UNIVERSITETET I BERGEN

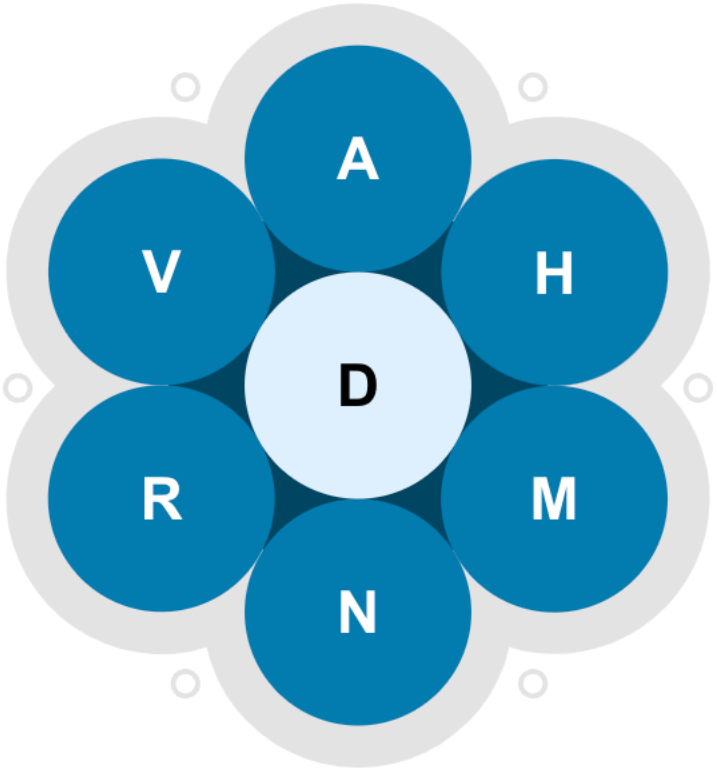
LISTER

INF100

VÅR 2024

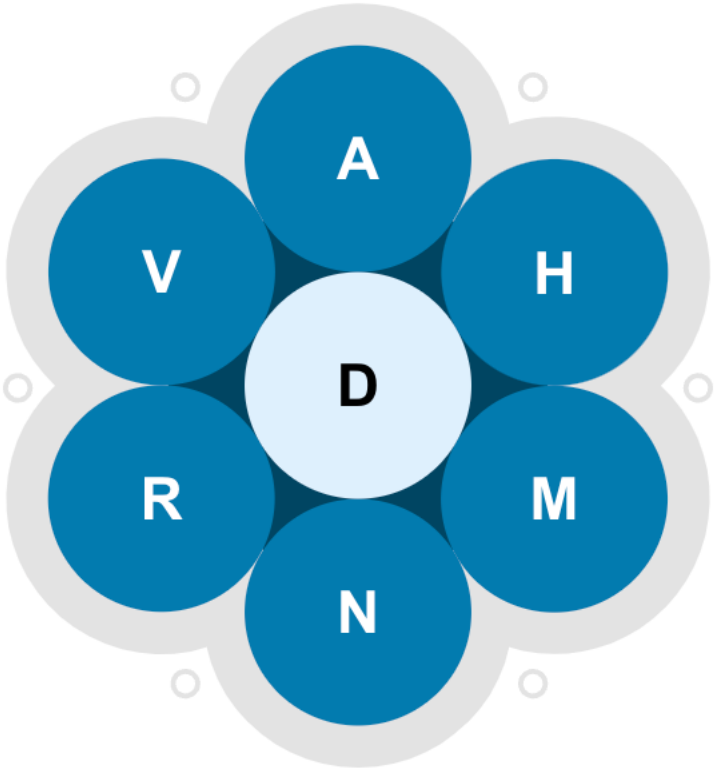
Torstein Strømme

ORDKNUTE



- Finn så mange ord du kan
- Regler:
  - Alle ord må inneholde bokstaven i midten
  - Du kan bruke samme bokstav flere ganger
  - Ordene må ha minst fire bokstaver

ORDKNUTE

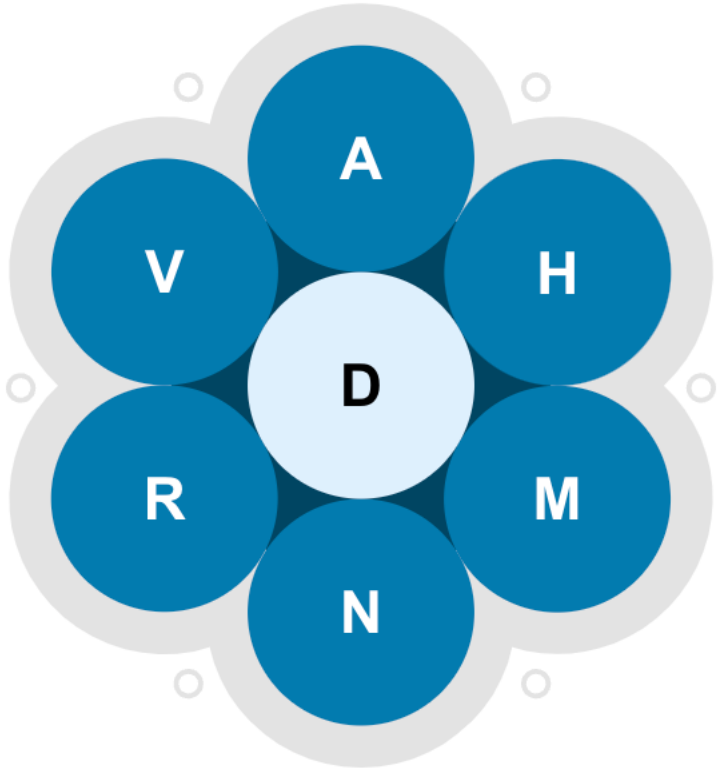


- Finn så mange ord du kan
- Regler:
  - Alle ord må inneholde bokstaven i midten
  - Du kan bruke samme bokstav flere ganger
  - Ordene må ha minst fire bokstaver

- Steg 1: en funksjon som avgjør om ett ord er godkjent eller ikke

```
def word_is_legal(word, required_char, available_chars):
    ...
```

ORDKNUTE



- Finn så mange ord du kan
- Regler:
  - Alle ord må inneholde bokstaven i midten
  - Du kan bruke samme bokstav flere ganger
  - Ordene må ha minst fire bokstaver

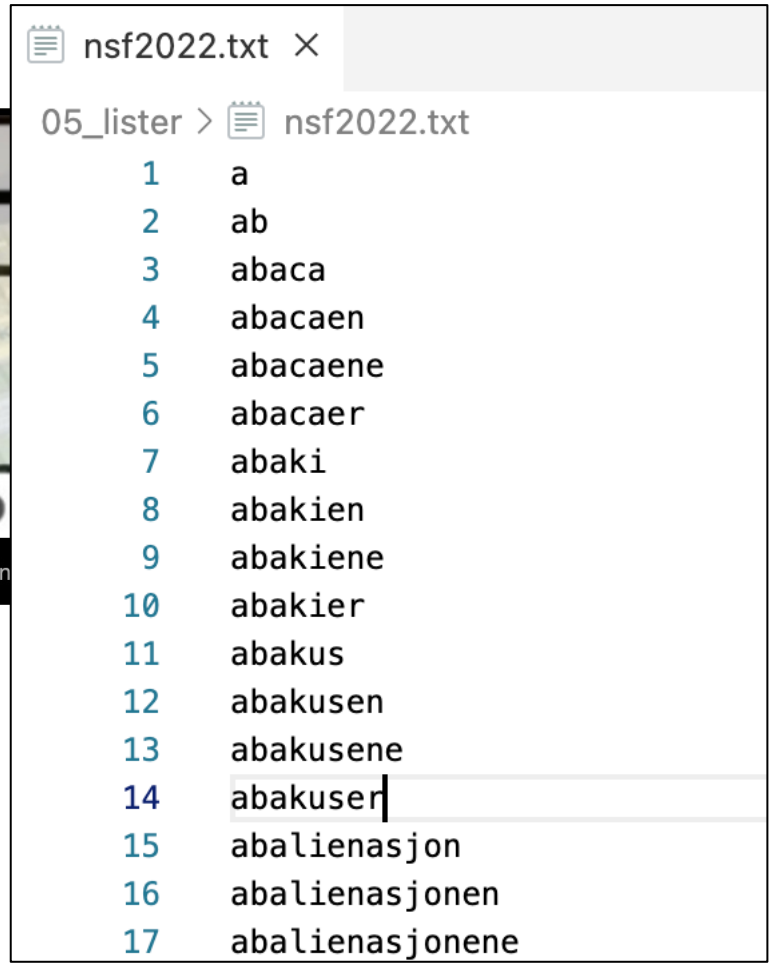
- Steg 1: en funksjon som avgjør om ett ord er godkjent eller ikke
- Steg 2: prøv alle mulige ord

ALLE MULIGE ORD

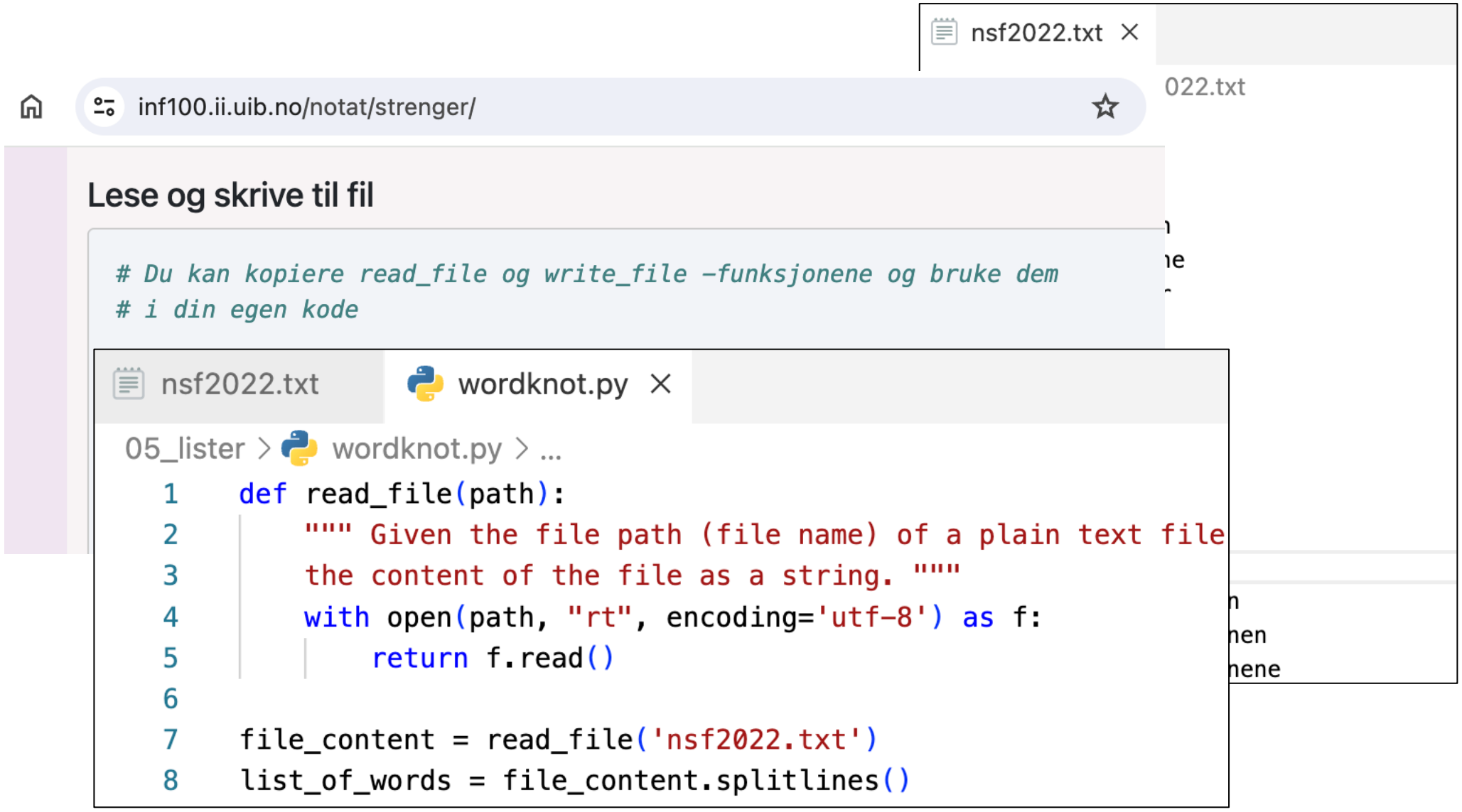
Ordliste og søkeside

19. august 2023 publiserte vår dyktige språkkomite ny utgave av ordlista. Det ble lagt til 22 431 ord, og fjernet 560.

Klikk her for å laste ned en zippet (pakket) utgave av hele lista.



ALLE MULIGE ORD



PRØV ALLE MULIGE ORD

oppretter tom liste

løkke over alle mulige ord

```
legal_words = []
for candidate_word in list_of_words:
    if word_is_legal(candidate_word, required, available):
        legal_words.append(candidate_word)
```

legger til `candidate_word` *på slutten av* `legal_words`

PRØV ALLE MULIGHETER

oppretter tom liste

løkke over alle potensielle muligheter

```
filtered_things = []
for candidate_thing in collection_of_everything:
    if matches_critera(candidate_thing):
        filtered_things.append(candidate_thing)
```

sjekk om tingen er en sånn ting vi vil ha

legg til `candidate_thing` *på slutten av* `filtered_stuff`

SAMLINGER

- Strenger, range, tupler og lister

```
s = "abc"
r = range(3, 12, 2)
t = (1, 2, "abc")
a = [9, 8, "abc"]
```

Indeksering

```
s[0]
r[1]
t[1]
a[2]
```

Beskjæring

```
s[:2]
r[1:3]
t[::-1]
a[:2]
```

Løkker

```
for thing in s/r/t/a :
    ...
```

Medlemskap

```
"a" in s
5 in r
"abc" in t
8 in a
```

Funksjoner

```
min  max
len
.count
.index
```

SAMLINGER

- Strenger, ~~range~~, tupler og lister

```
s = "abc"
r = range(3, 12, 2)
t = (1, 2, "abc")
a = [9, 8, "abc"]
```

Repetisjon

```
s * 0
t * 2
a * 3
```

Konkatenasjon

```
s + "def"
t + (True, 0.0)
a + ["a", -3]
```

Hvordan opprette tupler

```
# Tom tuple
t = ()
t = tuple()

# Tuple med ett element
t = (42,)


# Tuple med flere elementer
t = (42, 95, "abc",)
t = (42, 95)

# Konvertere andre samlinger til tuple
t = tuple("abc")
t = tuple([1, 2, 3])
t = tuple(range(5))
```

Hvordan opprette lister

```
# Tom liste
a = []
a = list()

# Liste med ett element
a = [42,]
a = [42]

# Liste med flere elementer
a = [42, 95, "abc",]
a = [42, 95]

# Konvertere andre samlinger til liste
a = list("abc")
a = list((1, 2, 3))
a = list(range(5))
```

```
# Splitting av en streng
a = "1 2 5 9 4".split(" ")
a = "1\n5 9\n4\n".splitlines()
```

```
for line in file_contents.splitlines():
    ...
```

en liste

INDEKSERING

menti.com

3137 5844

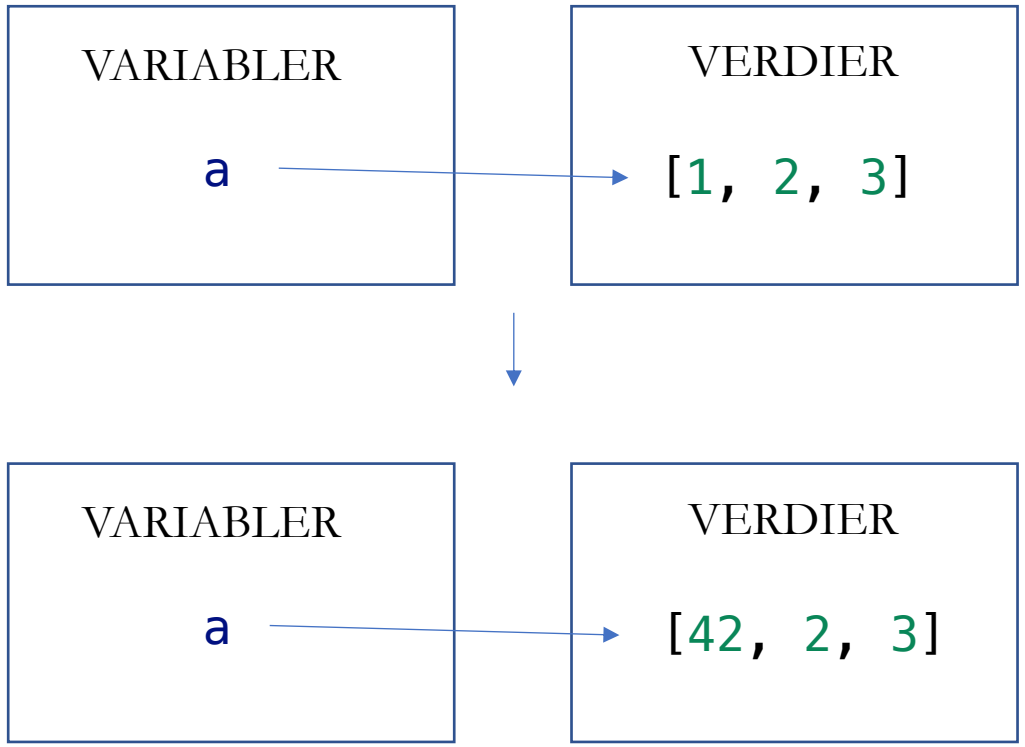
LISTER vs. TUPLER

- En liste kan *muteres*
- En tupel kan *ikke* muteres

- Tidligere: *variabler* har endret seg
- Mutasjon: selve *verdien* endrer seg

MUTASJON

```
a = [1, 2, 3]
a[0] = 42
```



MUTASJON

```
t = (1, 2, 3)
t[0] = 42
```

TypeError: 'tuple' object does not support item assignment

STRENG

```
a = "abcdef"
b = "xyz"
a += b
```

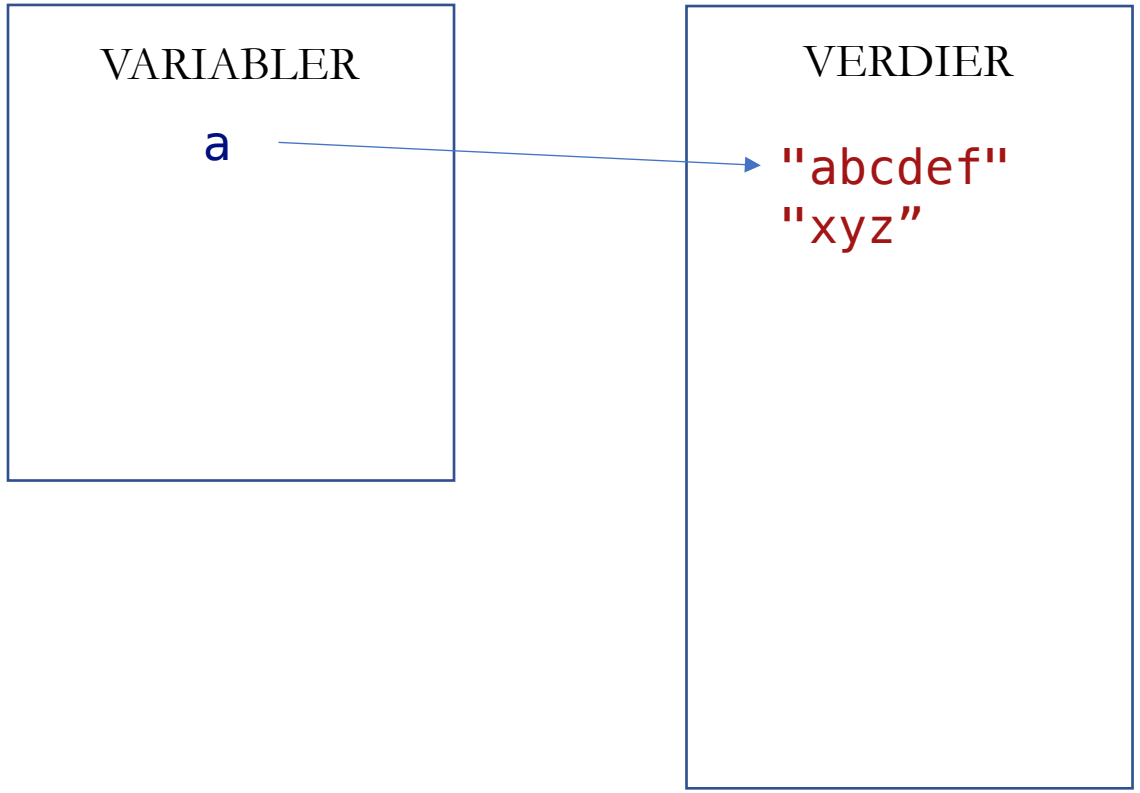
VARIABLER

VERDIER

"abcdef"
"xyz”

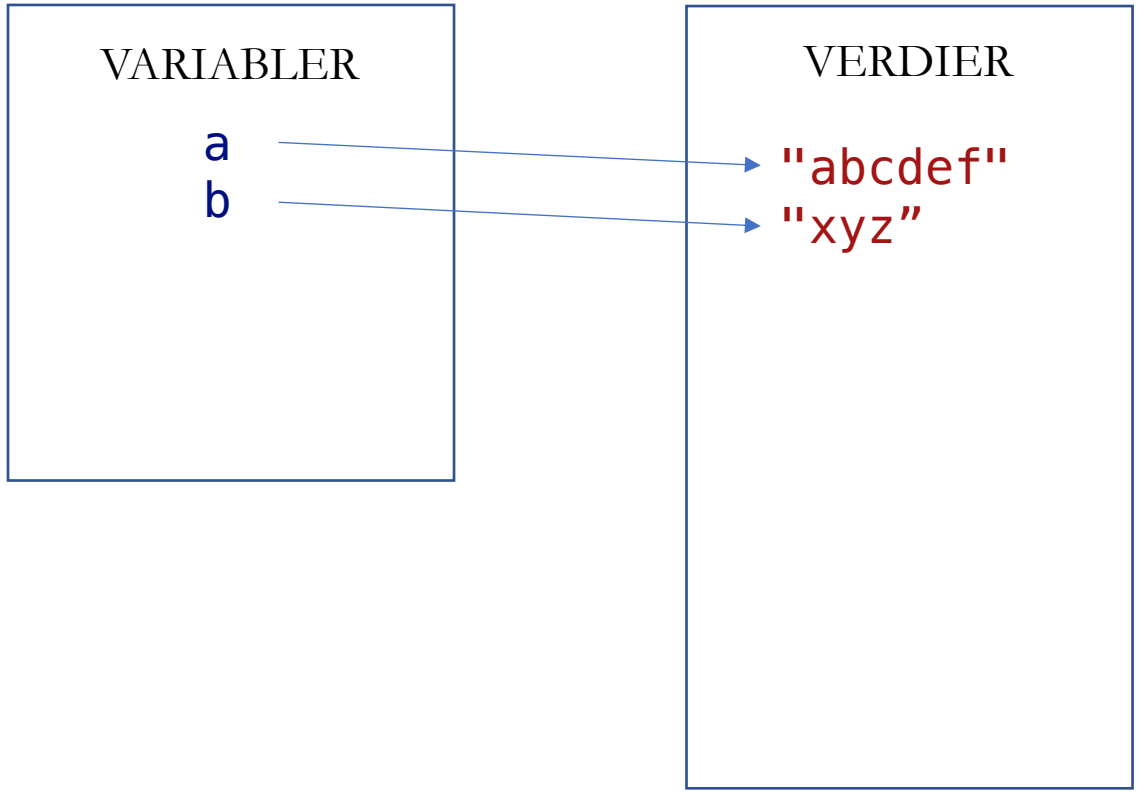
STRENG

```
a = "abcdef"
b = "xyz"
a += b
```



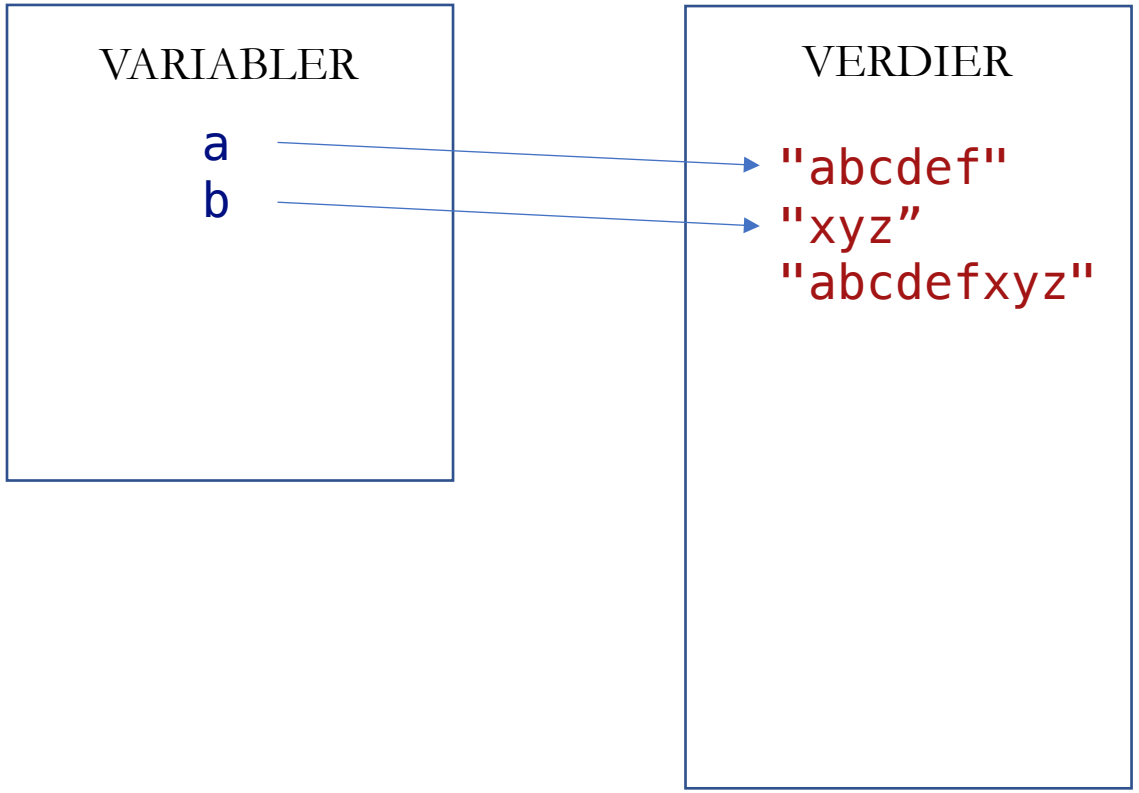
STRENG

```
a = "abcdef"
b = "xyz"
a += b
```



