LAB2

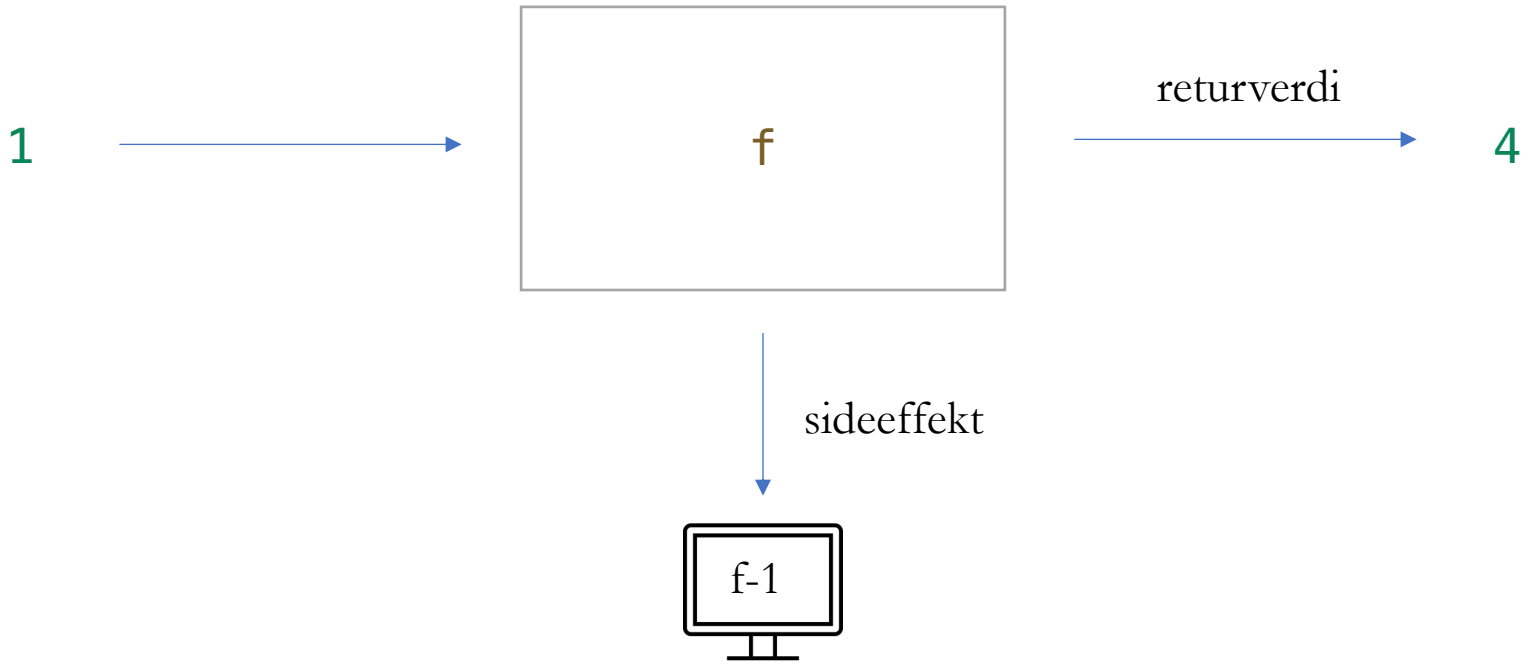
- return vs print
- if-if-if vs. if-elif-elif
- programflyt

- du må ha 100% riktig på CodeGrade for å få bestått
- du kan levere så mange ganger du vil (siste innlevering telles)
- ekstra gruppetimer i dag fra 16 – 18

SIDEEFFEKT vs. RETURVERDI

```
def f(x):
    print(f"f-{x}")
    x += 1
    return 2*x
```

```
f(1)
```



IF-IF-IF vs. IF-ELIF-ELIF

```
def print_longest_word(s1, s2, s3):
  longest_len = max(len(s1), len(s2), len(s3))
  if len(s1) == longest_len:
    print(s1)
  if len(s2) == longest_len:
    print(s2)
  if len(s3) == longest_len:
    print(s3)

print_longest_word("five", "four", "nine")
```

```
five
four
nine
```

```
def print_longest_word(s1, s2, s3):
  longest_len = max(len(s1), len(s2), len(s3))
  if len(s1) == longest_len:
    print(s1)
  elif len(s2) == longest_len:
    print(s2)
  elif len(s3) == longest_len:
    print(s3)

print_longest_word("five", "four", "nine")
```

```
five
```

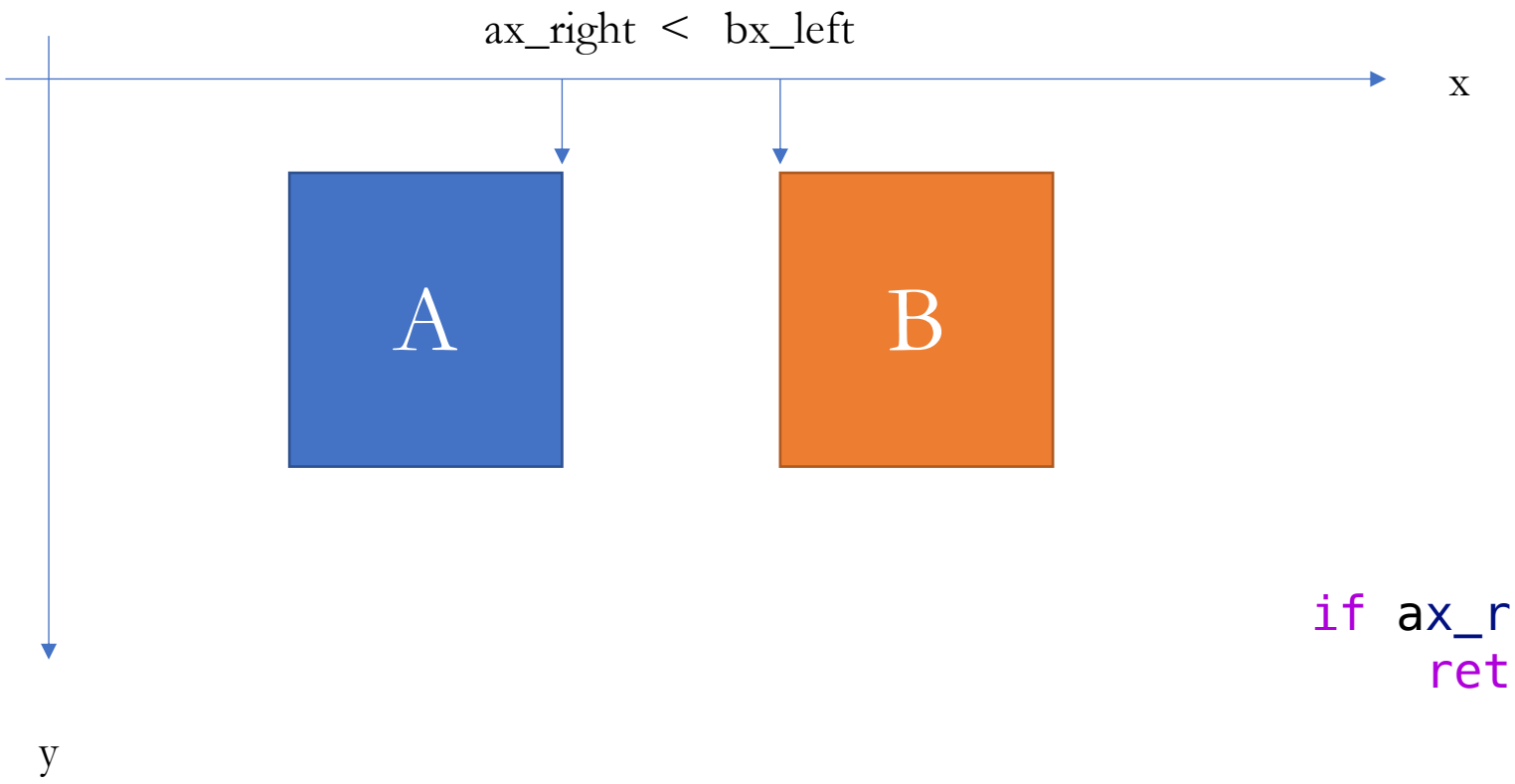
PROGRAMFLYT

```
def is_leap_year(year):
  if year % 4 == 0:
    return True
  if year % 100 == 0:
    return False
  if year % 400 == 0:
    return True
  else:
    return False
```

```
def is_leap_year(year):
  if year % 4 == 0:
    if year % 100 == 0:
      if year % 400 == 0:
        return True
      return False
    return True
  return False
```

PROBLEMLØSNING

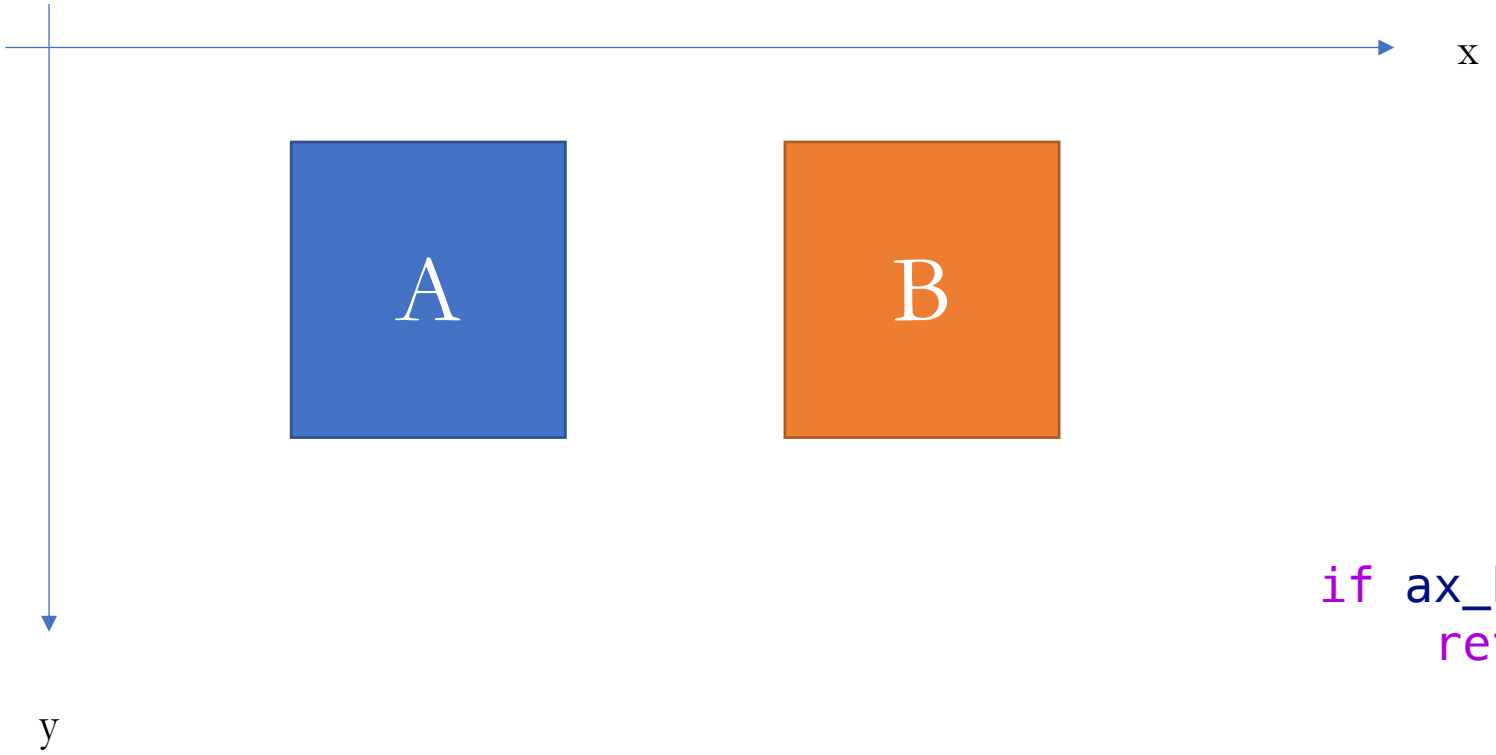
- To rektangler overlapper *ikke* hvis




```
if ax_right < bx_left:
    return False
```

PROBLEMLØSNING

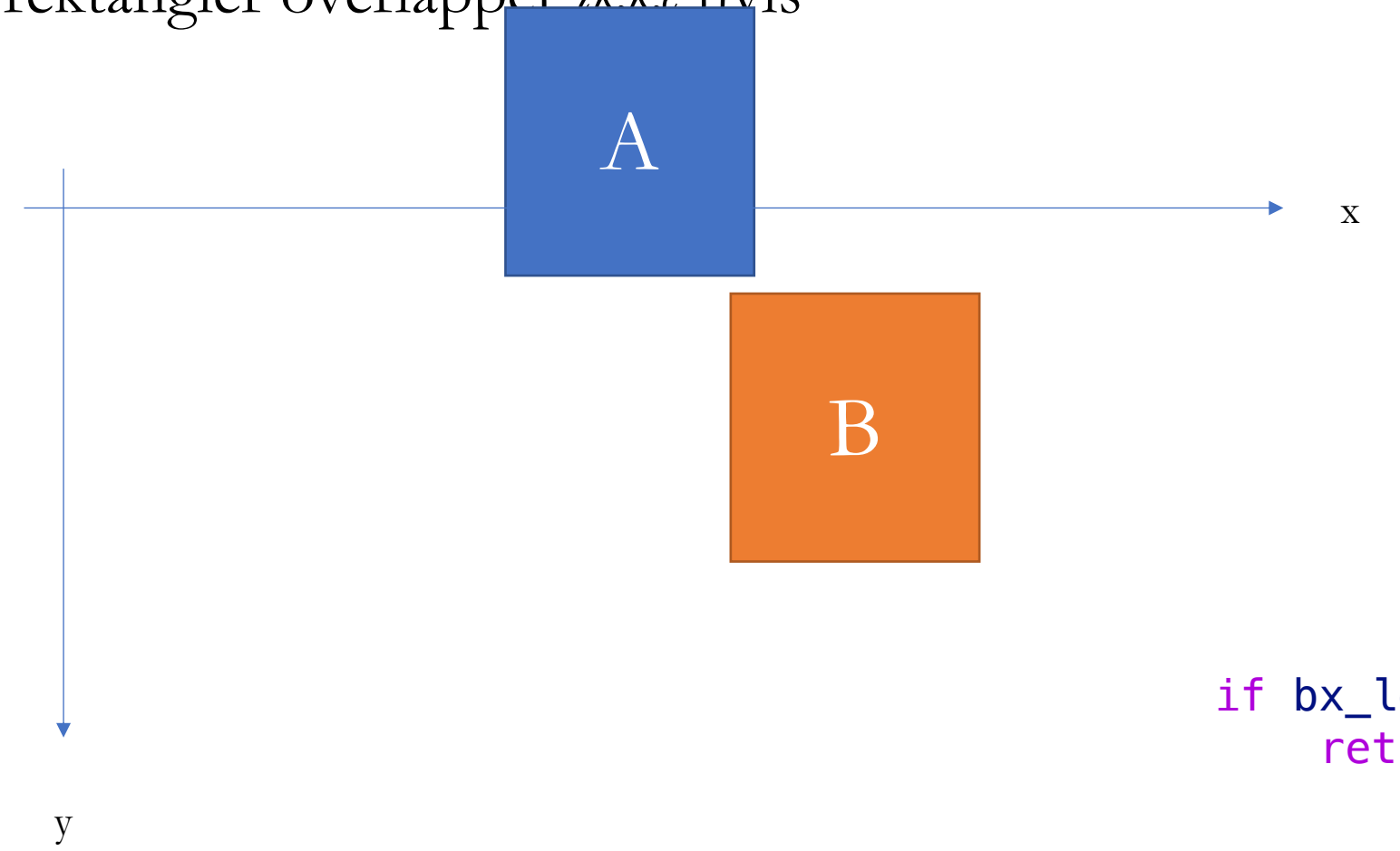
- To rektangler overlapper *ikke* hvis




```
if ax_bottom < bx_top:
    return False
```

