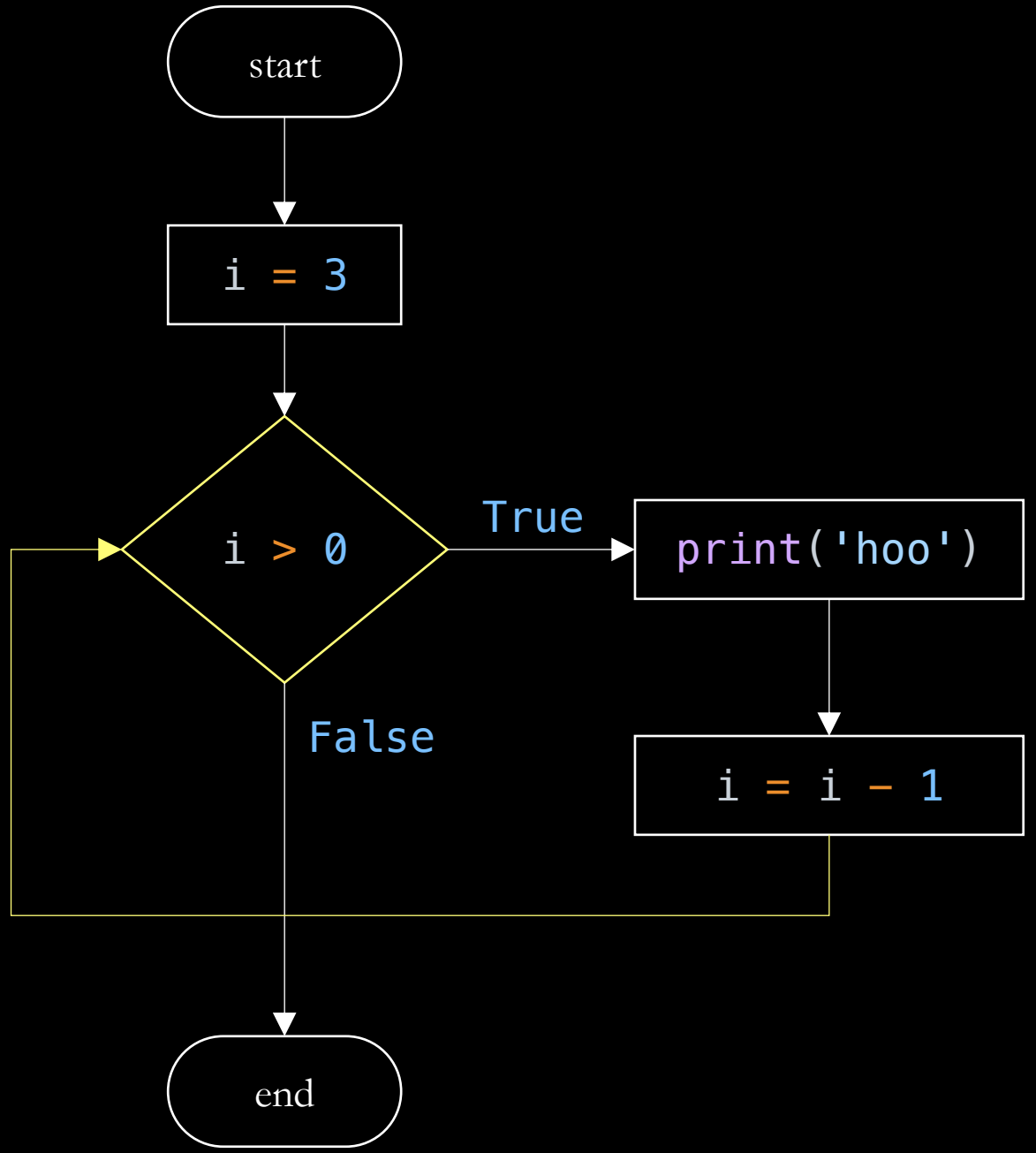
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```