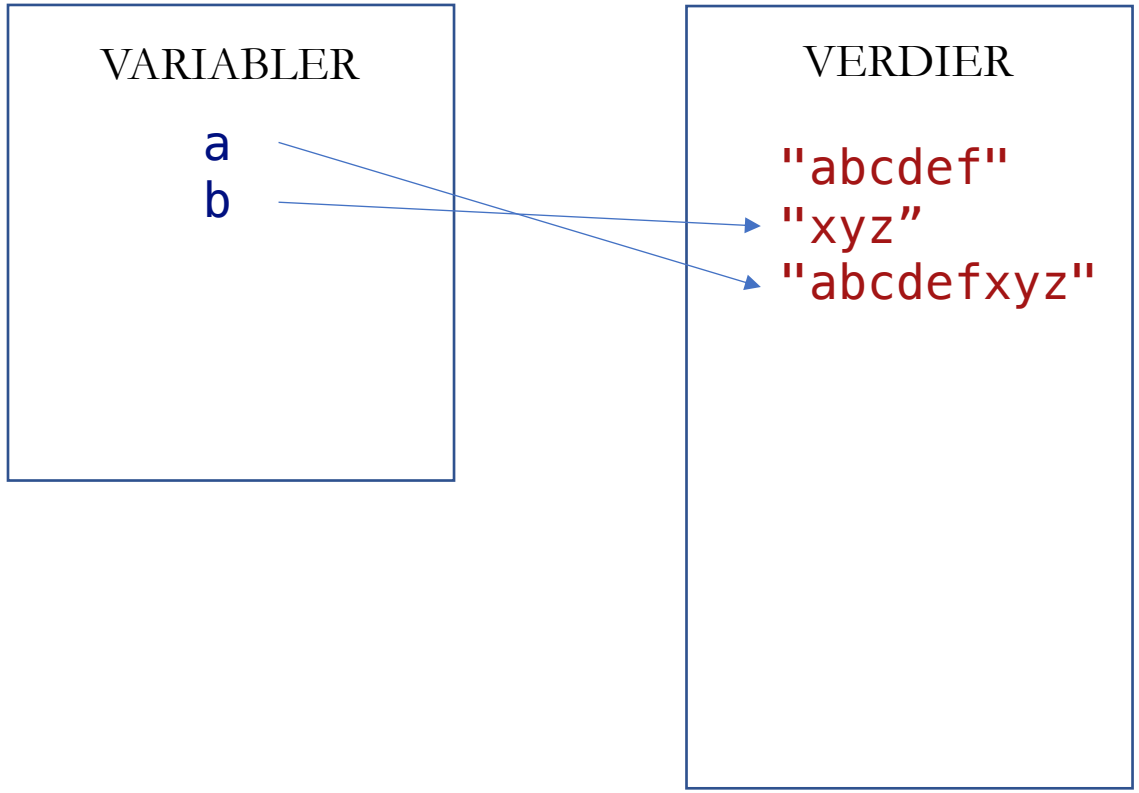
STRENG

```
a = "abcdef"
b = "xyz"
a += b
```



STRENG

```
a = "abcdef"
b = "xyz"
a += b
```



TUPLE

VARIABLER

VERDIER

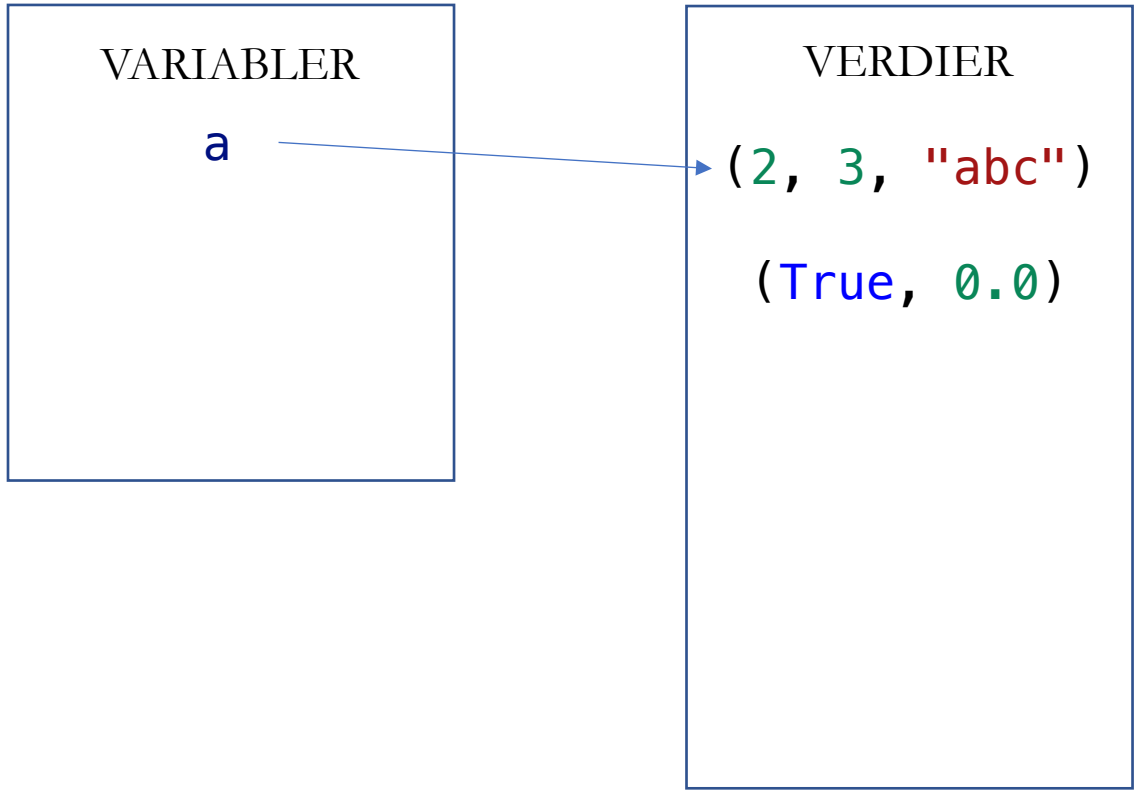
(2, 3, "abc")

(True, 0.0)

```
a = (2, 3, "abc")
b = (True, 0.0)
a += b
```

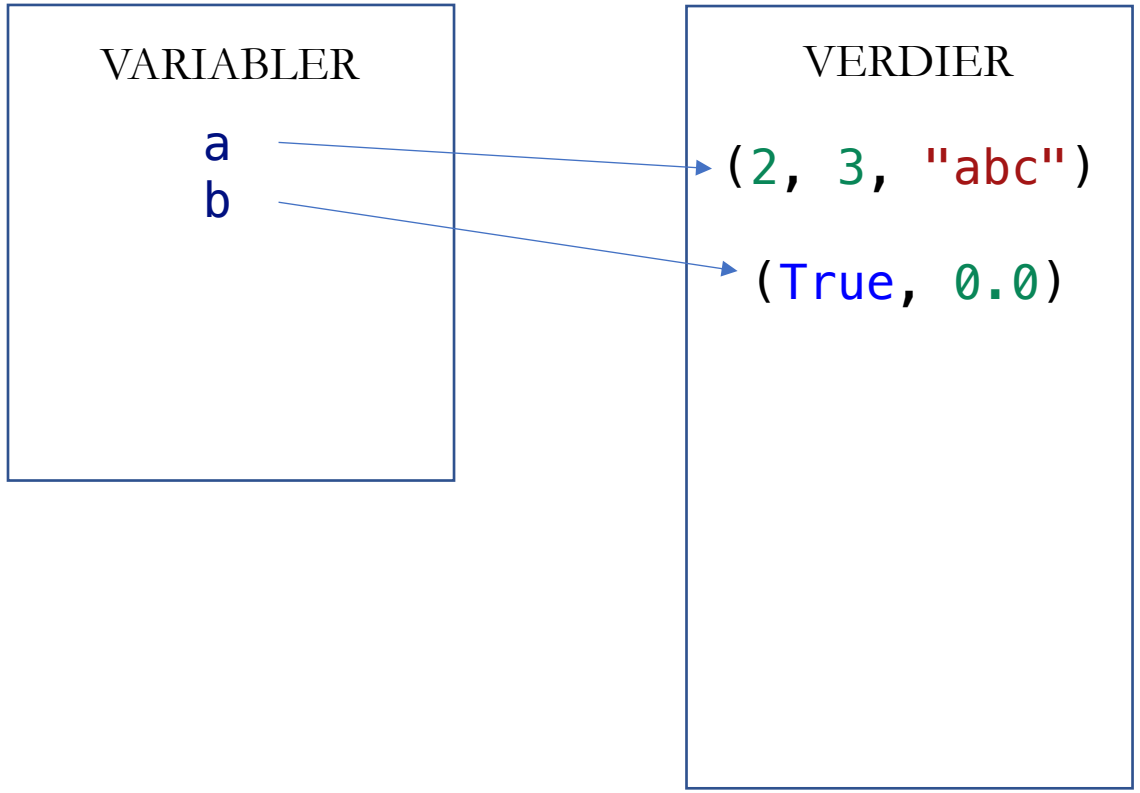
TUPLE

```
a = (2, 3, "abc")
b = (True, 0.0)
a += b
```



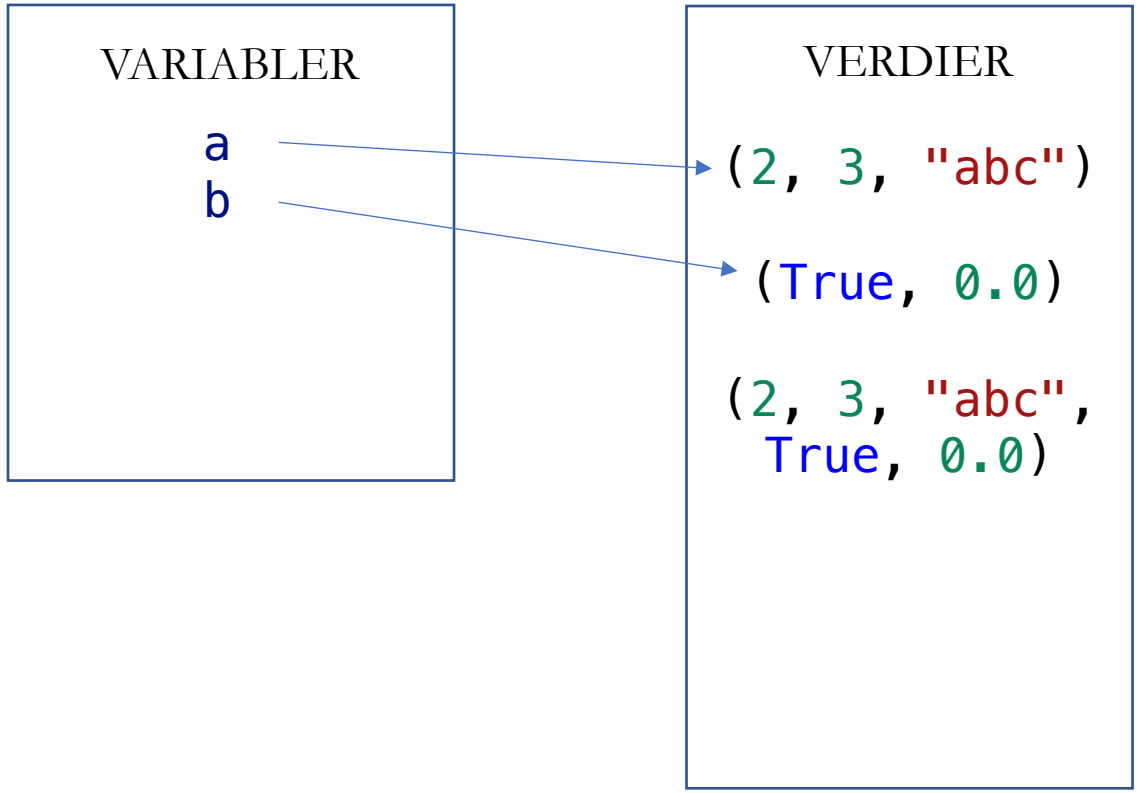
TUPLE

```
a = (2, 3, "abc")
b = (True, 0.0)
a += b
```



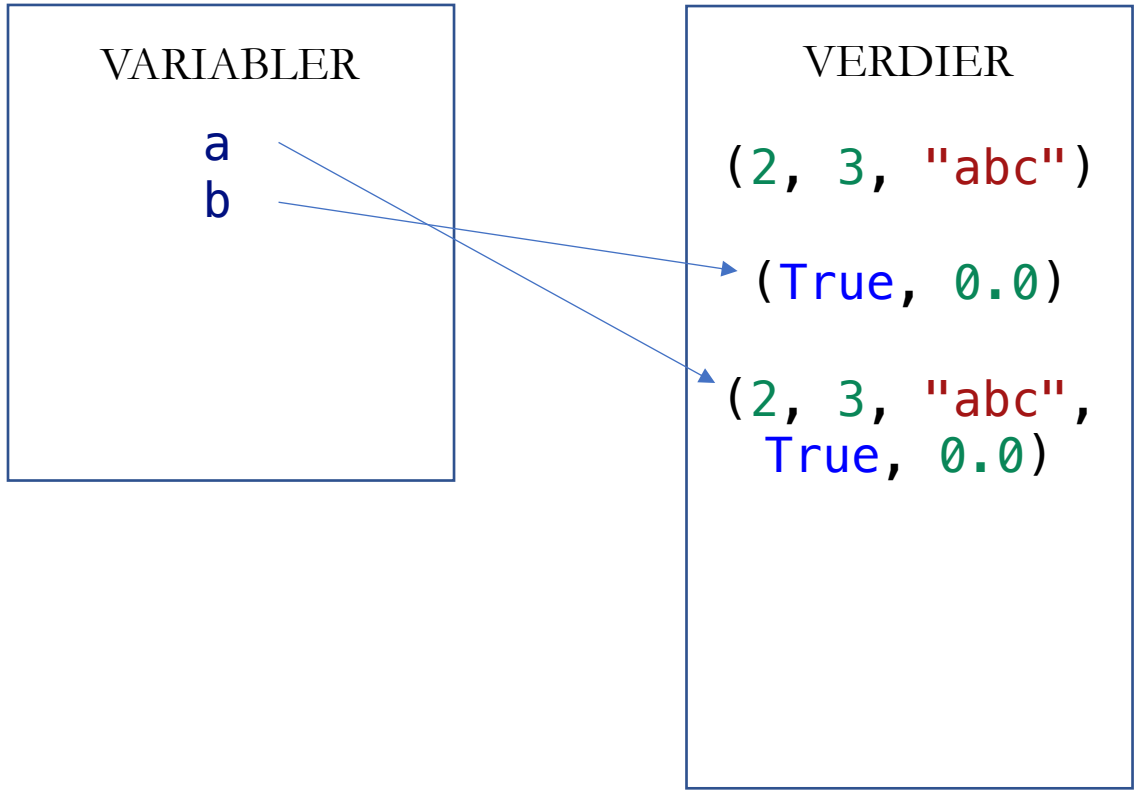
TUPLE

```
a = (2, 3, "abc")
b = (True, 0.0)
a += b
```



TUPLE

```
a = (2, 3, "abc")
b = (True, 0.0)
a += b
```



LISTE

```
a = [2, 3, "abc"]
b = [True, 0.0]
a += b
```

VARIABLER

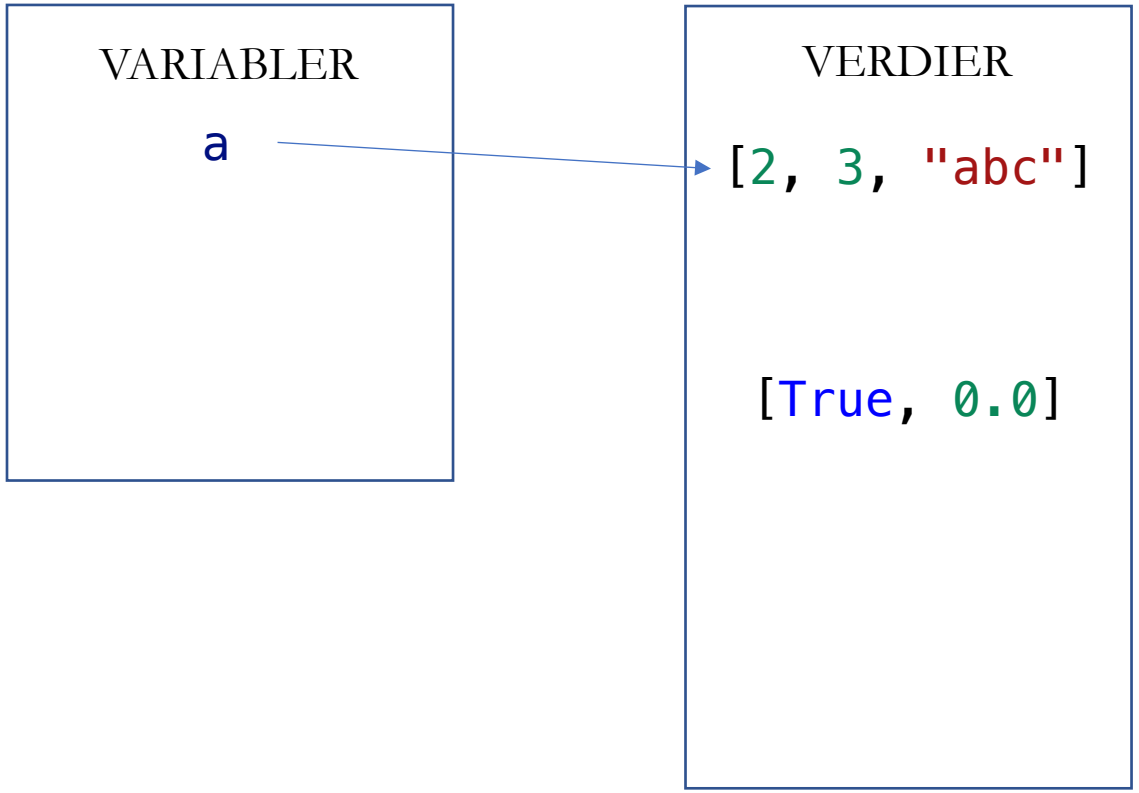
VERDIER

[2, 3, "abc"]

[True, 0.0]

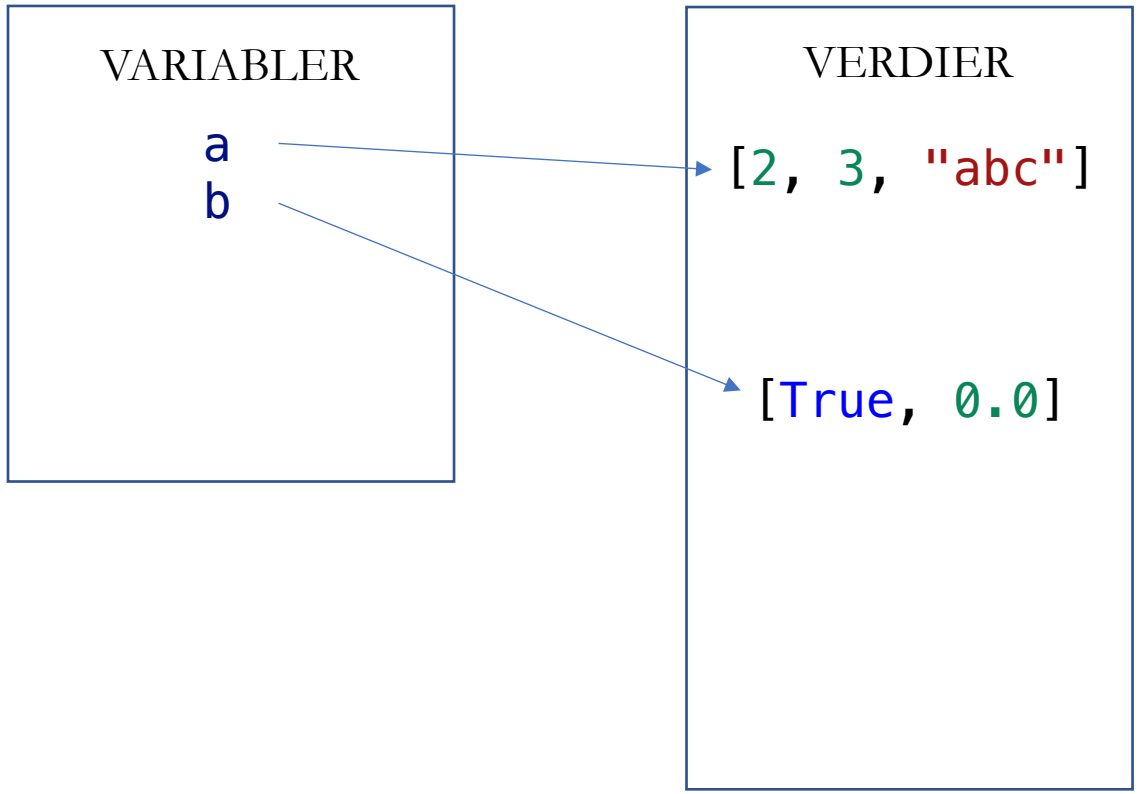
LISTE

```
a = [2, 3, "abc"]
b = [True, 0.0]
a += b
```



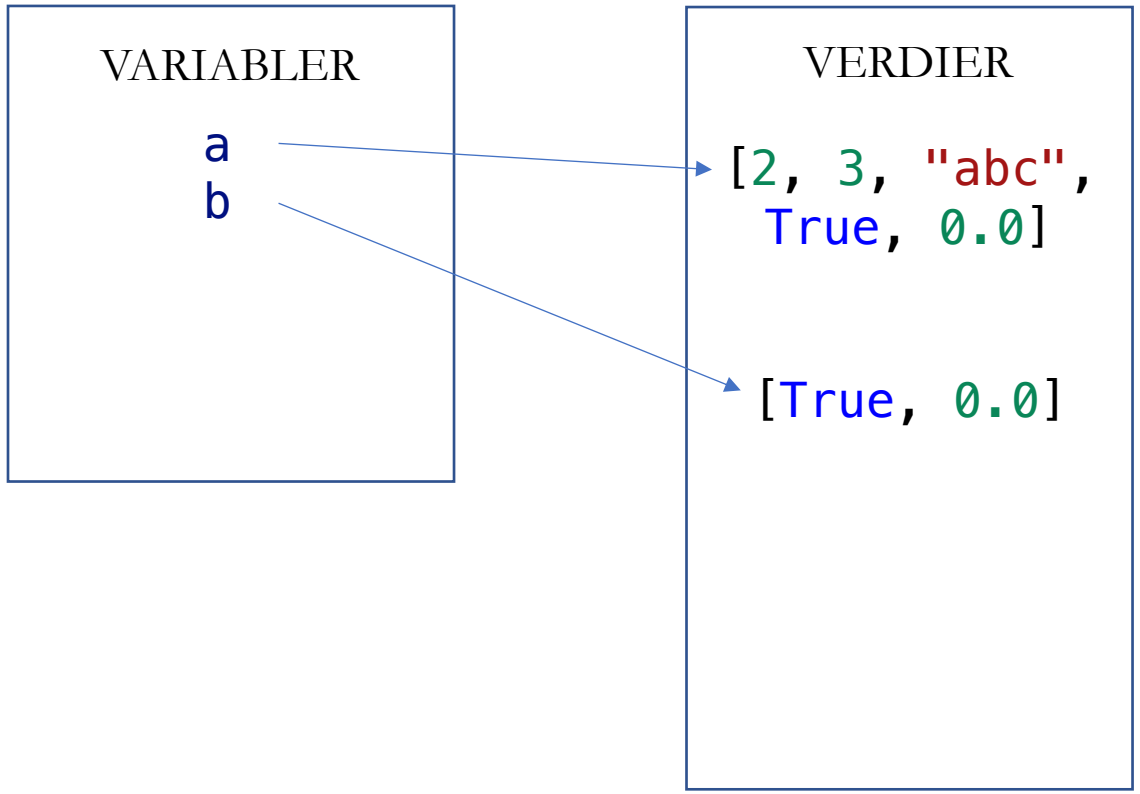
LISTE

```
a = [2, 3, "abc"]
b = [True, 0.0]
a += b
```



