



UNIVERSITETET I BERGEN

LØKKER

INF100

VÅR 2024

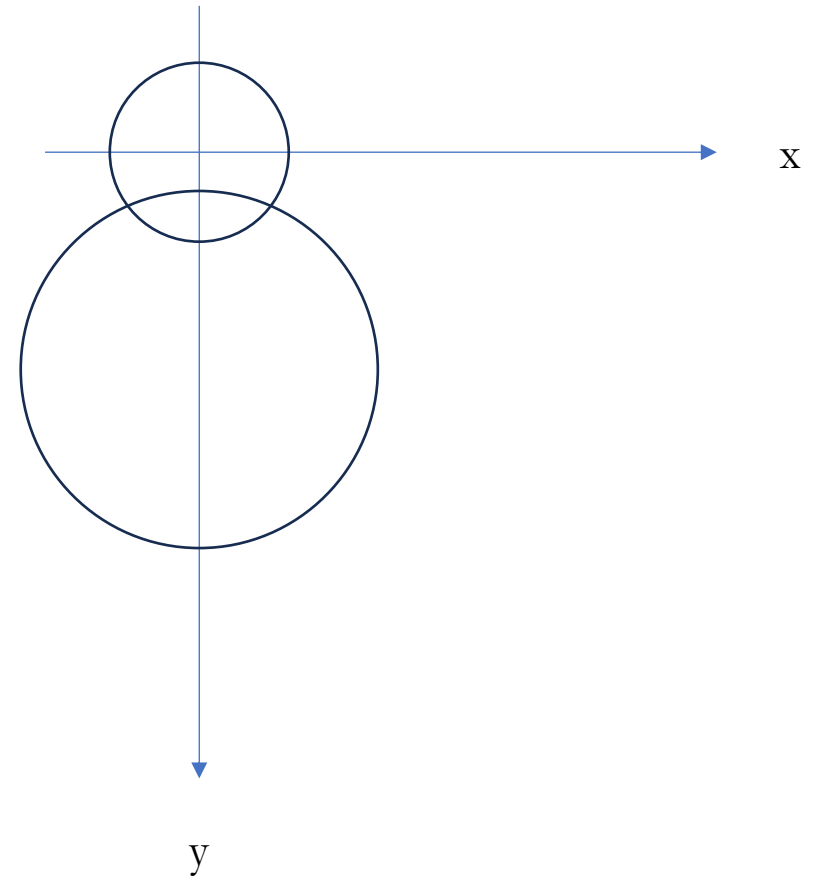
Torstein Strømme

VANLIGE FEIL

```
def circles_overlap(x1, y1, r1, x2, y2, r2):  
    distance = get_distance(x1, y1, x2, y2)  
    if distance <= r1 + r2:  
        print('True')  
    else:  
        print('False')
```

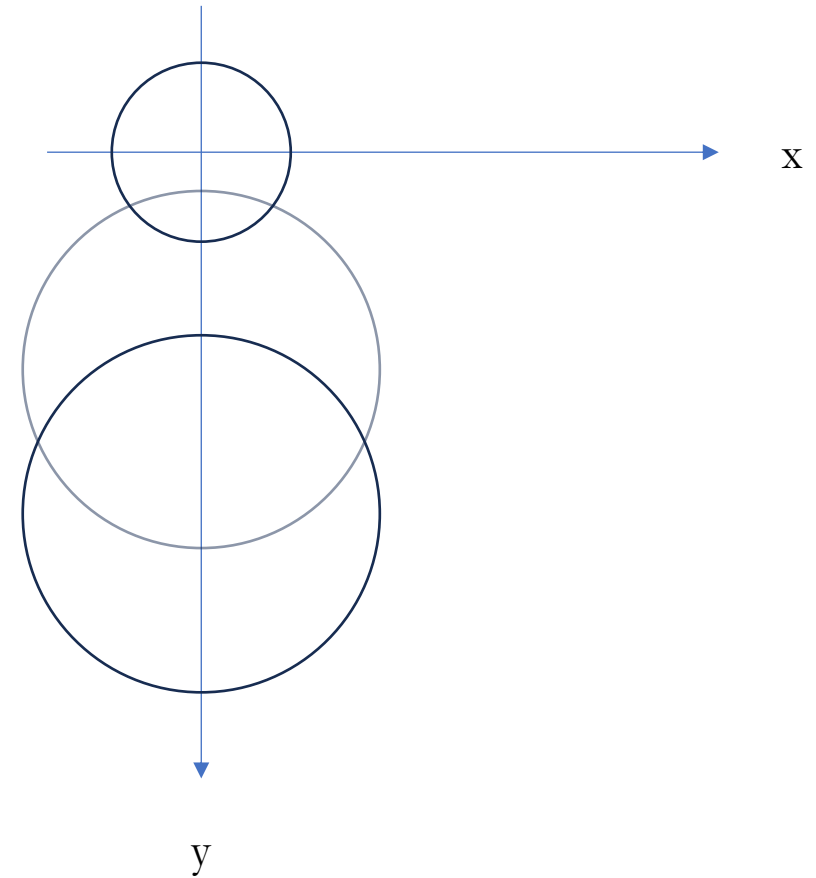


```
print('Tester circles_overlap...', end=' ')  
assert circles_overlap(0, 6, 5, 0, 0, 2)  
print('OK')
```



VANLIGE FEIL

```
def circles_overlap(x1, y1, r1, x2, y2, r2):  
    distance = get_distance(x1, y1, x2, y2)  
    if distance <= r1 + r2:  
        return 'True'  
    else:  
        return 'False'  
  
print('Tester circles_overlap...', end=' ')  
assert circles_overlap(0, 6, 5, 0, 0, 2)  
assert not circles_overlap(0, 10, 5, 0, 0, 2)  
print('OK')
```

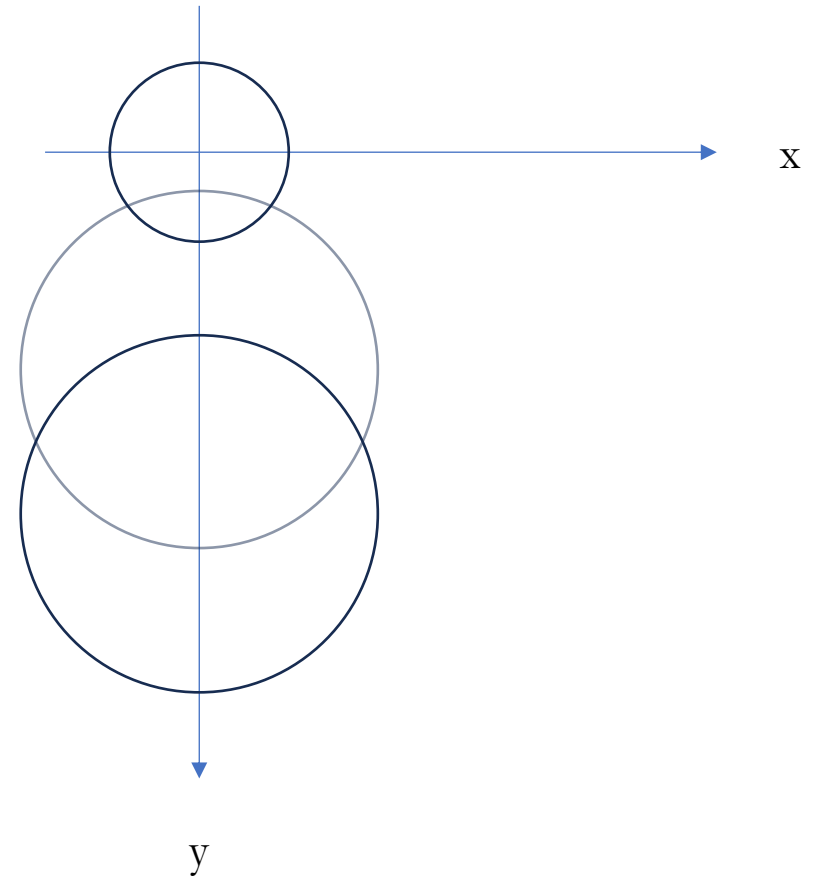


VANLIGE FEIL

```
def circles_overlap(x1, y1, r1, x2, y2, r2):  
    distance = get_distance(x1, y1, x2, y2)  
    if distance <= r1 + r2:  
        return True  
    else:  
        return False
```



```
print('Tester circles_overlap...', end=' ')  
assert circles_overlap(0, 6, 5, 0, 0, 2)  
assert not circles_overlap(0, 10, 5, 0, 0, 2)  
print('OK')
```