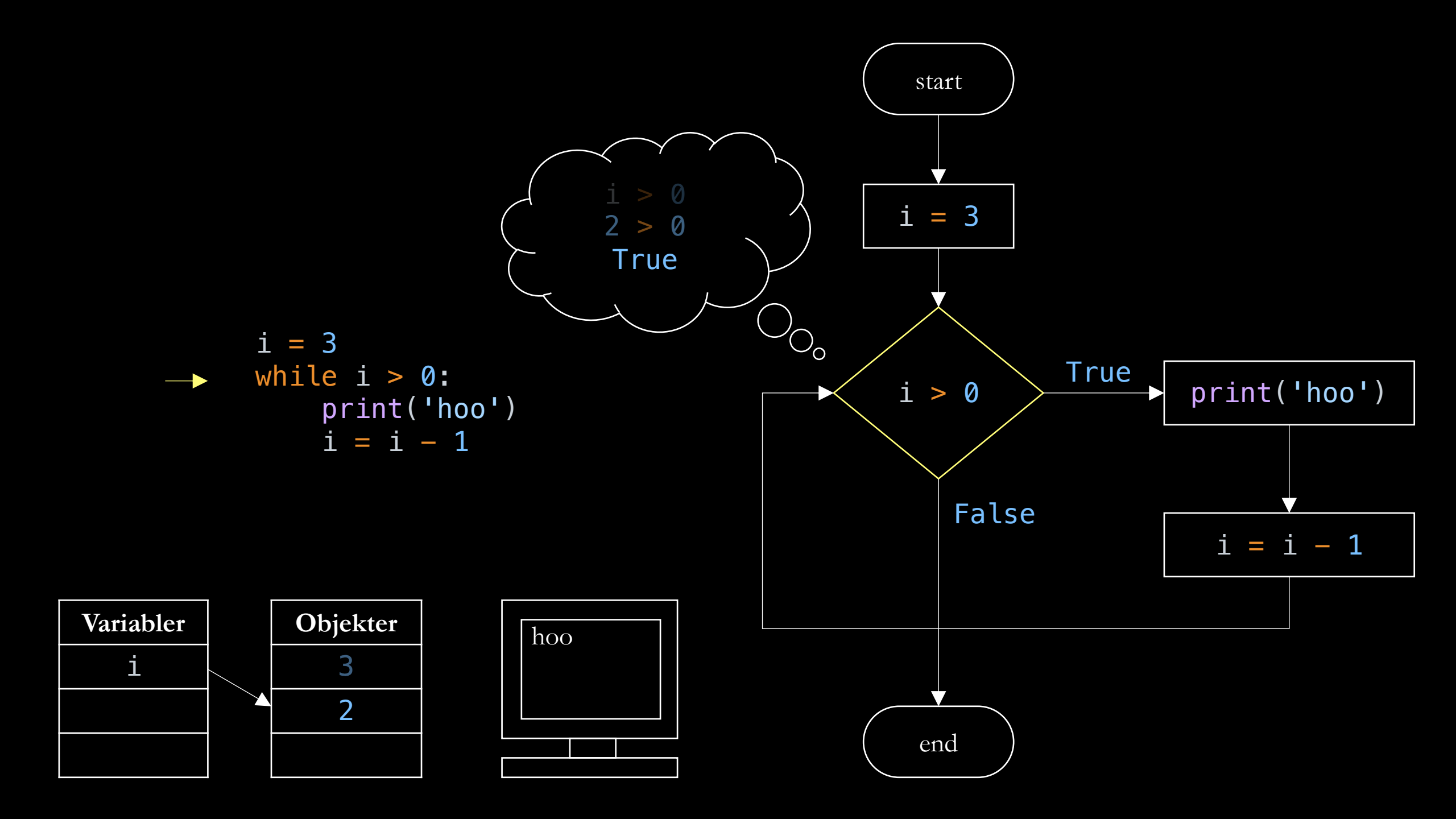


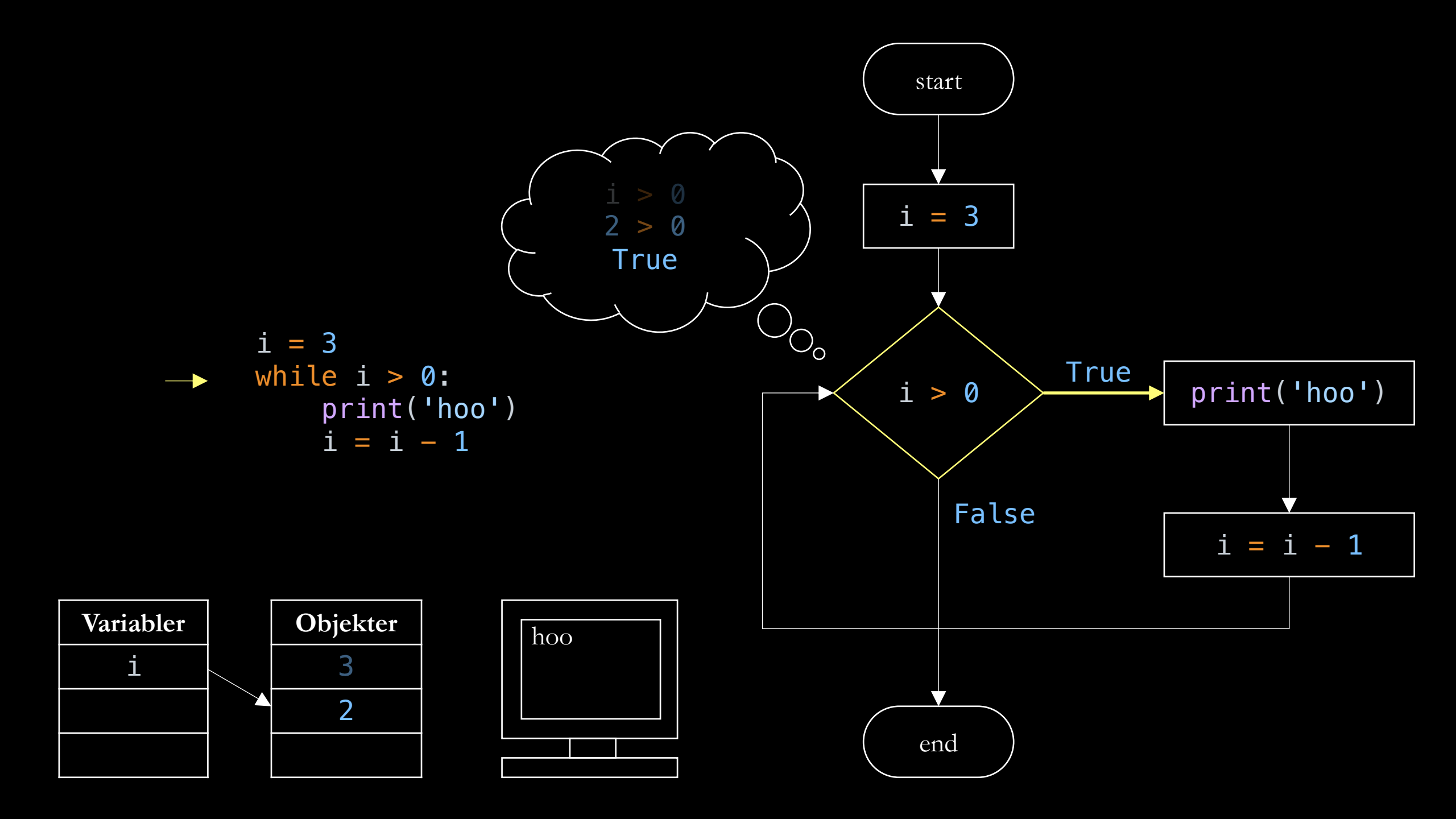


```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```

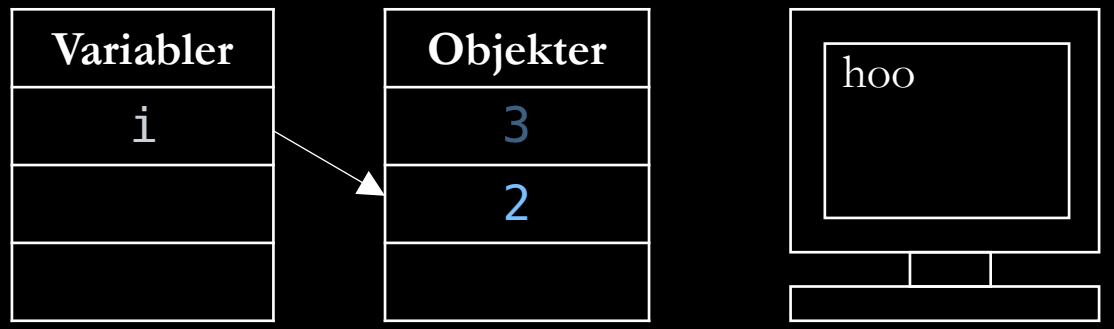


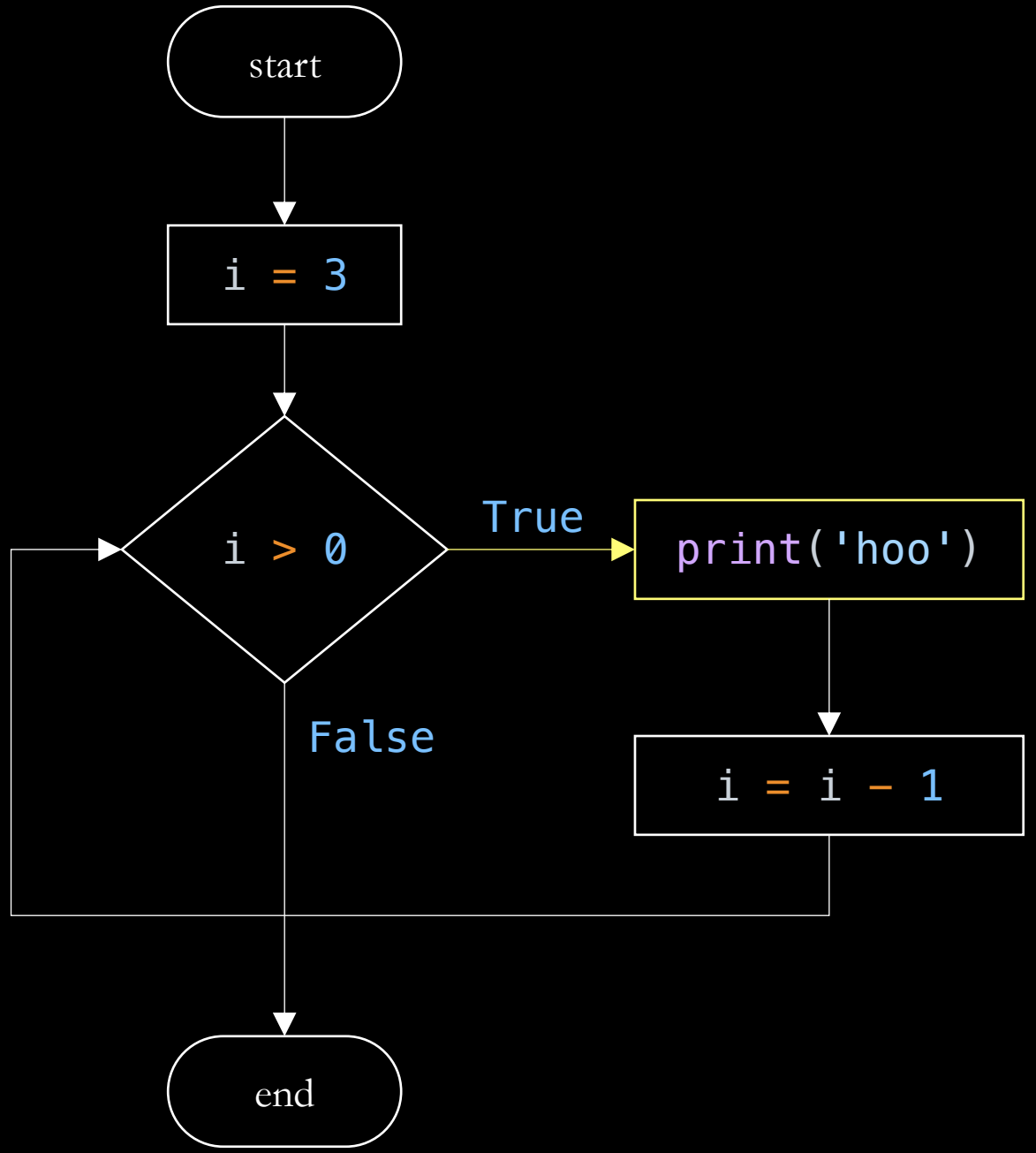



start
i = 3
i > 0
True
False
print('hoo')
i = i - 1
end
i = 3
while i > 0:
print('hoo')
i = i - 1
i > 0
2 > 0
True
Variabler
i
Objekter
3
2
hoo