LISTE

```
a = [2, 3, "abc"]
b = [True, 0.0]
a += b
```



MUTATIVE OPERASJONER PÅ LISTER

- Operasjoner som *muterer*

```
a = [1, 2, 3]

a[i] = 42
a += [4, 5]
a *= 2
```

- Operasjoner som oppretter ny verdi

```
a = [1, 2, 3]

a = a[:i] + [42] + a[i+1:]
a = a + [4, 5]
a = a * 2
```

DESTRUKTIVE FUNKSJONER

- En destruktiv funksjon har en sideeffekt: den muterer en verdi
- En ikke-destruktiv funksjon muterer ingen verdier (utenom lokale verdier)

- Destruktive funksjoner trenger ikke returverdi (men kan ha likevel)
- Ikke-destruktive funksjoner må gi returverdi (ellers er den meningsløs)

FUNKSJONER PÅ LISTER

Hva?	Destruktive funksjoner	Ikke-destruktive alternative operasjoner
Legge til en ny verdi på slutten av listen	`a.append(42)`	`a = a + [42]`
Utvid listen med flere nye elementer på en gang	`a.extend([3, 4])`	`a = a + [3, 4]`
Putte inn en verdi på gitt posisjon	`a.insert(3, "foo")`	`a = a[:3] + ["foo"] + a[3:]`
Fjerne første forekomst av gitt verdi	`a.remove(2)`	`i = a.index(2)` `a = a[:i] + a[i + 1:]`
Fjerne element på en gitt posisjon	`a.pop(5)`	`a = a[:5] + a[5 + 1:]`
Fjerne alle elementer	`a.clear()`	`a = []`
Reverser	`a.reverse()`	`a = a[::-1]`
Sorter	`a.sort()`	`a = sorted(a)`

ALIAS

+ mutasjon

```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Mutasjon av listen
a[0] = 99
b[1] = 42
print(a)
print(b)
```

VARIABLER

UTSKRIFT

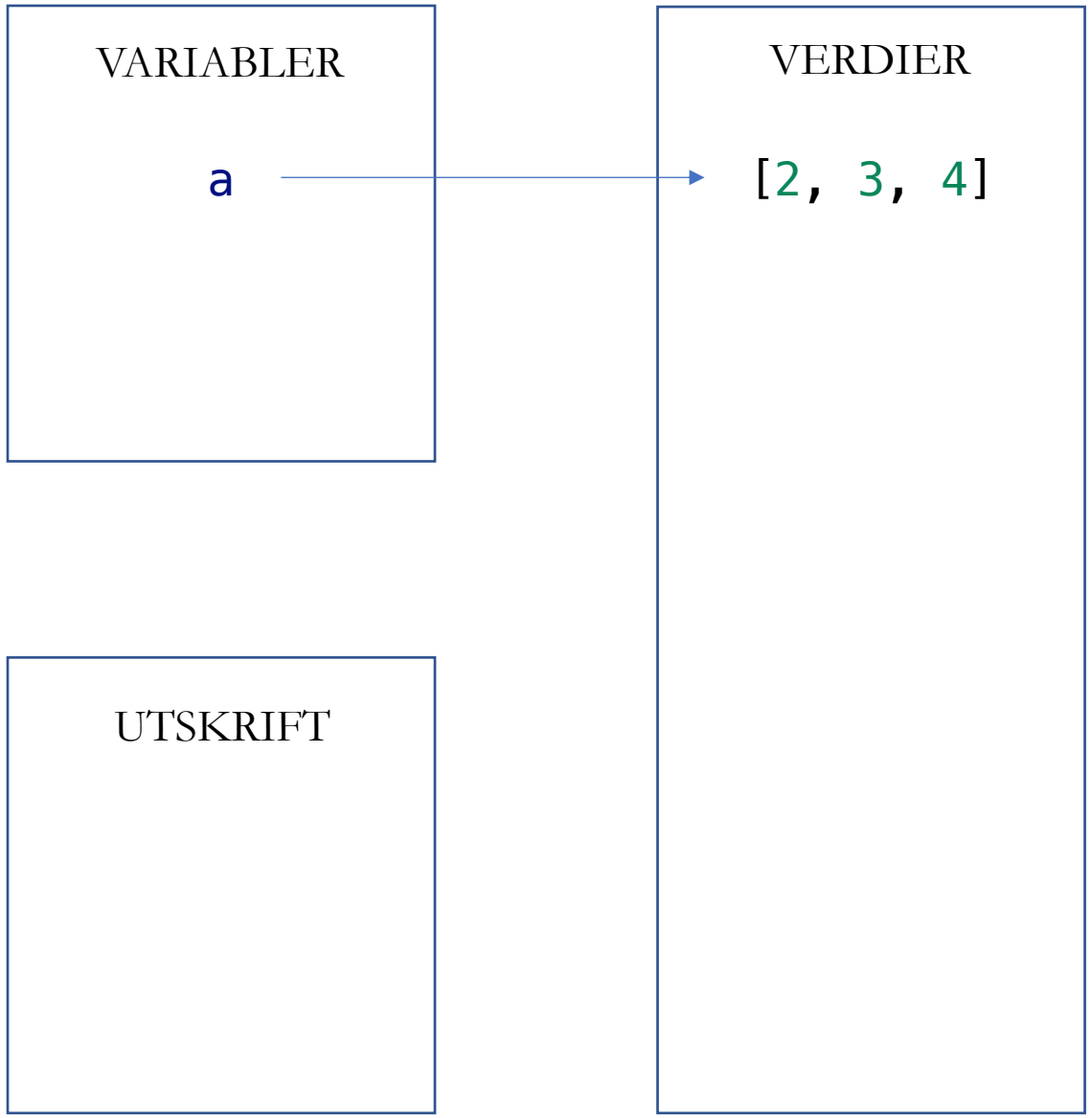
VERDIER

```
[2, 3, 4]
```

ALIAS

+ mutasjon

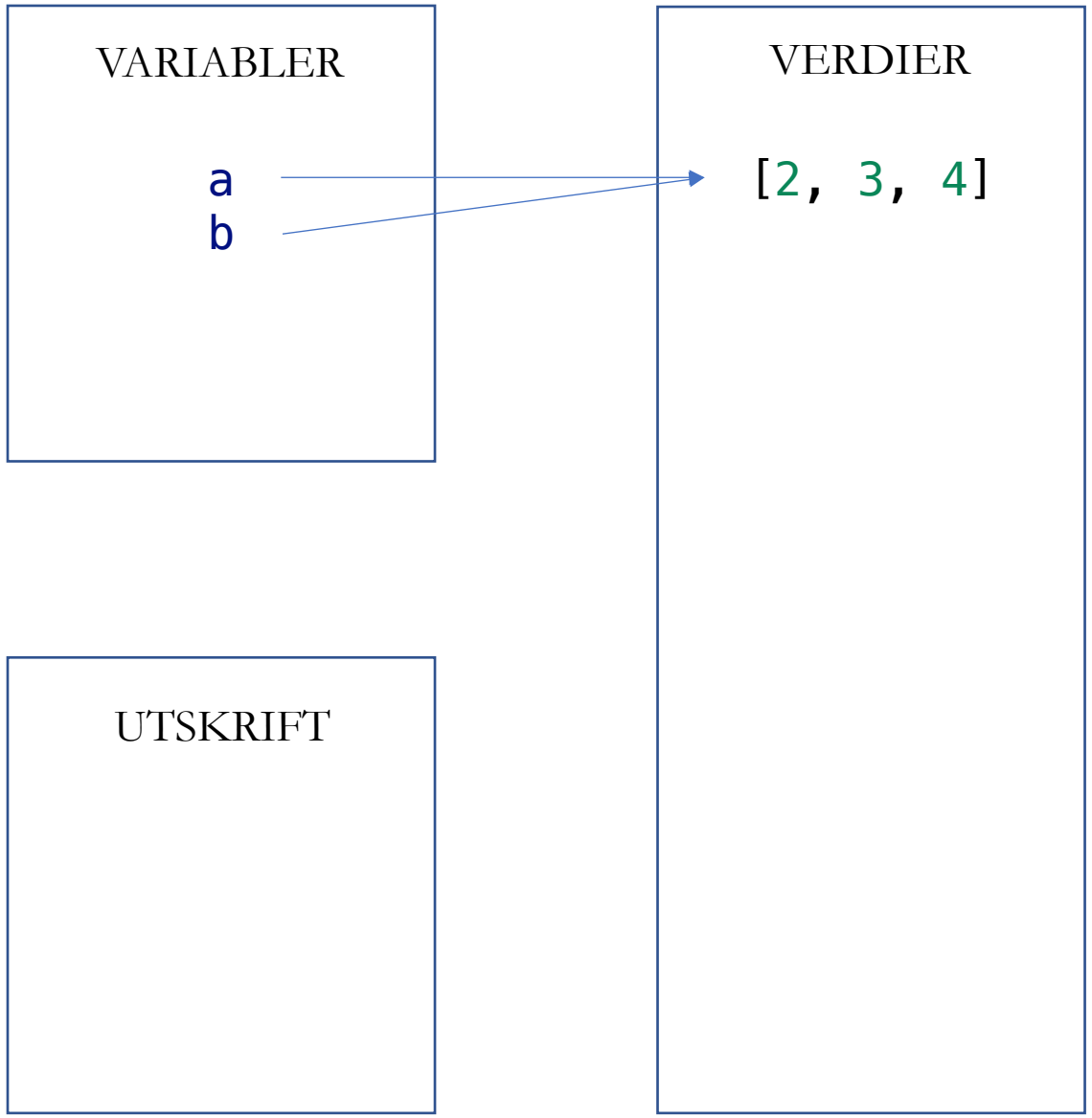
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Mutasjon av listen
a[0] = 99
b[1] = 42
print(a)
print(b)
```



ALIAS

+ mutasjon

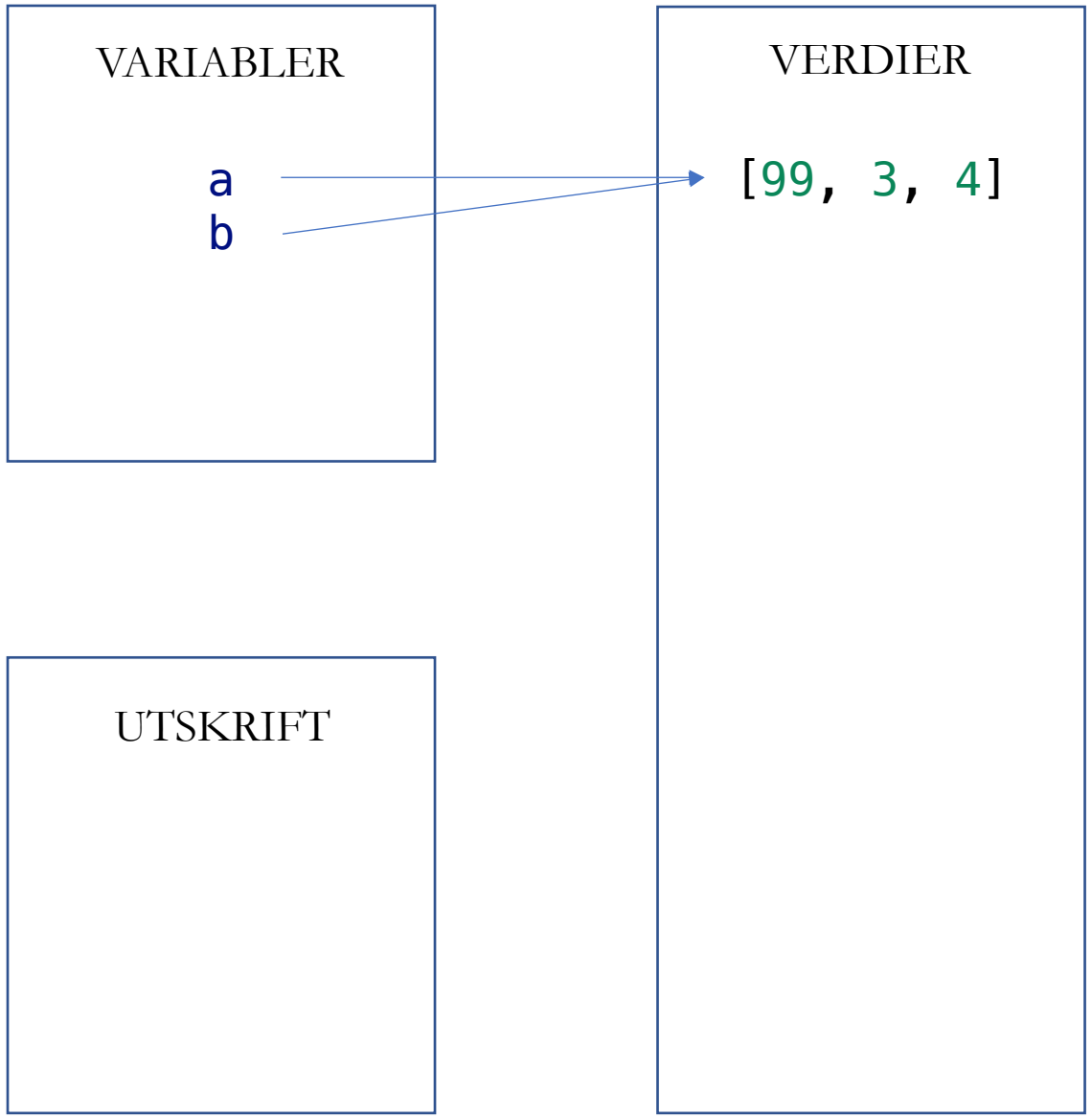
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Mutasjon av listen
a[0] = 99
b[1] = 42
print(a)
print(b)
```



ALIAS

+ mutasjon

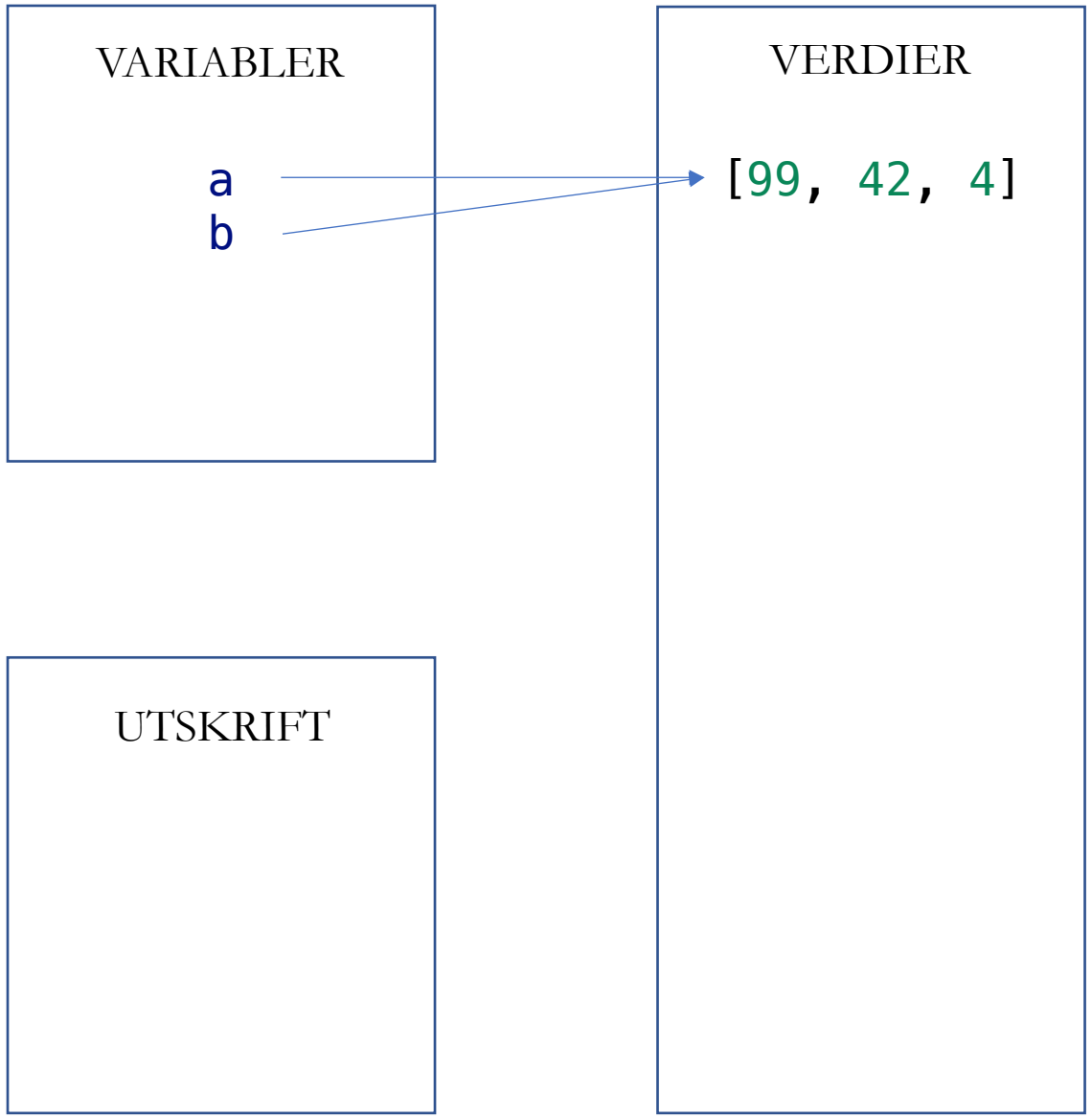
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Mutasjon av listen
a[0] = 99
b[1] = 42
print(a)
print(b)
```



ALIAS

+ mutasjon

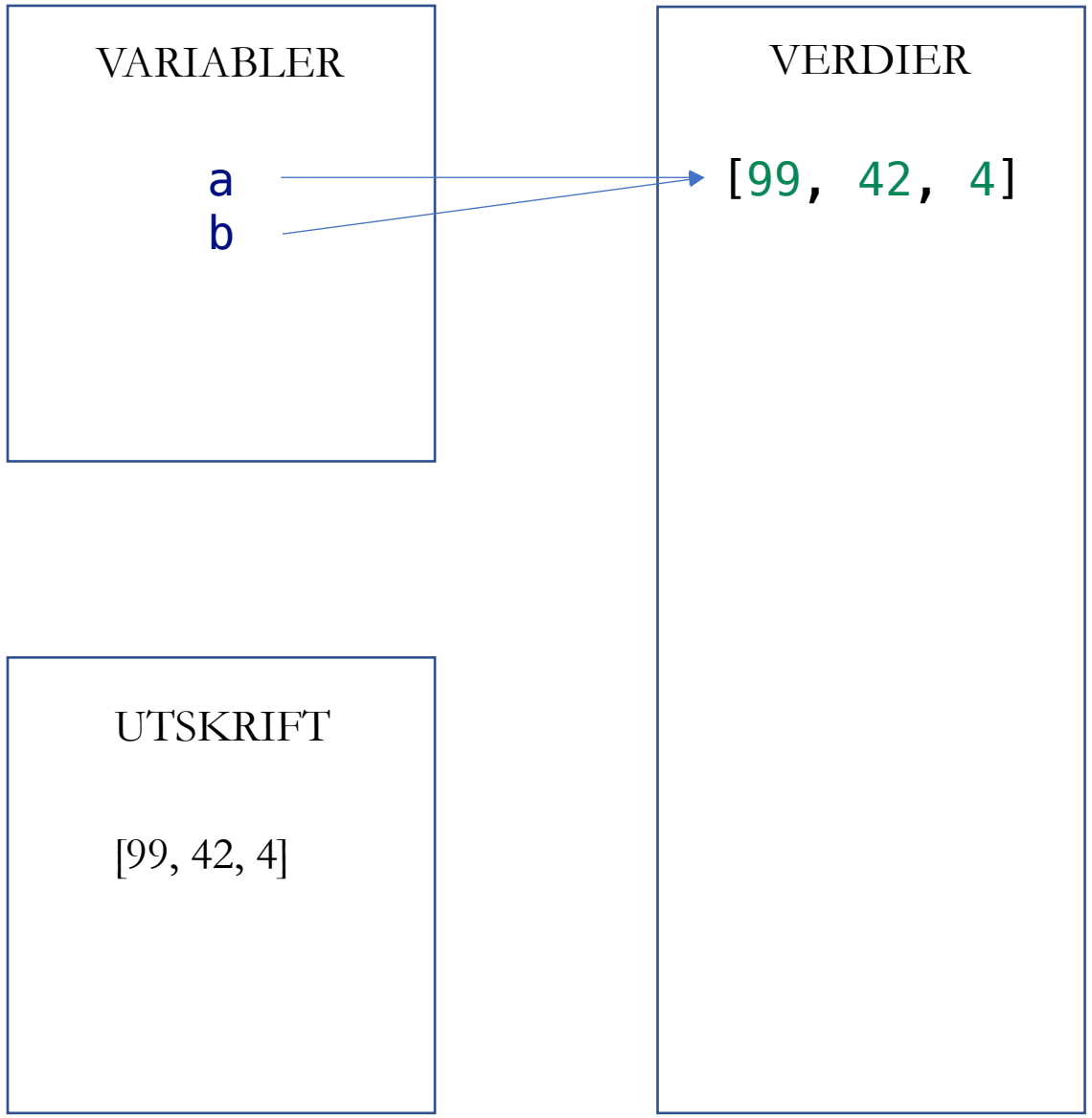
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Mutasjon av listen
a[0] = 99
b[1] = 42
print(a)
print(b)
```



ALIAS

+ mutasjon

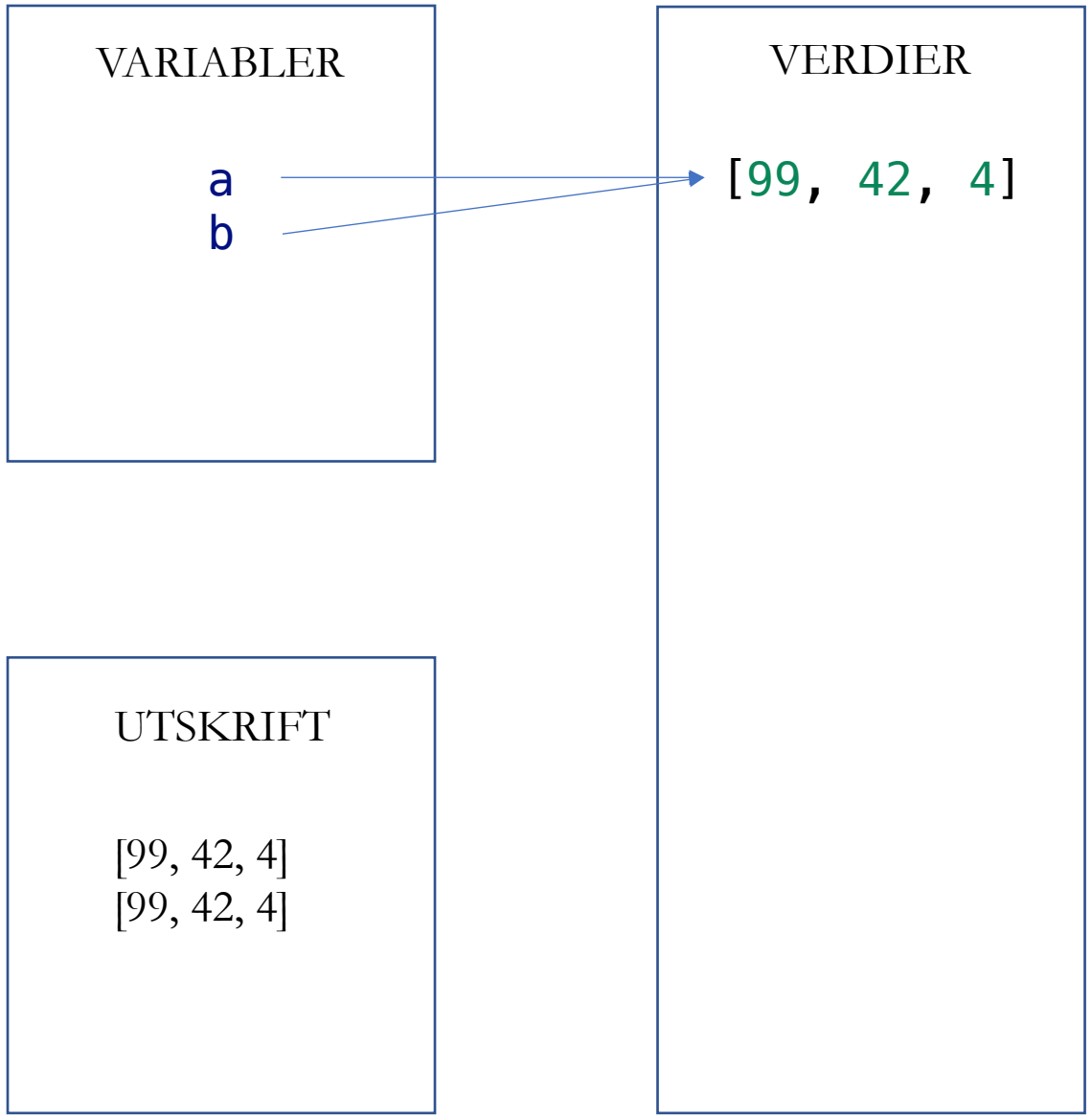
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Mutasjon av listen
a[0] = 99
b[1] = 42
print(a)
print(b)
```



ALIAS

+ mutasjon

```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Mutasjon av listen
a[0] = 99
b[1] = 42
print(a)
print(b)
```



ALIAS

+ ikke-destruktive endringer