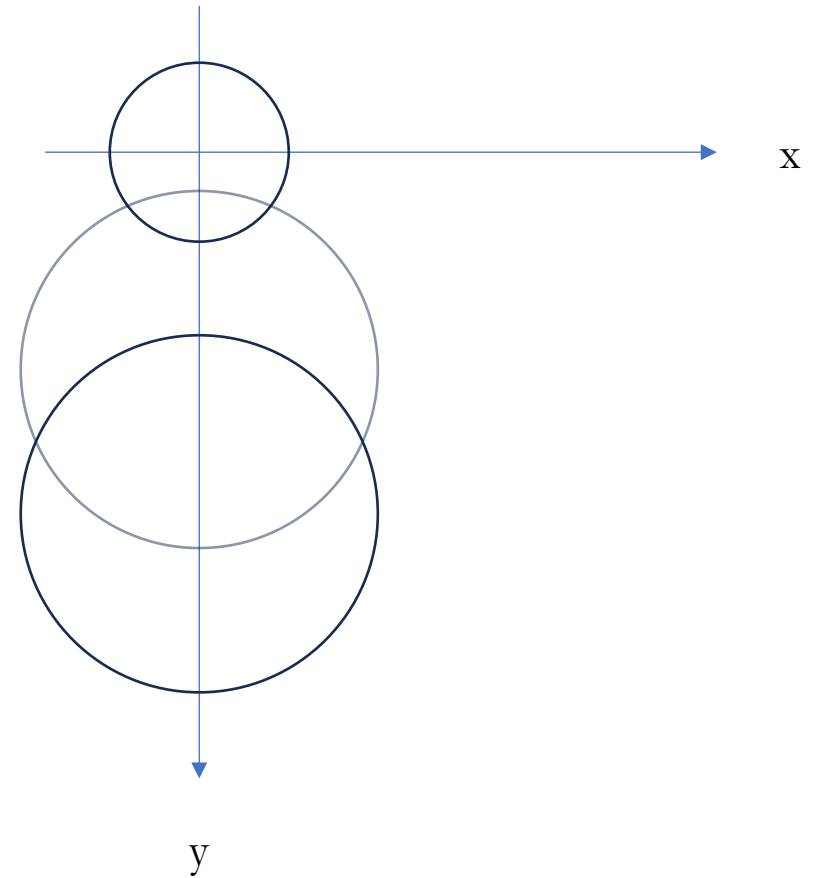


VANLIGE FEIL

```
def circles_overlap(x1, y1, r1, x2, y2, r2):  
    distance = get_distance(x1, y1, x2, y2)  
    return distance <= r1 + r2
```



```
print('Tester circles_overlap...', end=' ')  
assert not circles_overlap(0, 10, 5, 0, 0, 2)  
print('OK')
```



STRENGER

```
s = ' Dette er en streng '
```

- En streng er en samling av tegn
- Første tegn er på posisjon 0

```
c = s[1]  
print(c)  skriver ut: «e»
```

```
s[0] → 'D'  
s[1] → 'e'  
s[2] → 't'  
s[3] → 't'  
s[4] → 'e'  
s[5] → ' '  
s[6] → 'e'  
s[7] → 'r'  
s[8] → ' '  
...   ...
```

STRENGER

```
s = 'Dette er en streng'
```

- Hvor mange «e»'er er det i strengen?

```
count = 0
```

```
c = s[0]
```

```
if s[0] == 'e':  
    count += 1
```

```
c = s[1]
```

```
if c == 'e':  
    count += 1
```

```
c = s[2]
```

```
if c == 'e':  
    count += 1
```

```
...
```

```
print(f"Det er {count} «e»'er i '{s}'")
```

STRENGER OG LØKKER

```
s = ' Dette er en streng '
```

- Hvor mange «e»'er er det i strengen?

```
count = 0
```

```
for c in s:
```

```
    if c == 'e':
```

```
        count += 1
```

for hvert element «c» i samlingen «s», gjør følgende:

```
print(f"Det er {count} «e»'er i '{s}'")
```



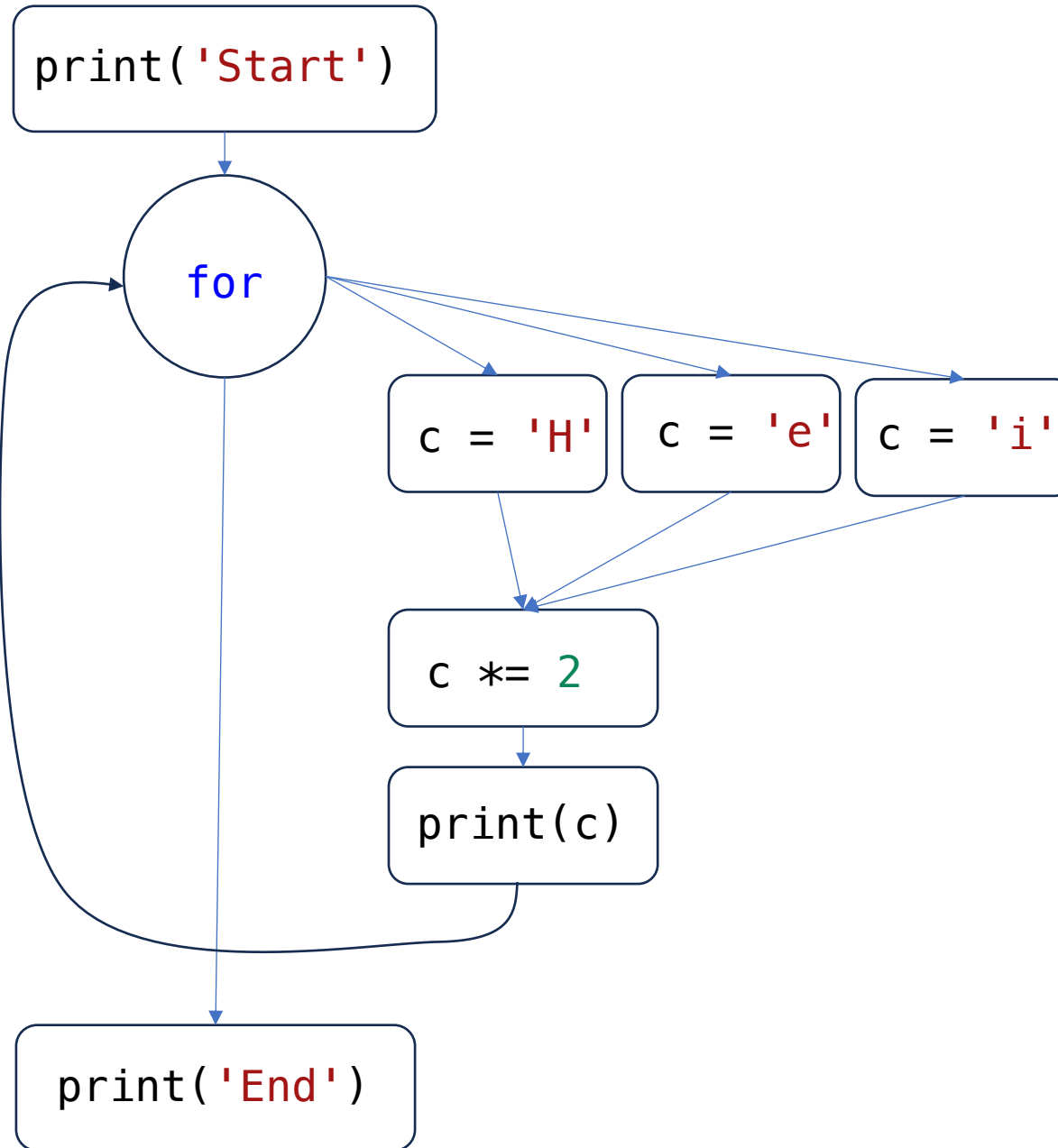
STRENGER OG LØKKER

```
s = 'Dette er en streng'
```

- Oppgave: skriv en funksjon som tar som input en streng s og et tegn c, og returnerer hvor mange ganger c opptrer i s

FOR

```
print('Start')  
for c in 'Hei':  
    c *= 2  
    print(c)  
print('End')
```