PROBLEMLØSNING

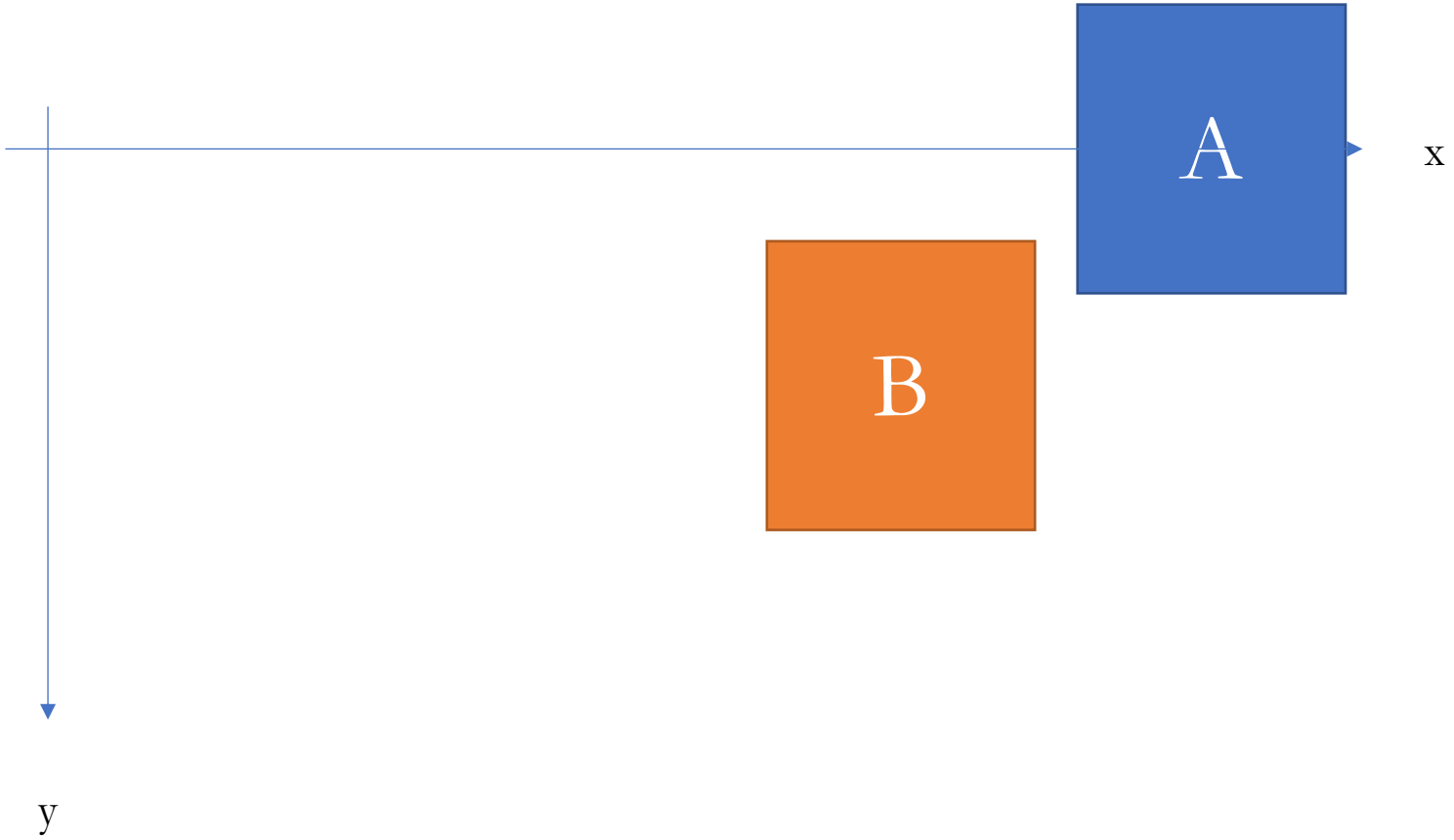
- To rektangler overlapper *ikke* hvis




```
if bx_left < ax_right:
    return False
```

PROBLEMLØSNING

- To rektangler overlapper *ikke* hvis



HJELP!

SENTRALE BEGREPER

- Verdier
  - Data lagret i minnet (RAM)
- Type
  - Alle verdier har en type (int, str, float, bool)
- Variabel
  - En navngitt referanse til en verdi
  - Tilordnes verdi med =
- Uttrykk og operasjoner
  - Et regnestykke som evaluerer til en verdi
  - Presedens og evaluering
- Funksjoner
  - Returverdi vs sideeffekter
- Betinget oppførsel
  - if/elif/else
- Løkker
  - while
  - for
    - range
- Lister
- Oppslagsverk og mengder

KODESPORING

```
def f(x):
    x += 1
    return 2 * x

def g(x):
    y = f(x)
    y += f(x)
    return f(y)

print(g(f(1)))
```

KODESPORING

```
def f(x):
    y = 0
    if x < 10:
        y = 10
    elif x < 20:
        y += 20
    else:
        y = x + 1
    return y

x = f(1)
y = 3*x
x += f(y + 1)
print(x)
```

UNIVERSITETET I BERGEN

LØKKER

INF100

HØST 2022

Torstein Strømme

WHILE

```
bla()
bla()

while <betingelse>:
        bla()
        bla()
        bla()

bla()
bla()
```

uttrykk

så lenge betingelsen er True

WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```

VARIABLER

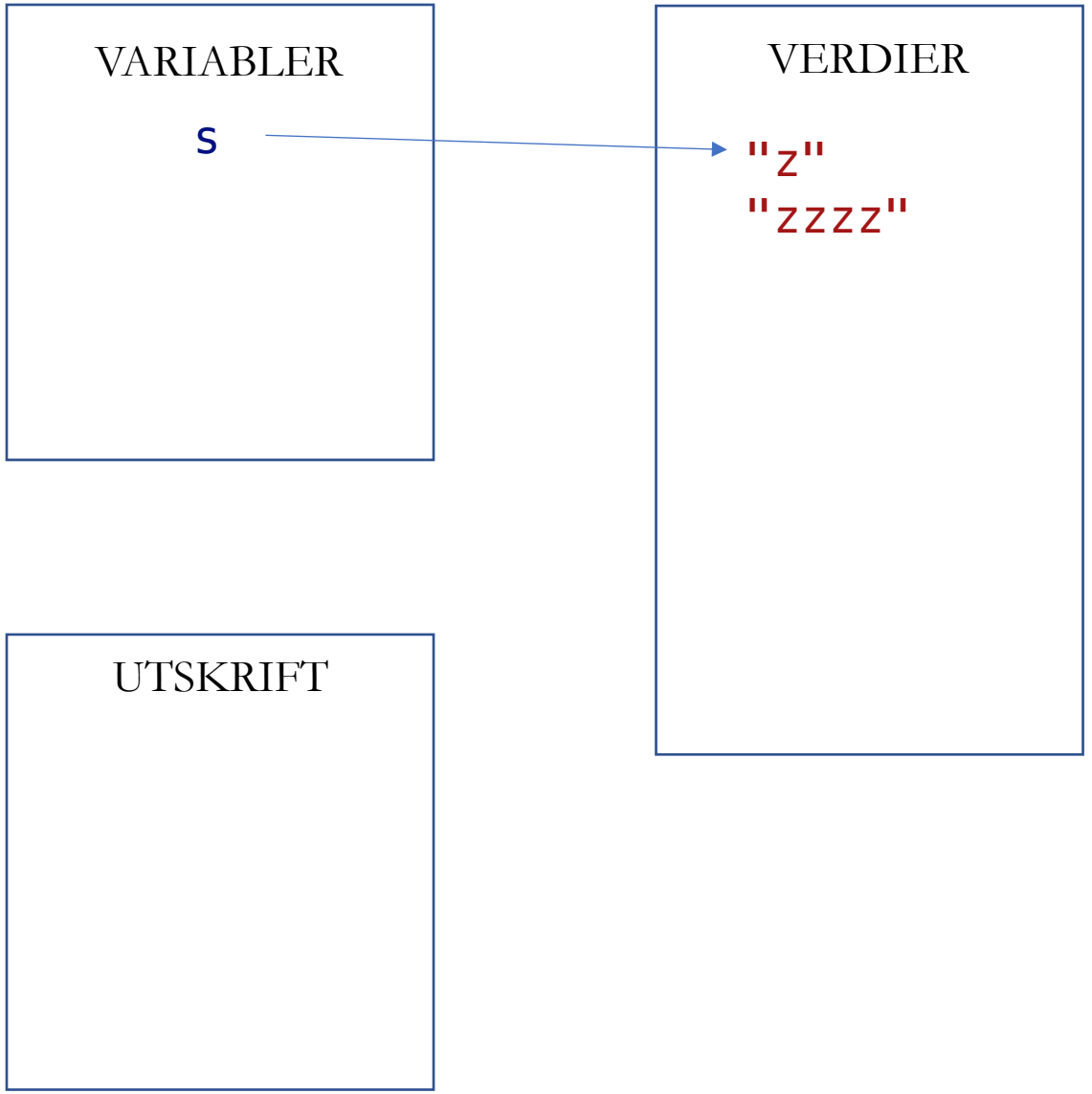
VERDIER

"z"
"zzzz"

UTSKRIFT

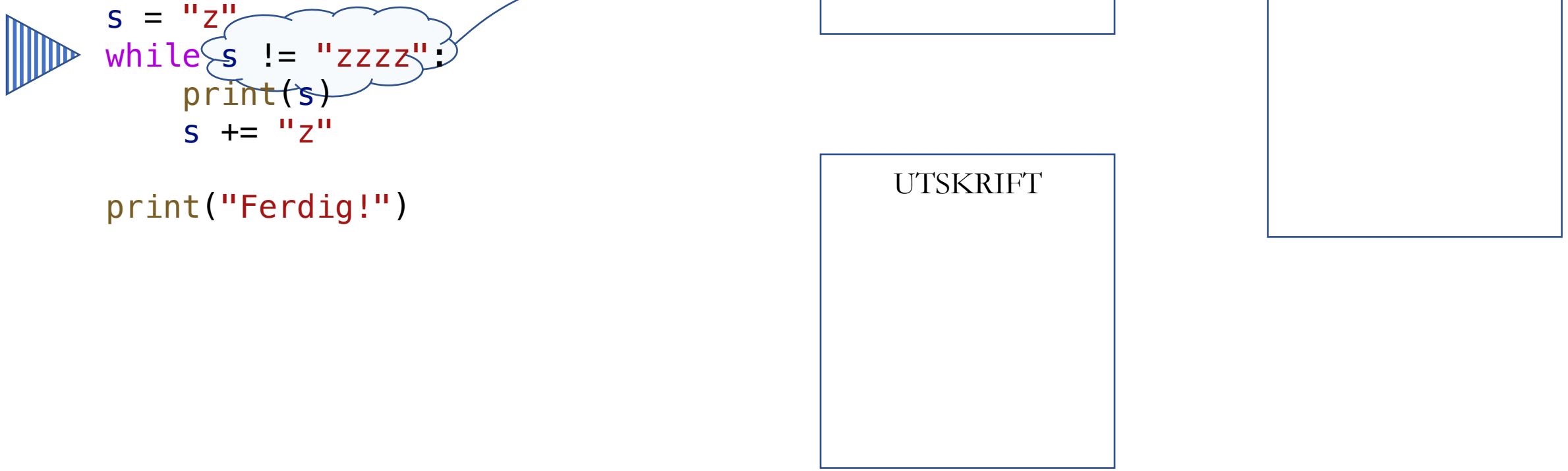
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



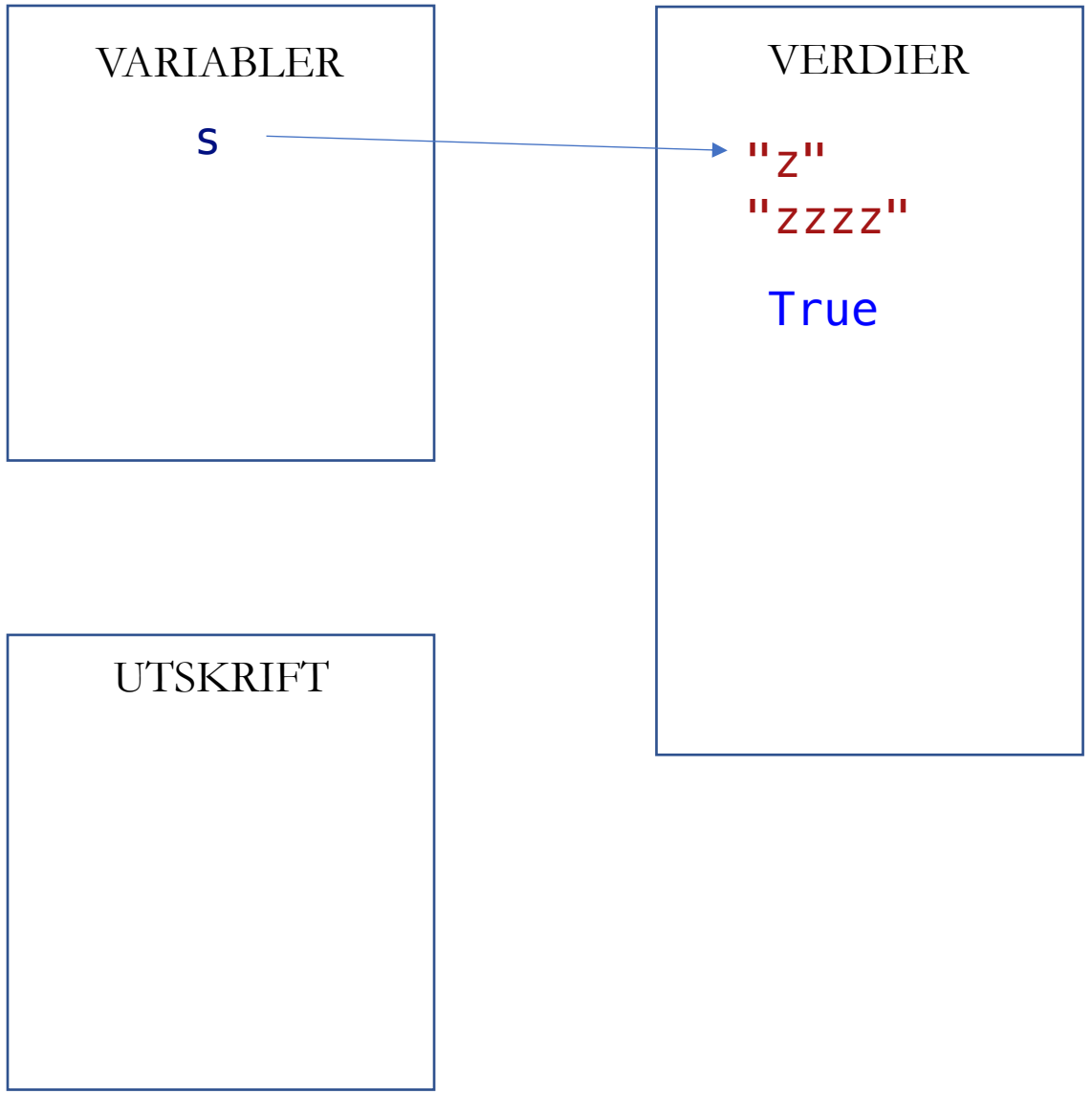
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