```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Ikke-destruktiv endring
a = [99] + a[1:]
b = b[:1] + [42] + b[2:]
print(a)
print(b)
```

VARIABLER

UTSKRIFT

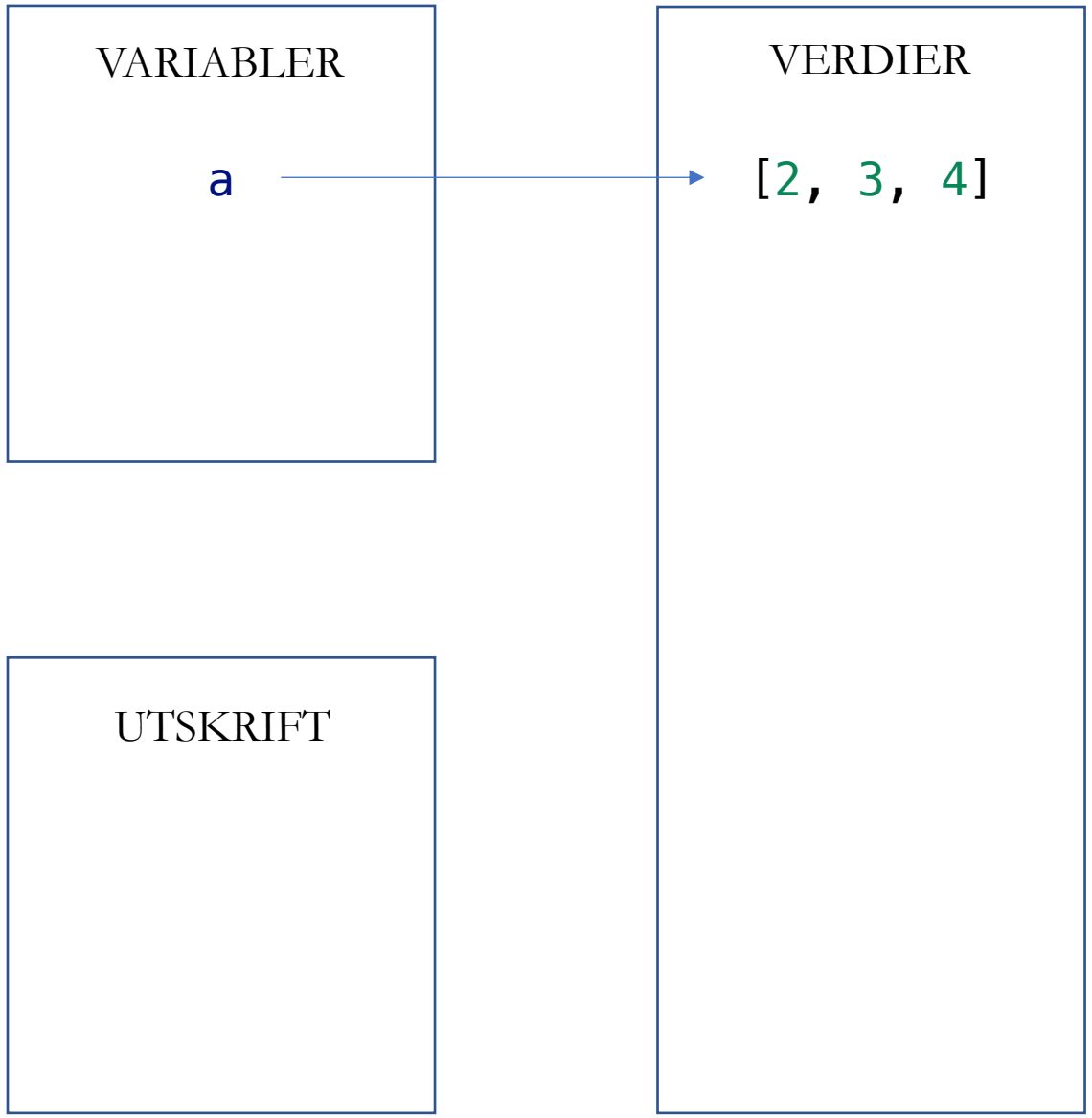
VERDIER

```
[2, 3, 4]
```

ALIAS

+ ikke-destruktive endringer

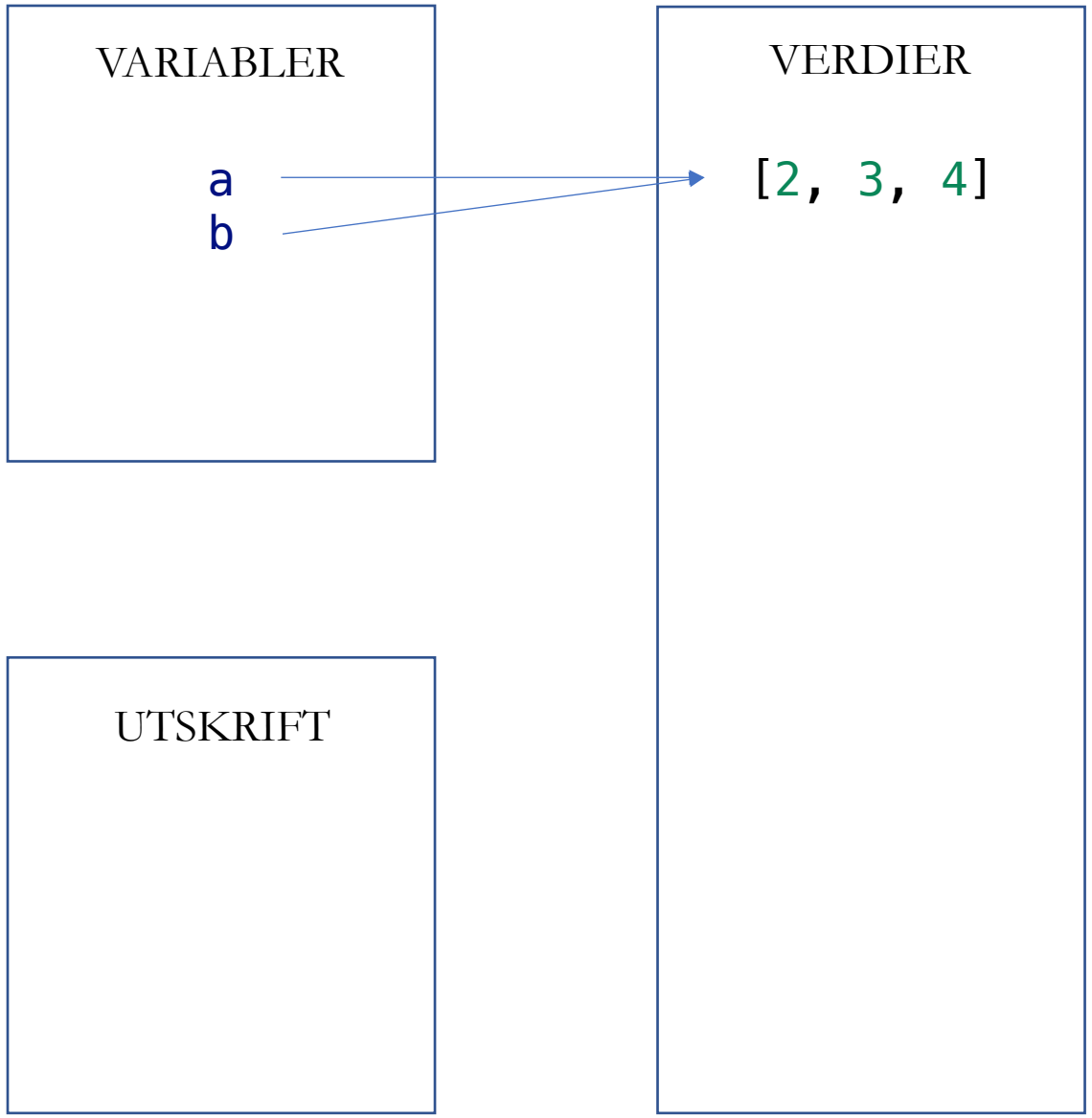
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Ikke-destruktiv endring
a = [99] + a[1:]
b = b[:1] + [42] + b[2:]
print(a)
print(b)
```



ALIAS

+ ikke-destruktive endringer

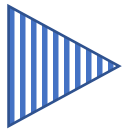
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Ikke-destruktiv endring
a = [99] + a[1:]
b = b[:1] + [42] + b[2:]
print(a)
print(b)
```

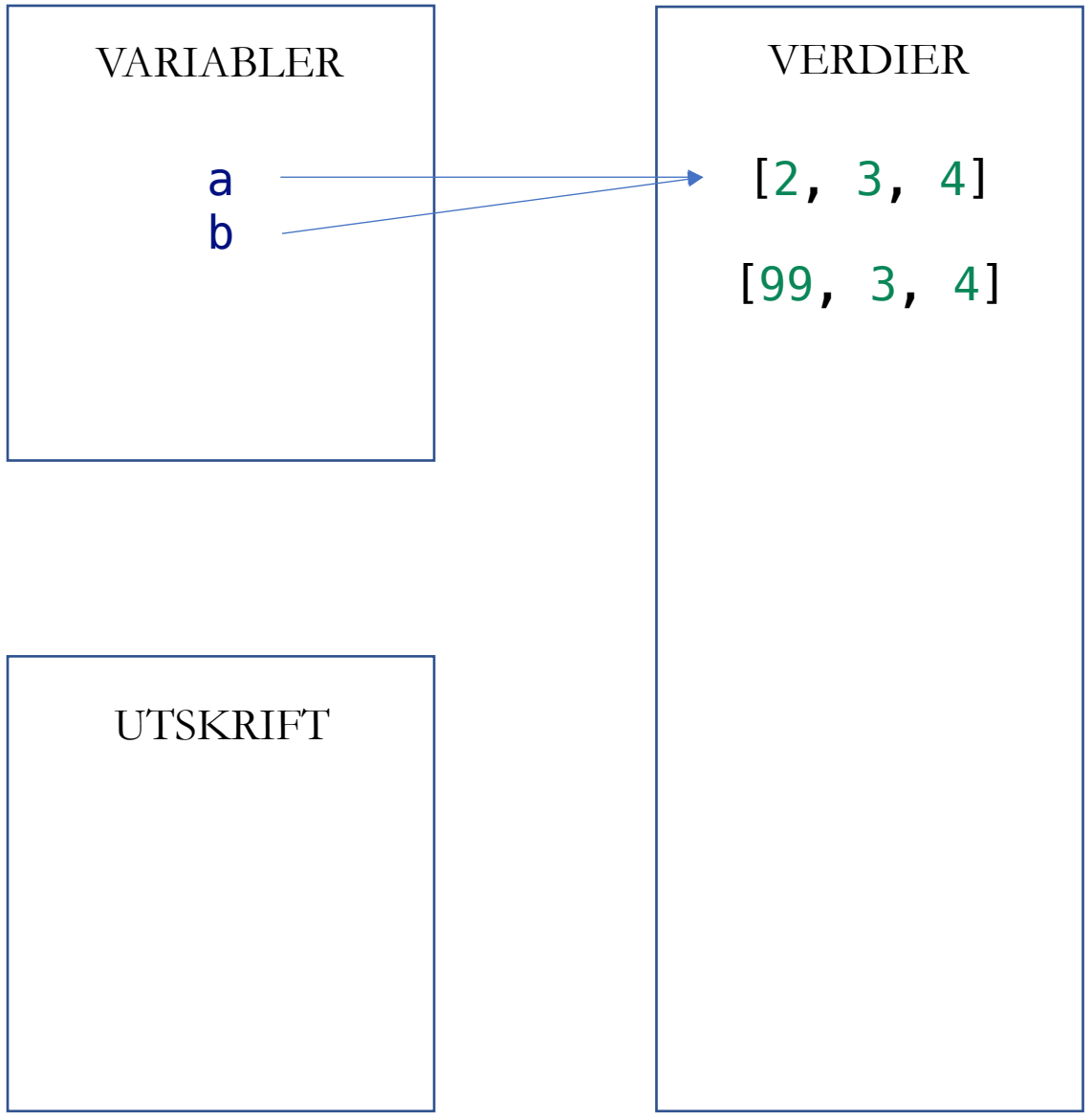


ALIAS

+ ikke-destruktive endringer

```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Ikke-destruktiv endring
a = [99] + a[1:]
b = b[:1] + [42] + b[2:]
print(a)
print(b)
```

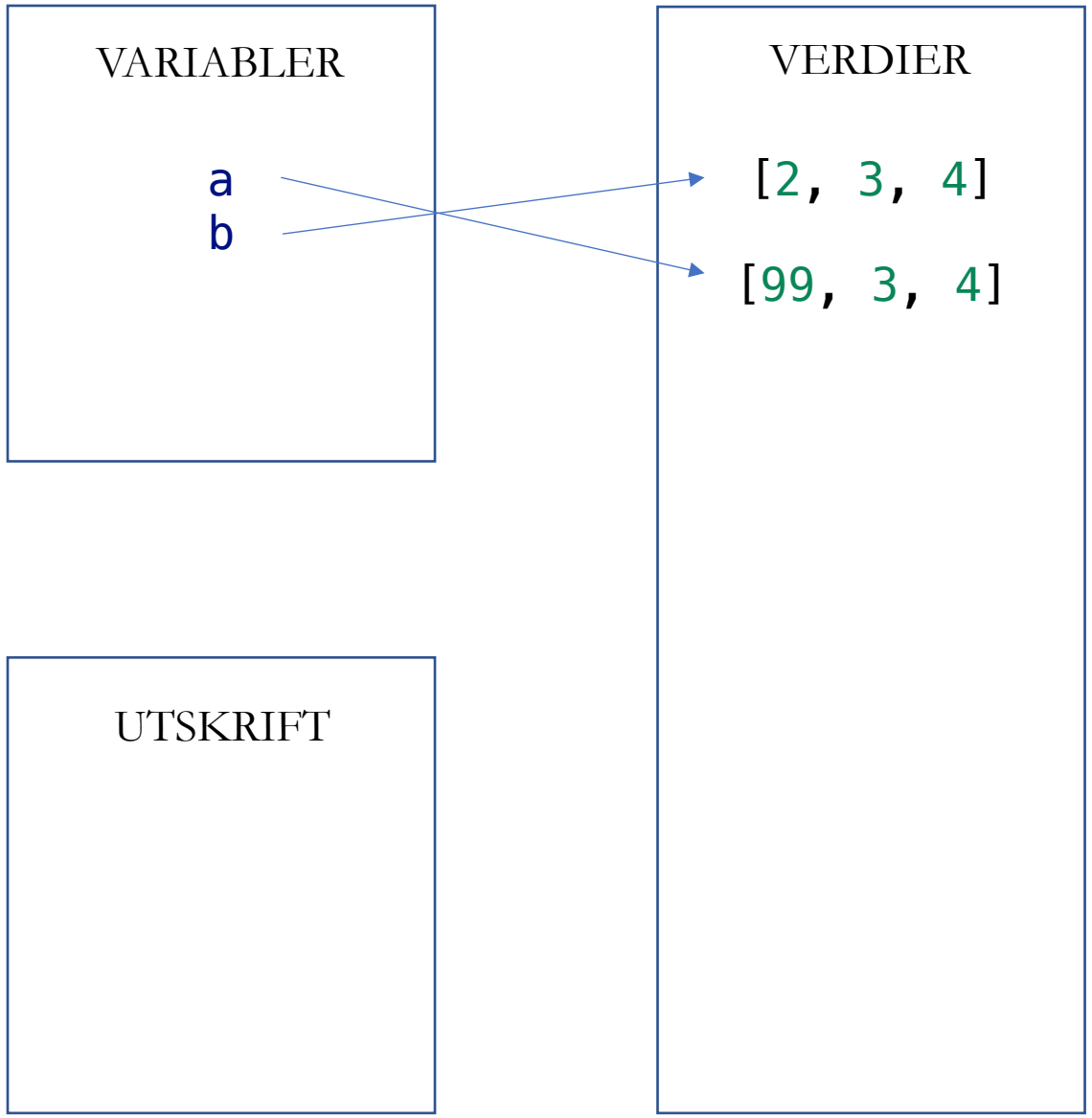




ALIAS

+ ikke-destruktive endringer

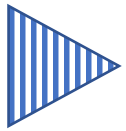
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Ikke-destruktiv endring
a = [99] + a[1:]
b = b[:1] + [42] + b[2:]
print(a)
print(b)
```

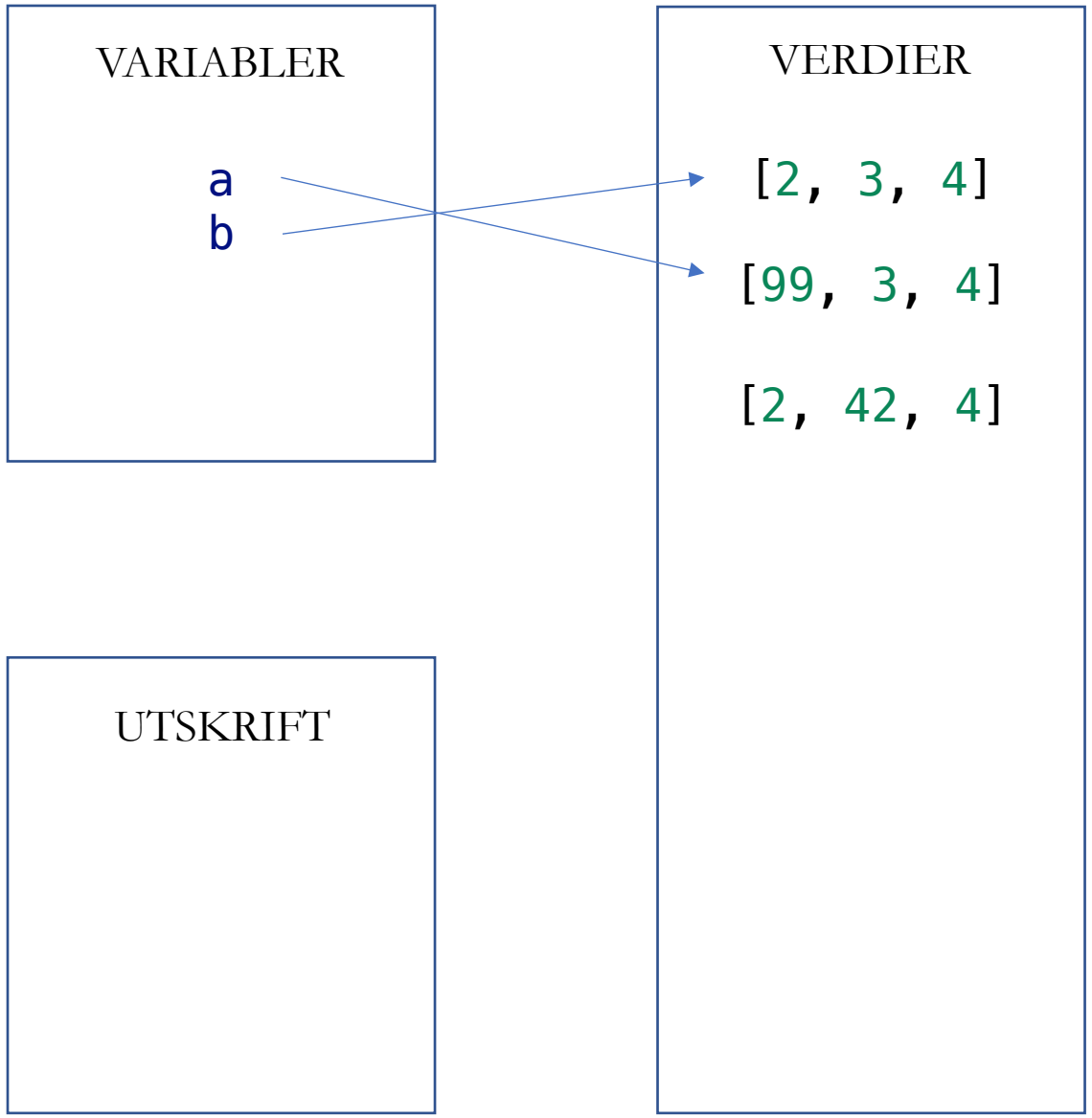


ALIAS

+ ikke-destruktive endringer

```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Ikke-destruktiv endring
a = [99] + a[1:]
b = b[:1] + [42] + b[2:]
print(a)
print(b)
```