FOR

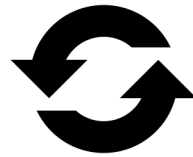
iterand/løkke-variabel

```
bla()  
bla()
```

f. eks

- range
- streng
- tuple
- liste

```
for i in <samling>:  
    bla()  
    bla(i)  
    bla()
```



én gang for hvert element i samlingen

```
bla()  
bla()
```

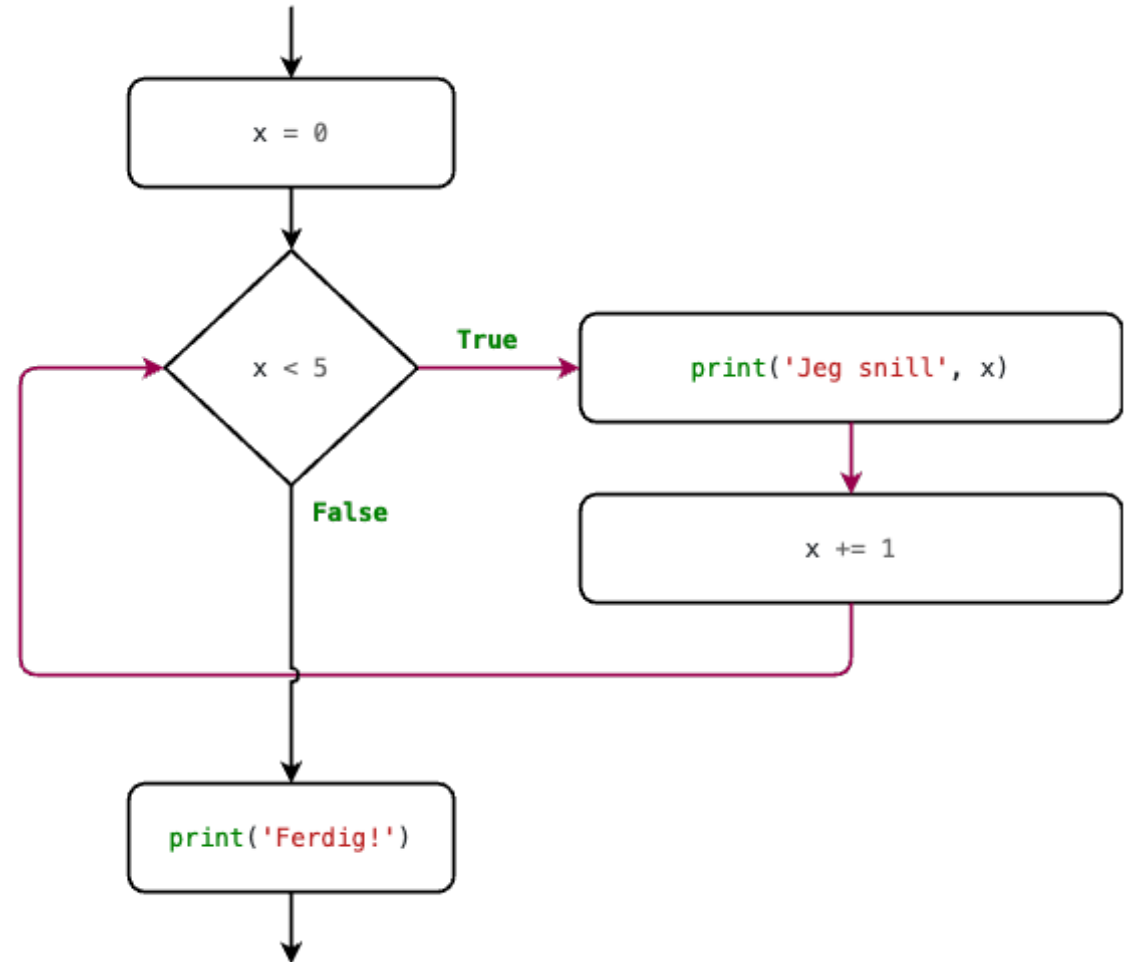
WHILE

```
x = 0
while x < 5:
    print('Jeg snill', x)
    x += 1
print('Ferdig!')
```

betingelse (points to `x < 5`)

kodeblokk (bracketed around the loop body)

- Kodeblokken gjentas så lenge betingelsen evaluerer til **True**



EKSEMPEL: ANTALL SIFFER

Gitt et heltall n , hvor mange siffer er det i tallet?

UENDELIG LØKKE

- Hvis betingelsen alltid er True

```
x = 1
while x < 10:
    print(x)
    x + 1
```

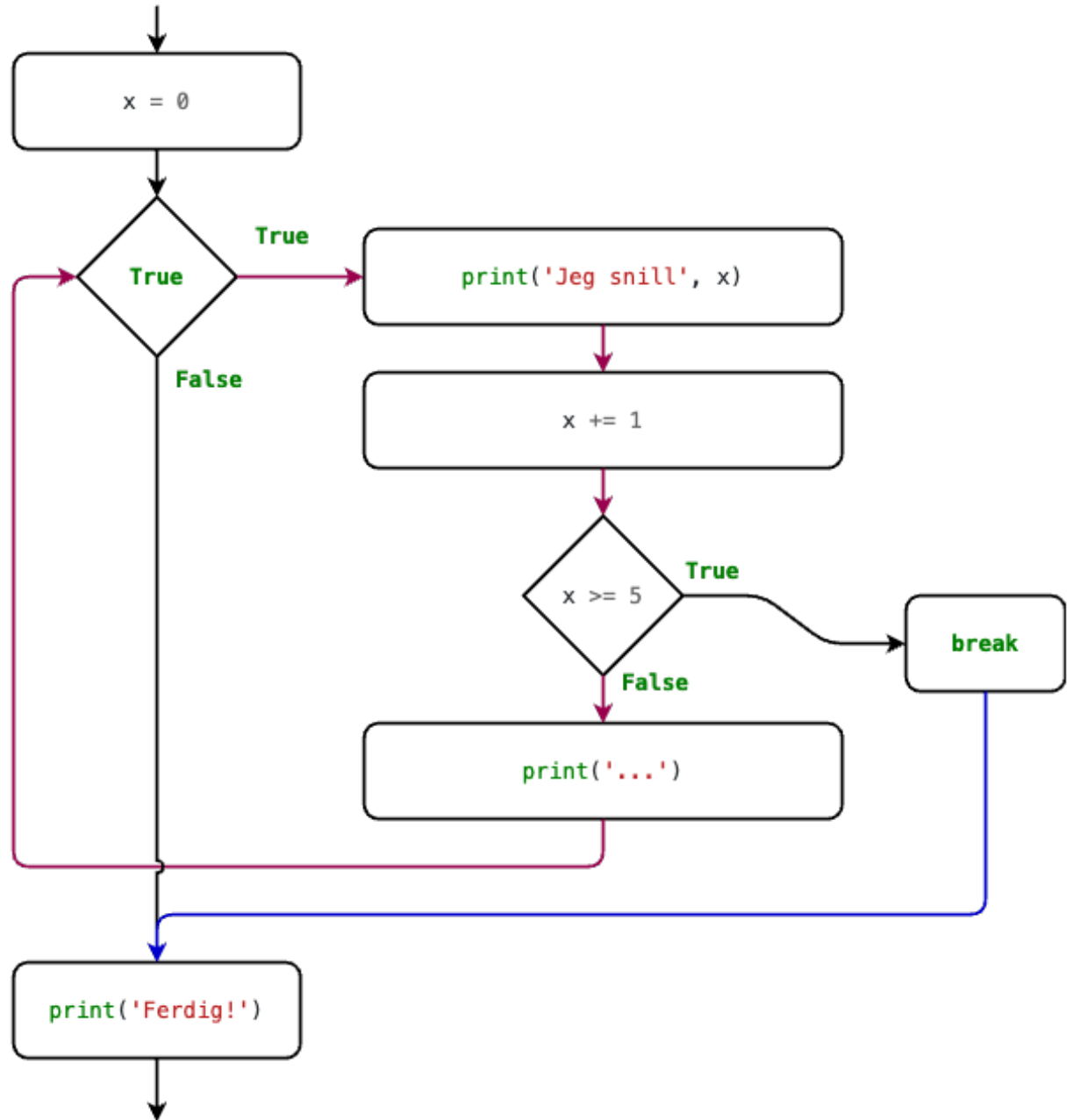
← glemt tilordning (x =)