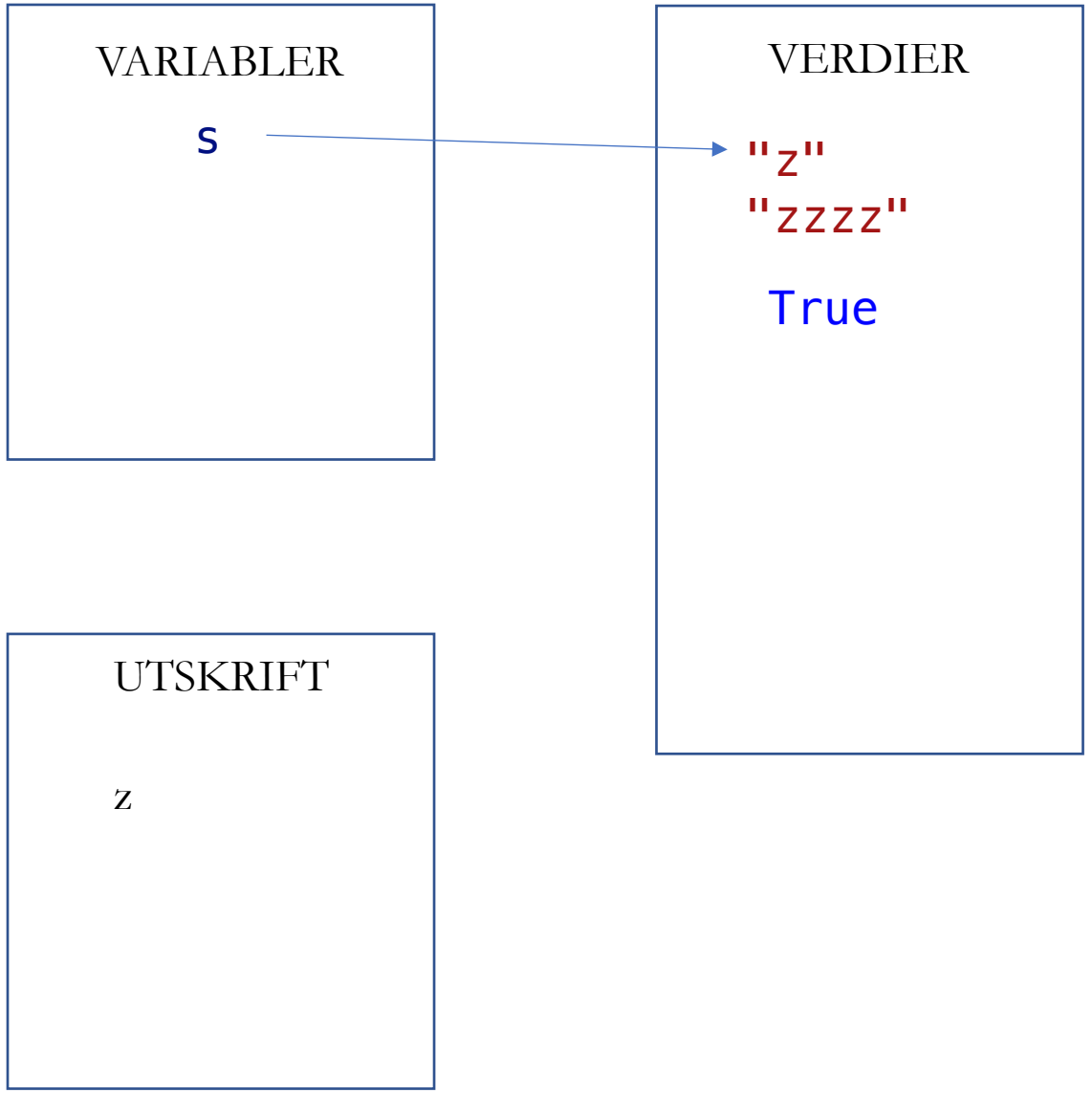
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



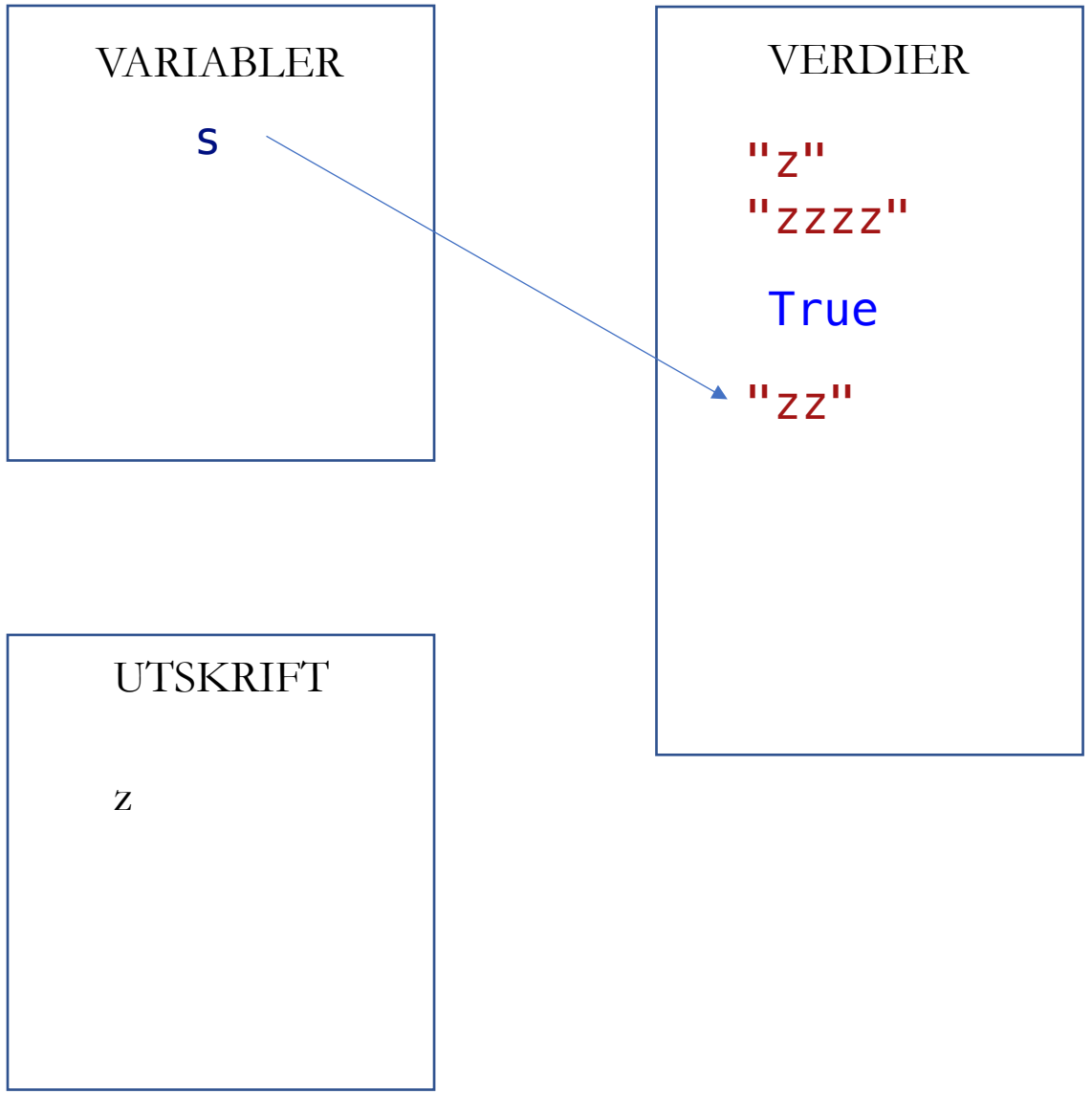
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



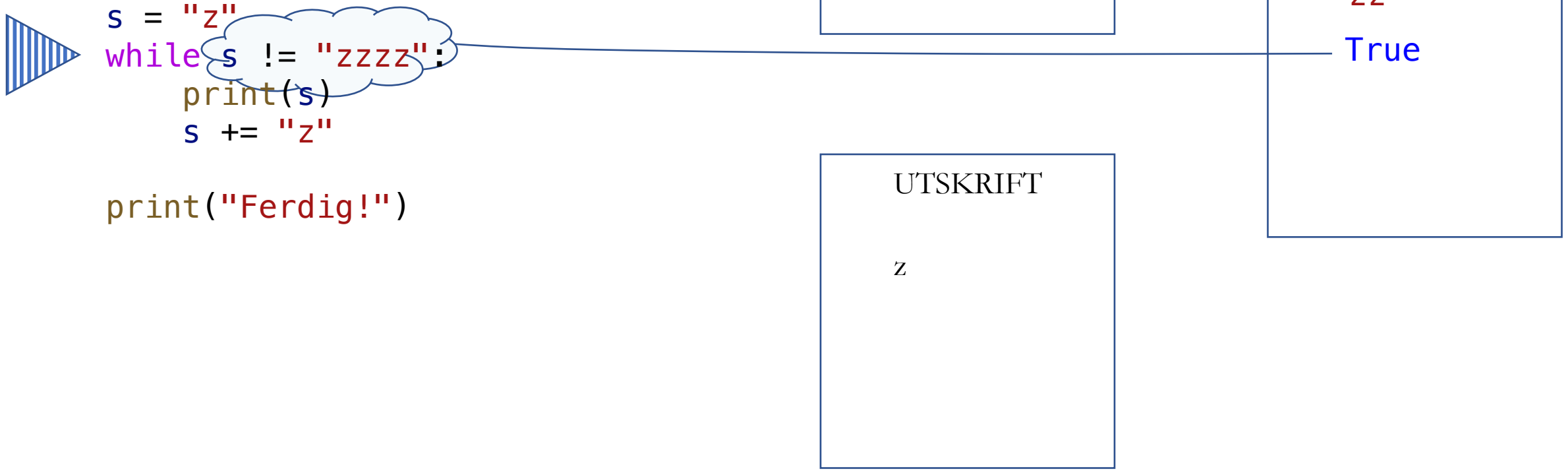
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



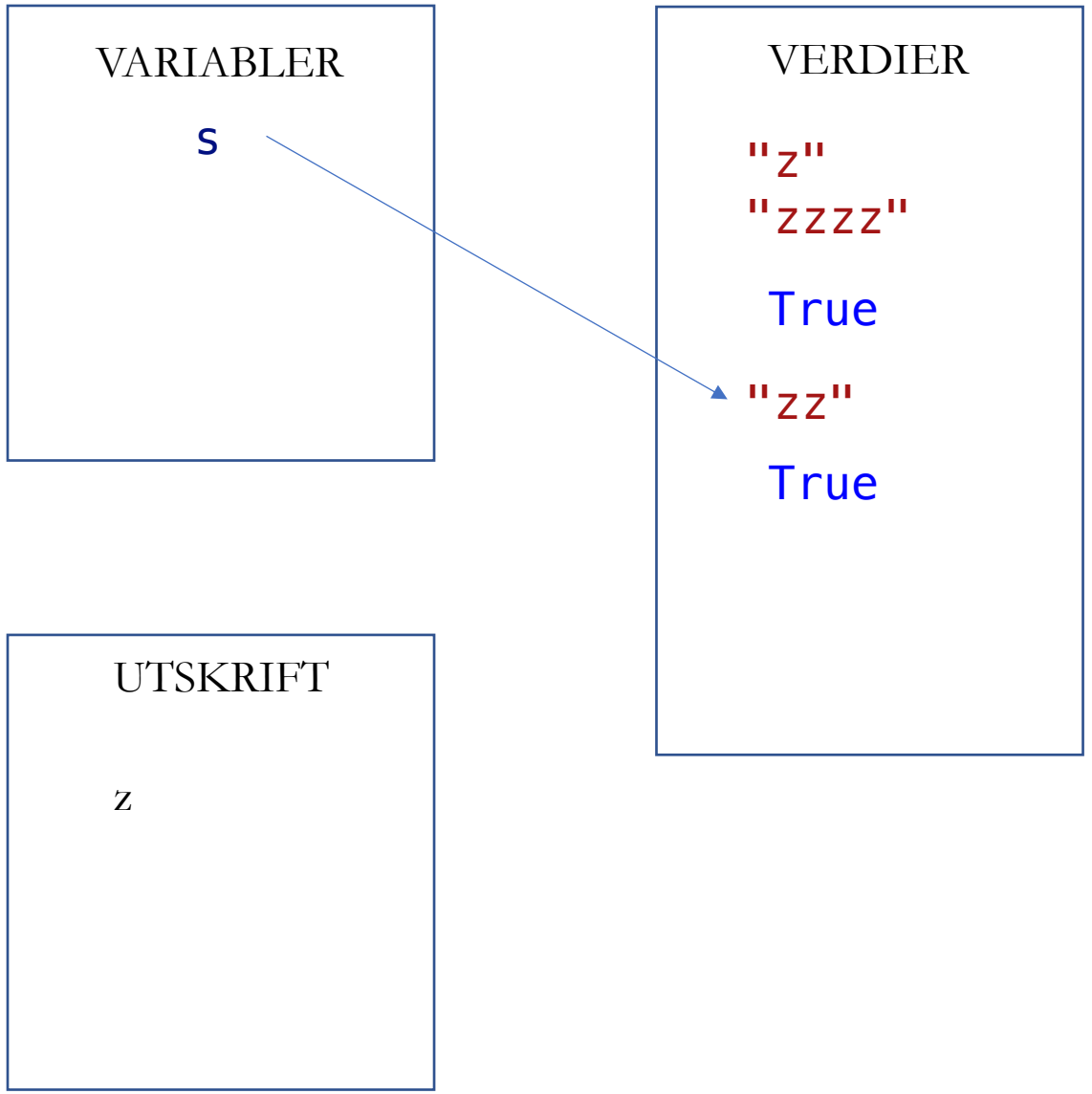
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