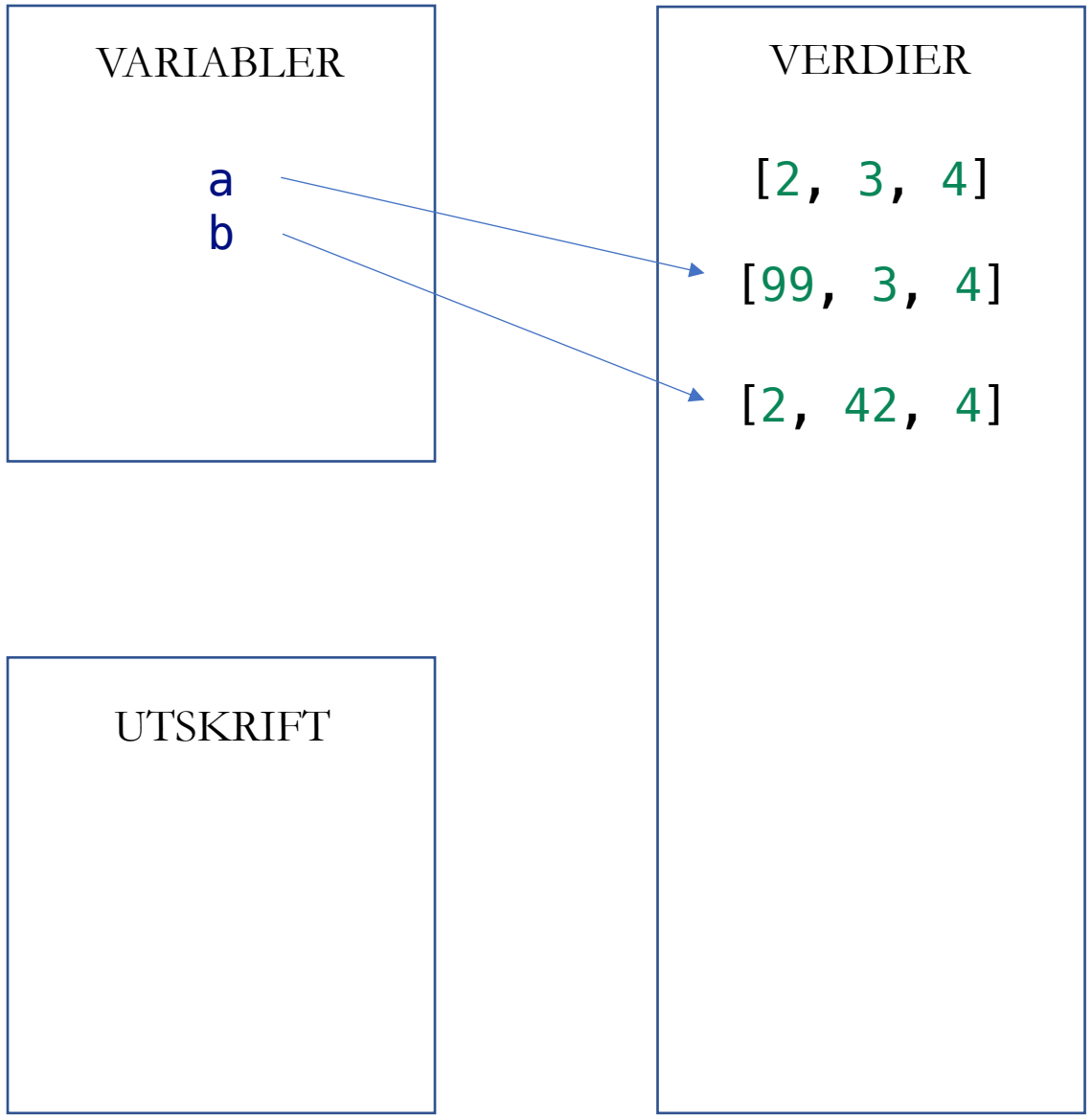




ALIAS

+ ikke-destruktive endringer

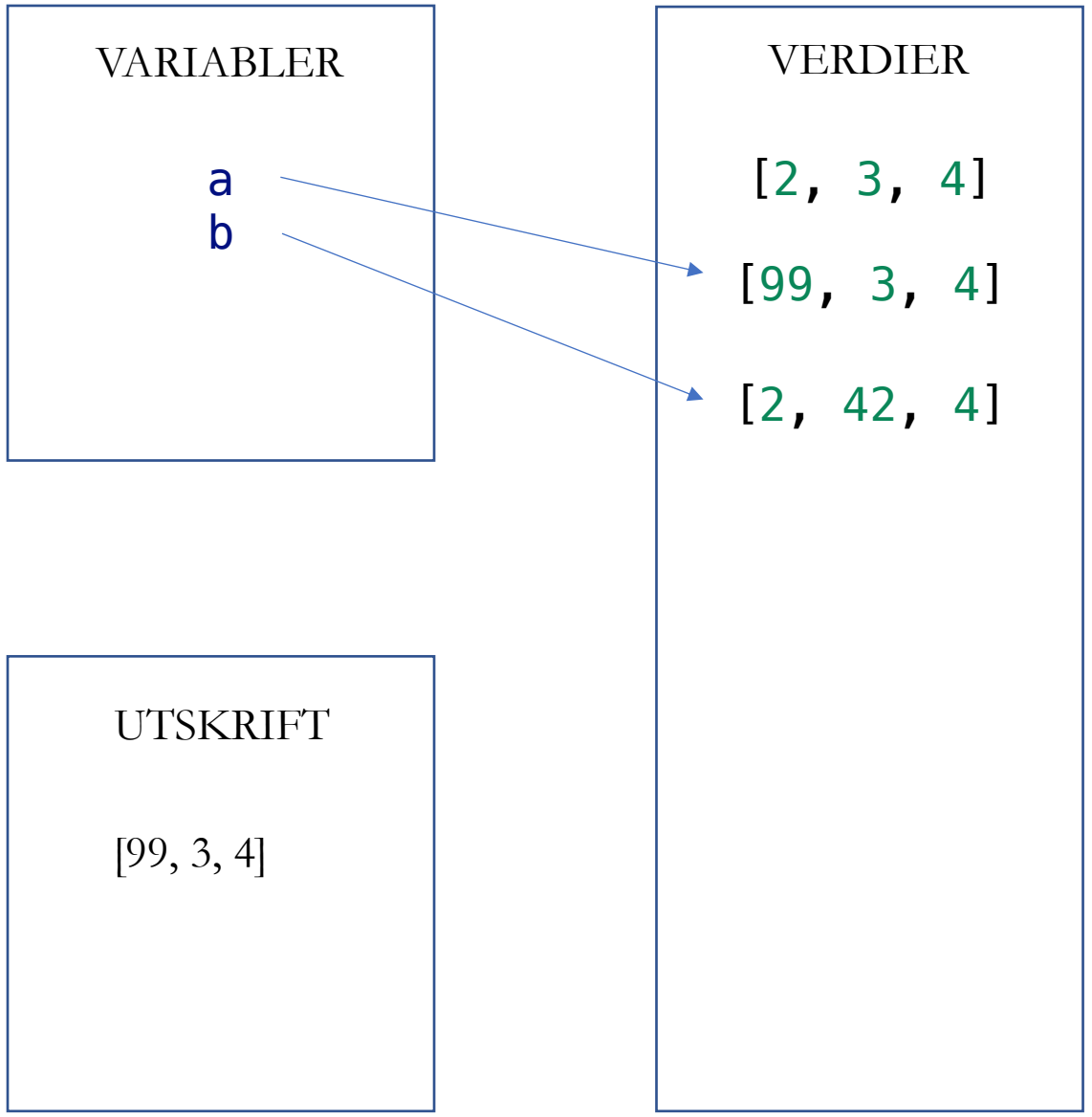
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Ikke-destruktiv endring
a = [99] + a[1:]
b = b[:1] + [42] + b[2:]
print(a)
print(b)
```



ALIAS

+ ikke-destruktive endringer

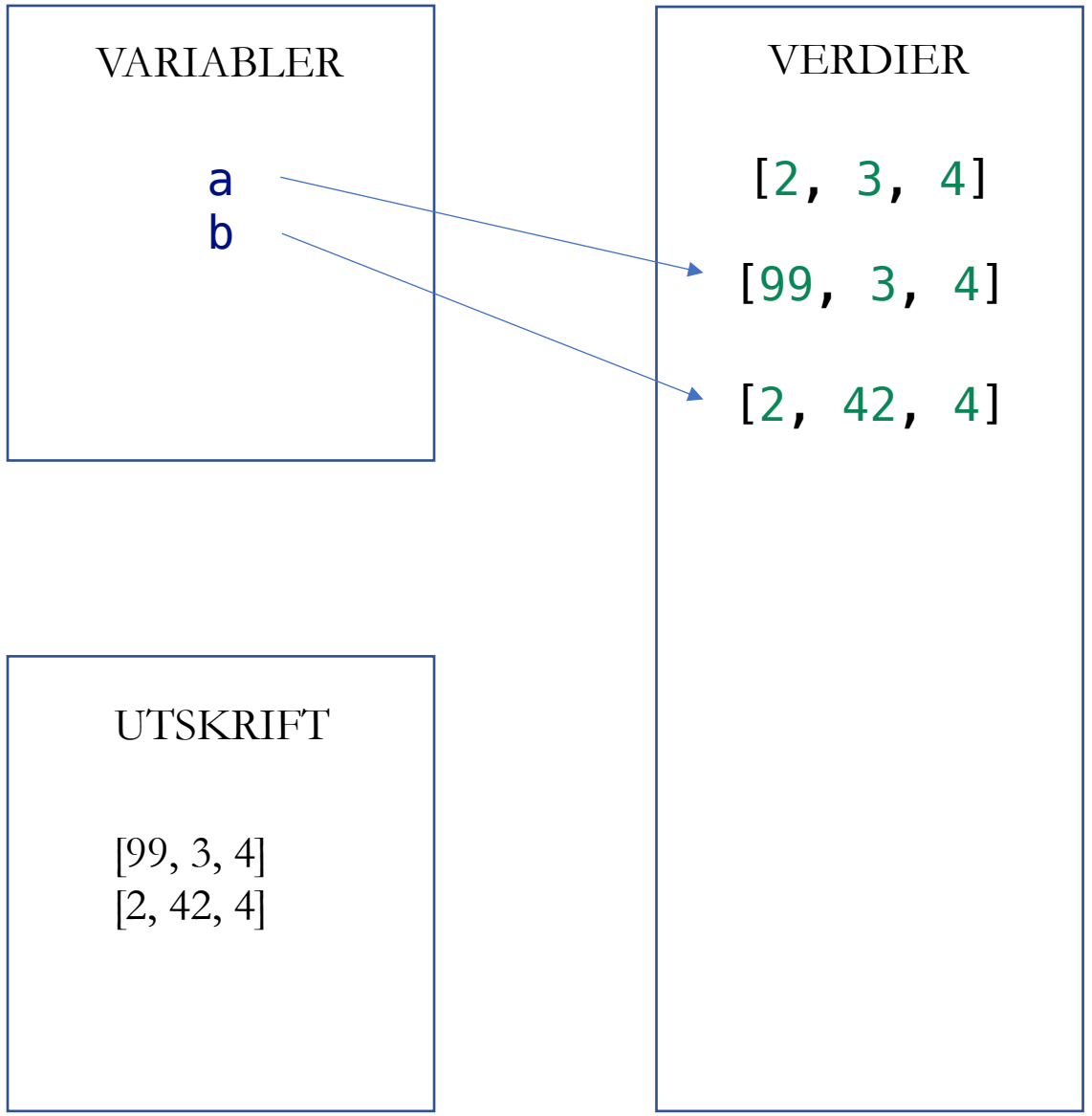
```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Ikke-destruktiv endring
a = [99] + a[1:]
b = b[:1] + [42] + b[2:]
print(a)
print(b)
```



ALIAS

+ ikke-destruktive endringer

```
a = [2, 3, 4]

# Oppretter et alias
b = a

# Ikke-destruktiv endring
a = [99] + a[1:]
b = b[:1] + [42] + b[2:]
print(a)
print(b)
```



ALIAS

+ destruktiv funksjon

```
def foo(bar):
    bar[2] = 42

a = ["p", "q", "r", "s"]
foo(a)
print(a)
```

VARIABLER

UTSKRIFT

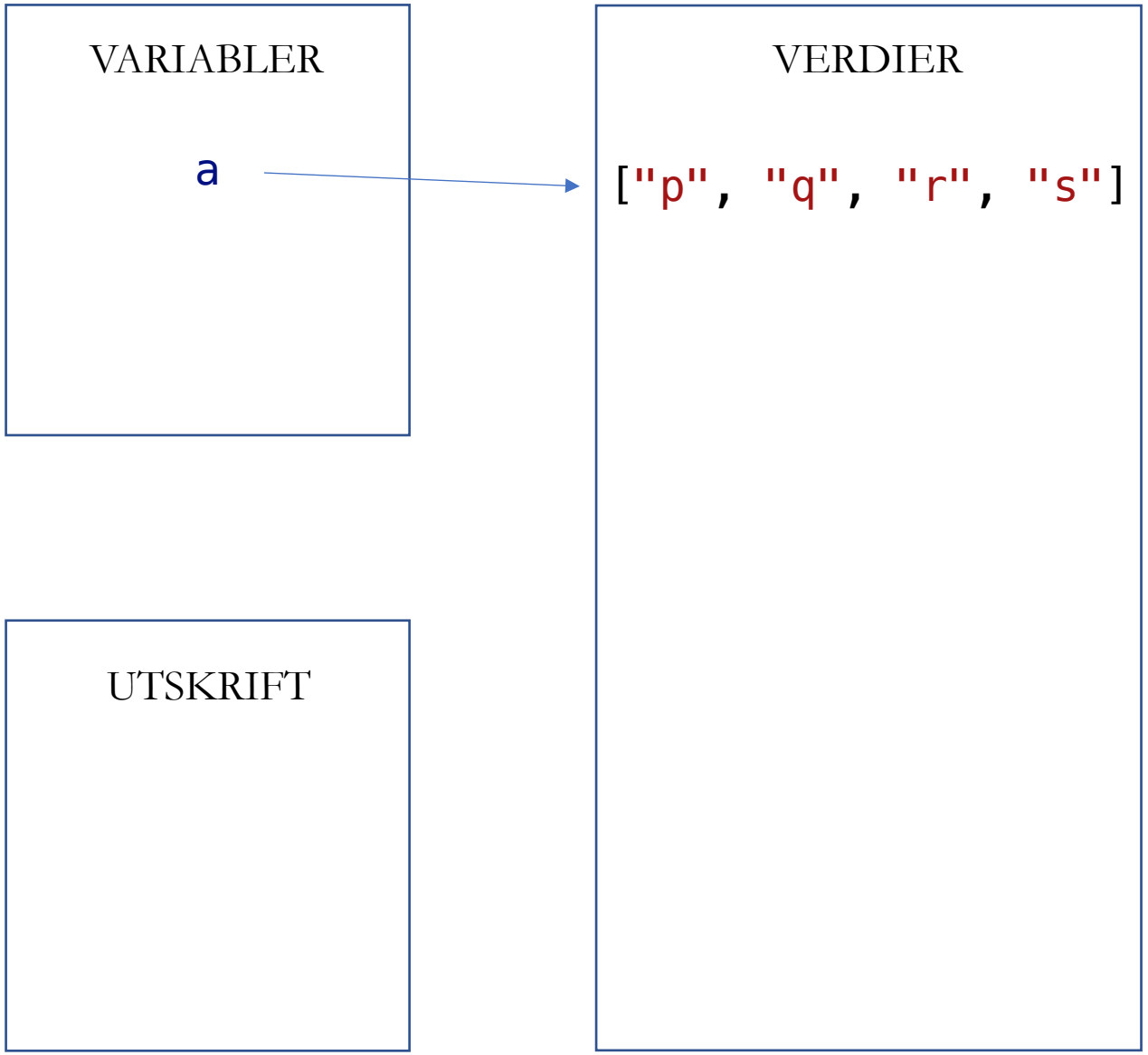
VERDIER

["p", "q", "r", "s"]

ALIAS

+ destruktiv funksjon

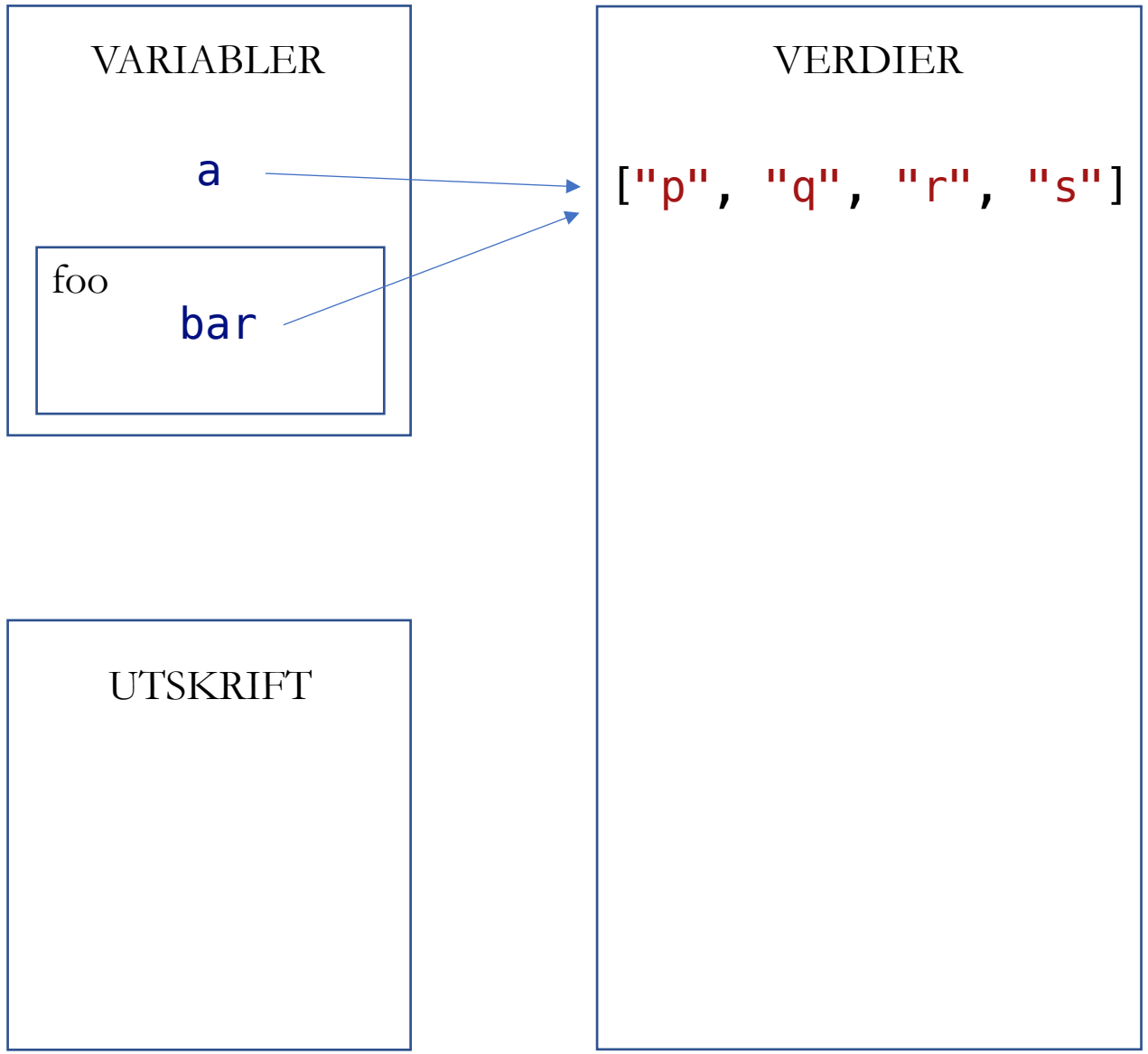
```
def foo(bar):
    bar[2] = 42

a = ["p", "q", "r", "s"]
foo(a)
print(a)
```



ALIAS

+ destruktiv funksjon

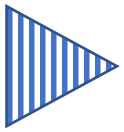
```
def foo(bar):
    bar[2] = 42

a = ["p", "q", "r", "s"]
foo(a)
print(a)
```

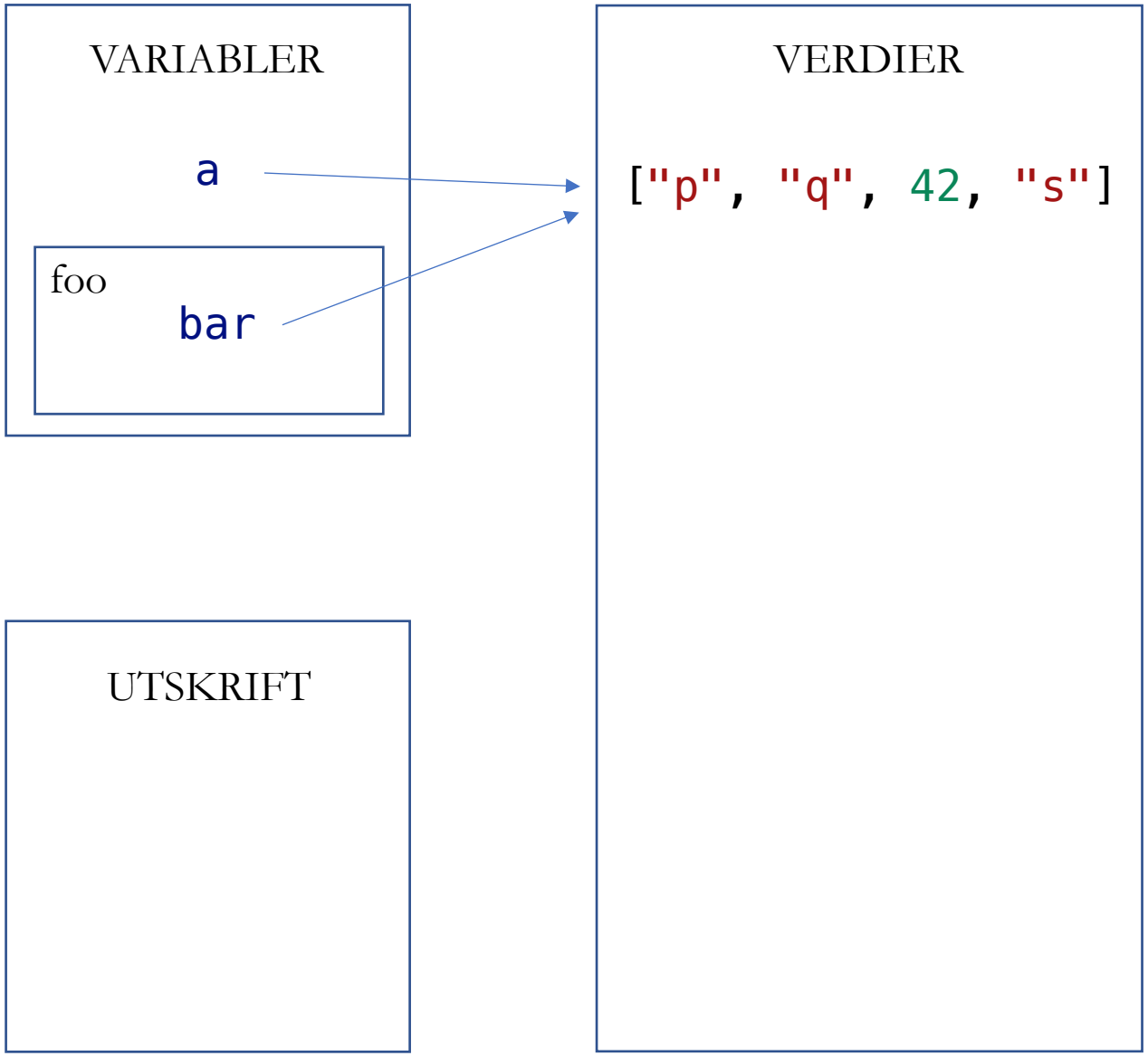


ALIAS

+ destruktiv funksjon

```
def foo(bar):
    bar[2] = 42

a = ["p", "q", "r", "s"]
foo(a)
print(a)
```

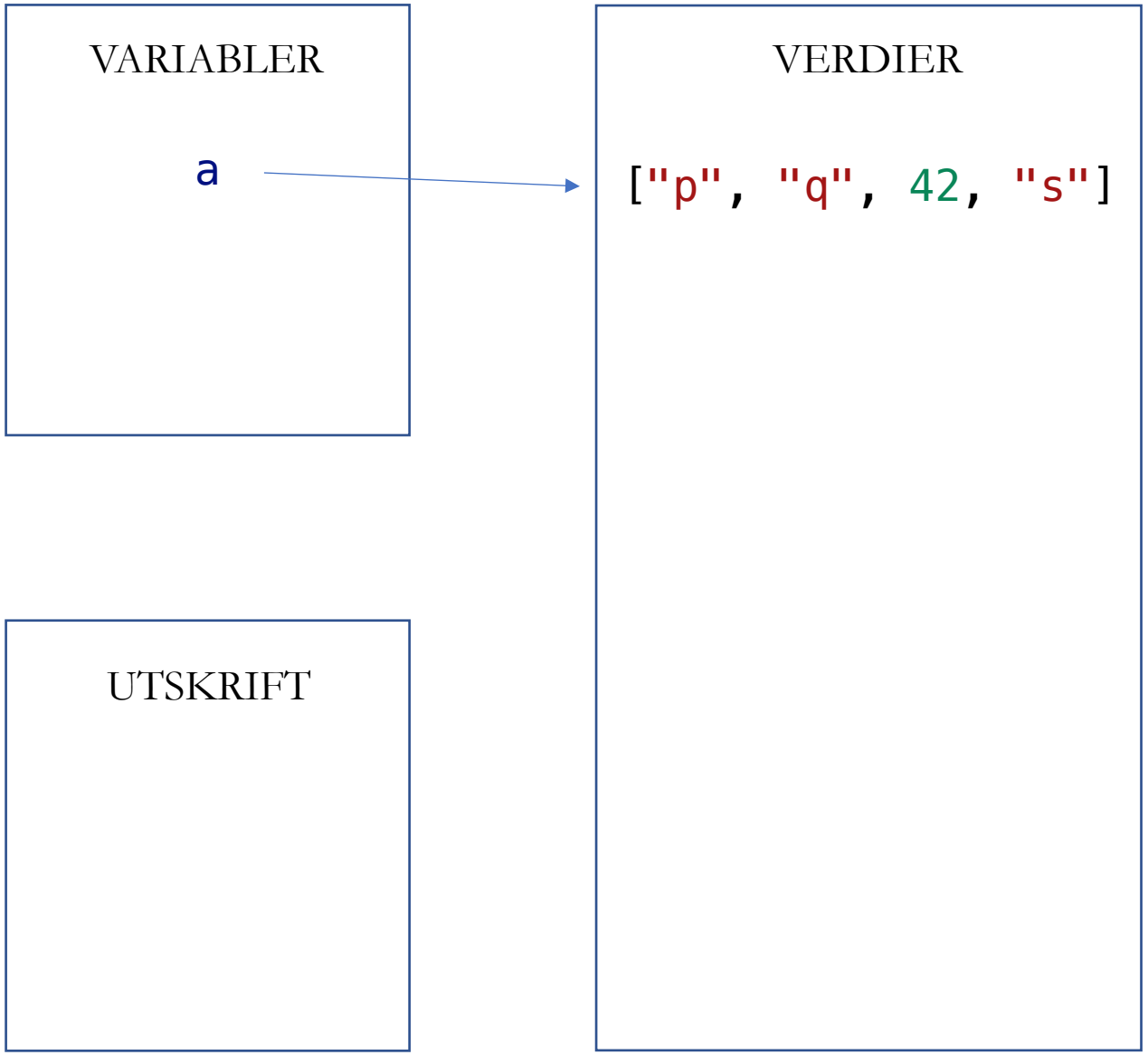




ALIAS

+ destruktiv funksjon

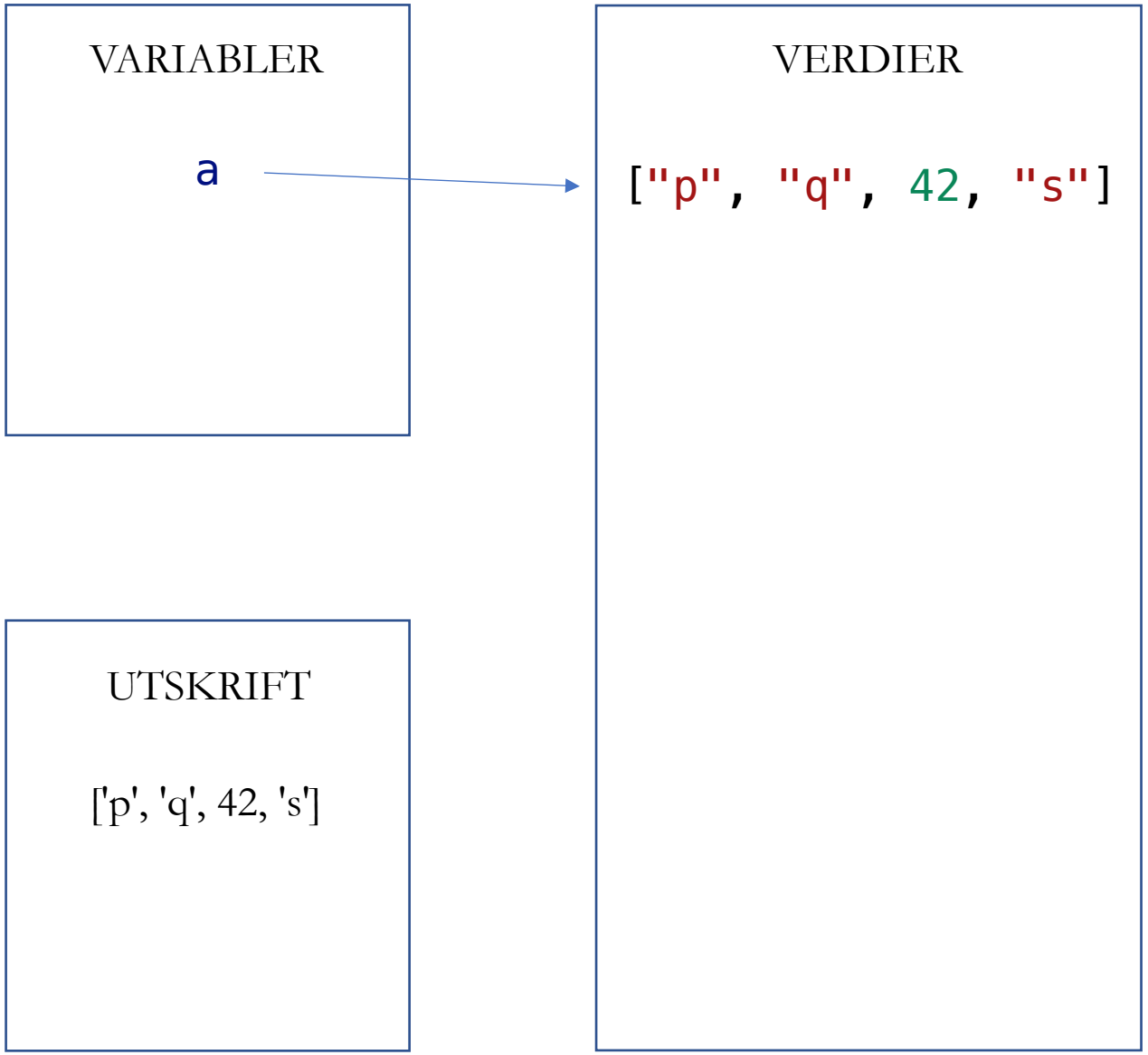
```
def foo(bar):
    bar[2] = 42

a = ["p", "q", "r", "s"]
foo(a)
print(a)
```



ALIAS

+ destruktiv funksjon

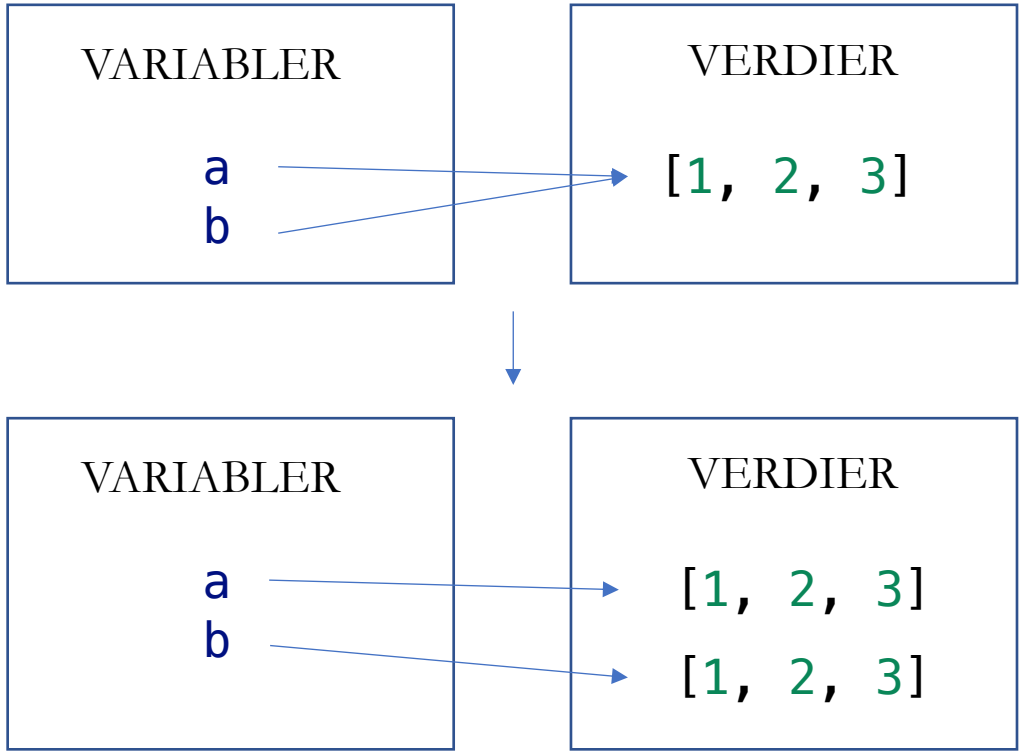
```
def foo(bar):
    bar[2] = 42

a = ["p", "q", "r", "s"]
foo(a)
print(a)
```



KOPIERING

- Vi kan bryte et alias ved å lage en kopi