- Brukeren kan avbryte et program underveis: CTRL + C

BREAK

- For å bryte ut av løkken umiddelbart

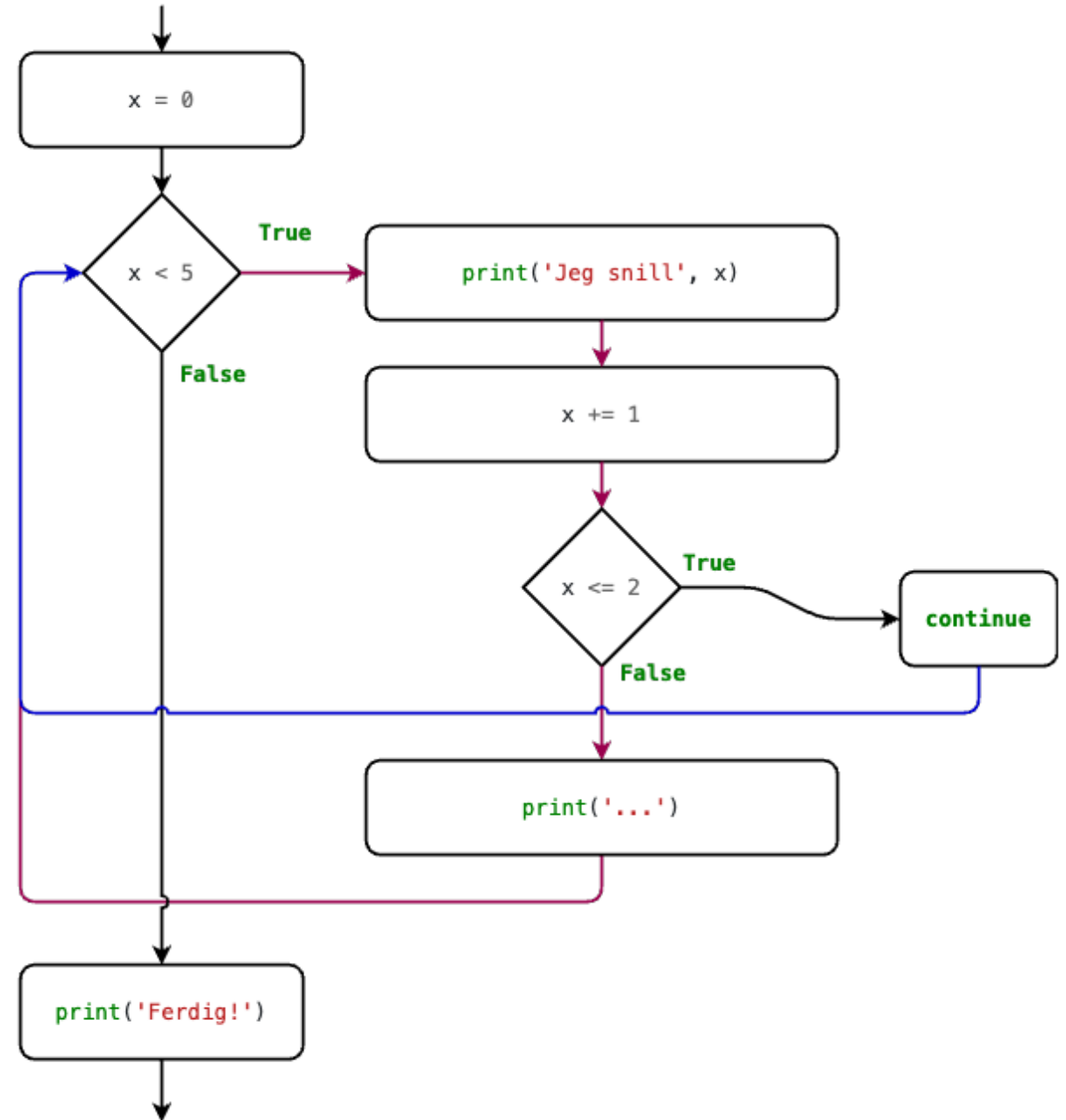
```
x = 0
while True:
    print('Jeg snill', x)
    x += 1
    if x >= 5:
        break
    print('...')
print('Ferdig!')
```



CONTINUE

- For å starte løkken på nytt umiddelbart

```
x = 0
while x < 5:
    print('Jeg snill', x)
    x += 1
    if x <= 2:
        continue
    print('...')
print('Ferdig!')
```



RANGE

`range(5)` → 0, 1, 2, 3, 4

`range(10, 14)` → 10, 11, 12, 13

`range(3, 15, 3)` → 3, 6, 9, 12

`range(15, 3, -2)` → 15, 13, 11, 9, 7, 5,

FOR



```
for i in range(1, 4):  
    print("z" * i)  
  
print("Ferdig")
```

VARIABLER

UTSKRIFT

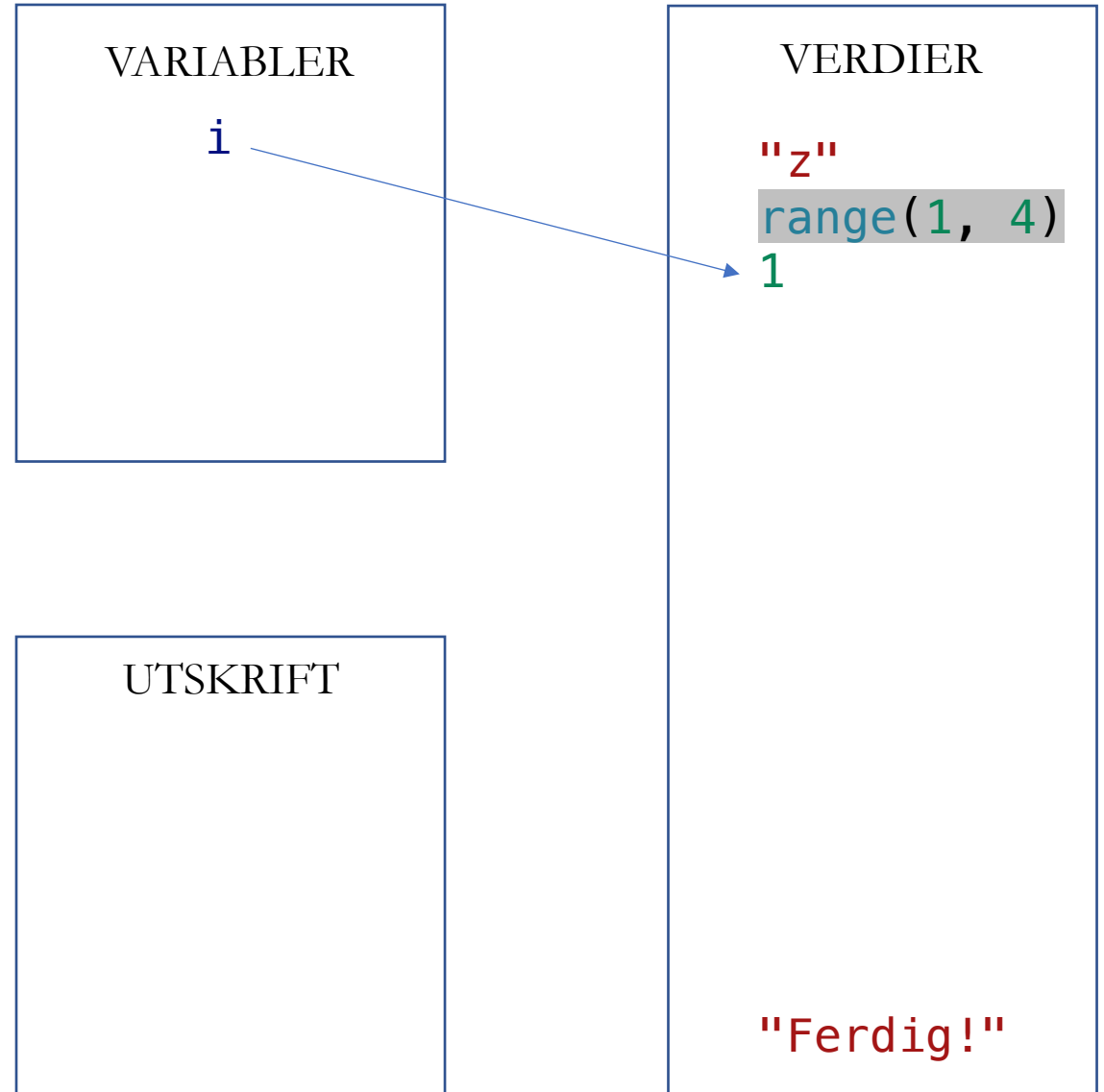
VERDIER

```
"z"  
range(1, 4)
```

```
"Ferdig!"
```