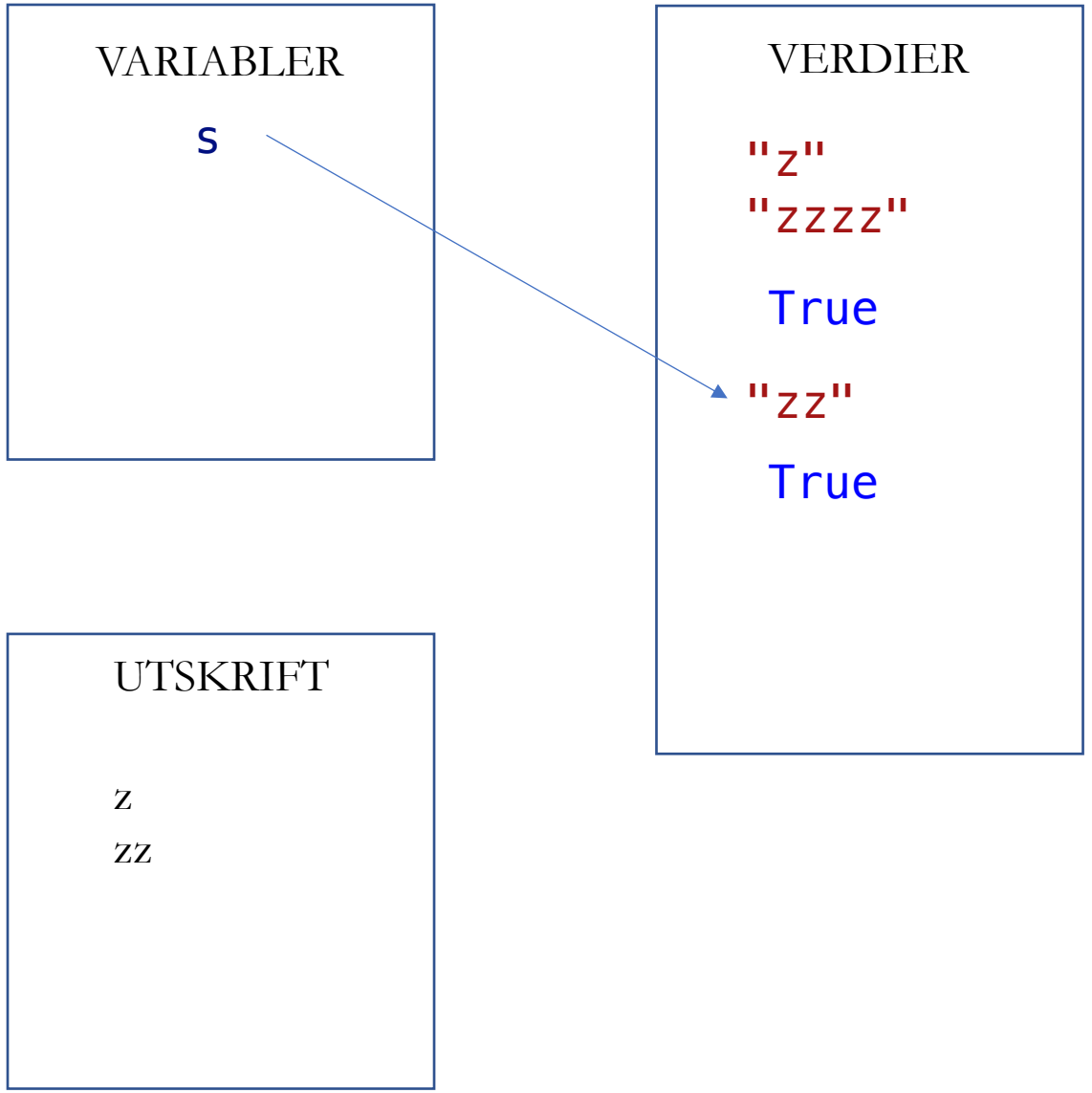
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



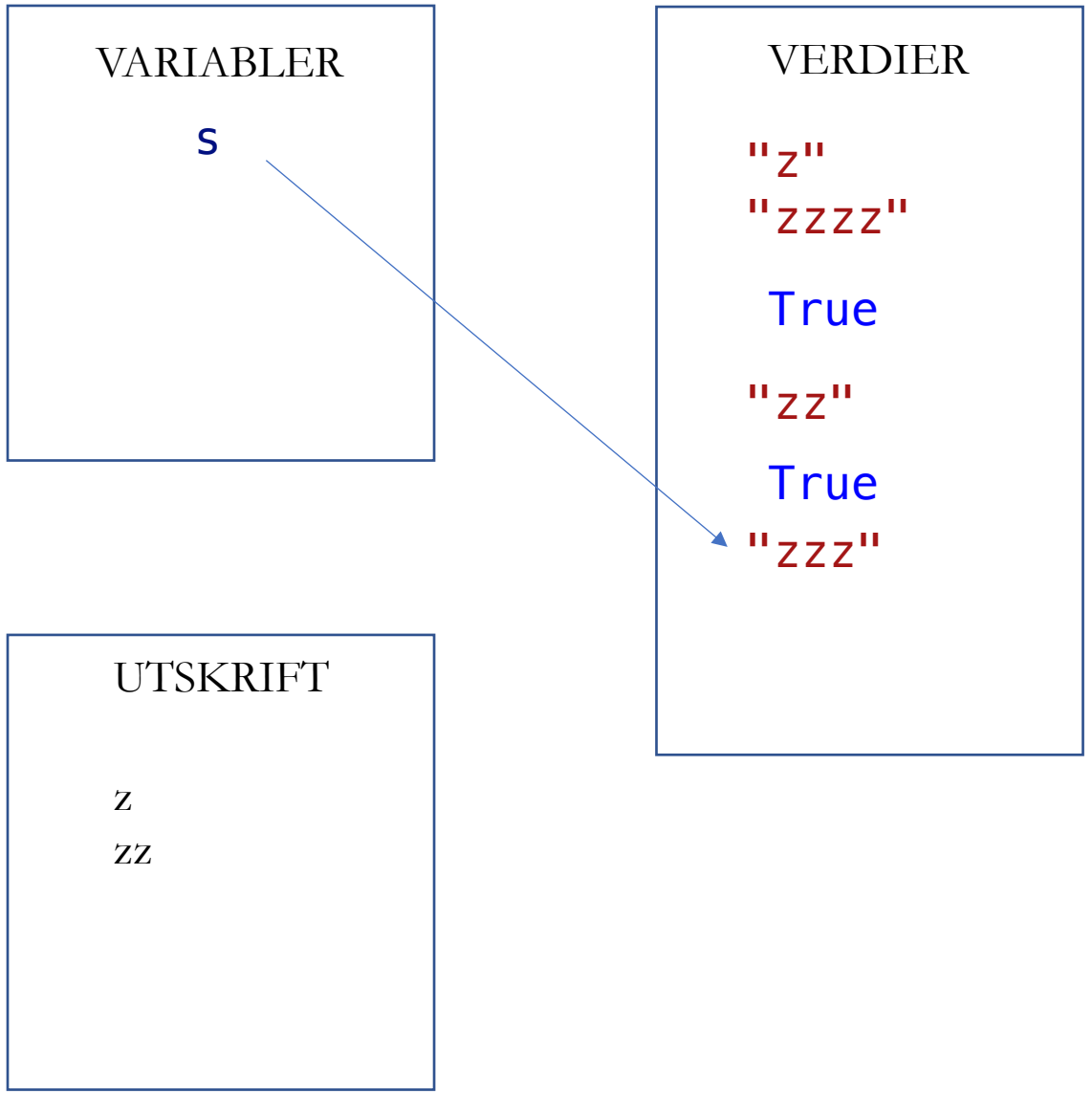
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



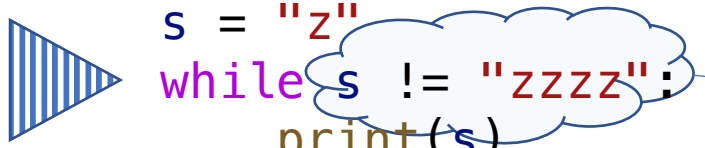
WHILE

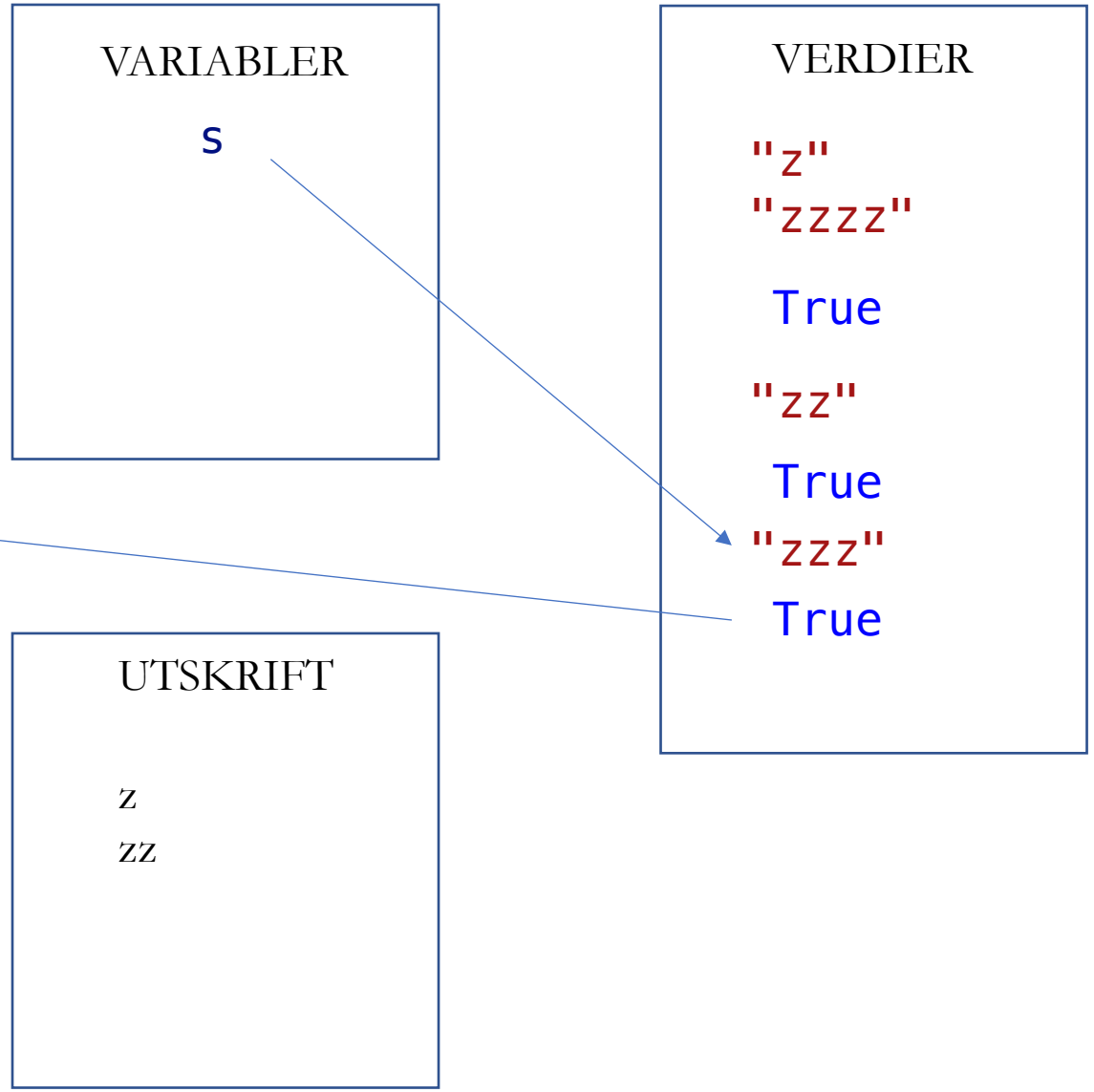
```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



WHILE

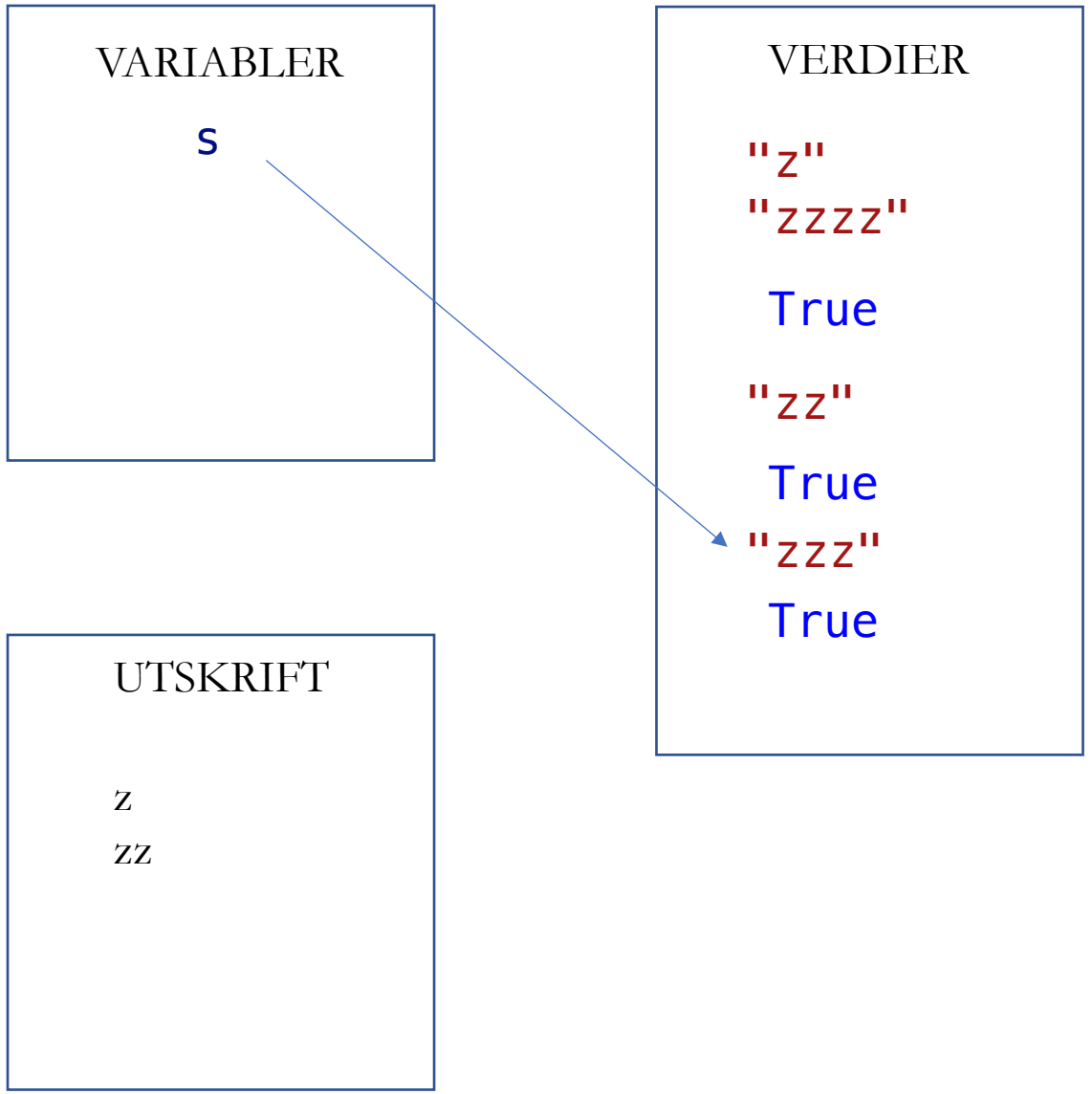
```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```