```
a = [1, 2, 3]
b = a
b = b.copy()
```



- Destruktive operasjoner på kopien er uten sideeffekter
- Bruker mer tid og minne

2D-LISTER

- En liste som inneholder lister

```
table = [
    ["00", "01"],
    ["10", "11"],
    ["20", "21"],
]

print(table[2])          # ['20', '21']
print(table[2][0])       # 20



print(table[2][0][1])    # 0
```

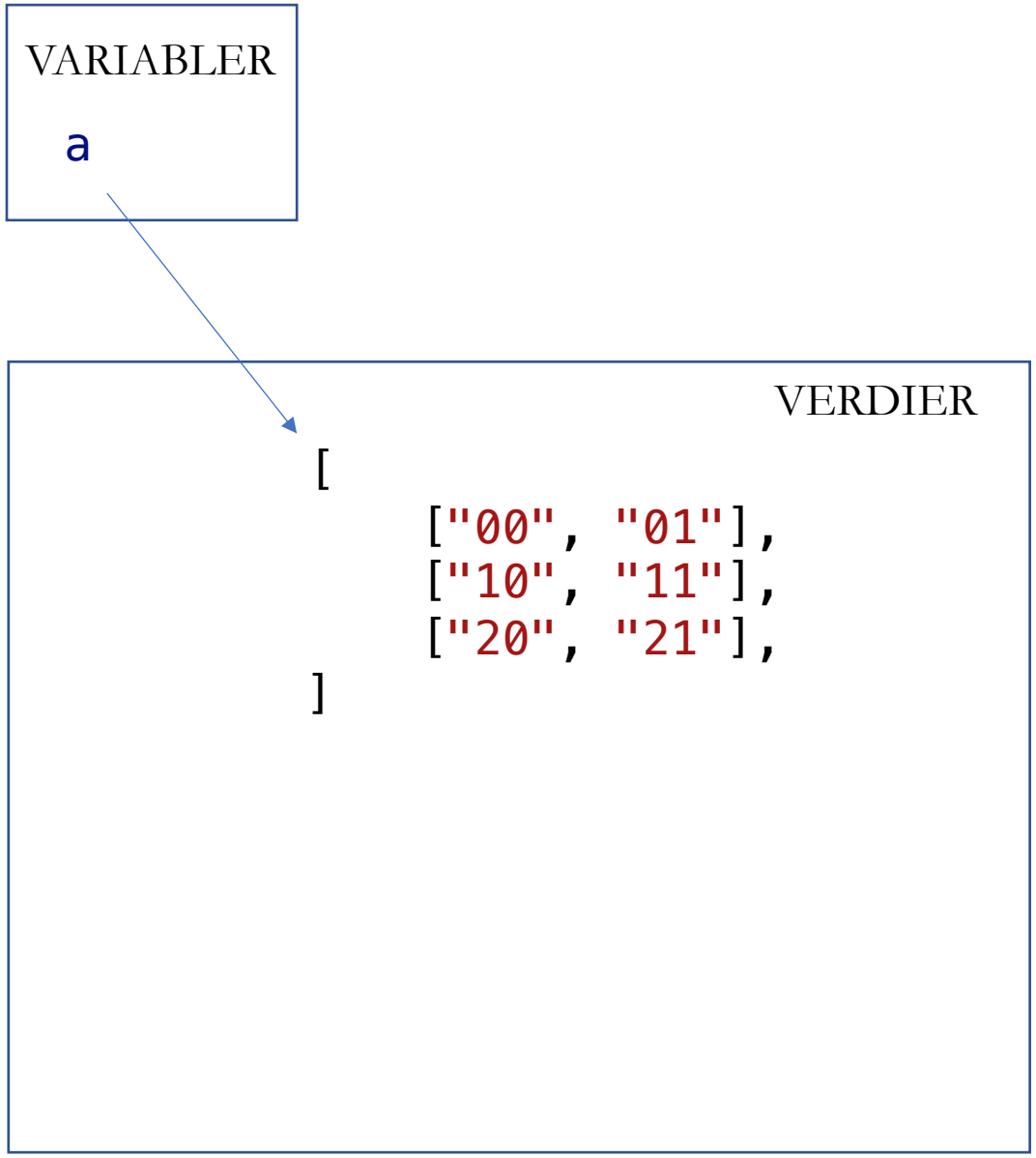
2D-LISTER

menti.com

6377 3840

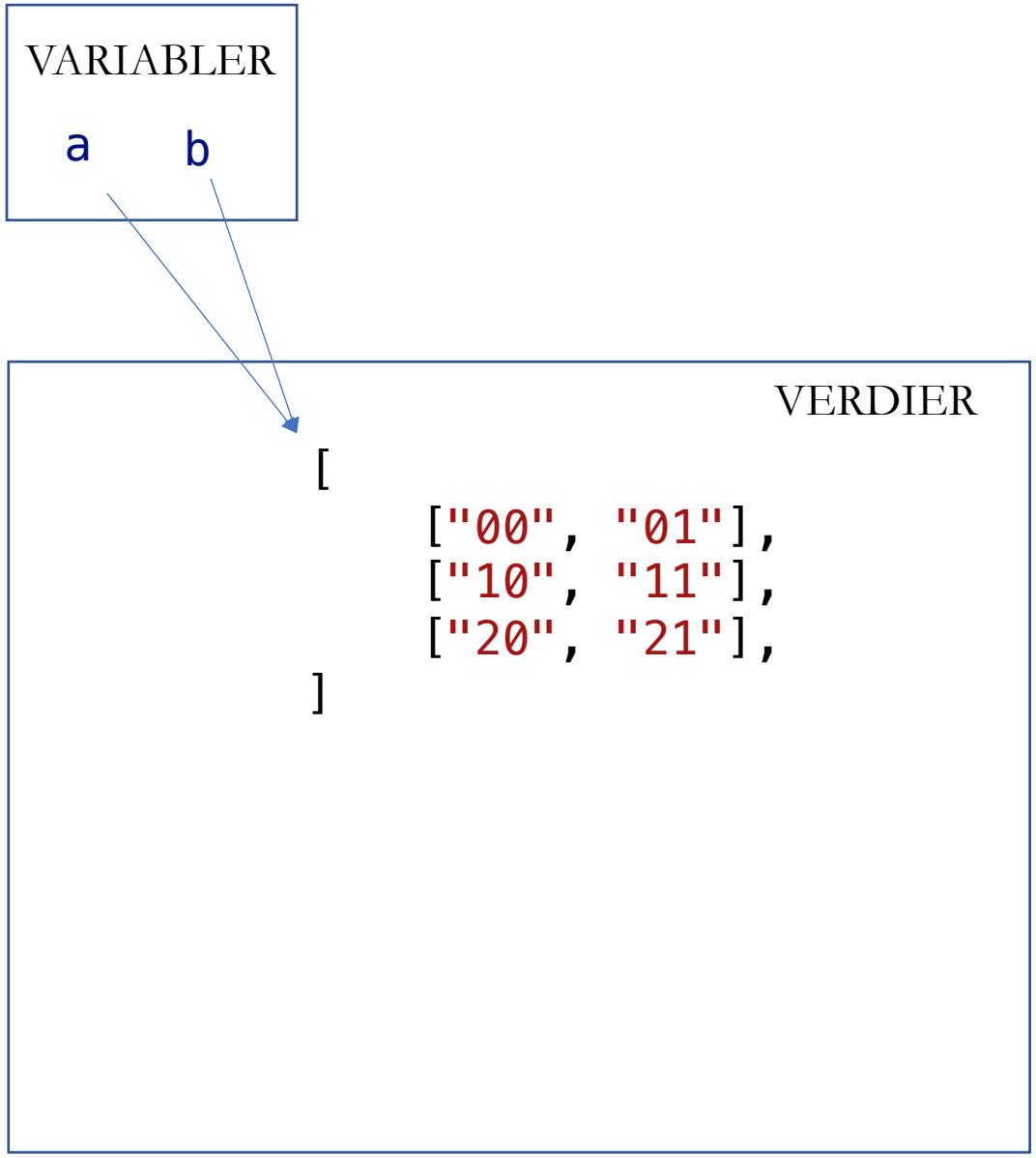
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a
b[1][0] = "foo"
print(a[1][0])
```



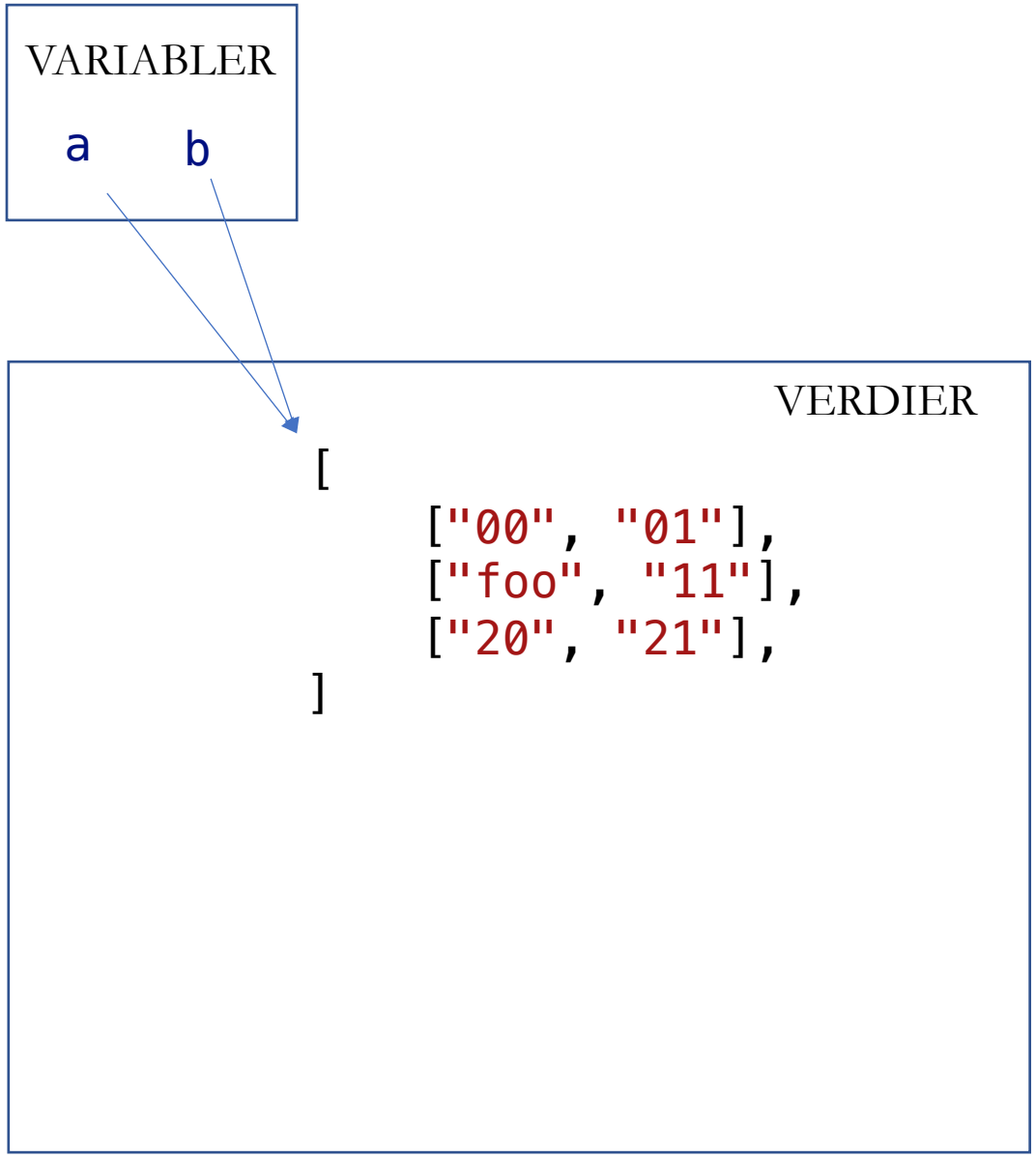
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a
b[1][0] = "foo"
print(a[1][0])
```



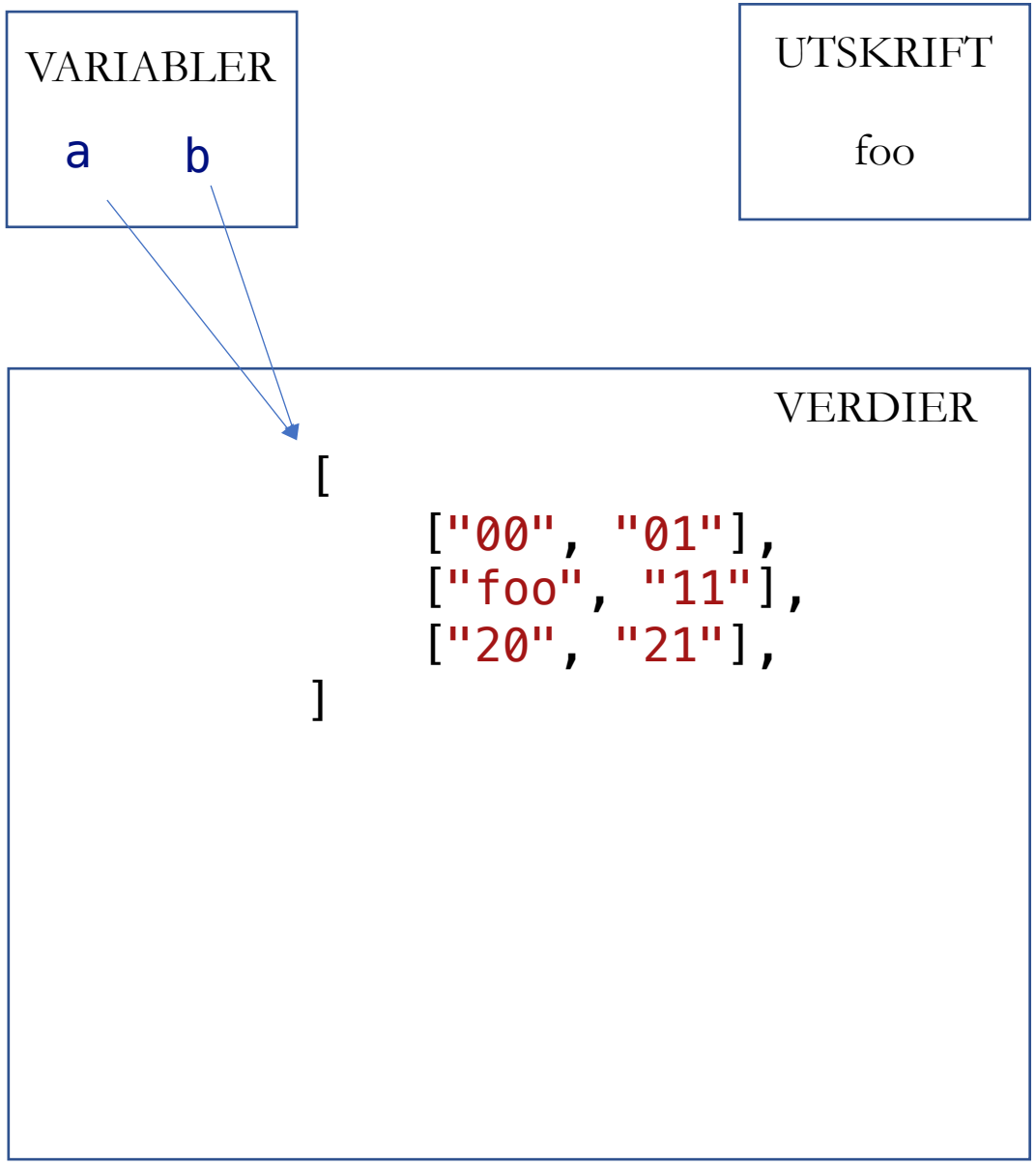
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a
b[1][0] = "foo"
print(a[1][0])
```



2D-LISTE

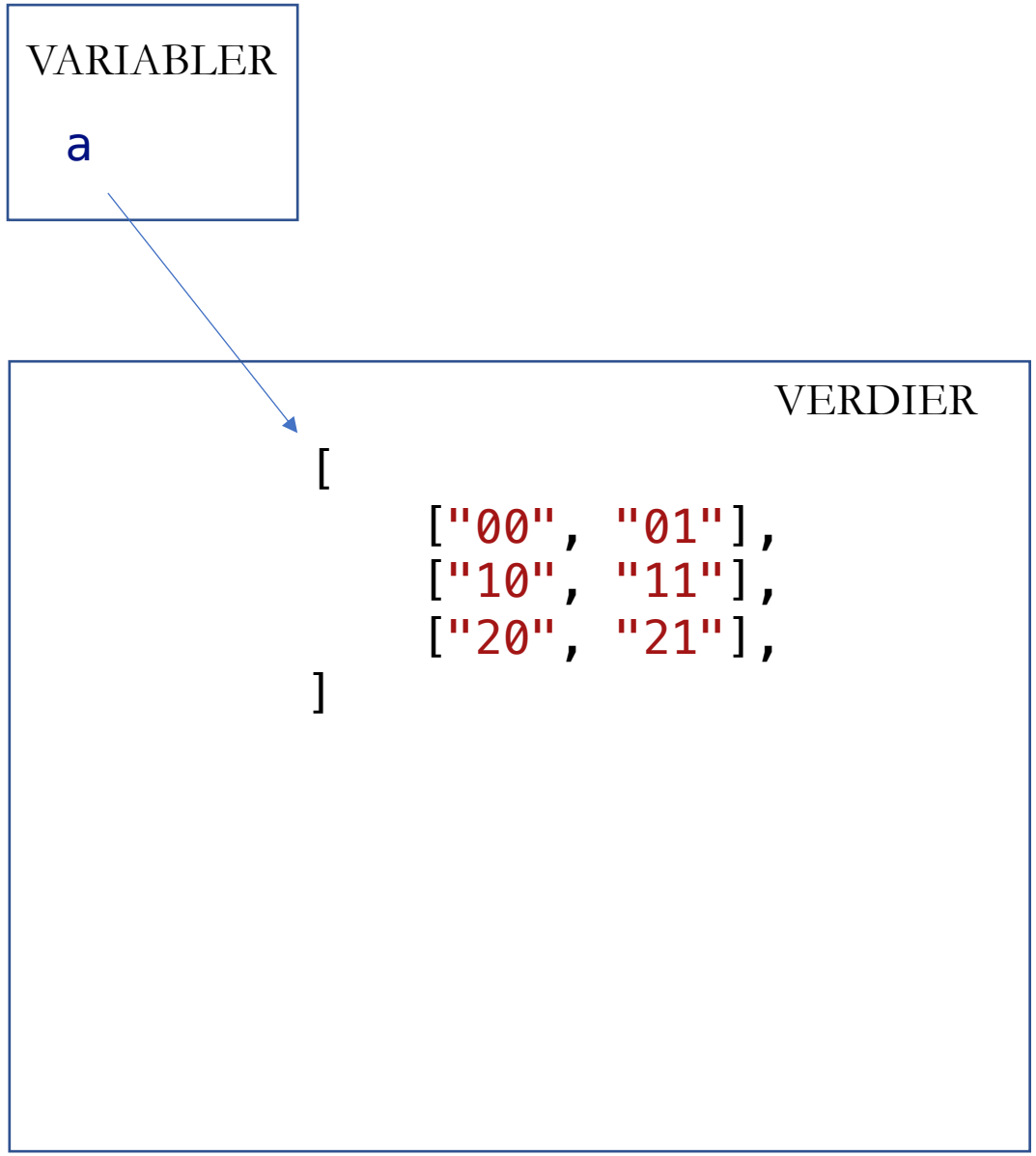
```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a
b[1][0] = "foo"
print(a[1][0])
```