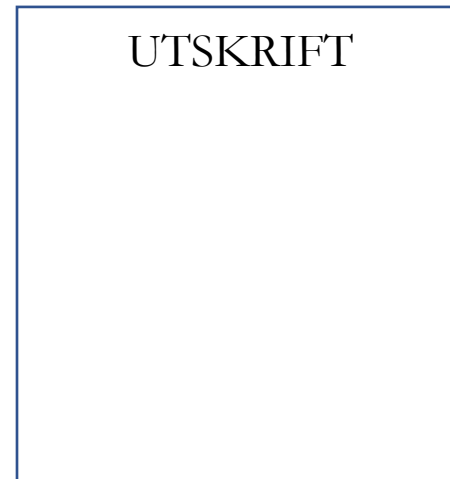
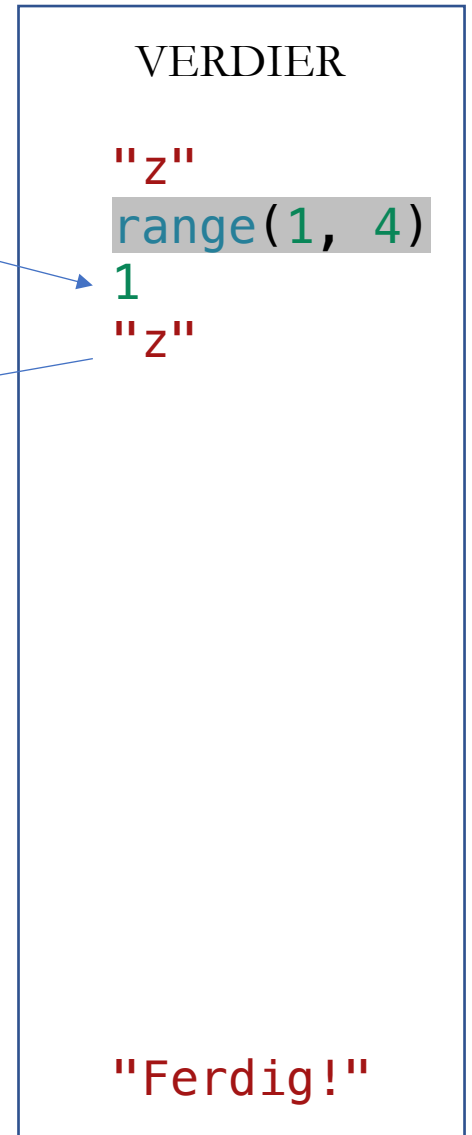
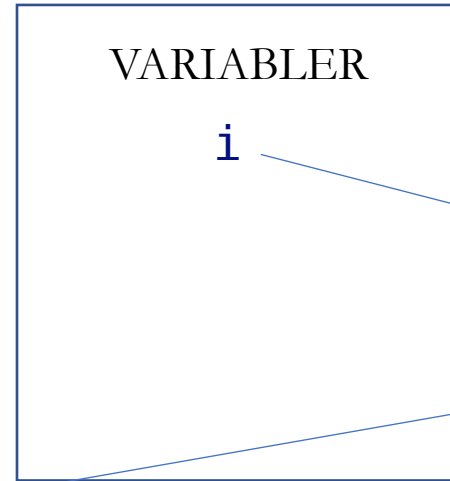

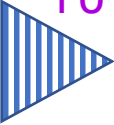
FOR

```
▶ for i in range(1, 4):  
    print("z" * i)  
print("Ferdig")
```