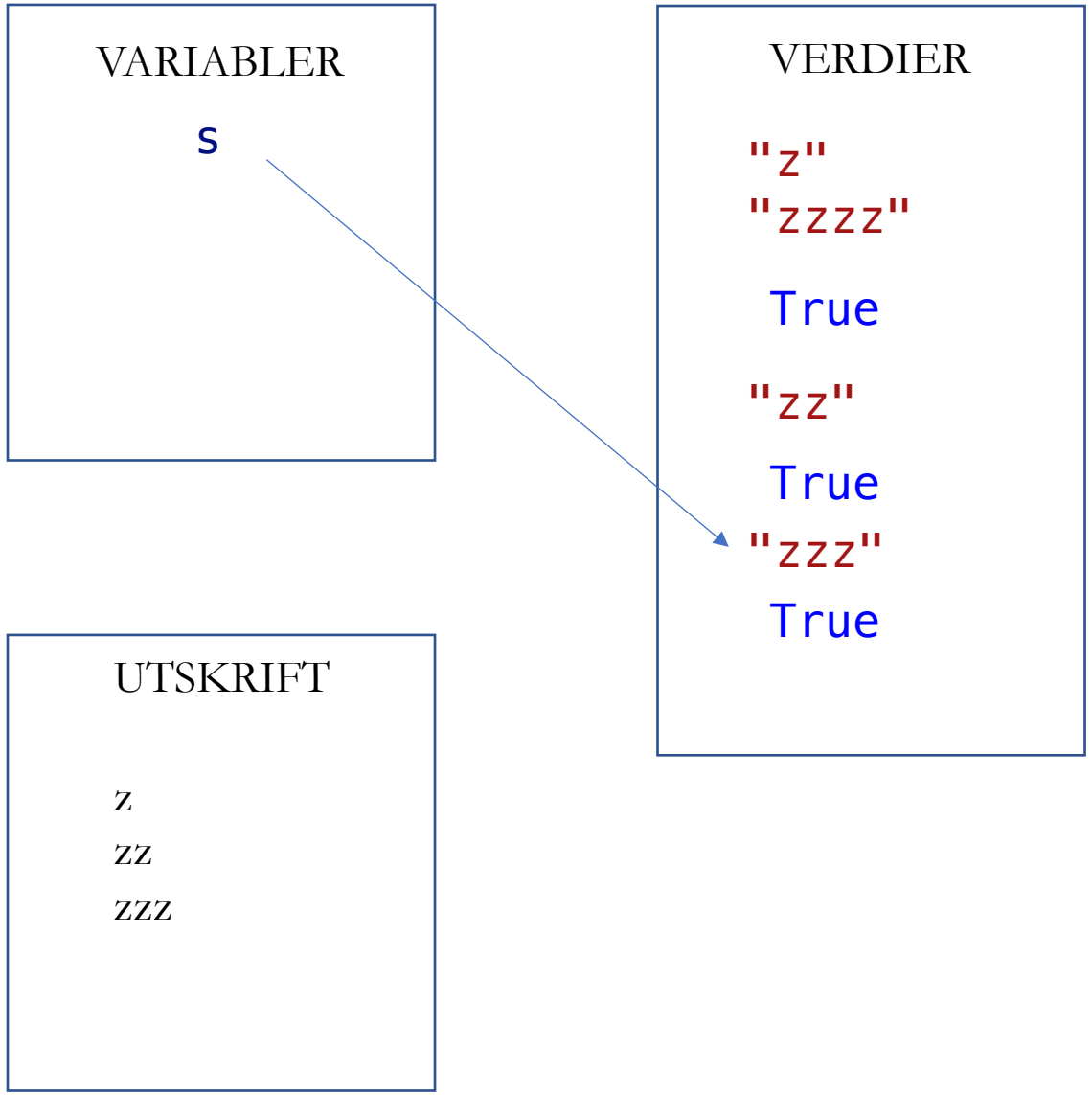
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



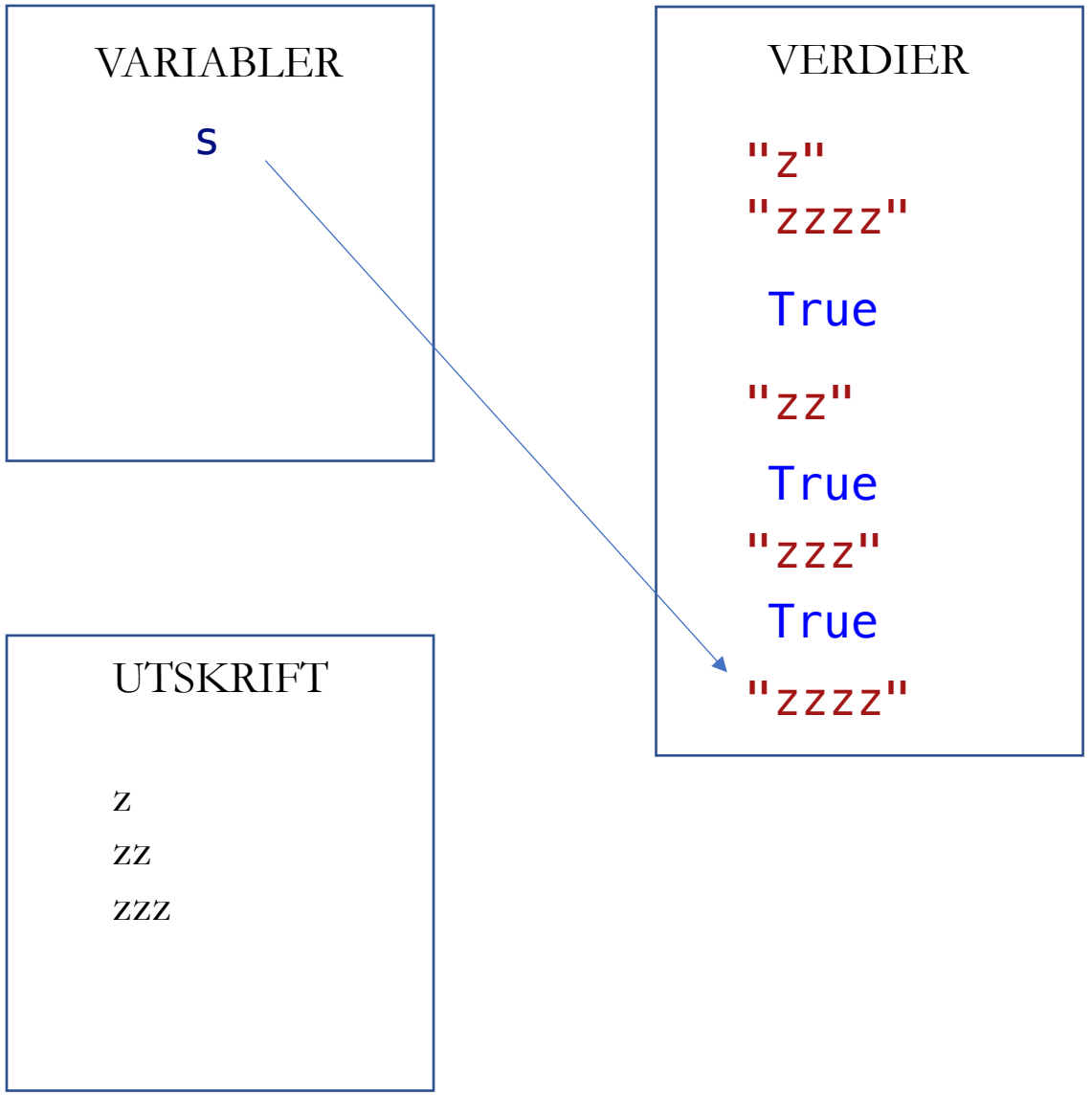
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



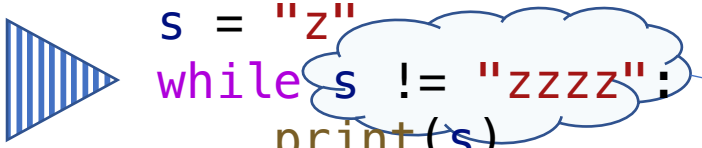
WHILE

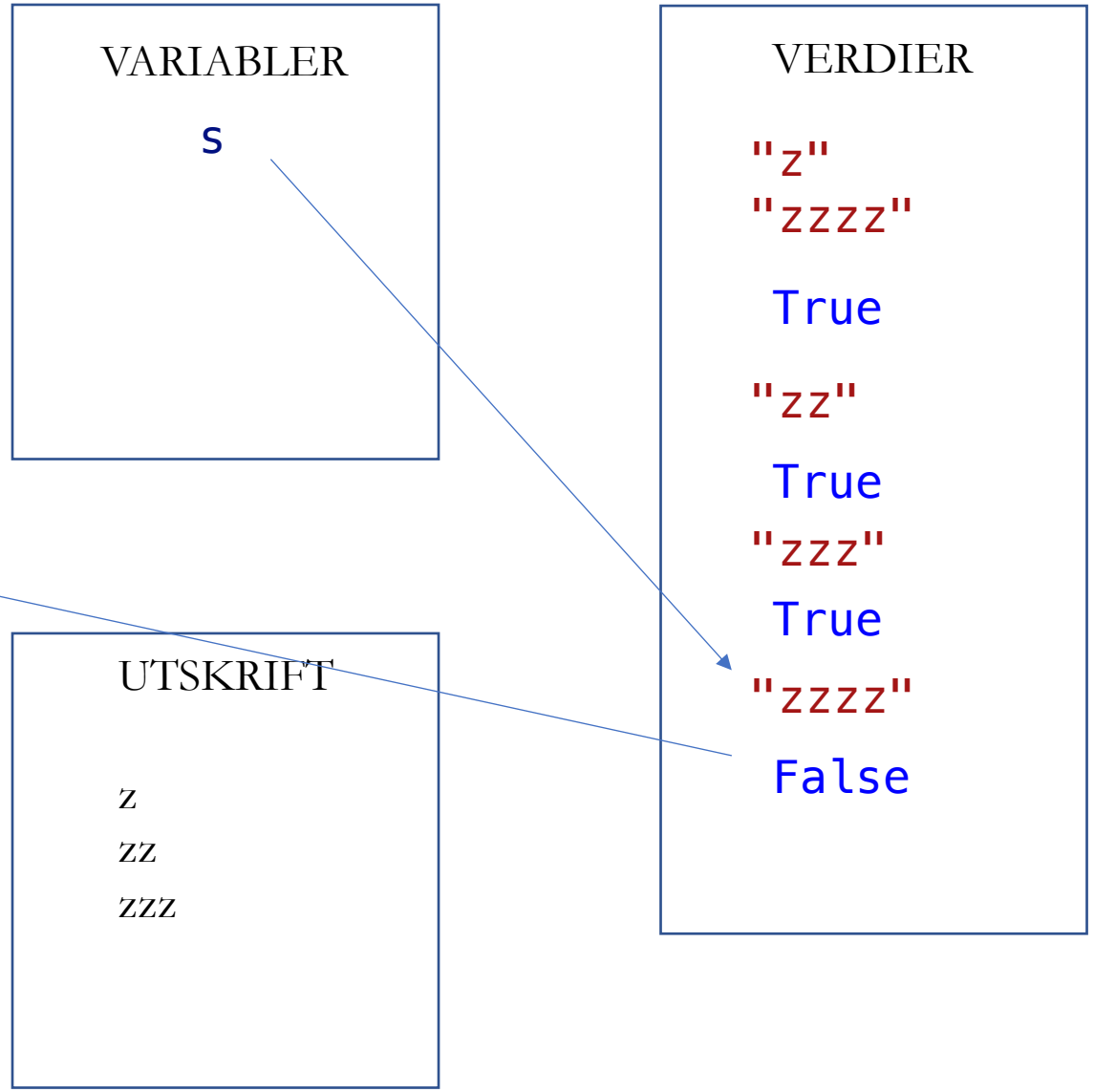
```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



WHILE

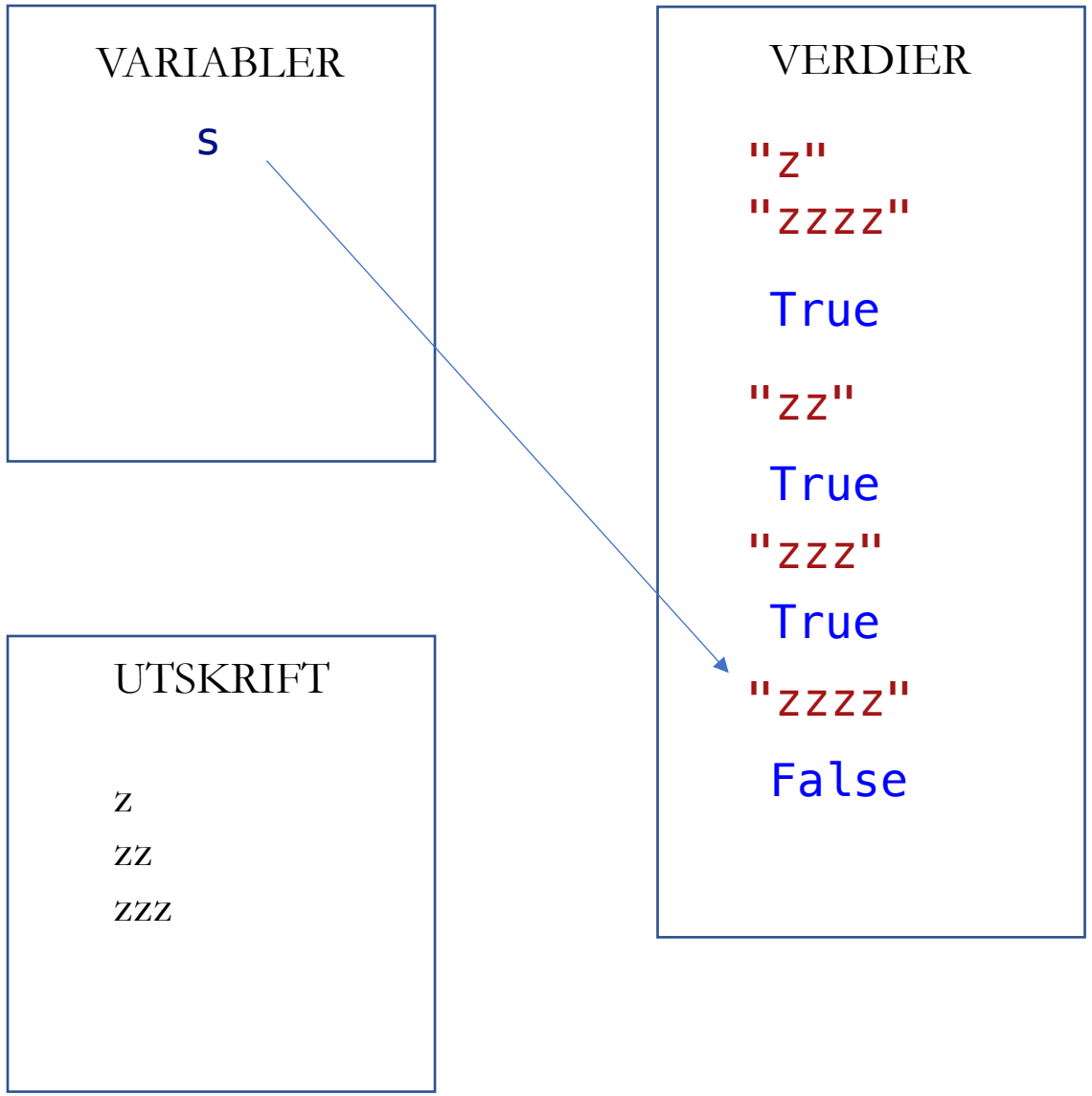
```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```