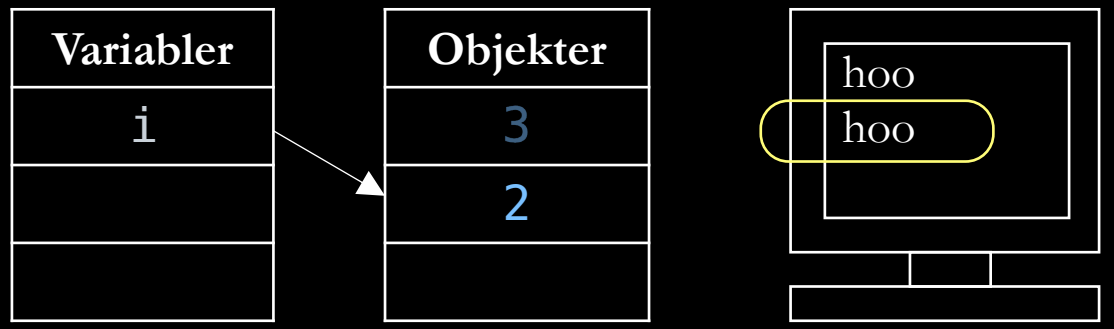

start
i = 3
i > 0
True
False
print('hoo')
i = i - 1
end
i = 3
while i > 0:
print('hoo')
i = i - 1
i > 0
2 > 0
True
Variabler
i
Objekter
3
2
hoo

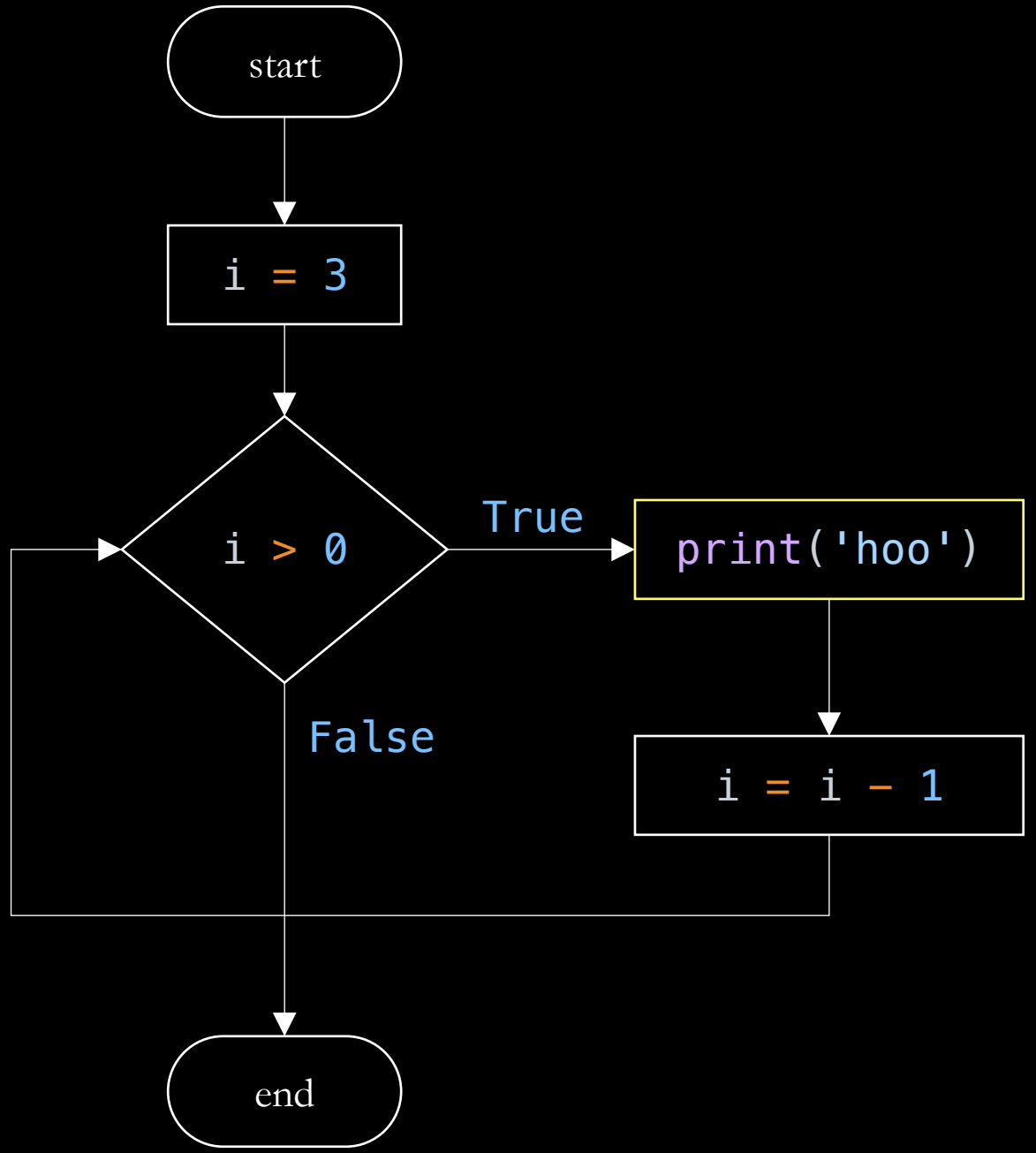
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



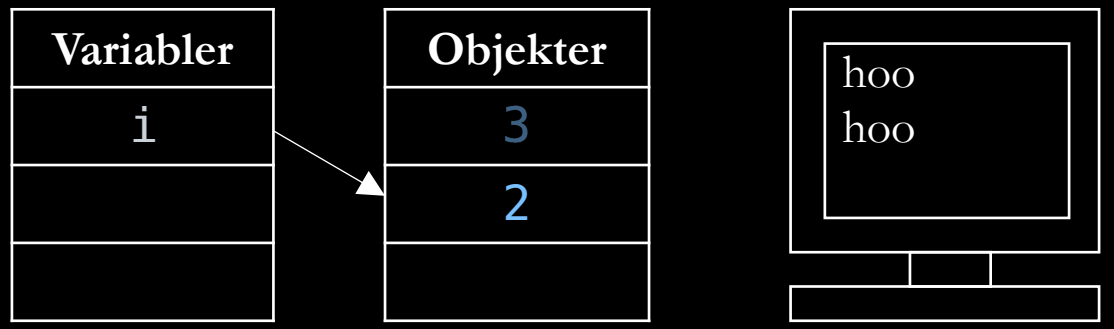


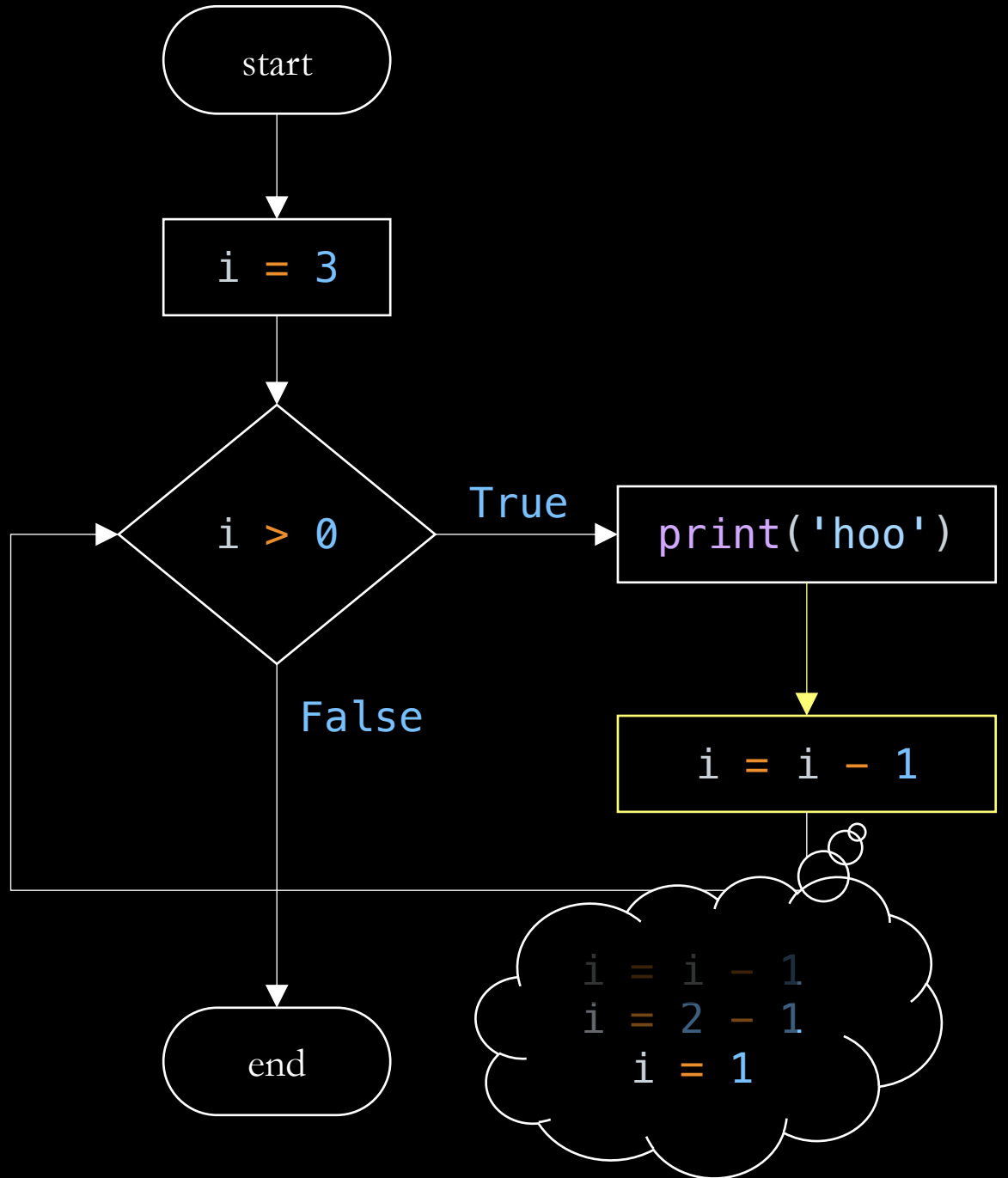
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