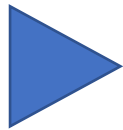


FOR

```
for i in range(1, 4):  
    print("z" * i)  
print("Ferdig")
```

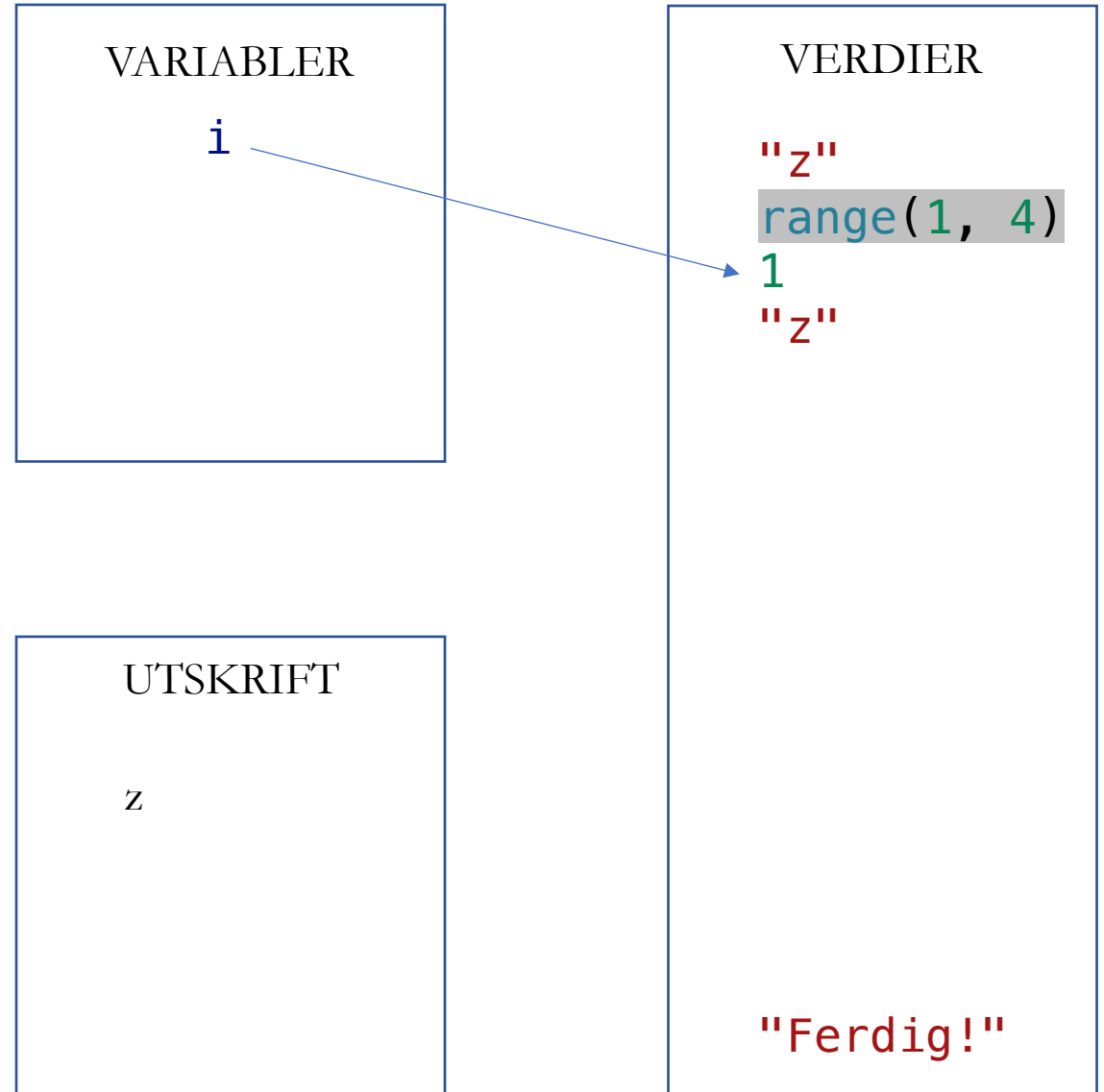


FOR



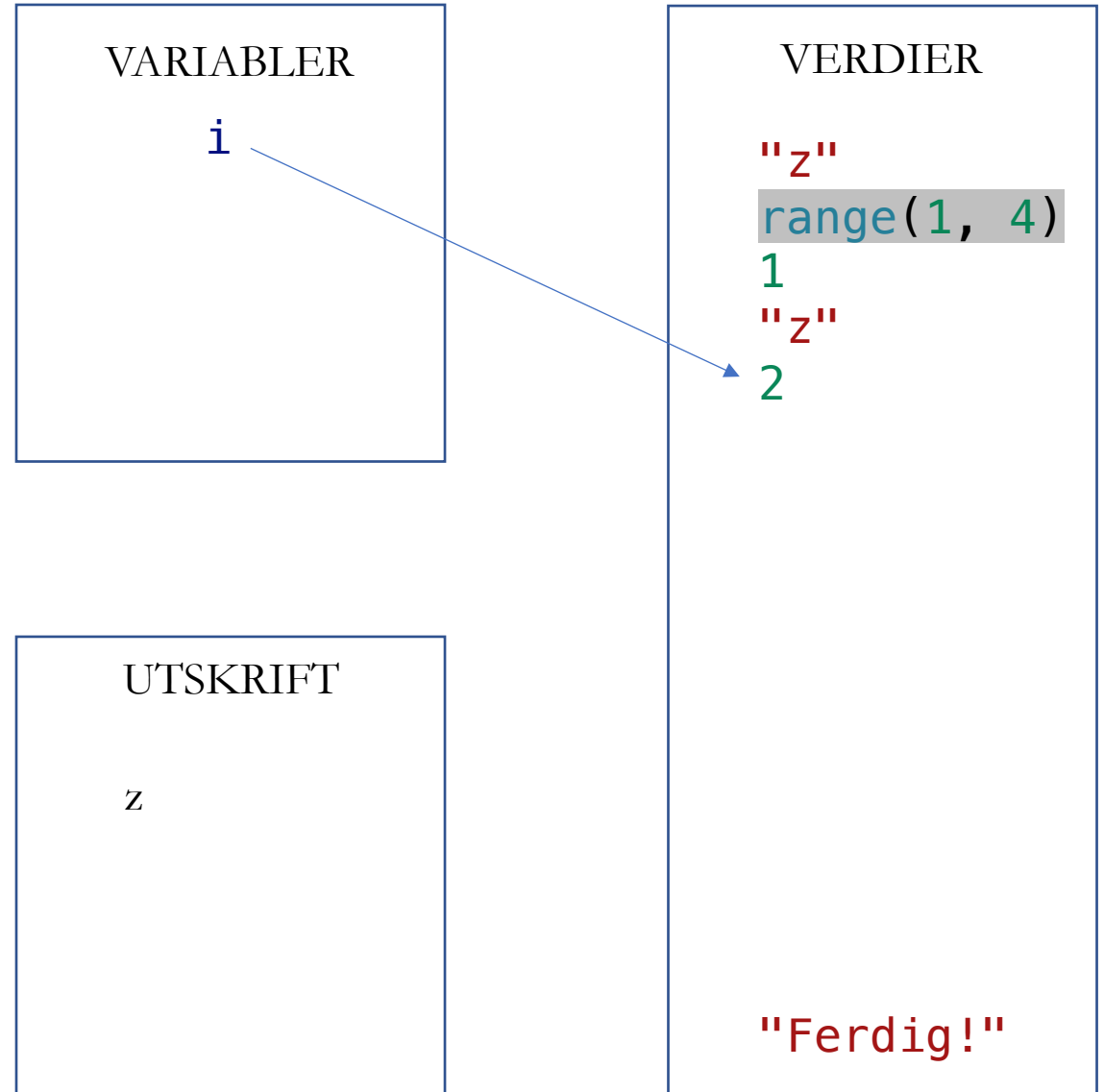
```
for i in range(1, 4):  
    print("z" * i)
```

```
print("Ferdig")
```



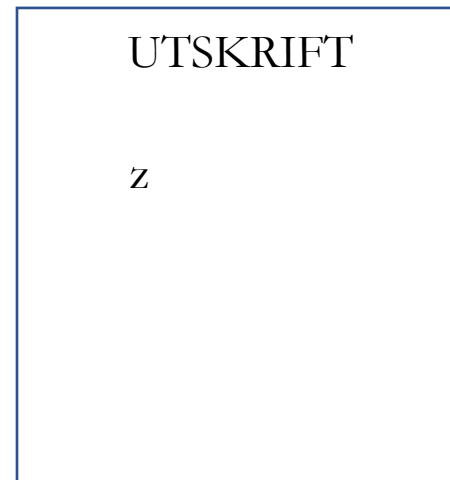
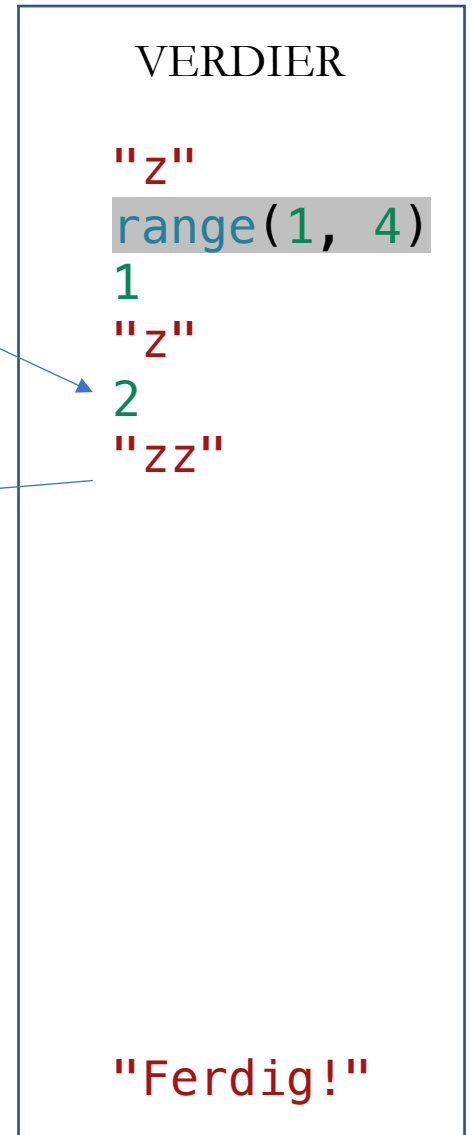
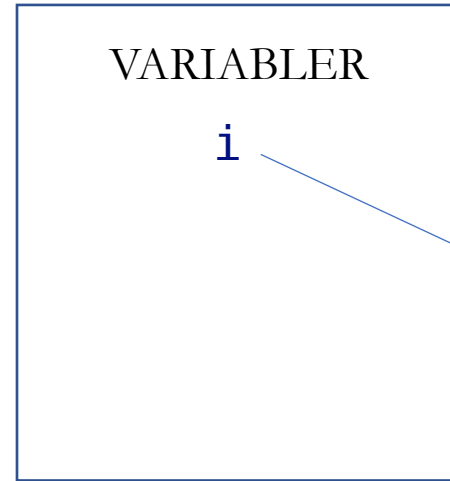

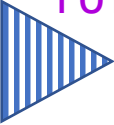
FOR

```
▶ for i in range(1, 4):  
    print("z" * i)  
print("Ferdig")
```