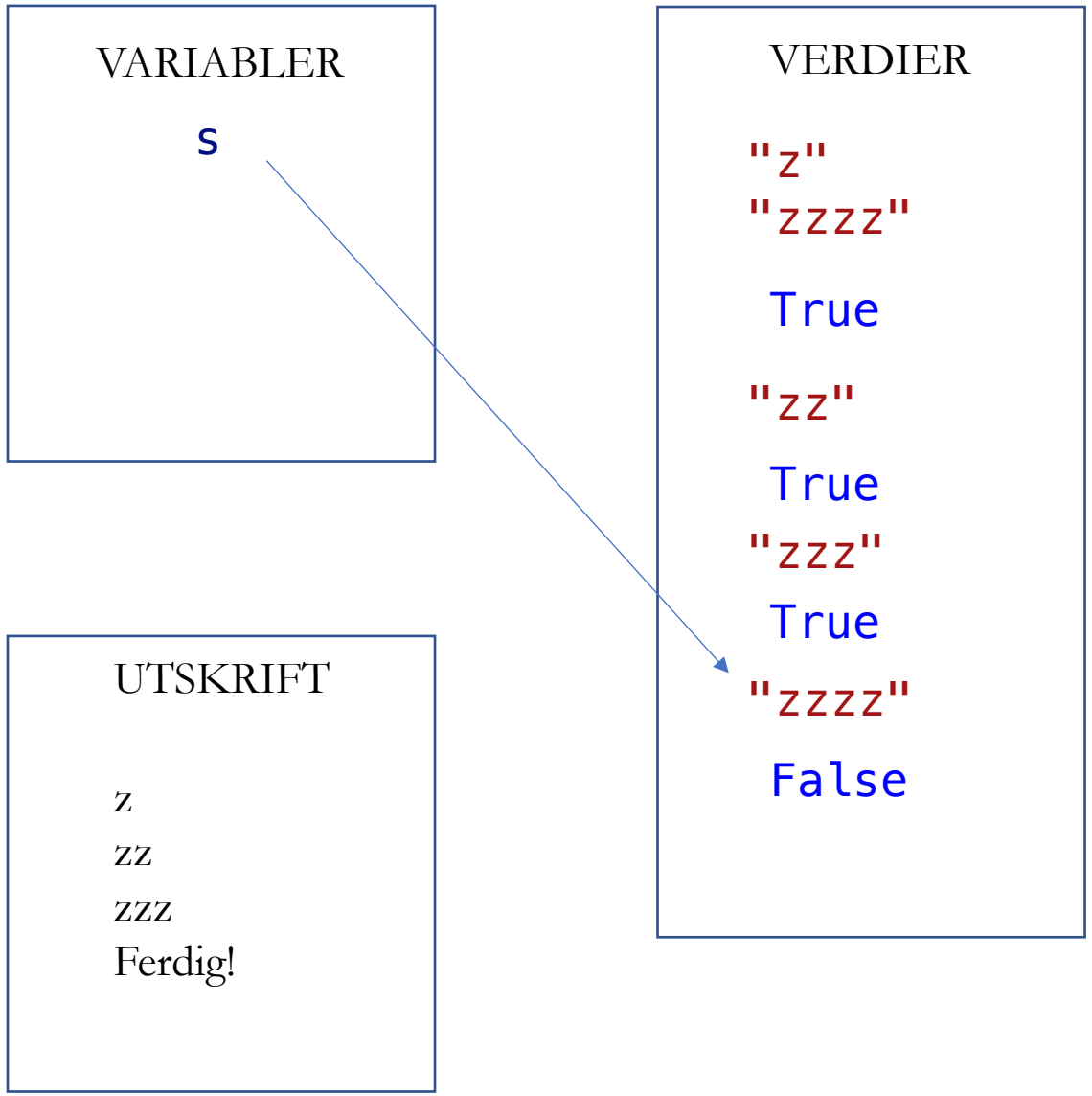
WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



WHILE

```
s = "z"
while s != "zzzz":
    print(s)
    s += "z"

print("Ferdig!")
```



FOR

iterand/løkke-variabel

f. eks
- range
- streng
- tuple
- liste

```
bla()
bla()

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

bla()
bla()
```

én gang for hvert element i samlingen

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

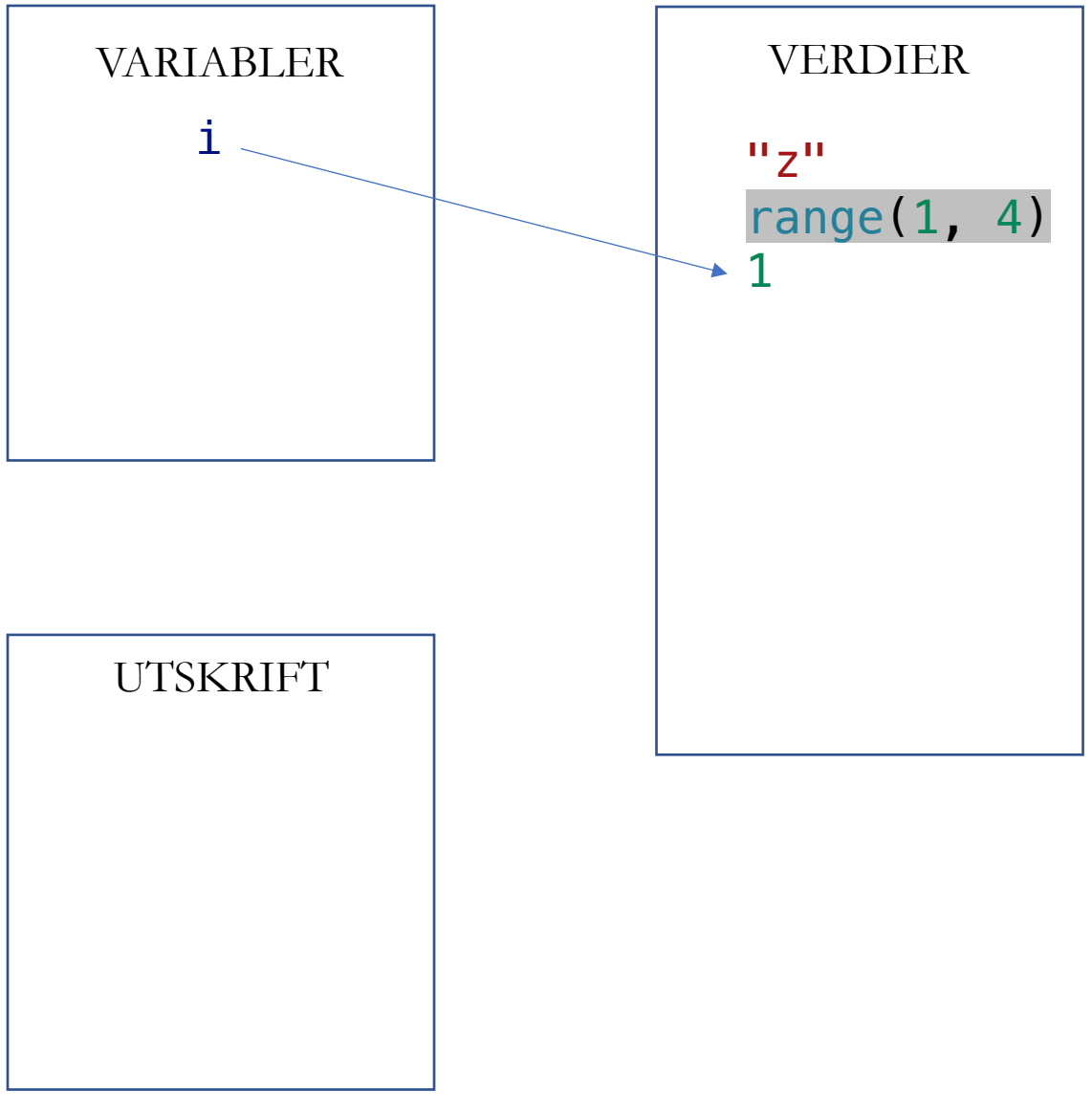
VERDIER

"z"
range(1, 4)

UTSKRIFT

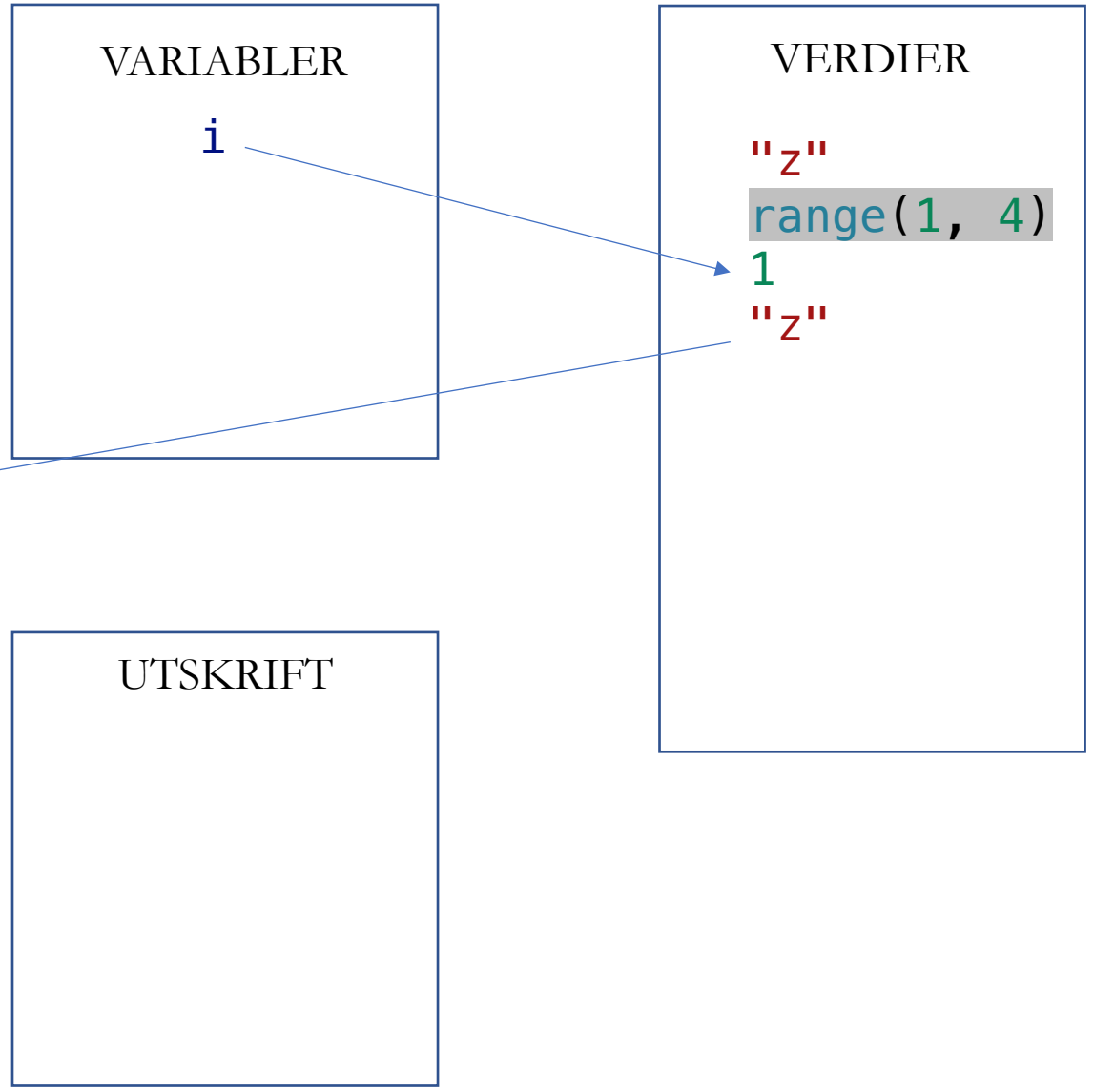
FOR

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

print("Ferdig")
```



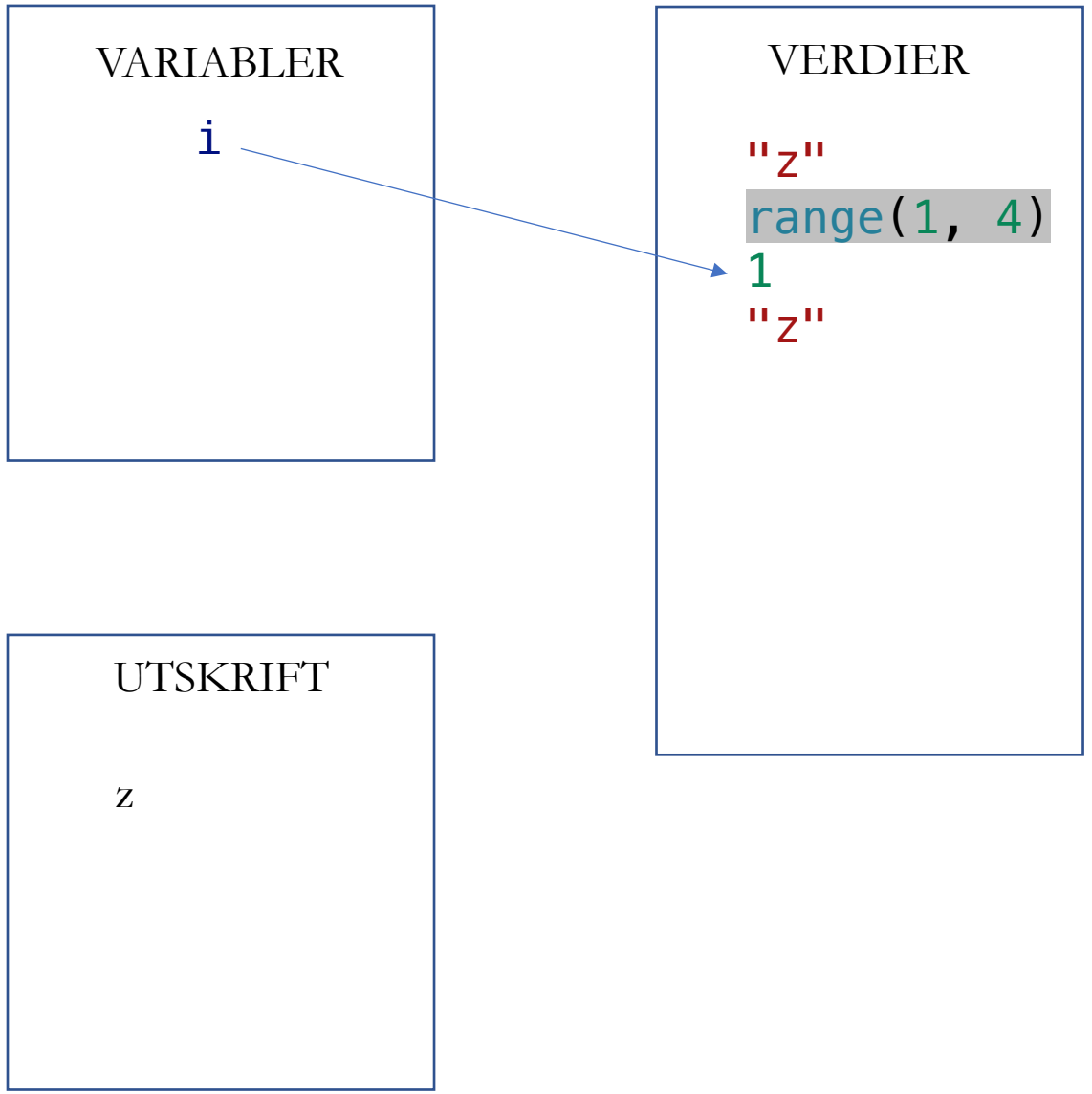
FOR

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

print("Ferdig")
```