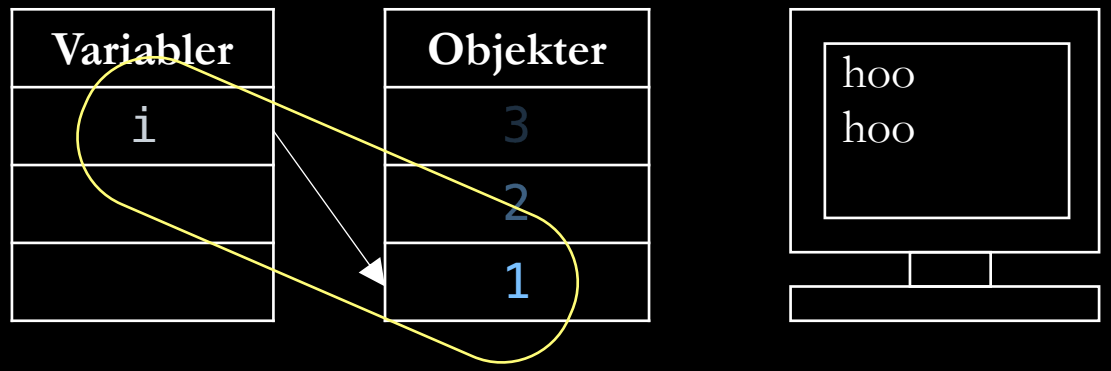


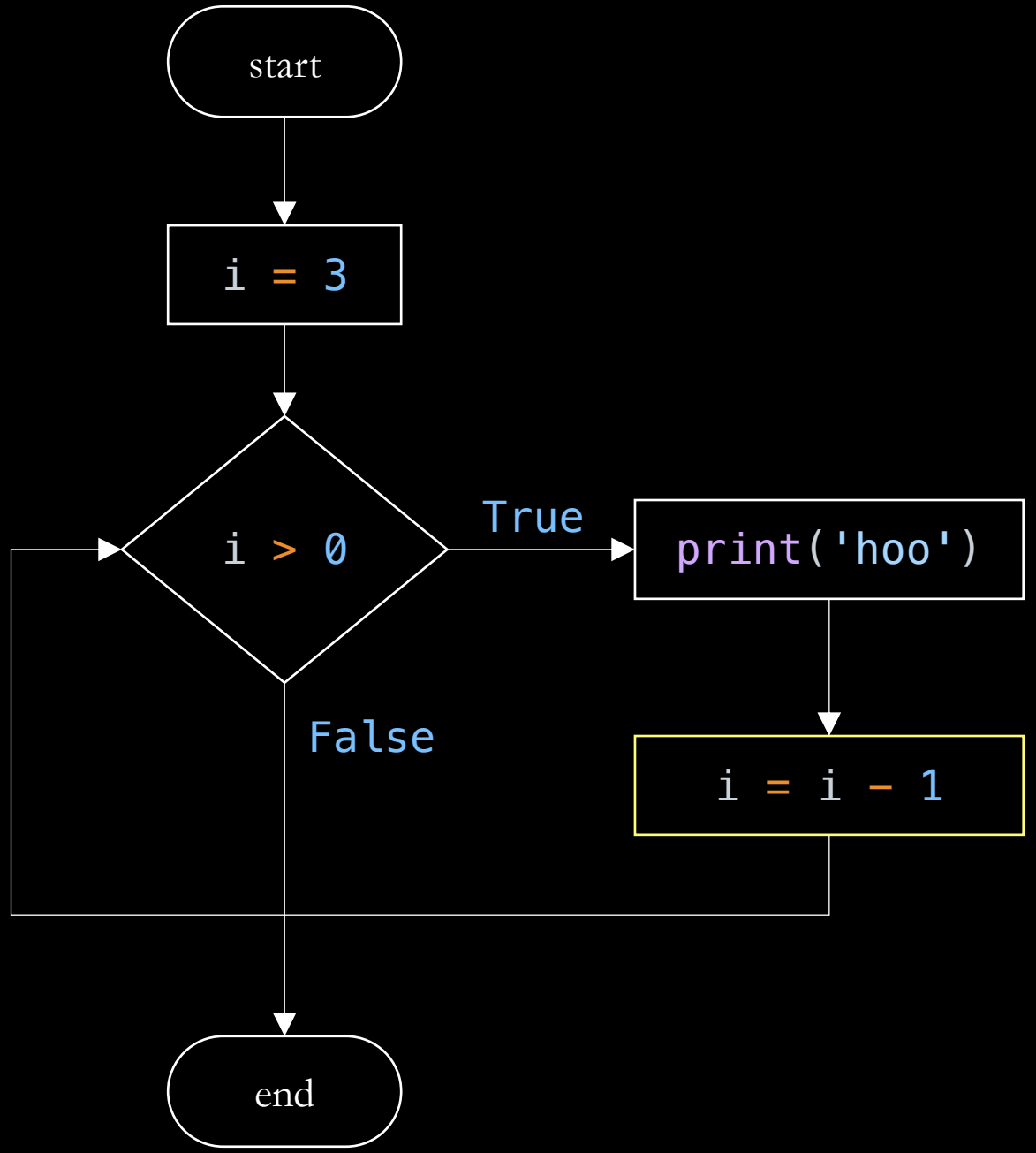
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



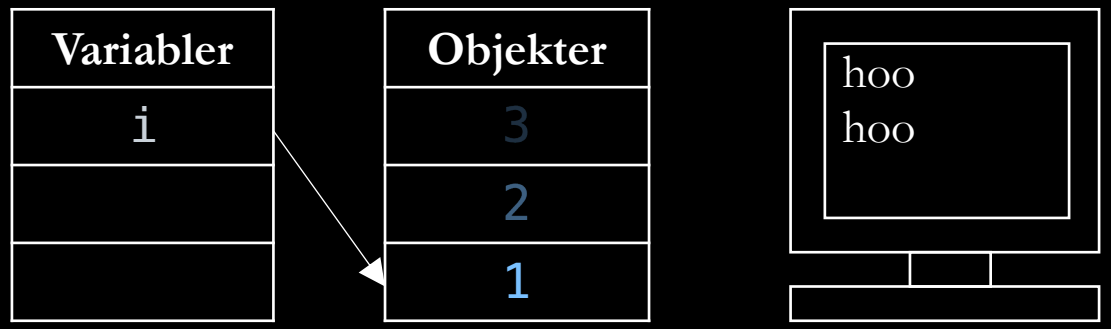


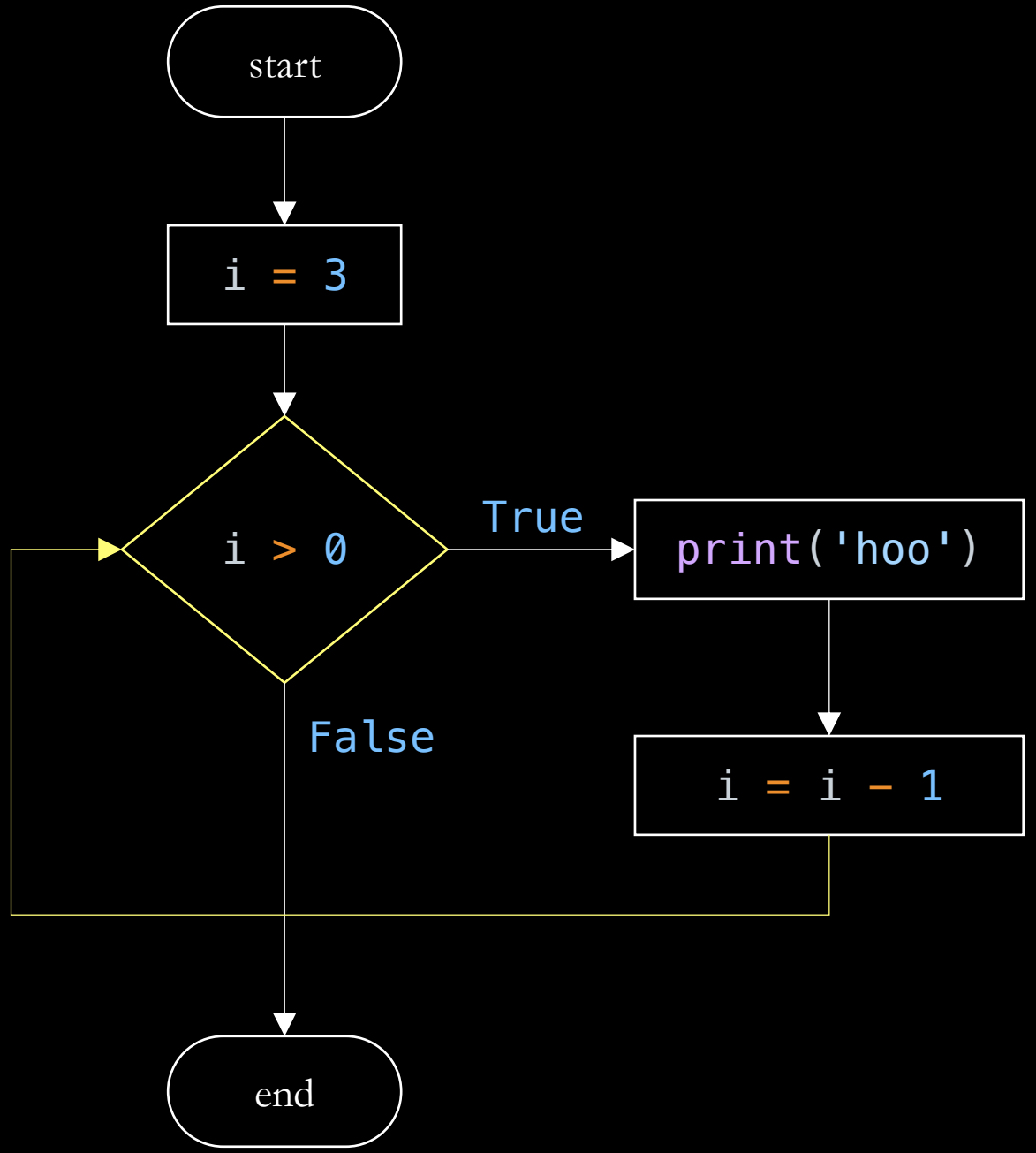
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```

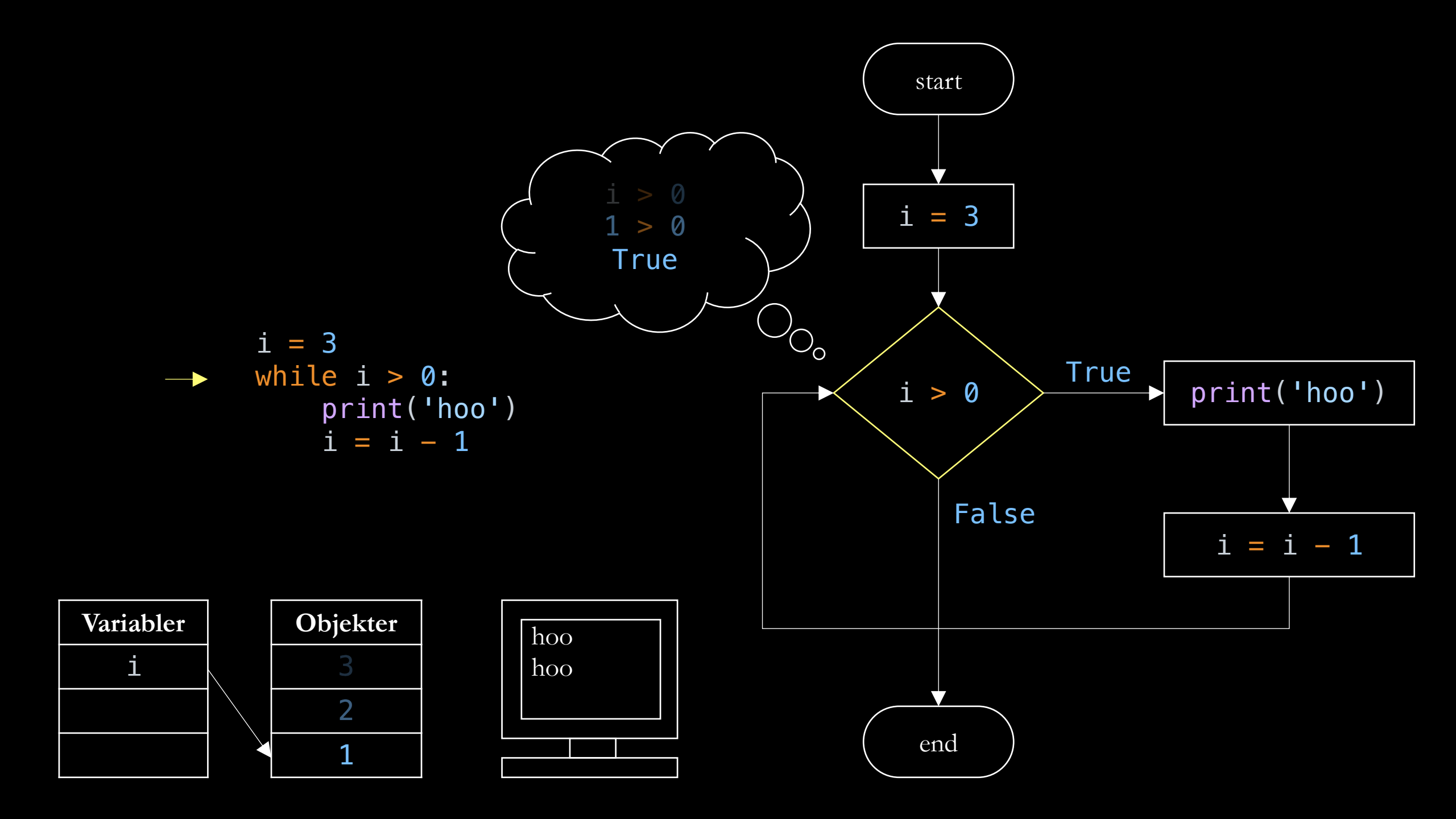


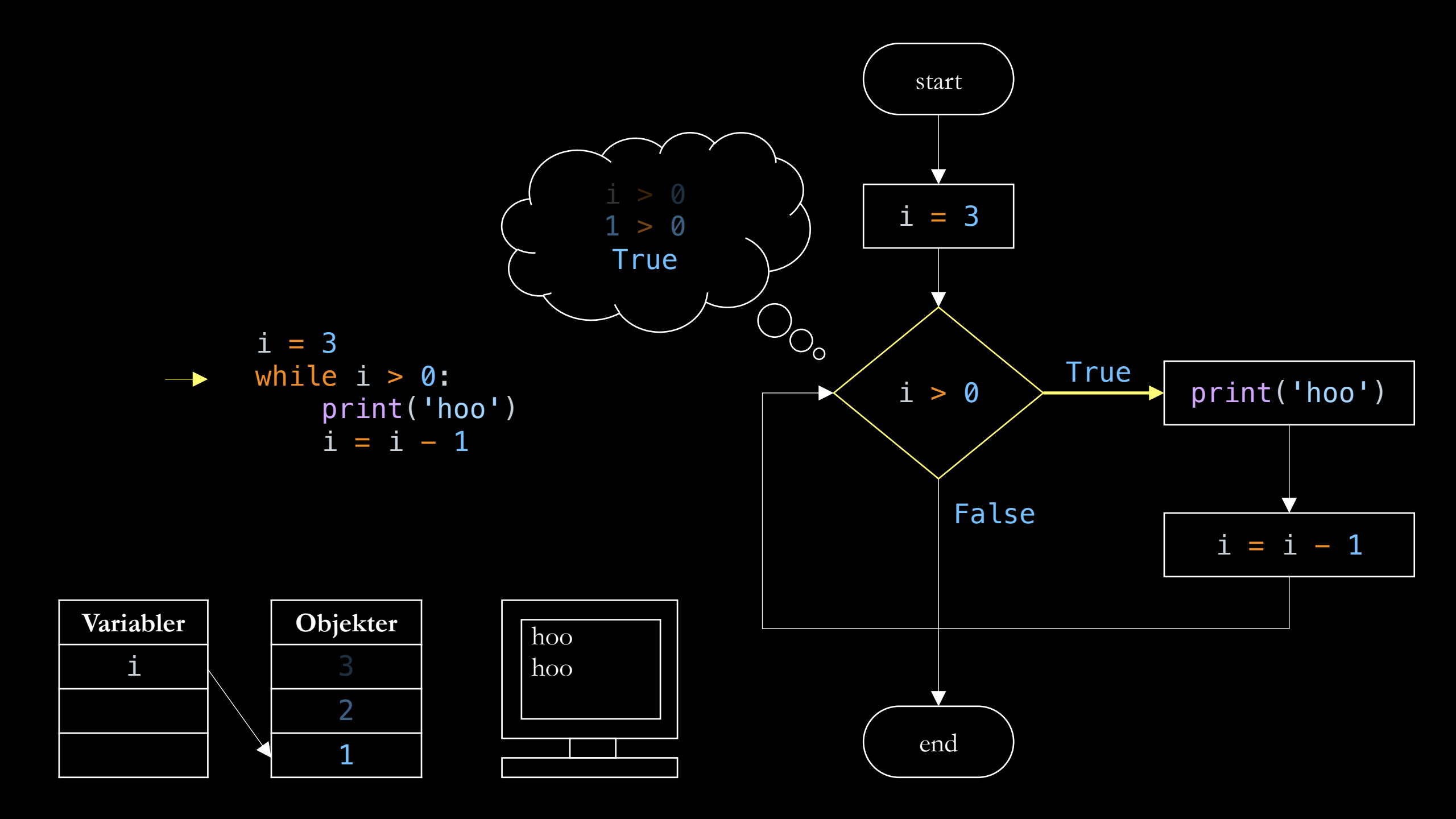