NYTT EKSEMPEL

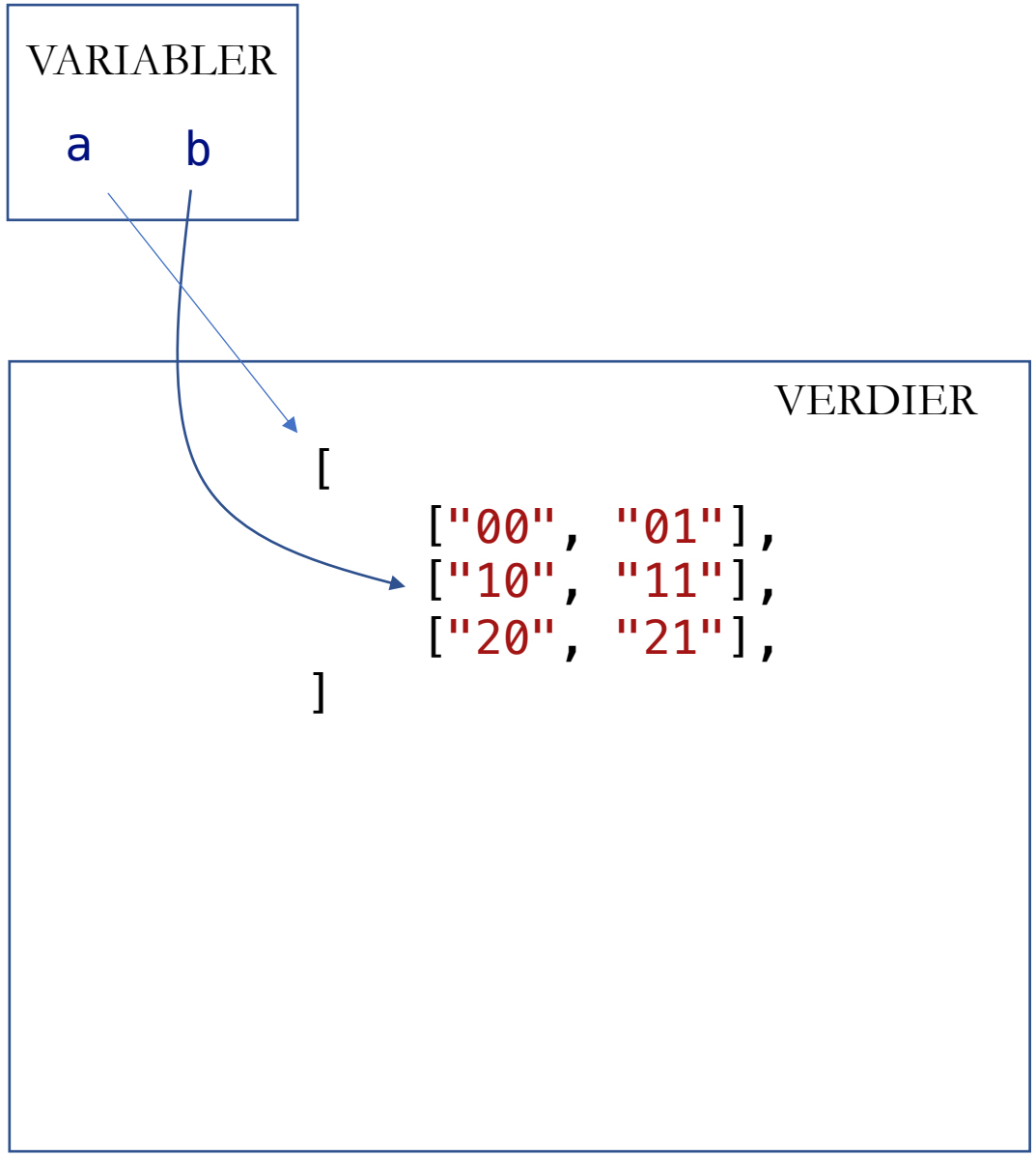
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
b[0] = "foo"
print(a[1][0])
```



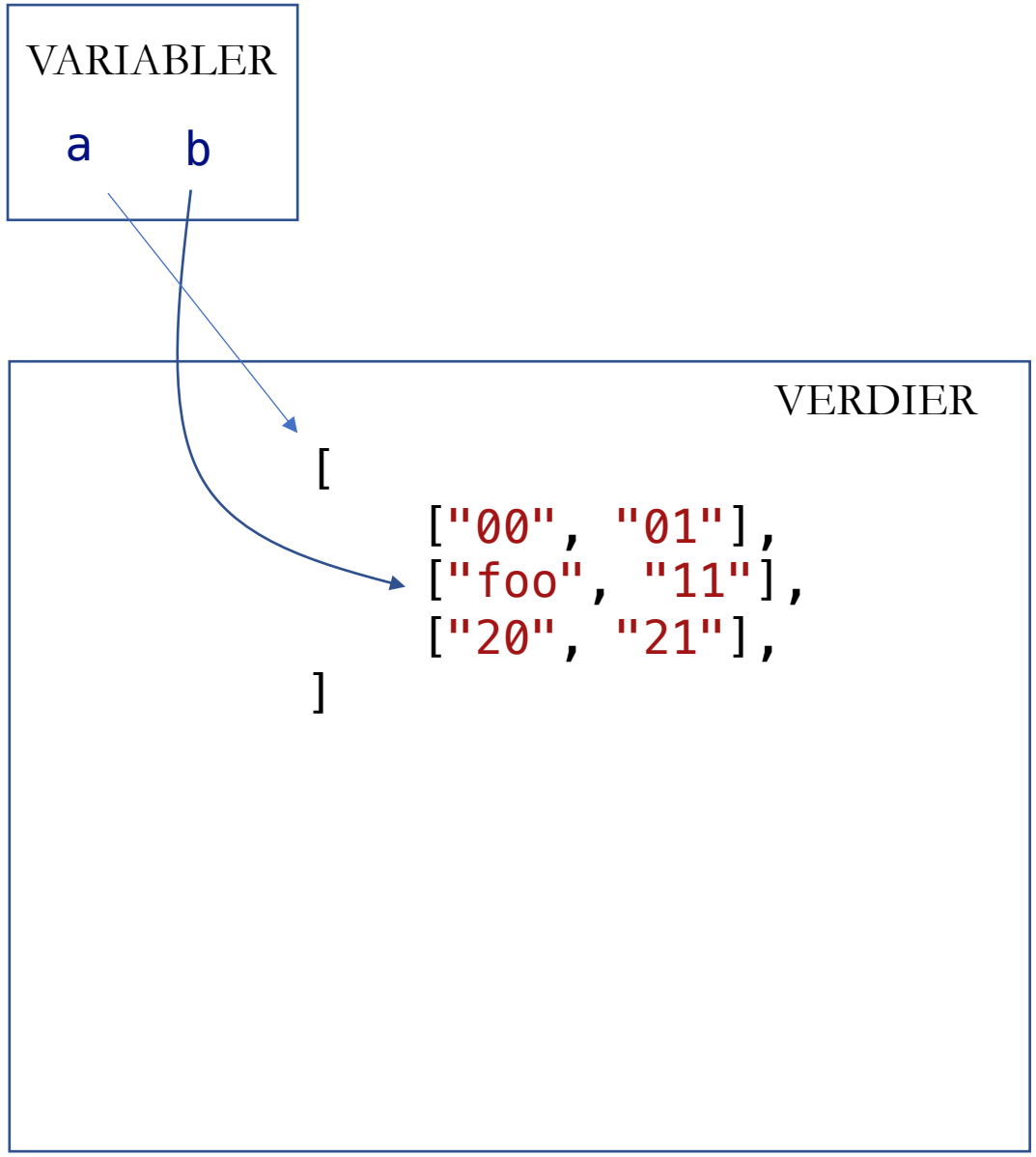
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
b[0] = "foo"
print(a[1][0])
```



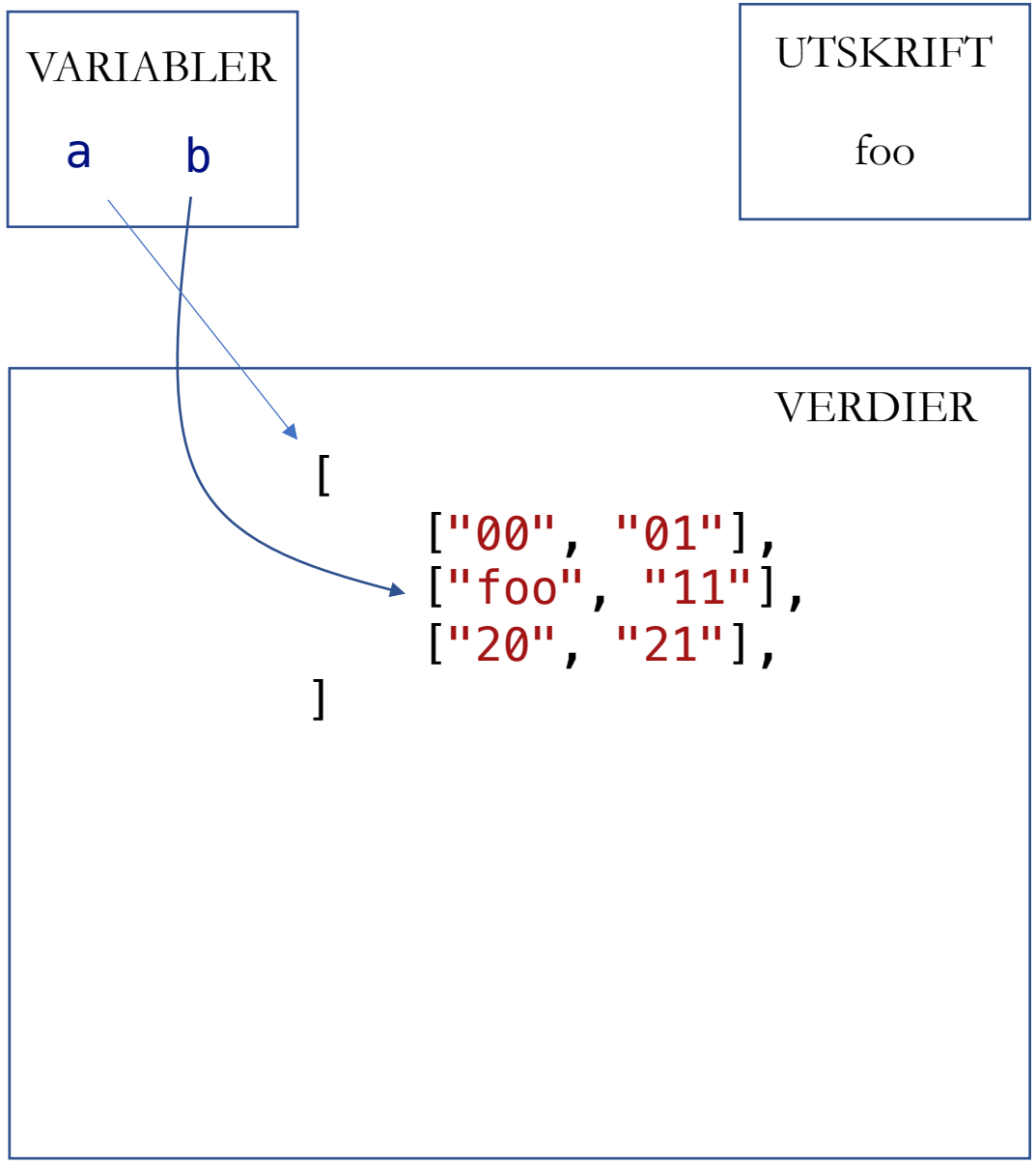
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
b[0] = "foo"
print(a[1][0])
```



2D-LISTE

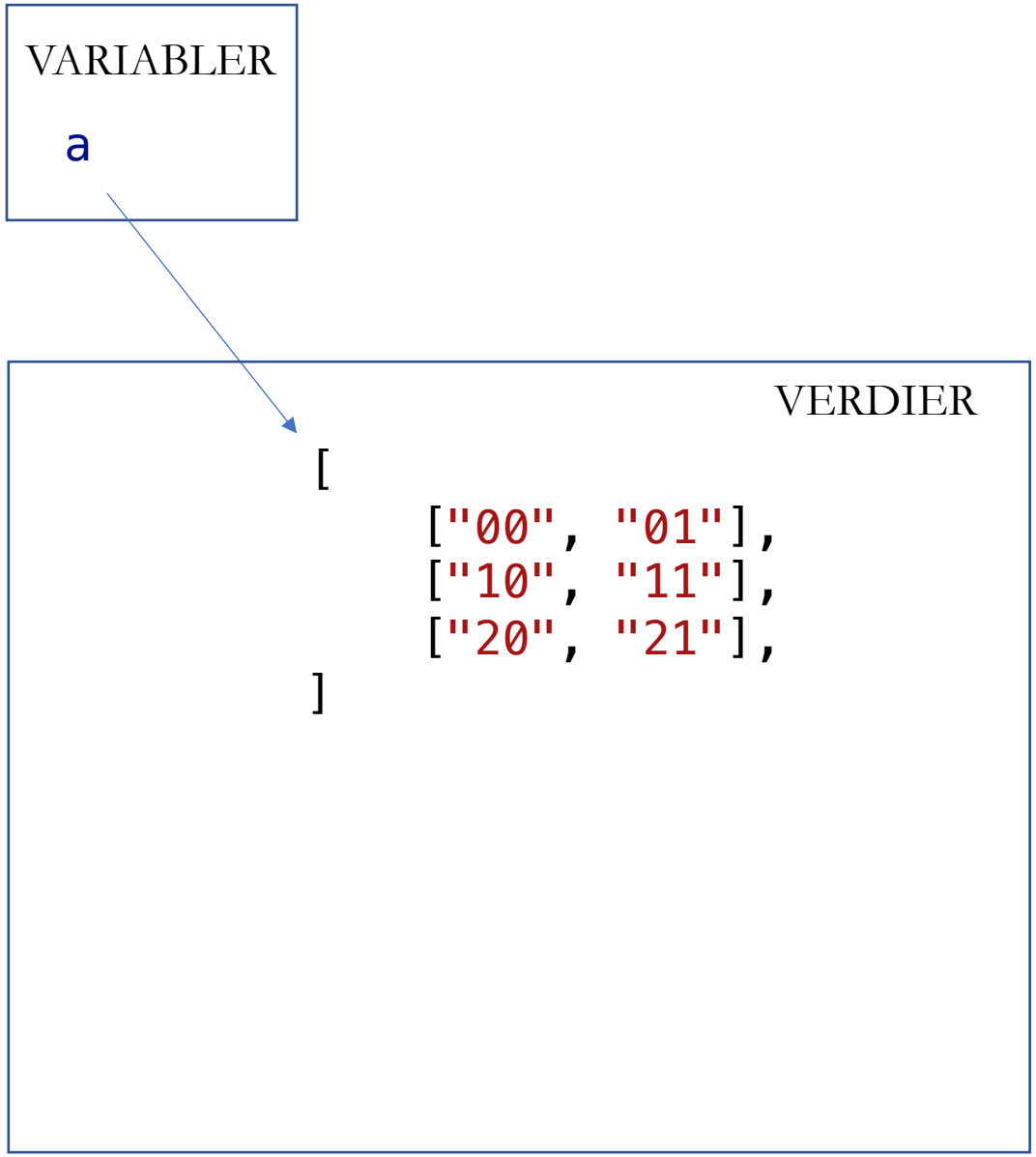
```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
b[0] = "foo"
print(a[1][0])
```



NYTT EKSEMPEL

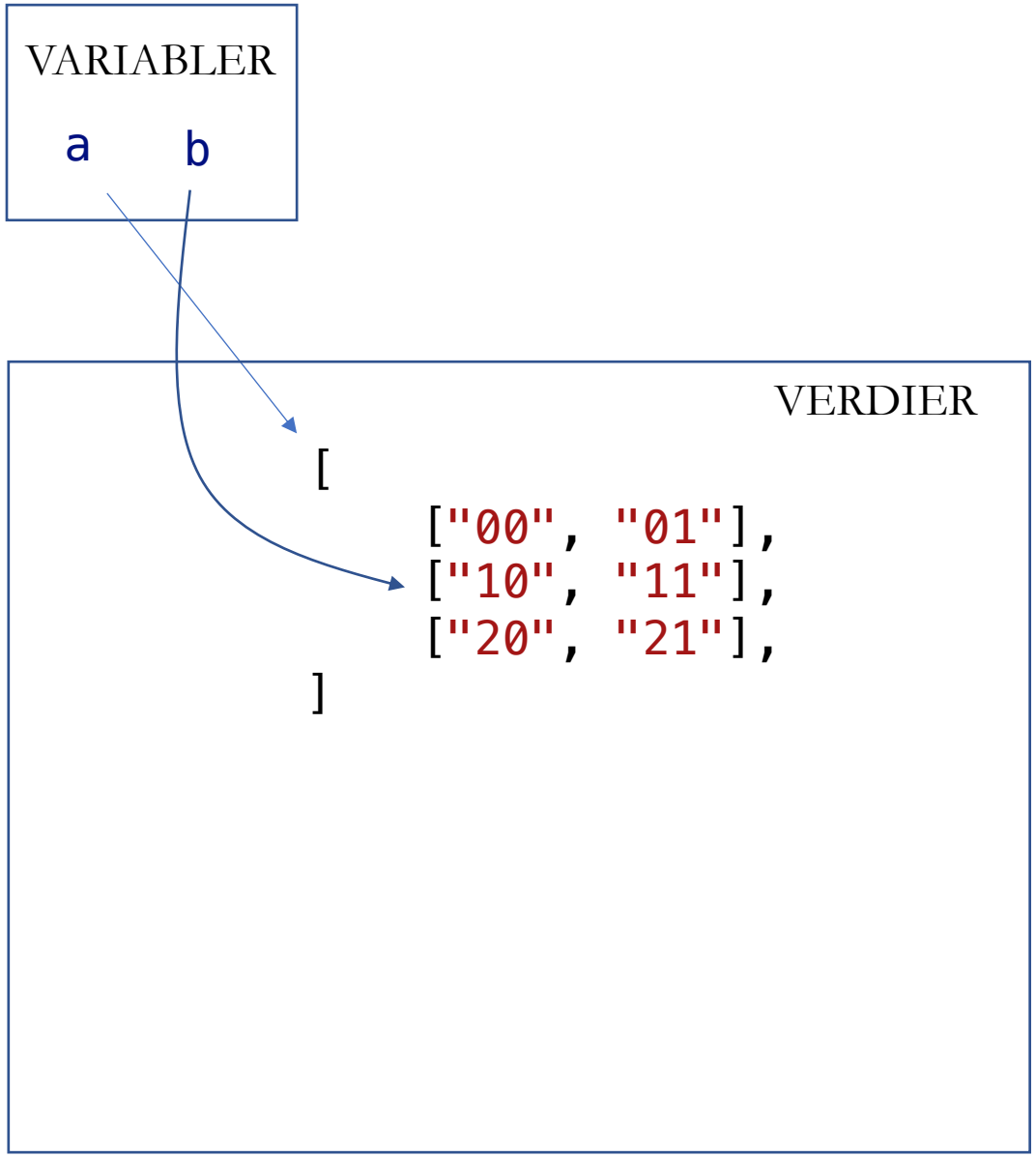
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
a[1] = ["HI", "HA"]
print(a[1][0])
print(b[0])
```



2D-LISTE

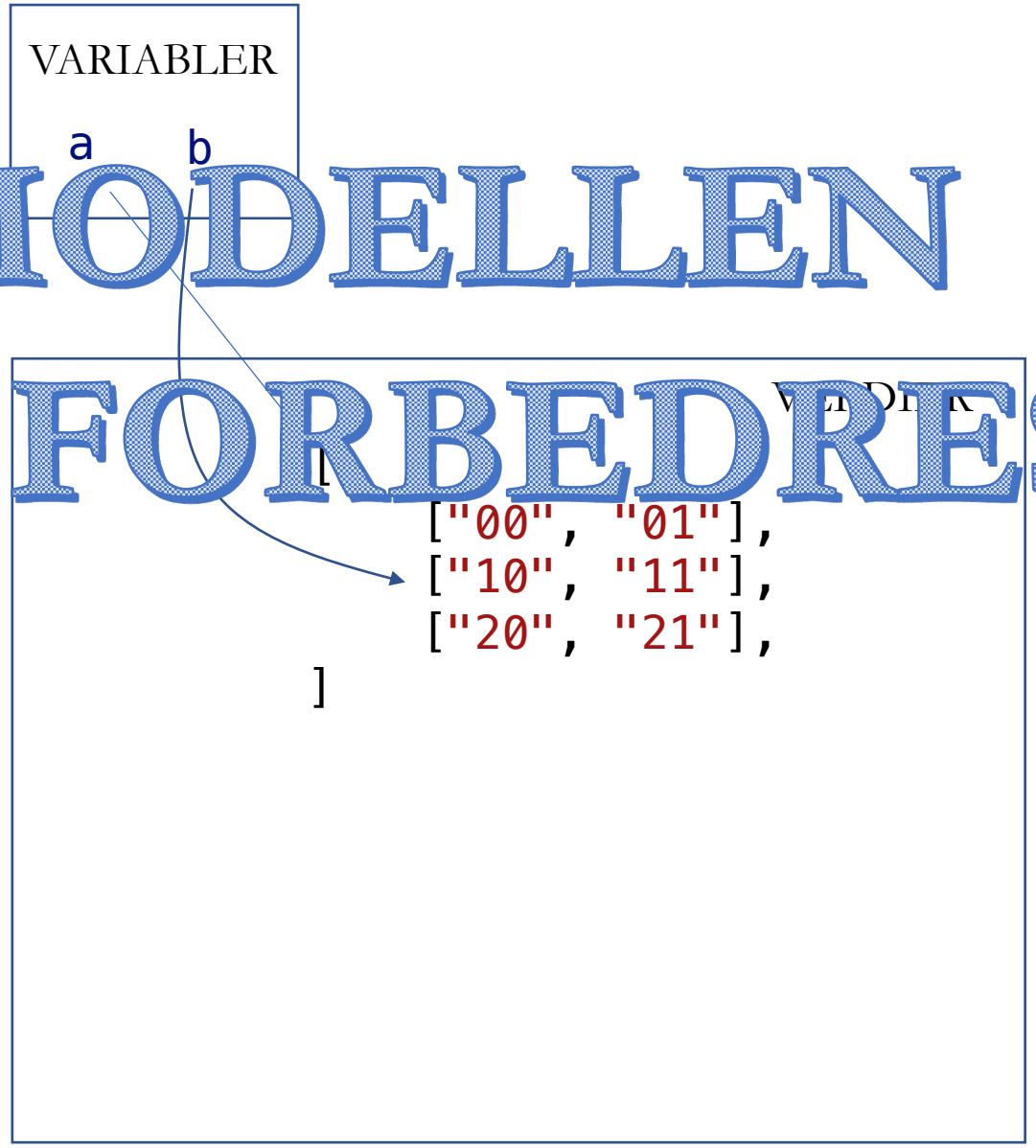
```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
a[1] = ["HI", "HA"]
print(a[1][0])
print(b[0])
```



2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
a[1] = ["HI", "HA"]
print(a[1][0])
print(b[0])
```

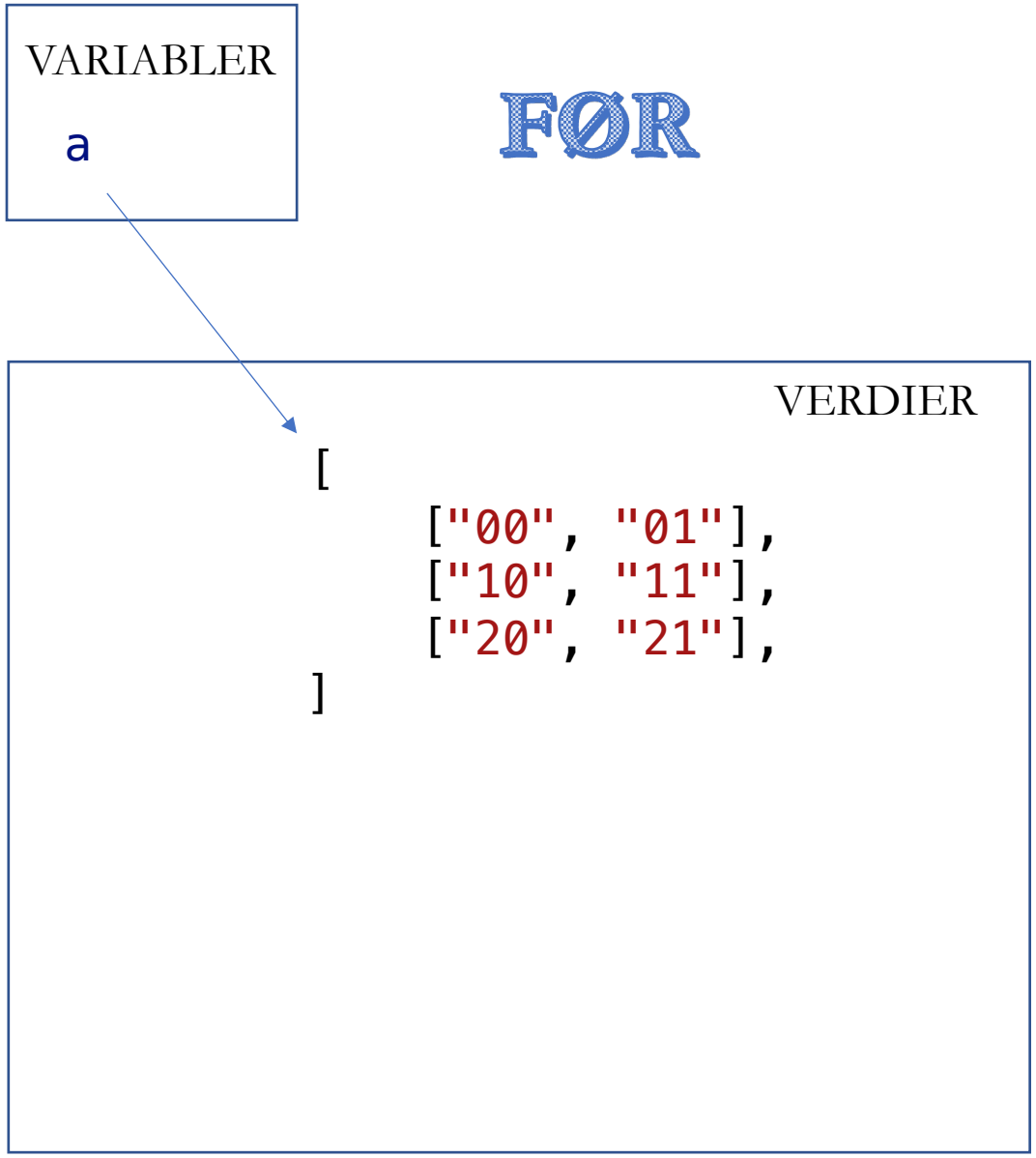
?



MODELLEN MÅ FORBEDRES!

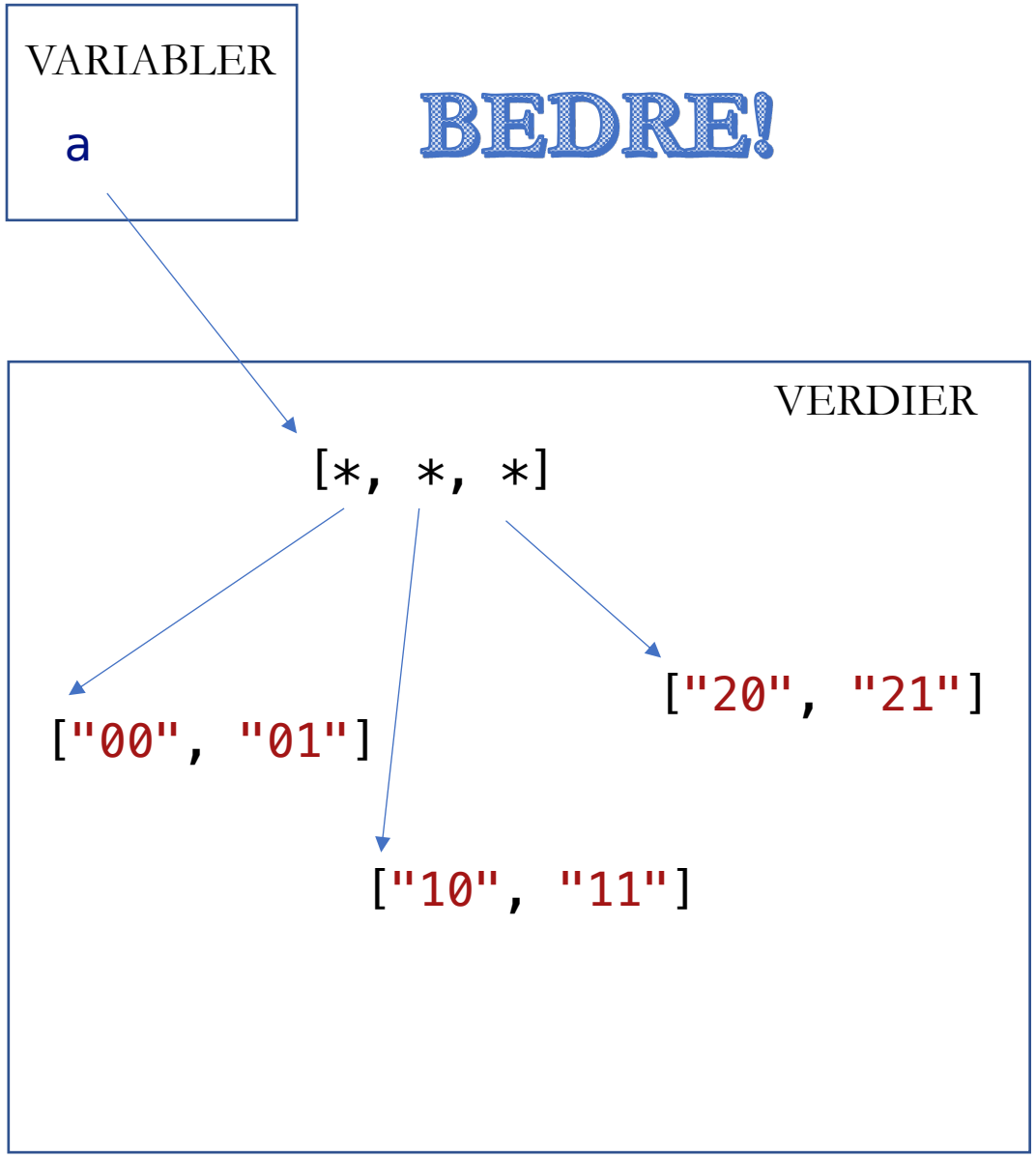
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]
```