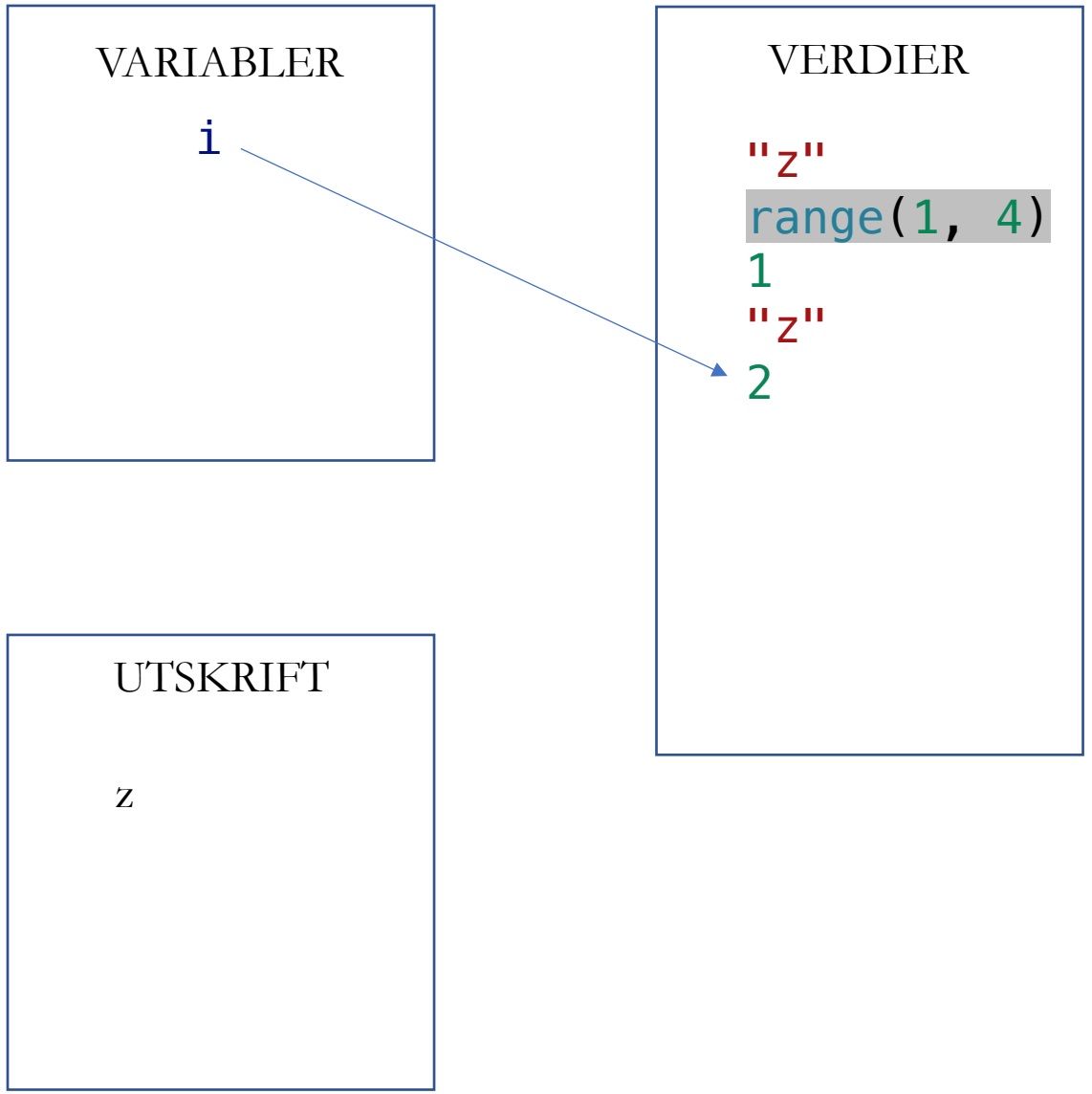
FOR

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

print("Ferdig")
```



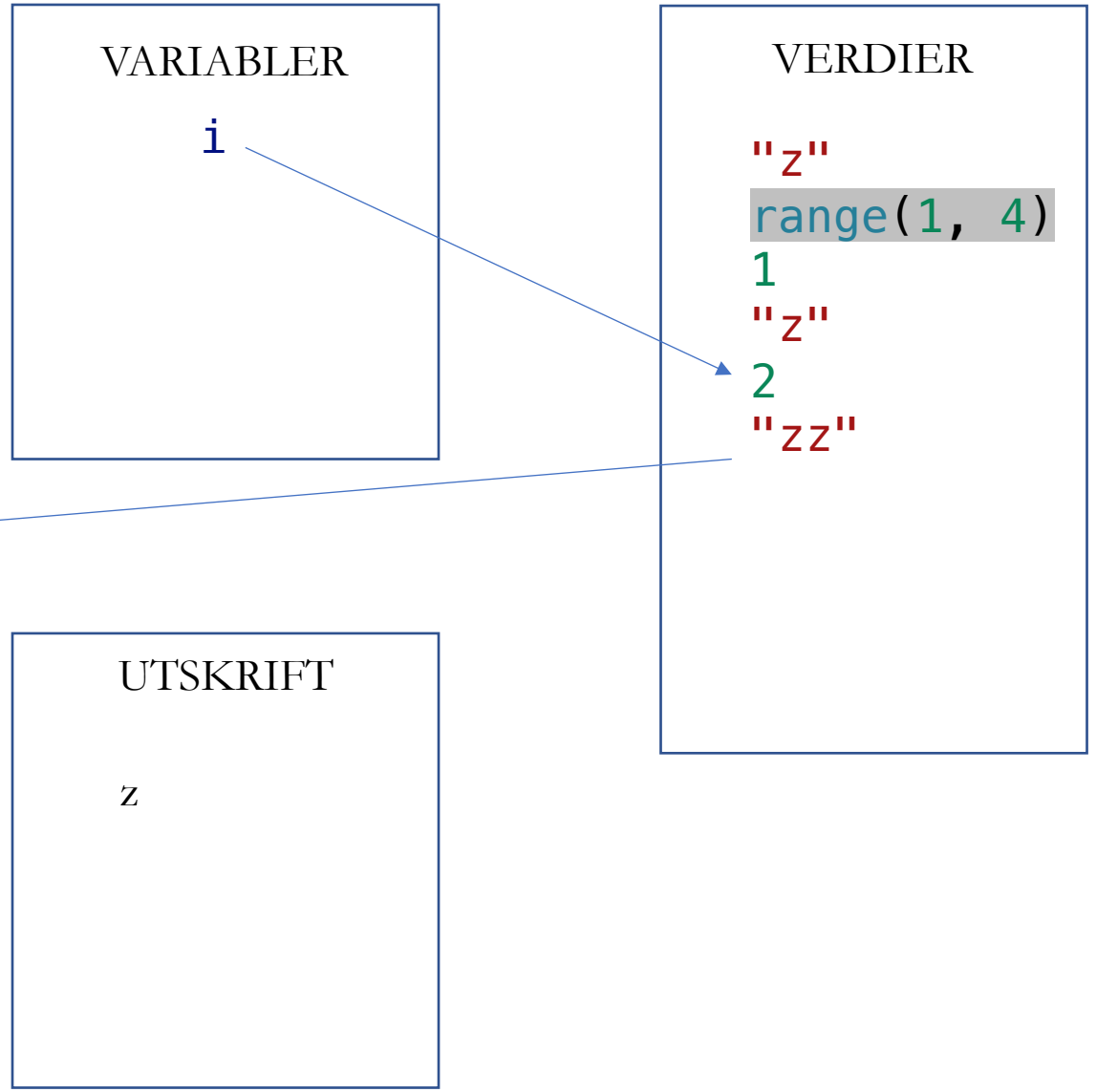
FOR

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

print("Ferdig")
```



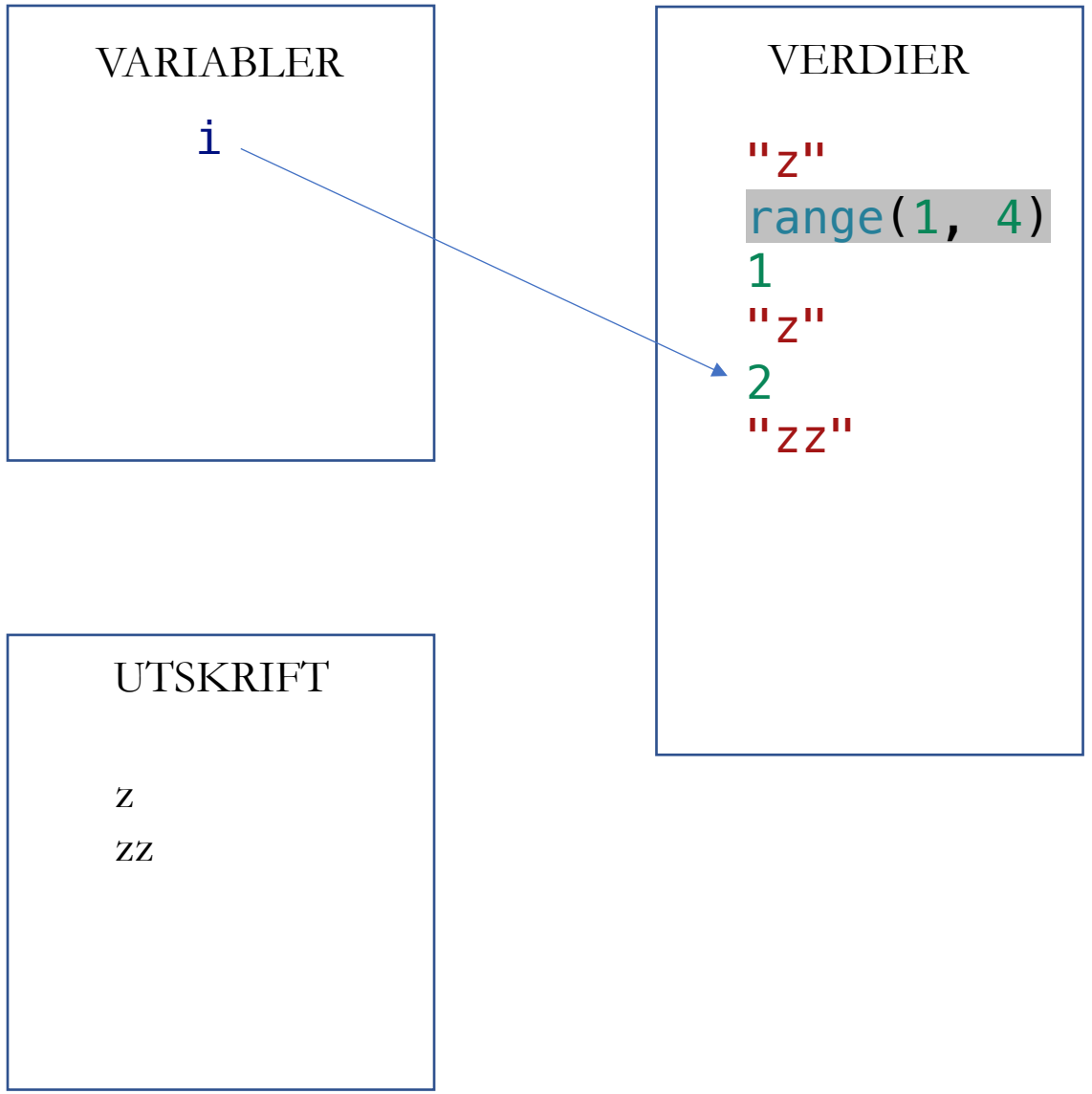
FOR

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

print("Ferdig")
```



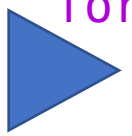
FOR

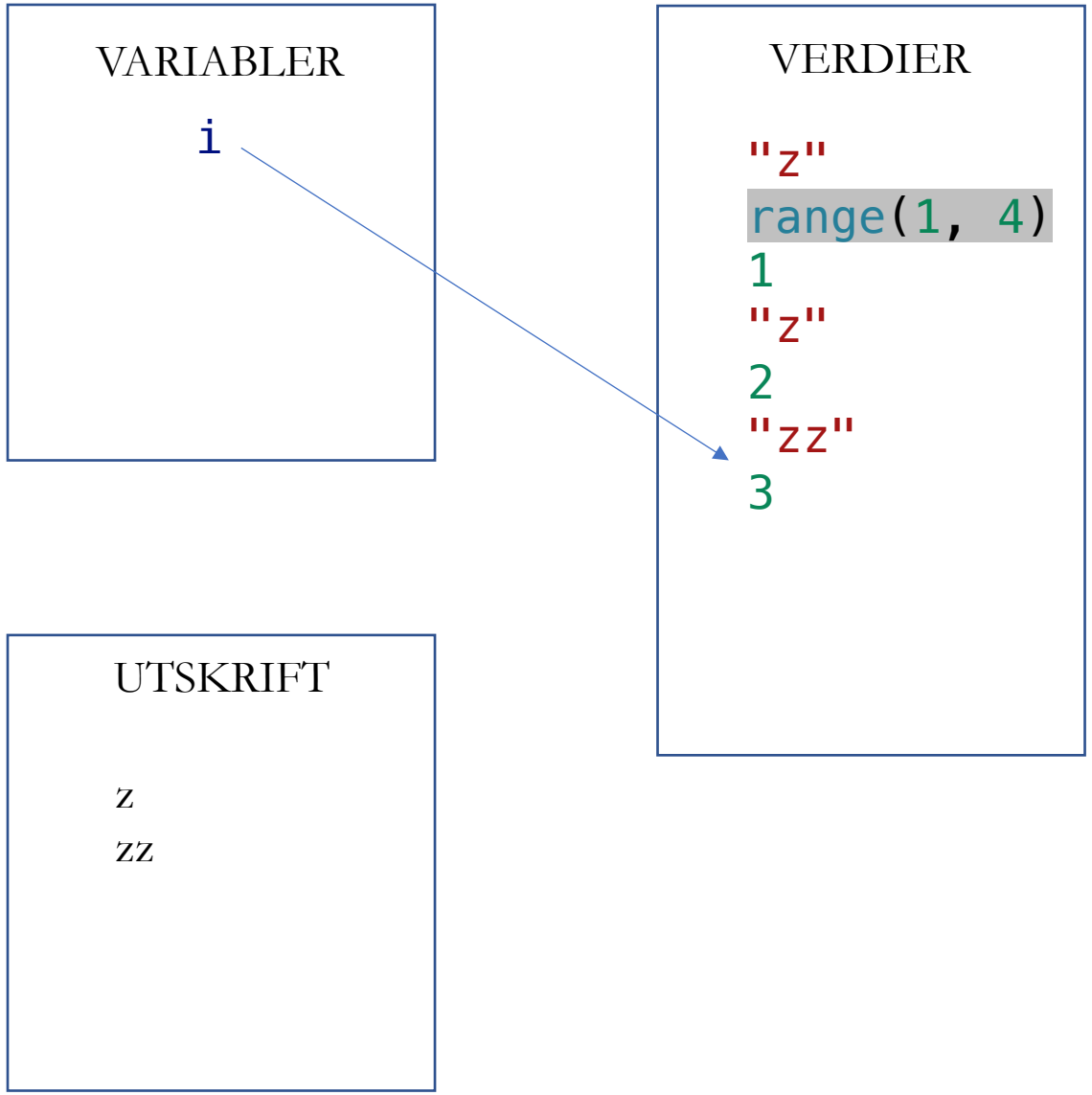
```
for i in range(1, 4):
    print("z" * i)