```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```

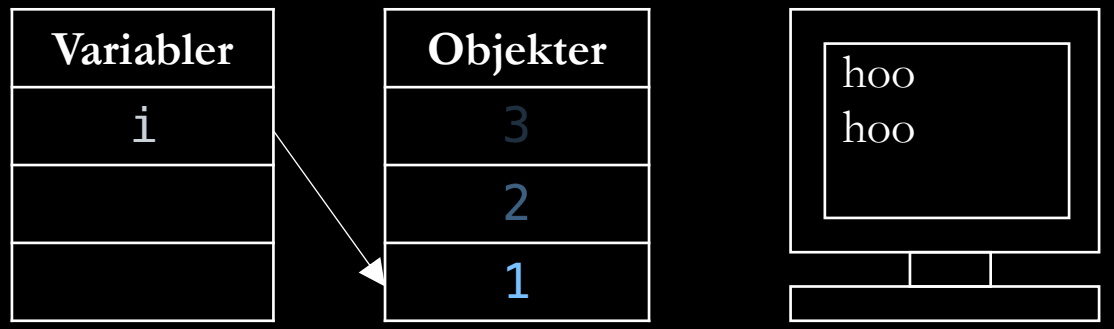


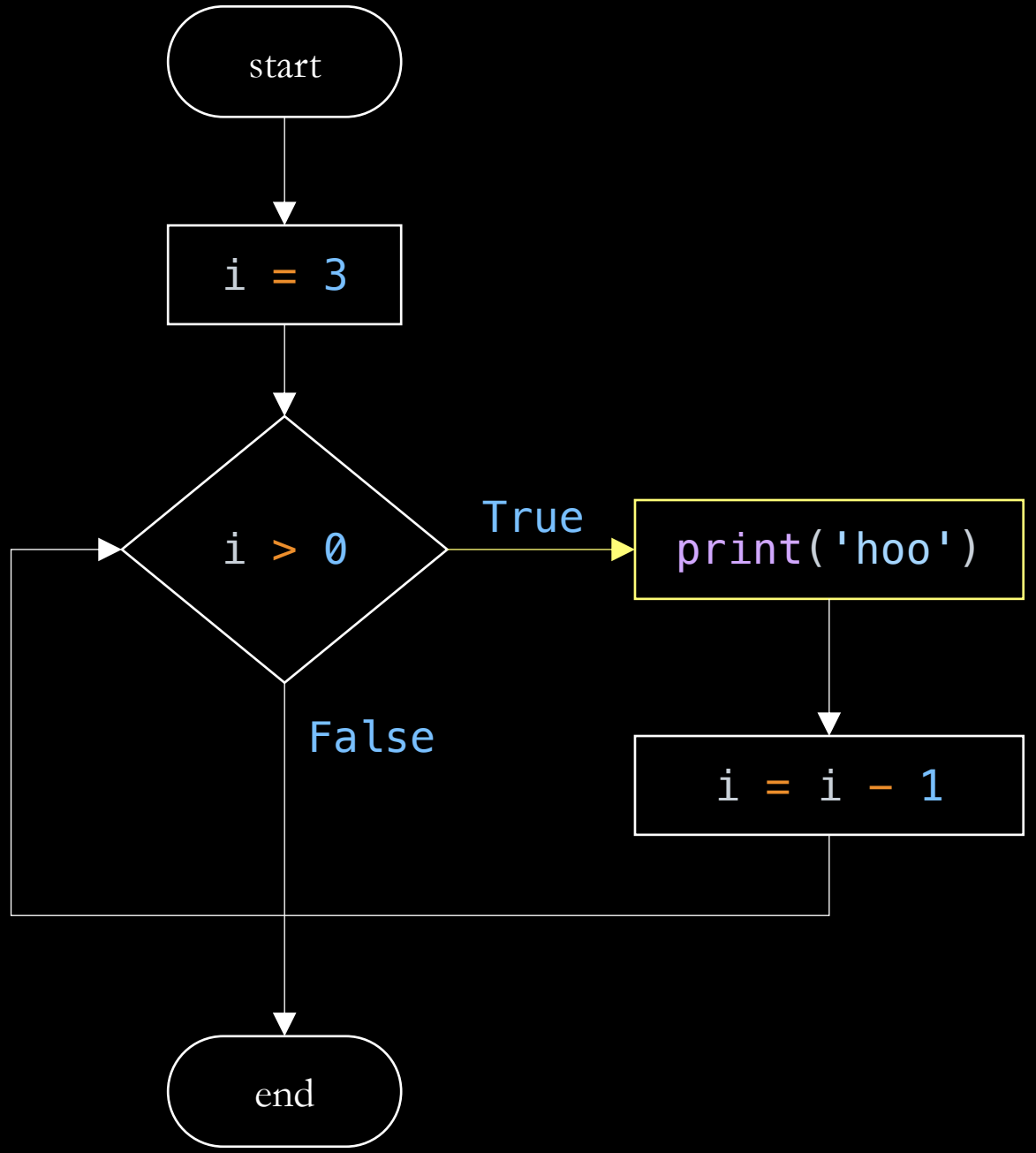



start
i = 3
i > 0
True
False
print('hoo')
i = i - 1
end
i = 3
while i > 0:
print('hoo')
i = i - 1
i > 0
1 > 0
True
Variabler
i
Objekter
3
2
1
hoo
hoo

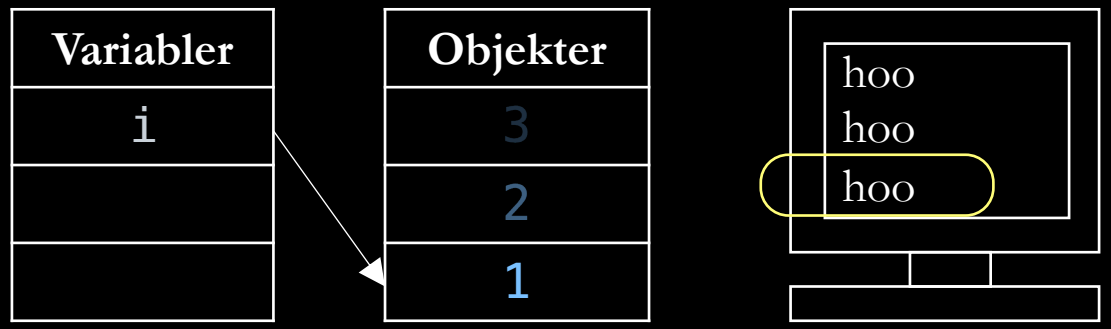

start
i = 3
i > 0
True
False
print('hoo')
i = i - 1
end
i = 3
while i > 0:
print('hoo')
i = i - 1
i > 0
1 > 0
True
Variabler
i
Objekter
3
2
1
hoo
hoo