FOR

```
for i in range(1, 4):  
    print("z" * i)  
print("Ferdig")
```

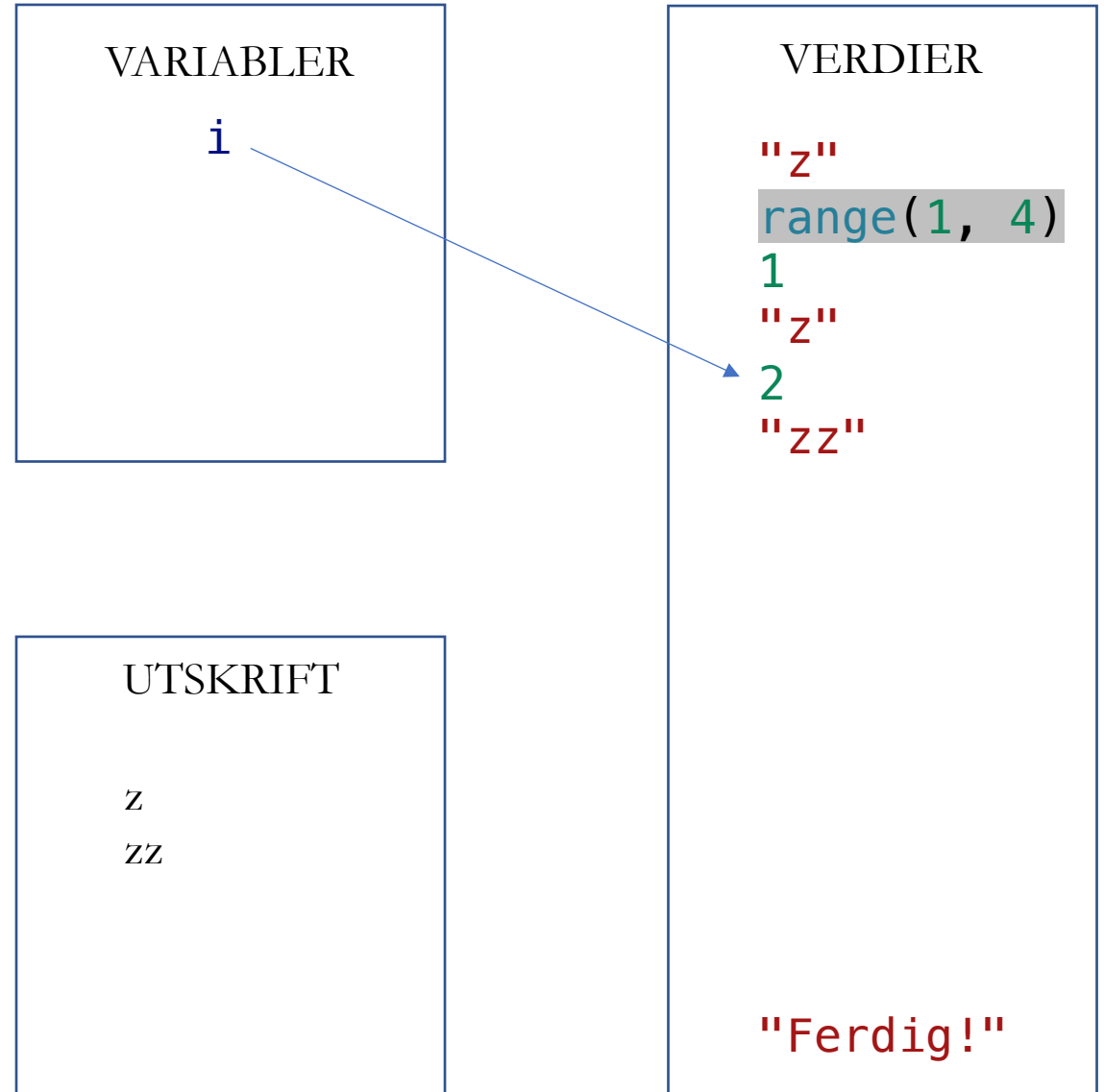


FOR



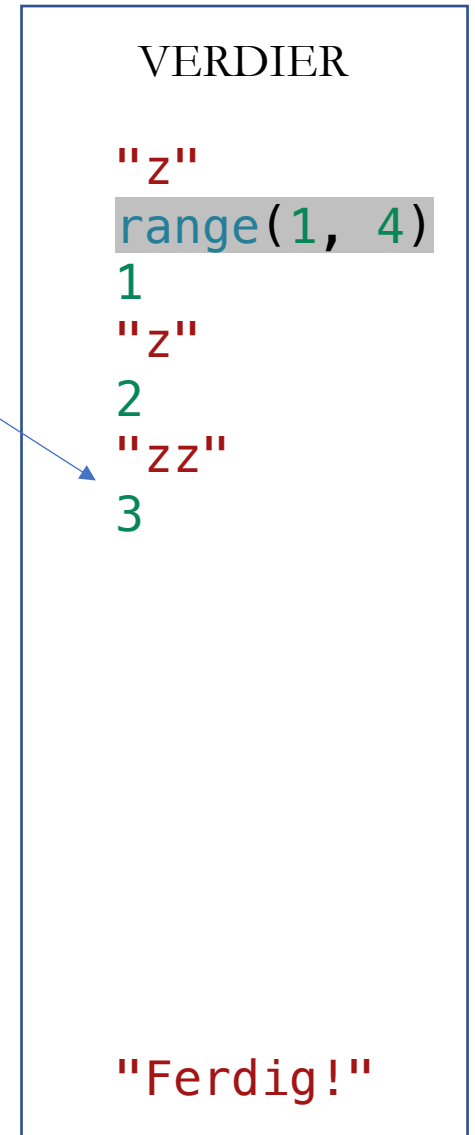
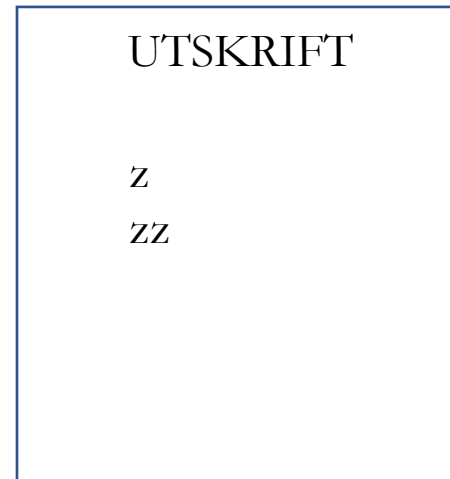
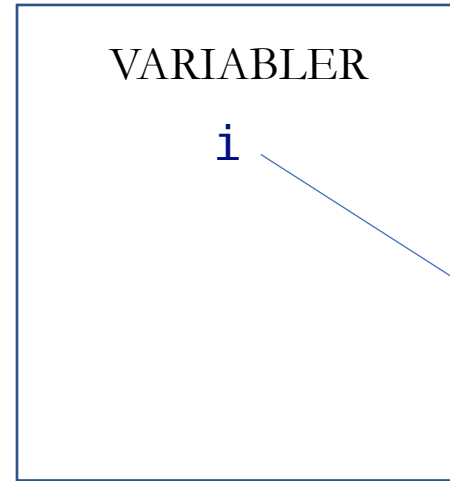
```
for i in range(1, 4):  
    print("z" * i)
```

```
print("Ferdig")
```



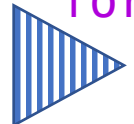
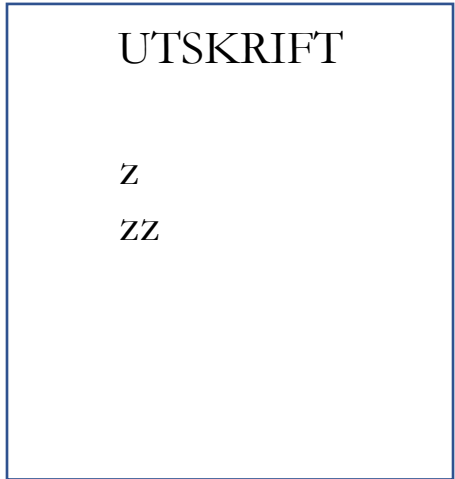
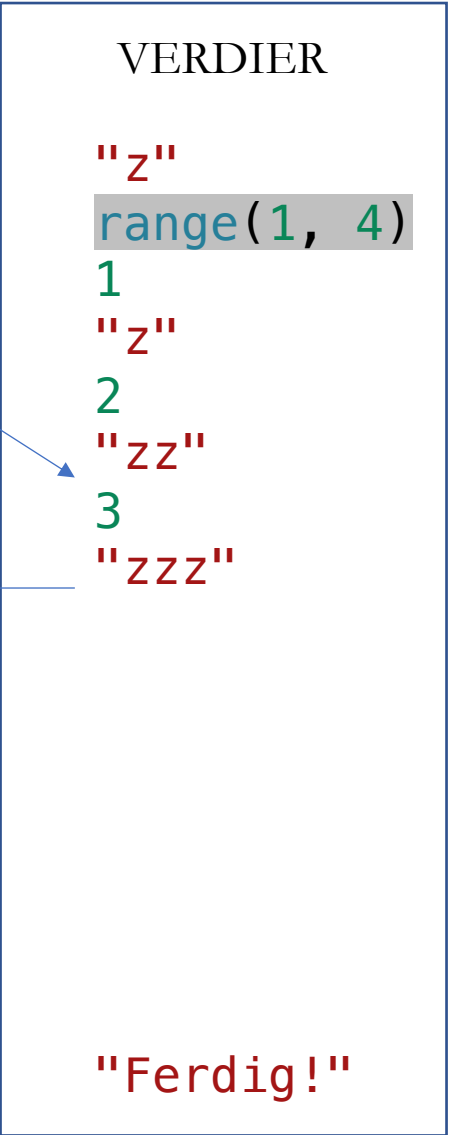
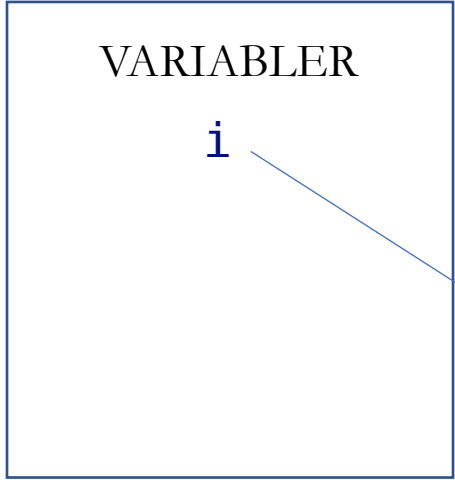
FOR

```
▶ for i in range(1, 4):  
    print("z" * i)  
print("Ferdig")
```