print("Ferdig")
```



FOR

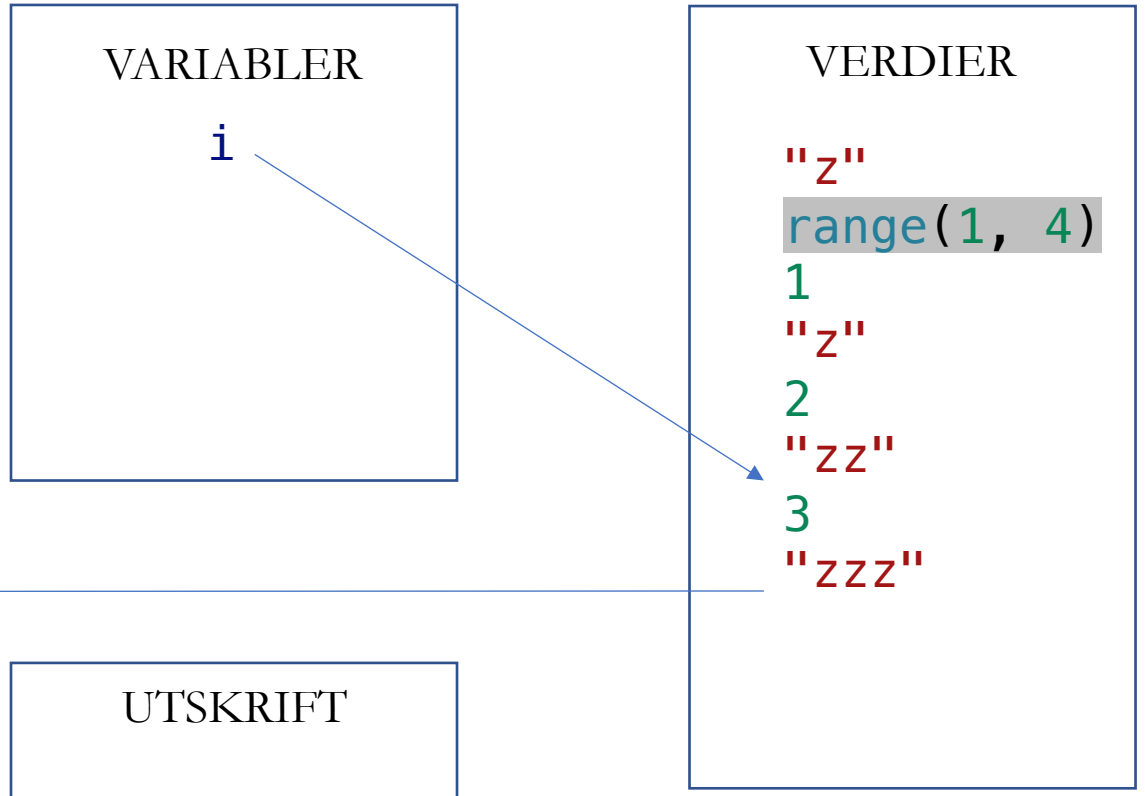
```
for i in range(1, 4):
    print("z" * i)

print("Ferdig")
```





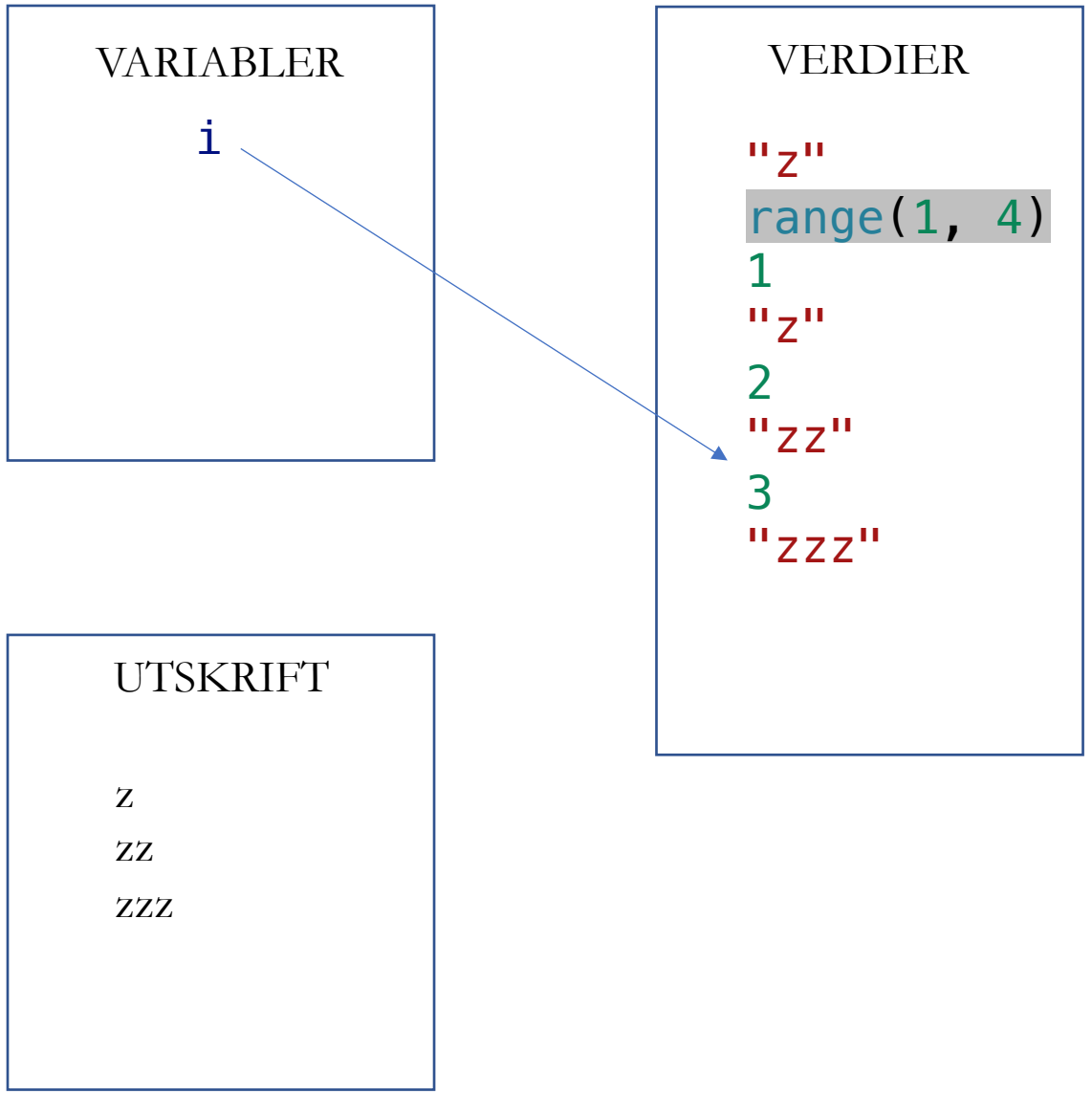
FOR

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

print("Ferdig")
```



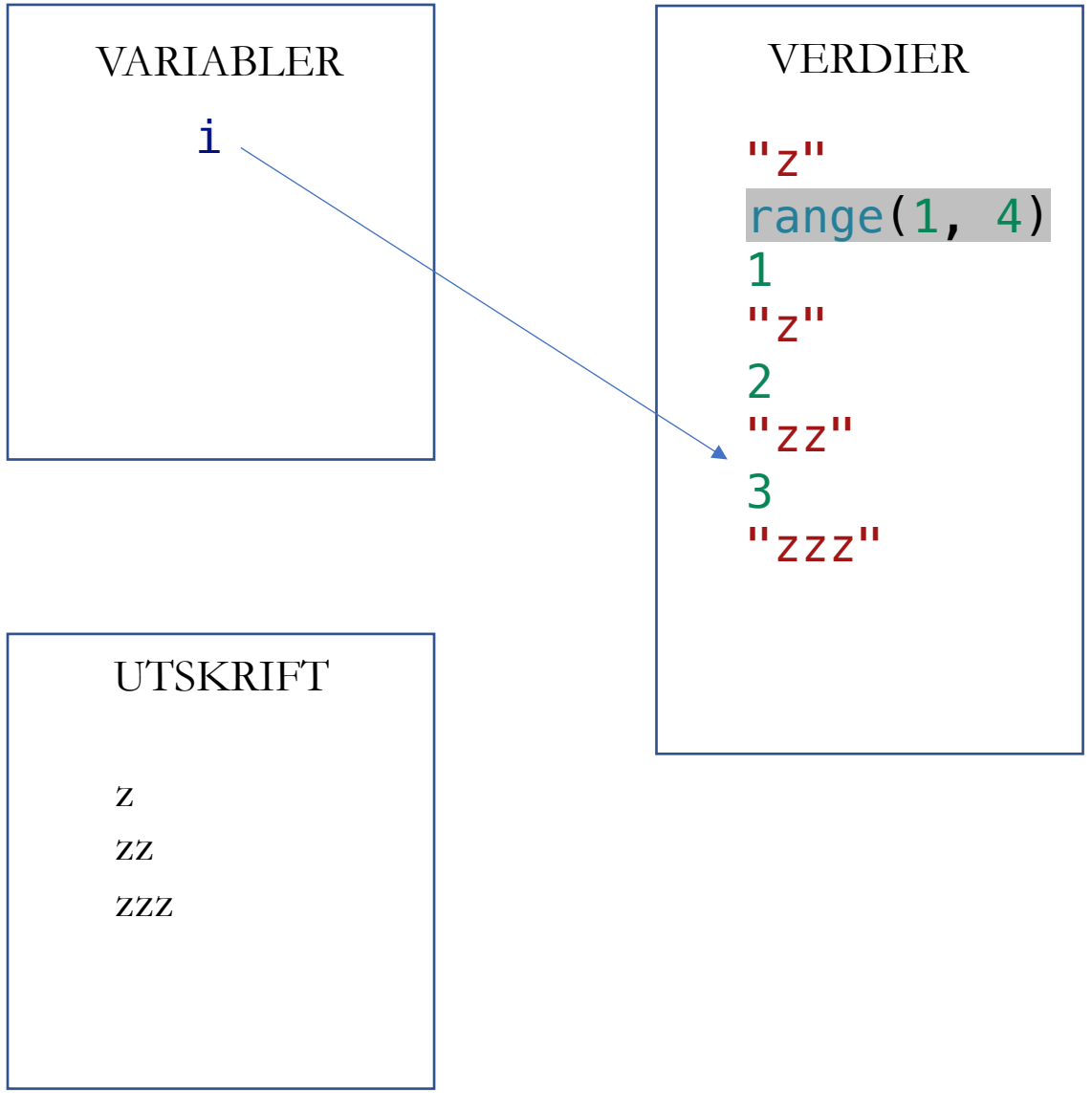
FOR

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

print("Ferdig")
```



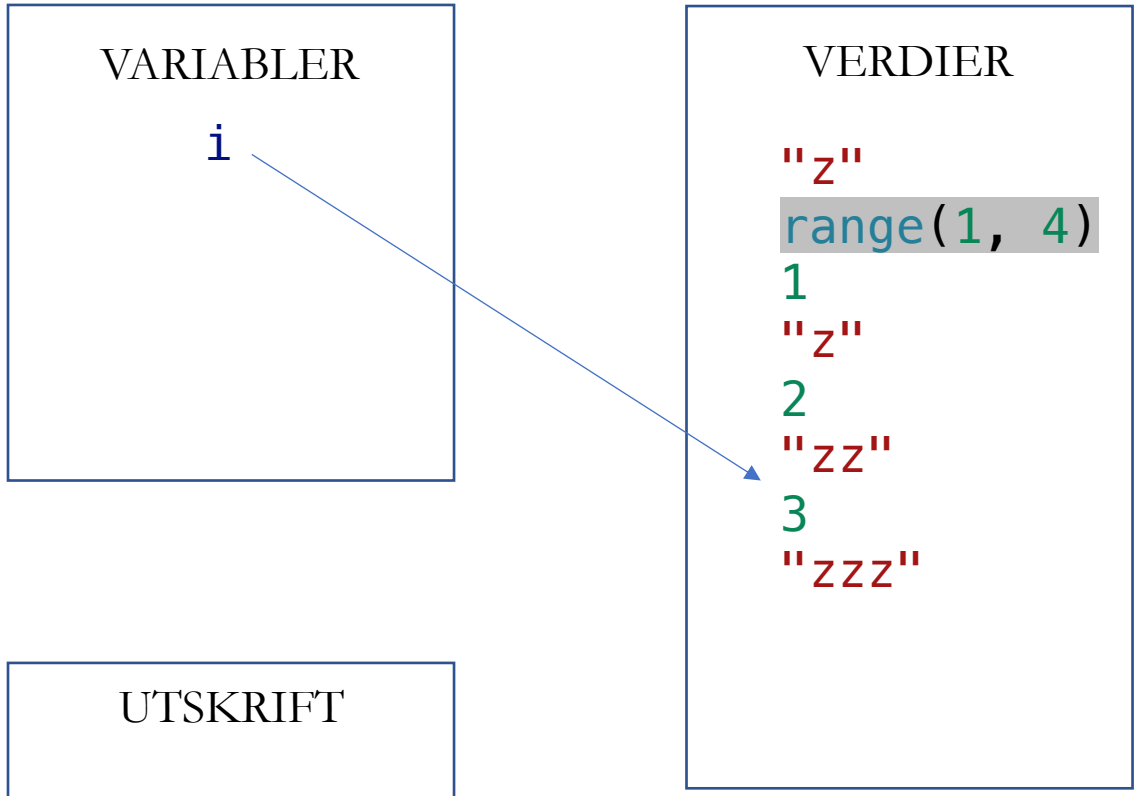
FOR

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

print("Ferdig")
```



FOR

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

print("Ferdig")
```



UTSKRIFT

z
zz
zzz
Ferdig

TUPLE

```
(42, 99, 123)
```

```
("foo", "bar", "baz")
```

```
(64, "INF100", None, True, 0.5)
```

```
(32, )
```

TUPLE

```
def print_longest_word(s1, s2, s3):
    longest_len = max(len(s1), len(s2), len(s3))
    for s in (s1, s2, s3):
        if len(s) == longest_len:
            print(s)
```

BREAK

```
def print_longest_word(s1, s2, s3):
    longest_len = max(len(s1), len(s2), len(s3))
    for s in (s1, s2, s3):
        if len(s) == longest_len:
            print(s)
            break
```

CONTINUE

```
def print_longest_word(s1, s2, s3):
    longest_len = max(len(s1), len(s2), len(s3))
    for s in (s1, s2, s3):
        if len(s) != longest_len:
            continue
        print(s)
```

PROBLEMLØSNING

https://open.kattis.com/contests/j7mmmy

- Opprett bruker
- Gå til Contests
- Join INF100 lecture 3