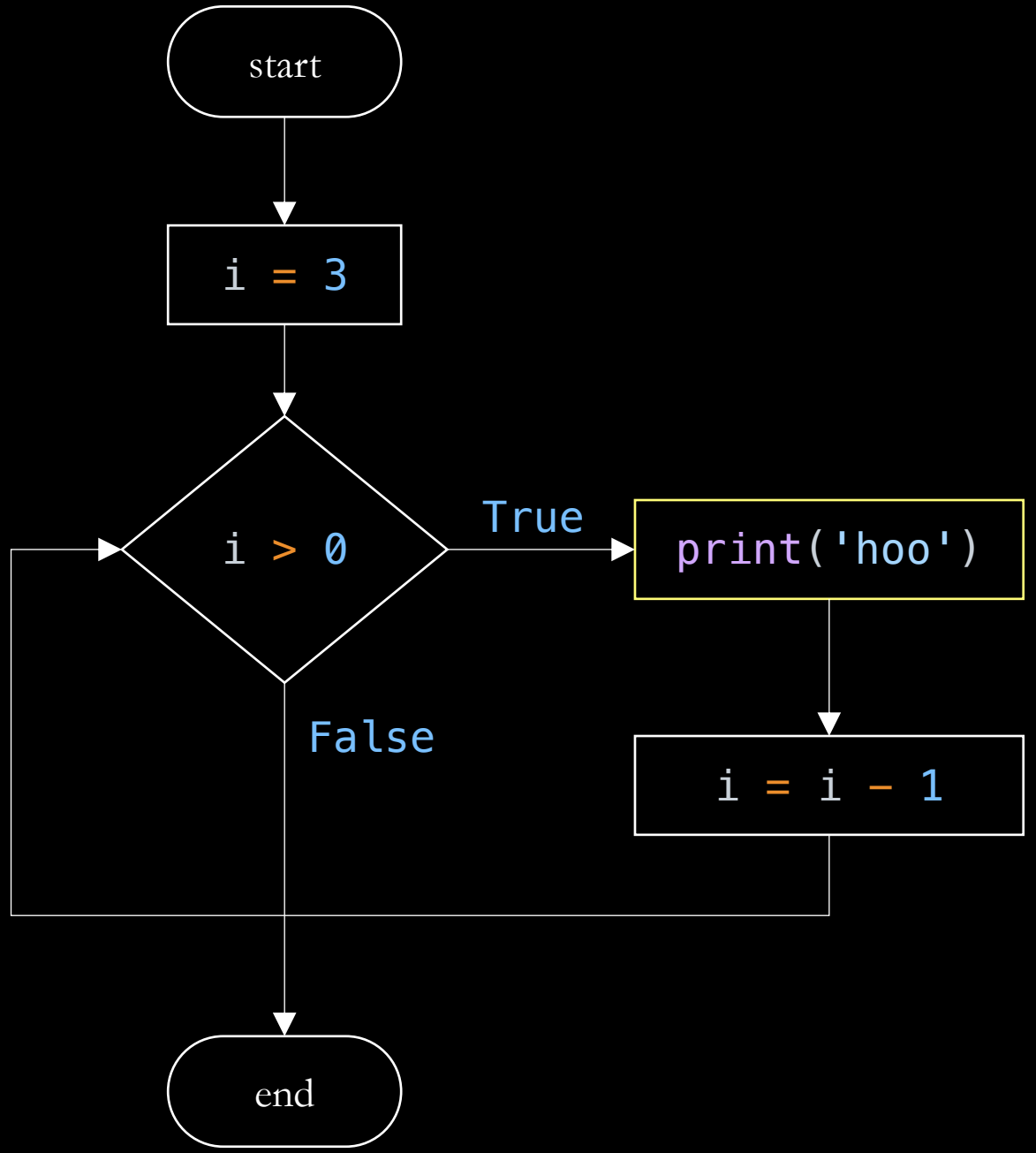
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



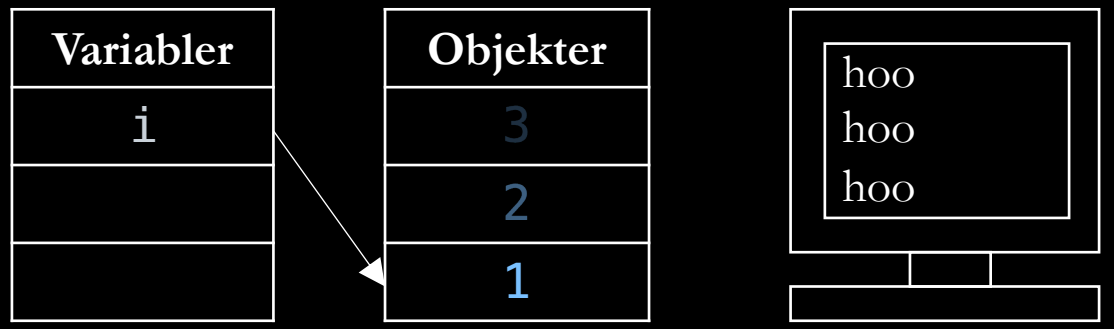


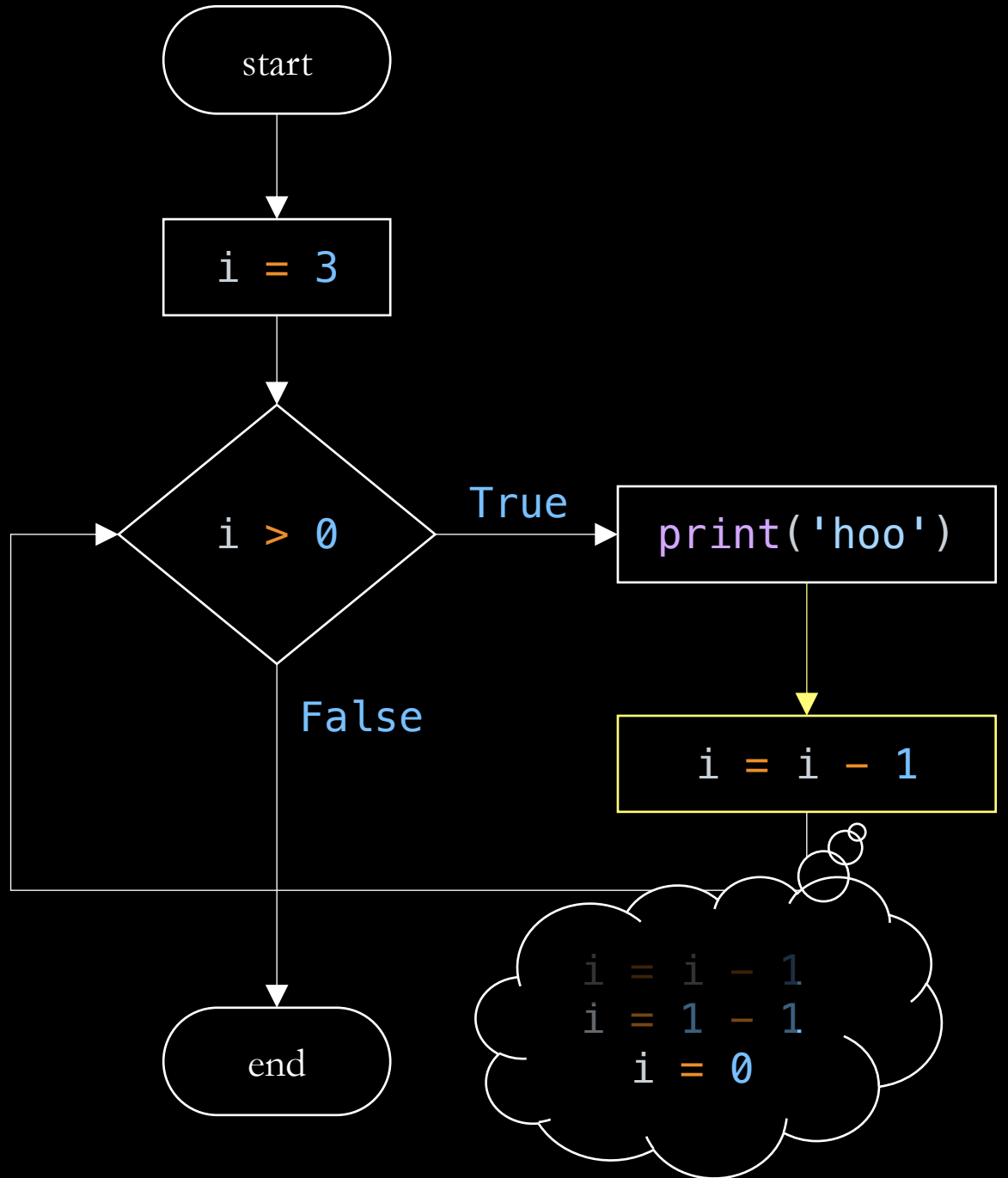
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