FOR

```
for i in range(1, 4):  
    print("z" * i)  
print("Ferdig")
```



VARIABLER

i

VERDIER

```
"z"  
range(1, 4)  
1  
"z"  
2  
"zz"  
3  
"zzz"
```

UTSKRIFT

```
z  
zz
```

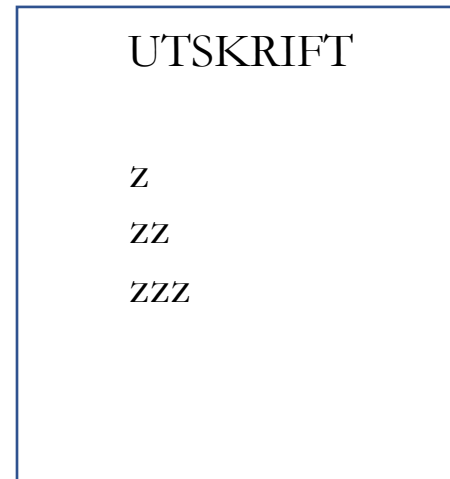
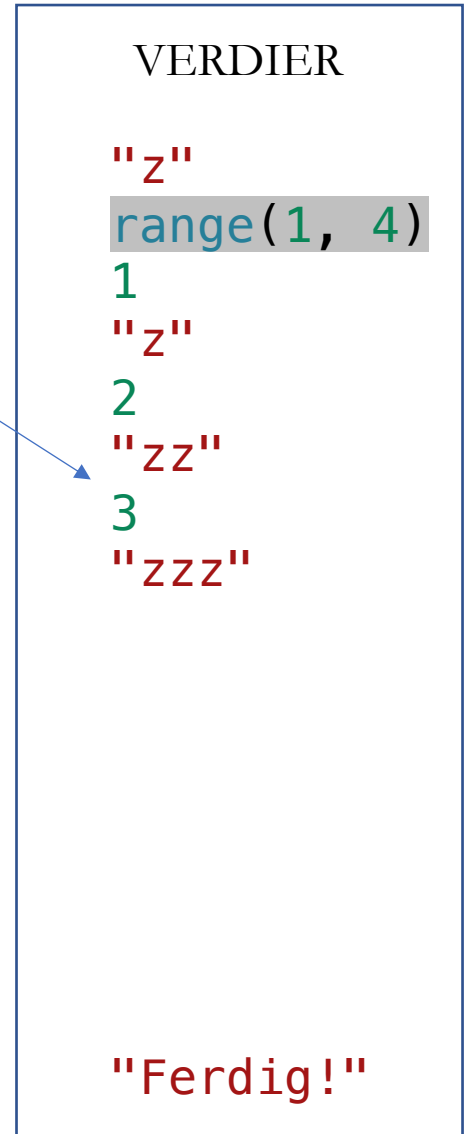
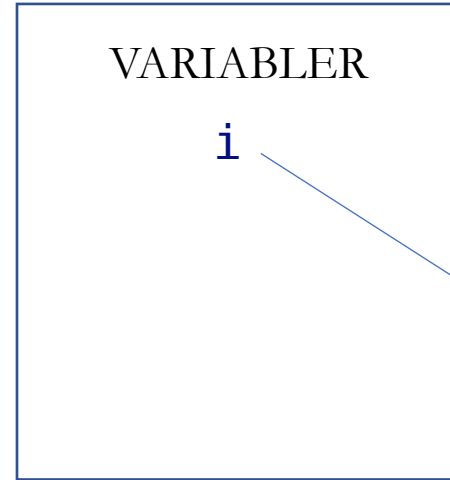
"Ferdig!"

FOR



```
for i in range(1, 4):  
    print("z" * i)
```

```
print("Ferdig")
```



FOR

```
for i in range(1, 4):  
    print("z" * i)
```



```
print("Ferdig")
```

VARIABLER

i

UTSKRIFT

z

zz

zzz

VERDIER

"z"

range(1, 4)

1

"z"

2

"zz"

3

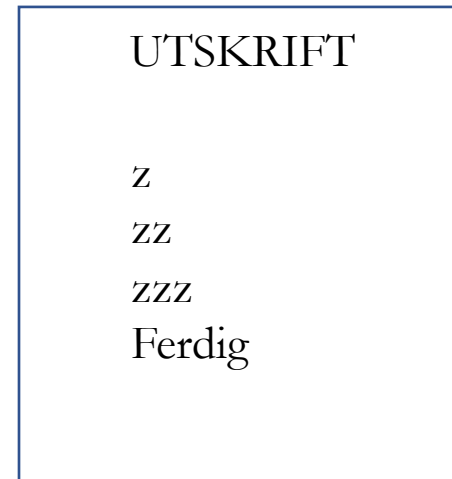
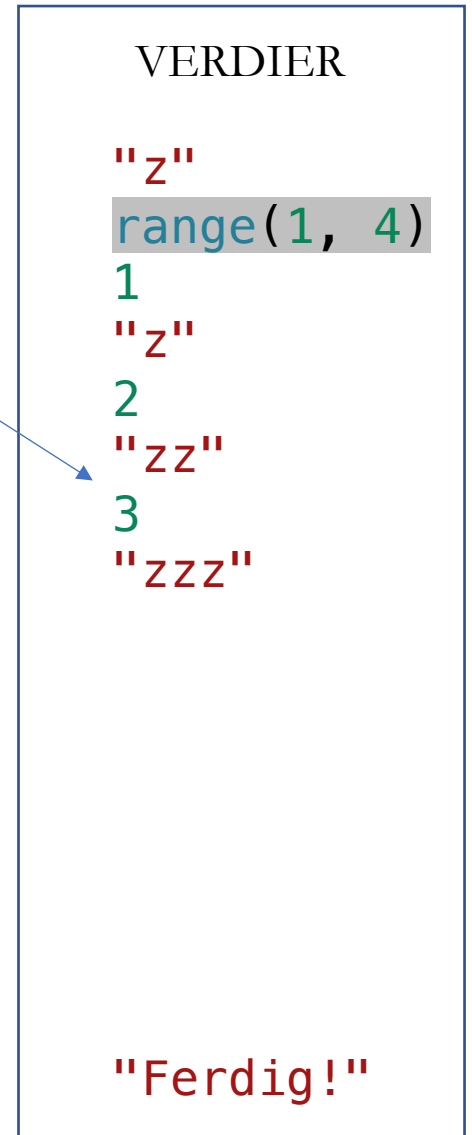
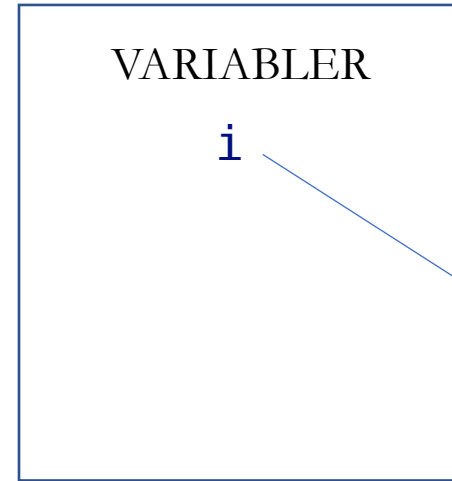
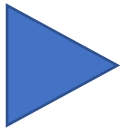
"zzz"

"Ferdig!"

FOR

```
for i in range(1, 4):  
    print("z" * i)
```

```
print("Ferdig")
```



FAKTOR

- Er 4 en faktor i 209414732?
- Hva er største faktor i 209414732?

PRIMTALL

- Er et tall et primtall eller ikke?
- Hva er det n 'te primtallet?