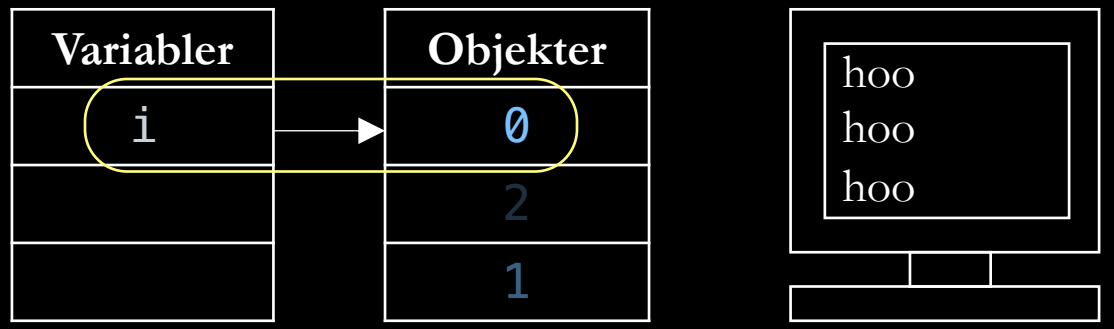


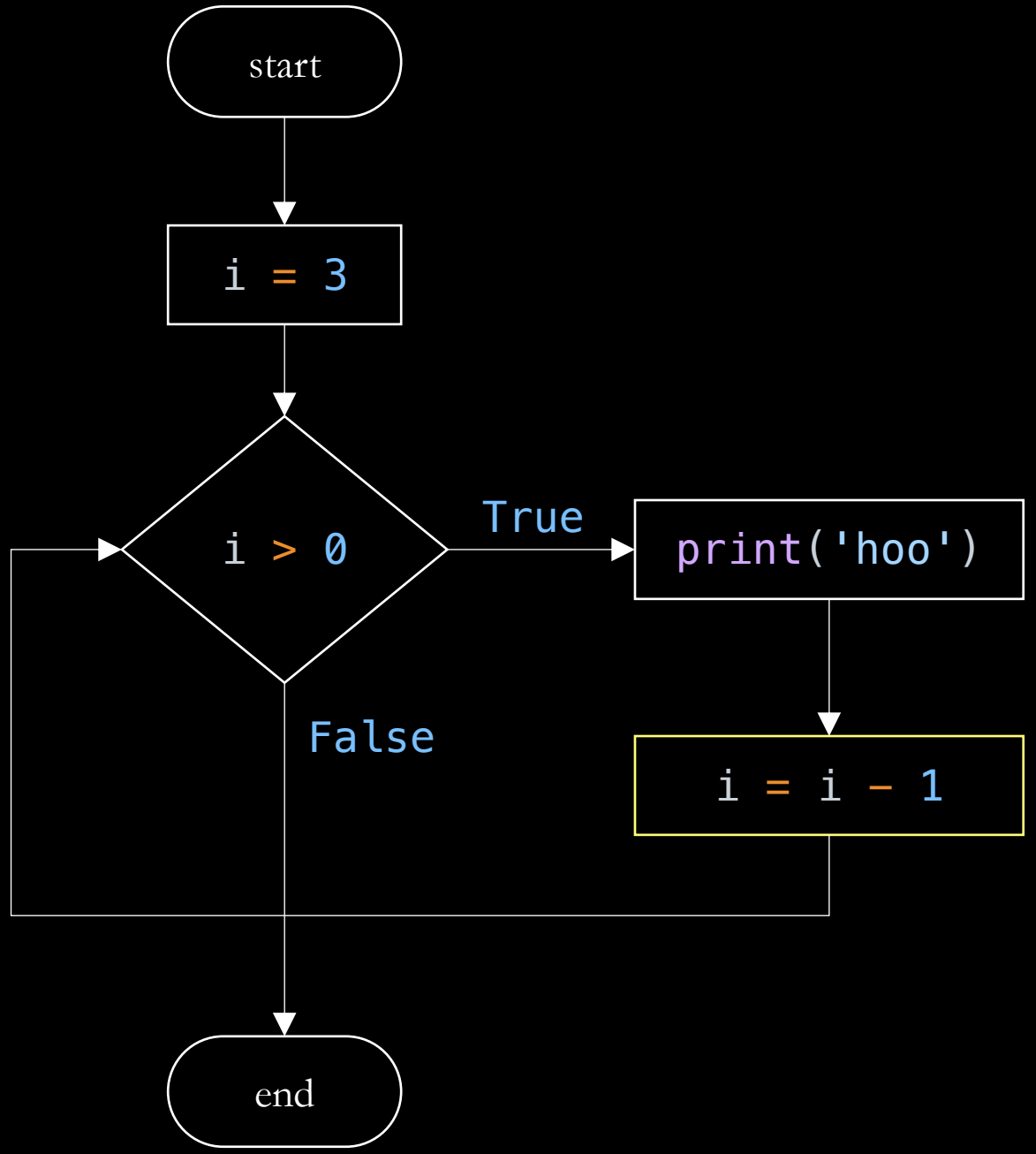
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



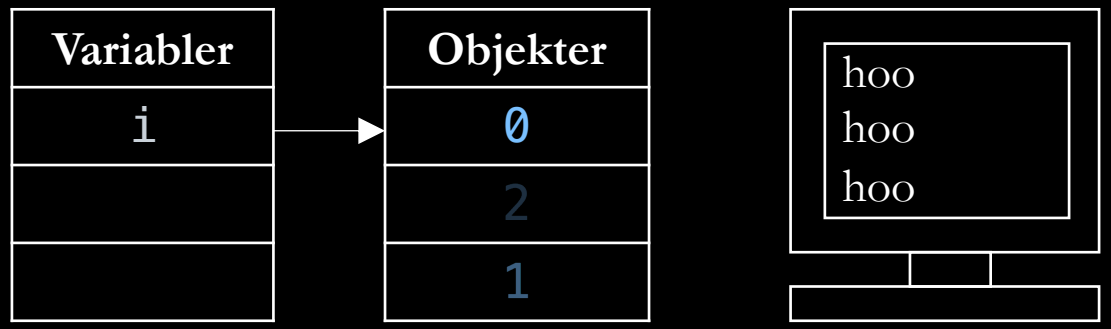


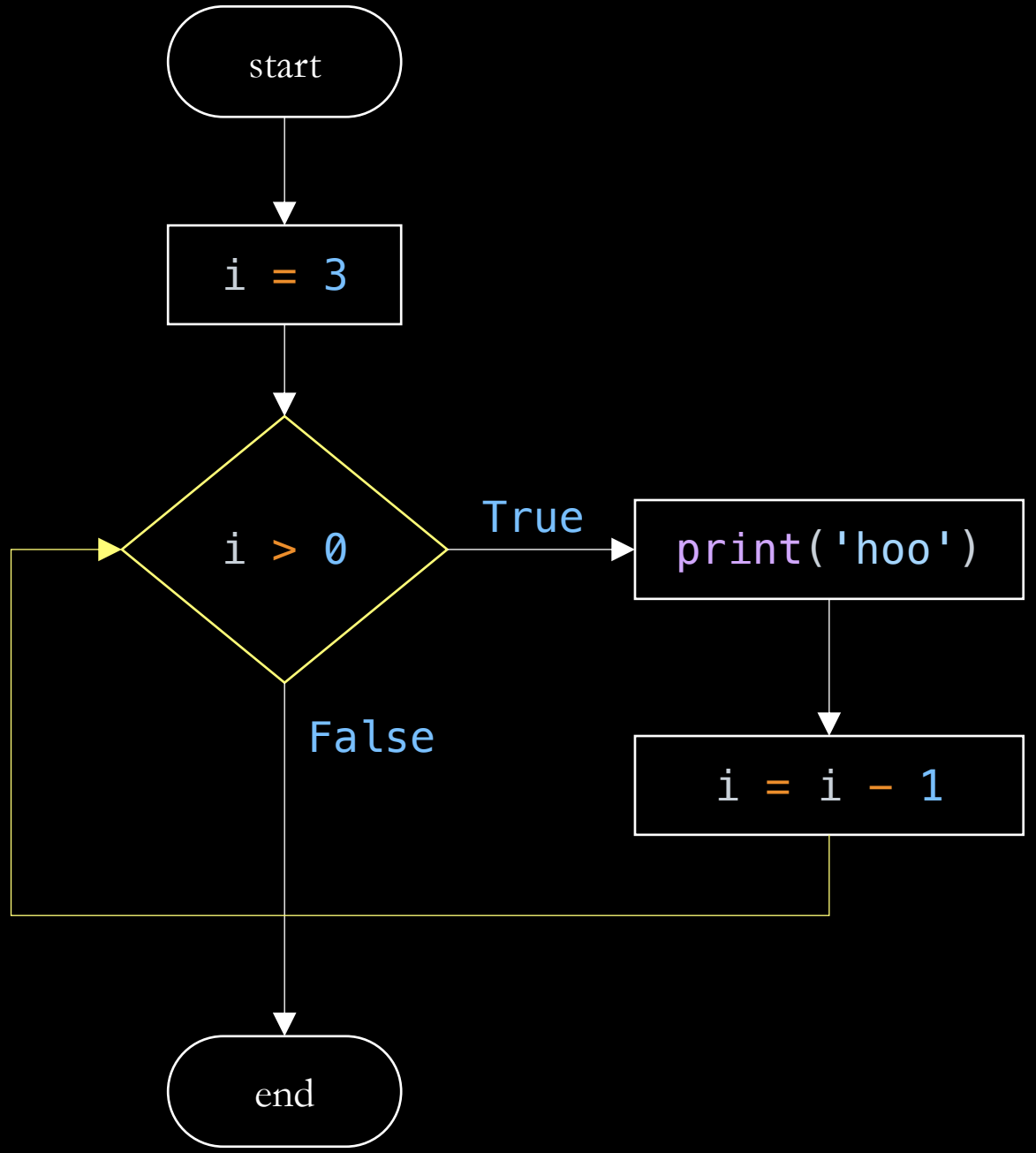
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```

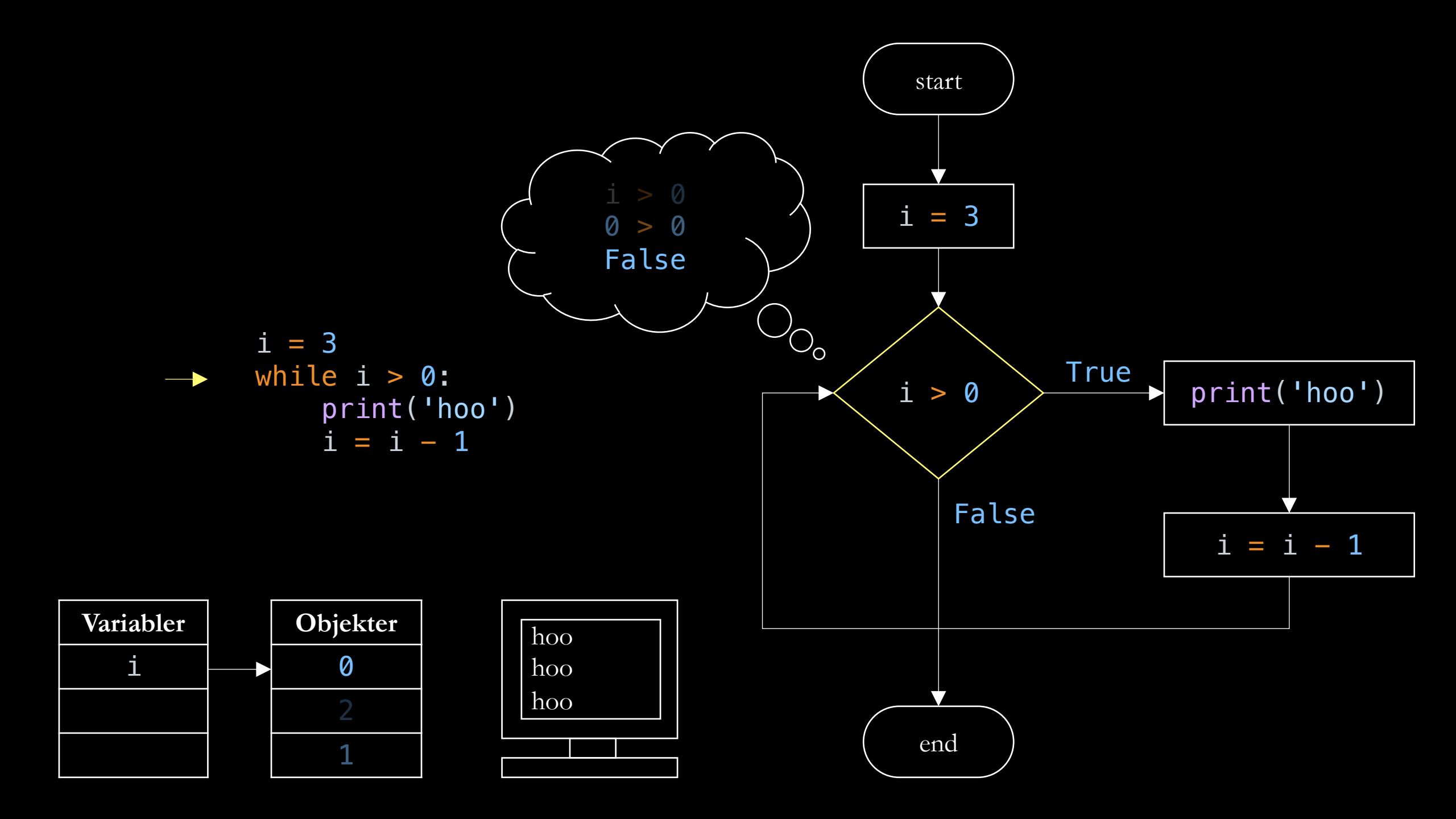


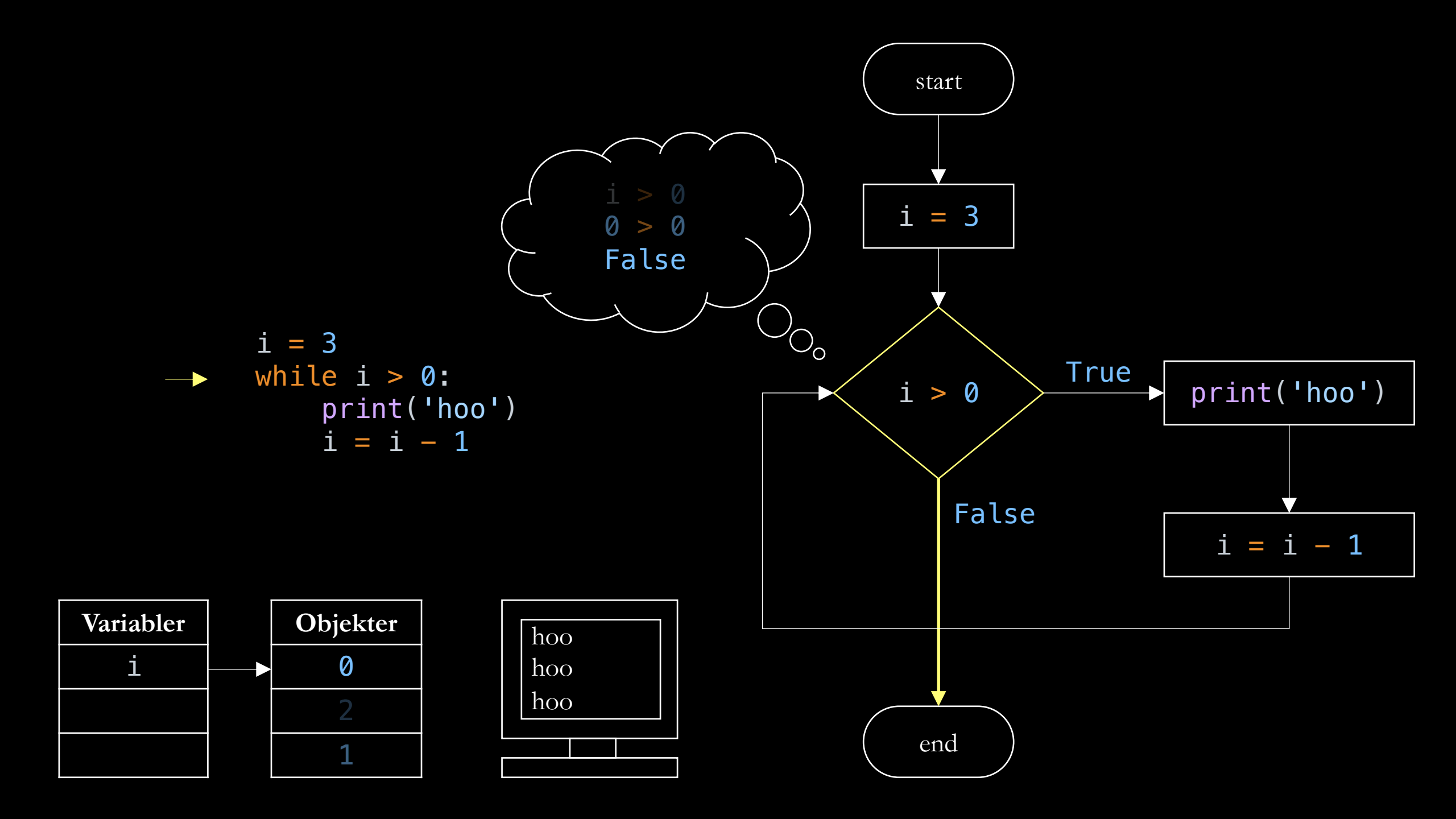


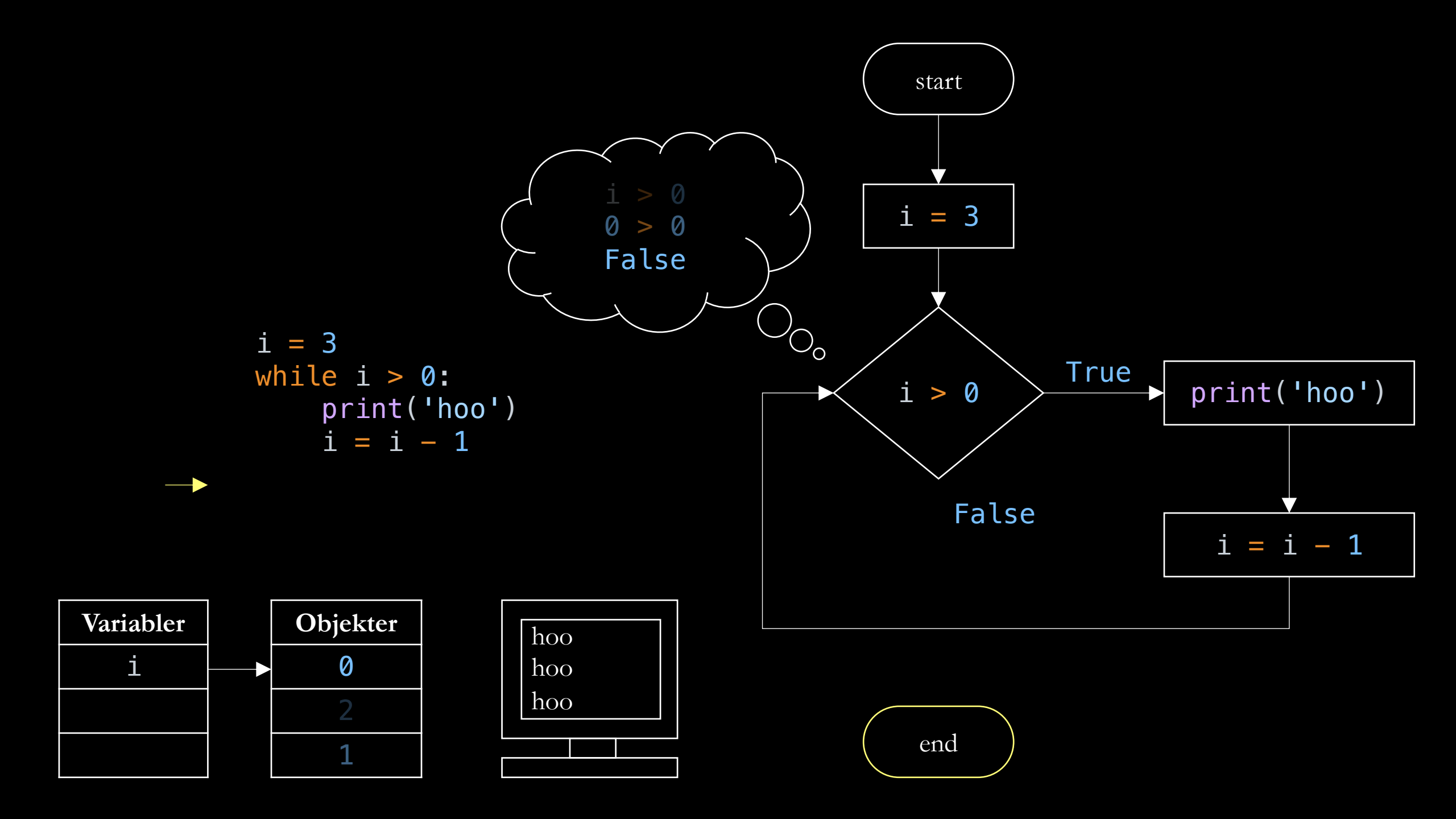
```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```






start
i = 3
i > 0
True
print('hoo')
i = i - 1
False
end
i > 0
0 > 0
False
i = 3
while i > 0:
print('hoo')
i = i - 1
Variabler
i
Objekter
0
2
1
hoo
hoo
hoo


start
i = 3
i > 0
0 > 0
False
i = 3
while i > 0:
print('hoo')
i = i - 1
i > 0
True
print('hoo')
i = i - 1
False
end
Variabler
i
Objekter
0
2
1
hoo
hoo
hoo


start
i = 3
i > 0
0 > 0
False
i > 0
True
print('hoo')
False
i = i - 1
i = 3
while i > 0:
print('hoo')
i = i - 1
Variabler
i
Objekter
0
2
1
hoo
hoo
hoo
end

```
i = 3
while i > 0:
    print('hoo')
    i = i - 1
```



```
i = 0
while i < 3:
    print('hoo')
    i += 1
```

```
i = 0
while i < 3:
    print('hoo')
    i += 1
```



```
for i in [0, 1, 2]:
    print('hoo')
```

```
for i in [0, 1, 2]:
    print('hoo')
```



```
for i in range(3):
    print('hoo')
```

```
for i in range(3):
    print('hoo')
```



```
for _ in range(3):
    print('hoo')
```

```
range(5)
```

```
0, 1, 2, 3, 4
```

```
range(3, 7)
```

```
3, 4, 5, 6
```

```
range(10, 20, 2)
```

```
10, 12, 14, 16, 18
```

```
range(20, 10, -2)
```

```
20, 18, 16, 14, 12
```

`for`

`while`

kan vare evig ☑

☑ kortere å skrive

☑ mindre bugs

Oppgave:

Les input fra brukeren helt til du får noe du vil ha

`break`

avbryt løkken umiddelbart

`continue`

avbryt denne iterasjon av løkken