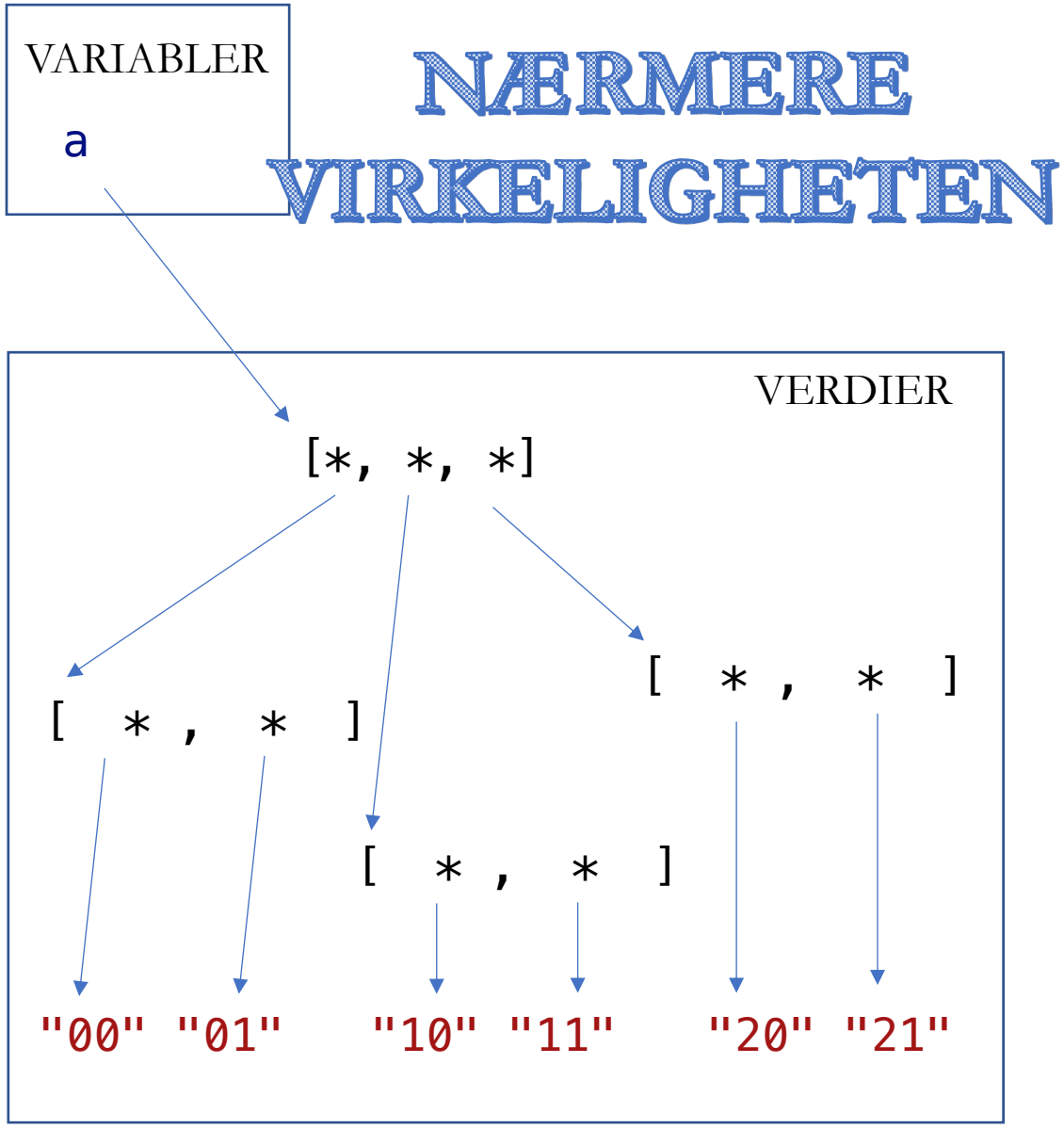
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]
```



2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]
```



2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]
```

ENDA NÆRMERE VIRKELIGHETEN

VARIABLER

Variabelnavn	Adresse
a	10240

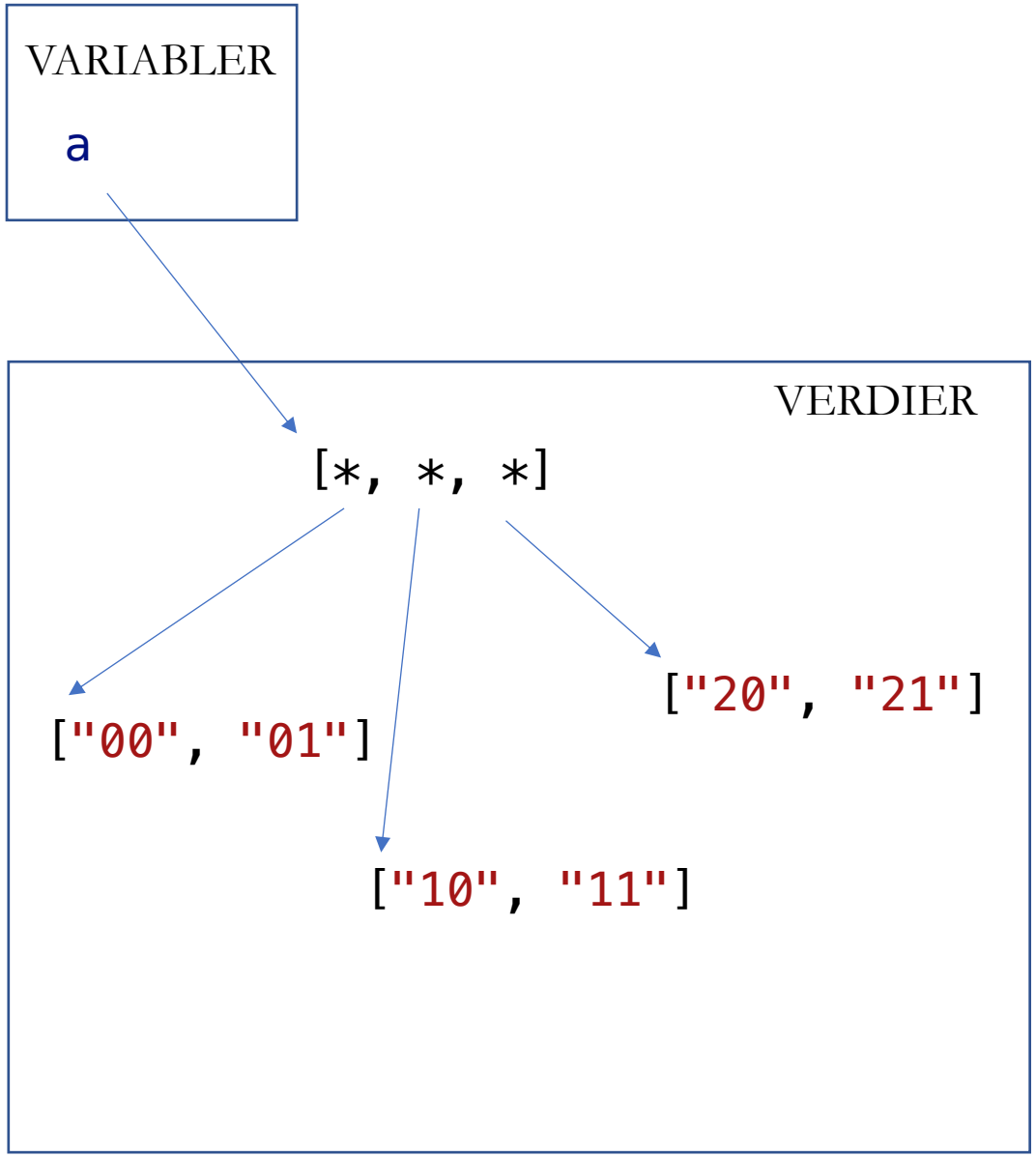
VERDIER

Adresse	Klasse	Verdi
1024	str	"00"
2048	str	"01"
3072	str	"10"
4096	str	"11"
5120	str	"20"
6144	str	"21"
7168	list	[1024, 2048]
8192	list	[3072, 4096]
9216	list	[5120, 6144]
10240	list	[7168, 8192, 9216]

NYTT FORSØK

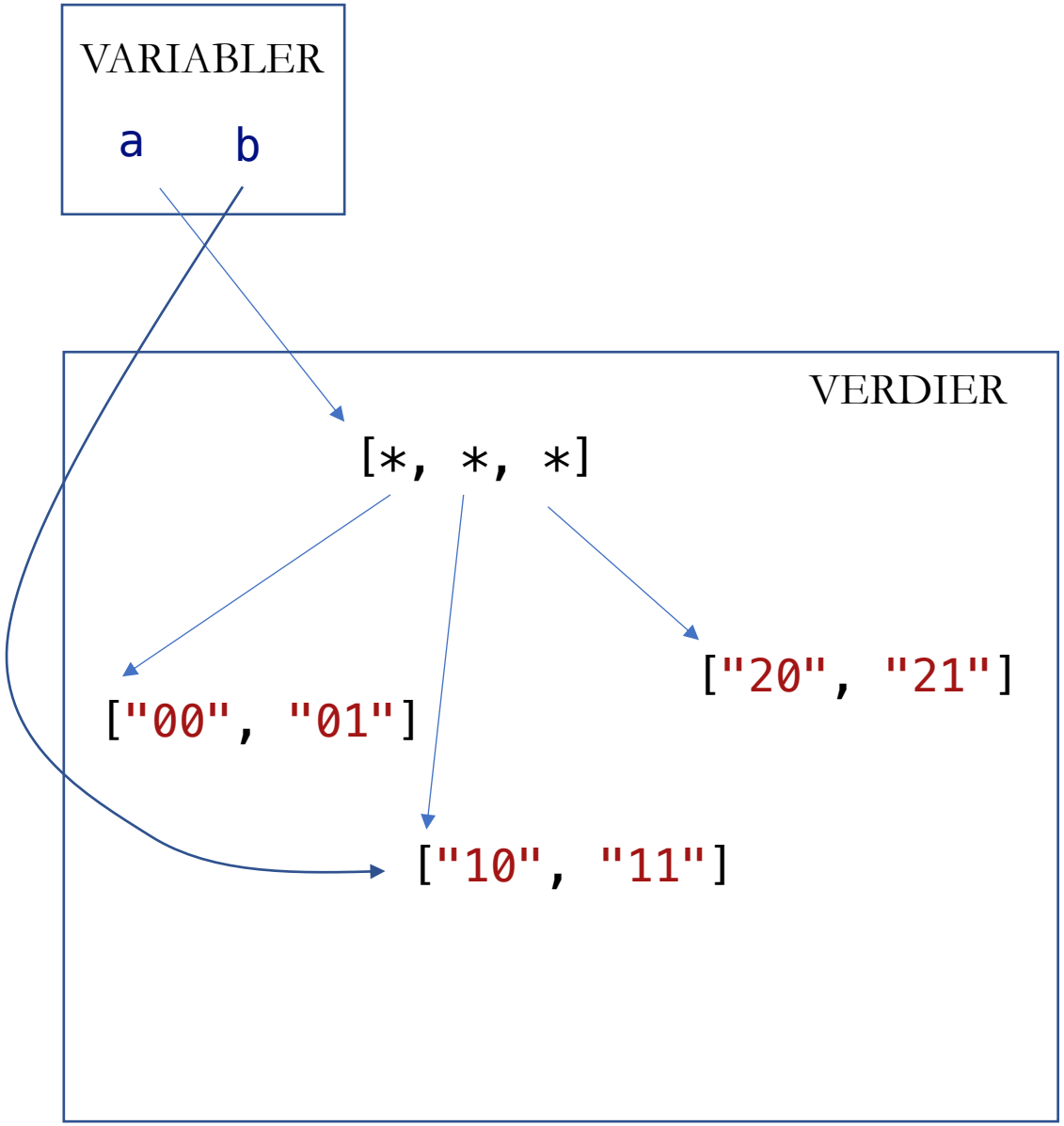
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
a[1] = ["HI", "HA"]
print(a[1][0])
print(b[0])
```



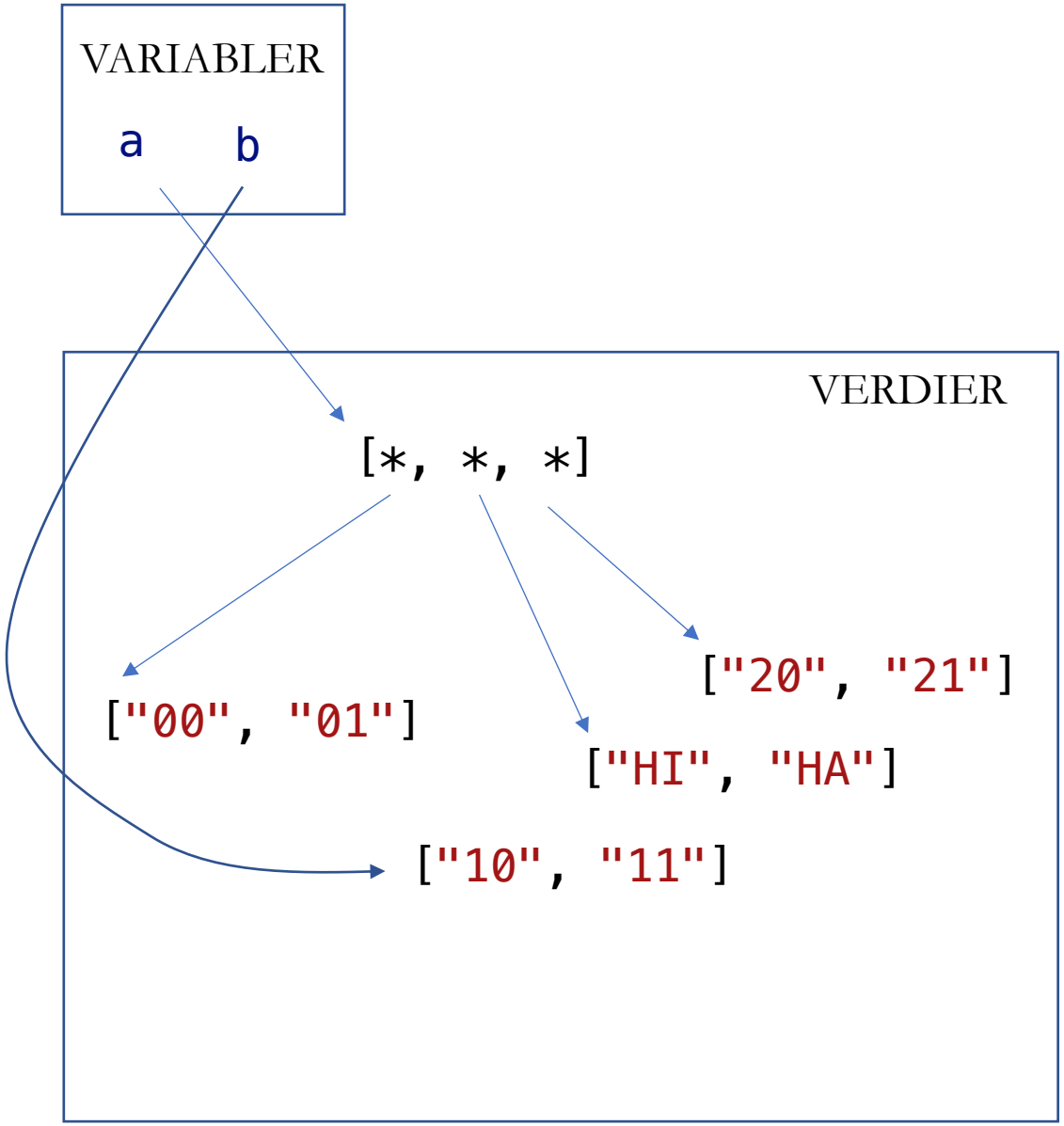
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
a[1] = ["HI", "HA"]
print(a[1][0])
print(b[0])
```



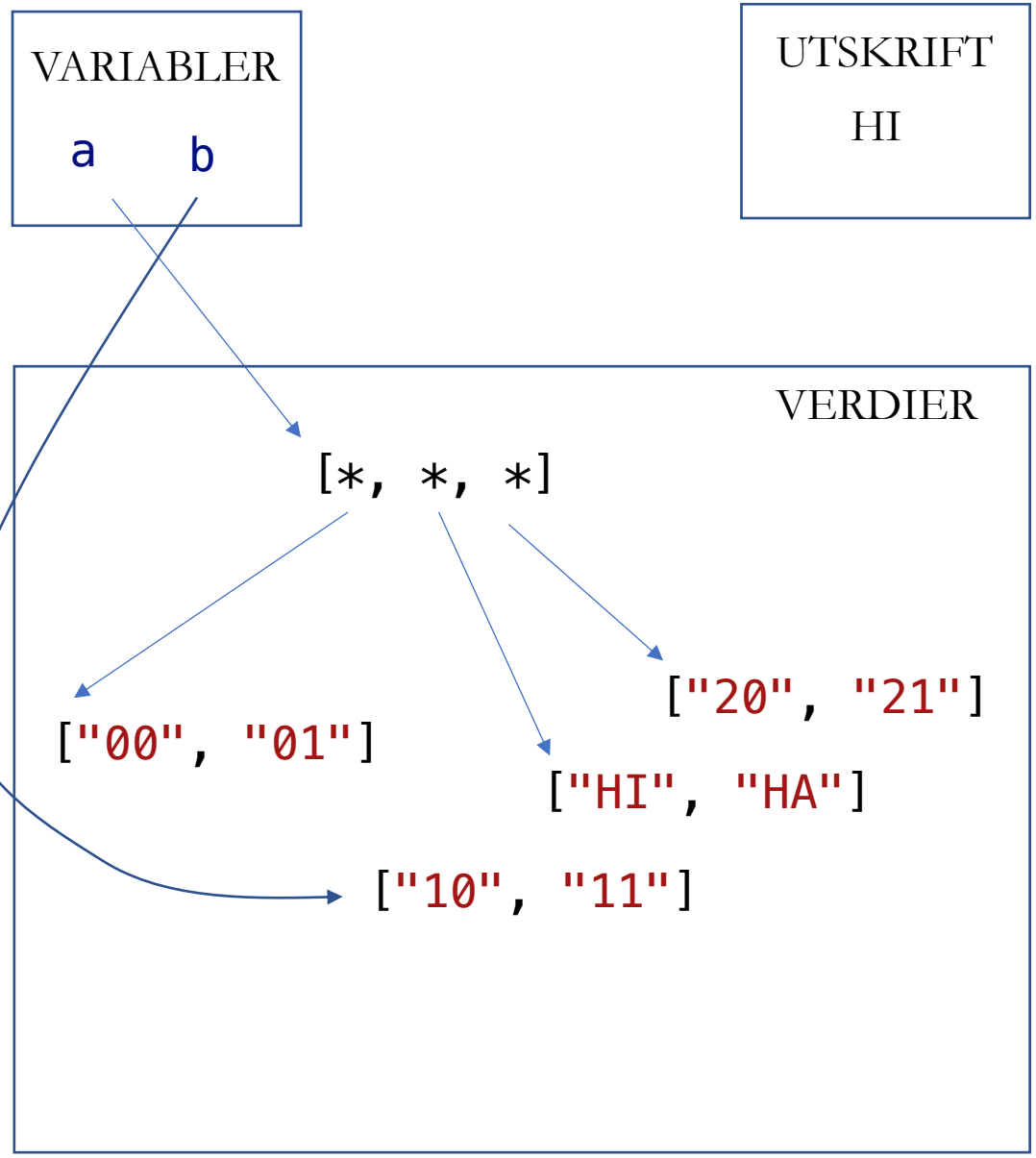
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
a[1] = ["HI", "HA"]
print(a[1][0])
print(b[0])
```



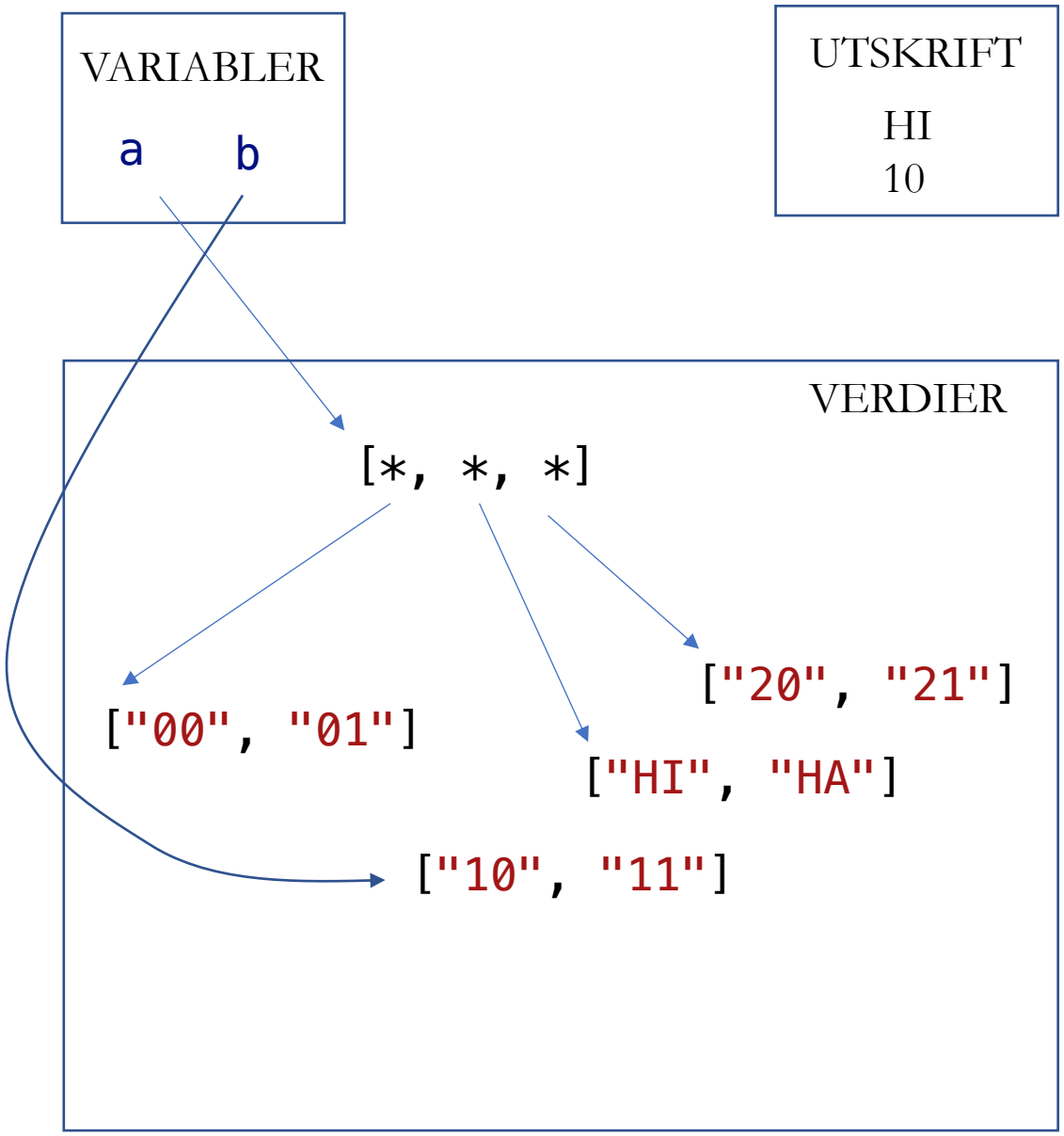
2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
a[1] = ["HI", "HA"]
print(a[1][0])
print(b[0])
```



2D-LISTE

```
a = [
        ["00", "01"],
        ["10", "11"],
        ["20", "21"],
    ]

b = a[1]
a[1] = ["HI", "HA"]
print(a[1][0])
print(b[0])
```



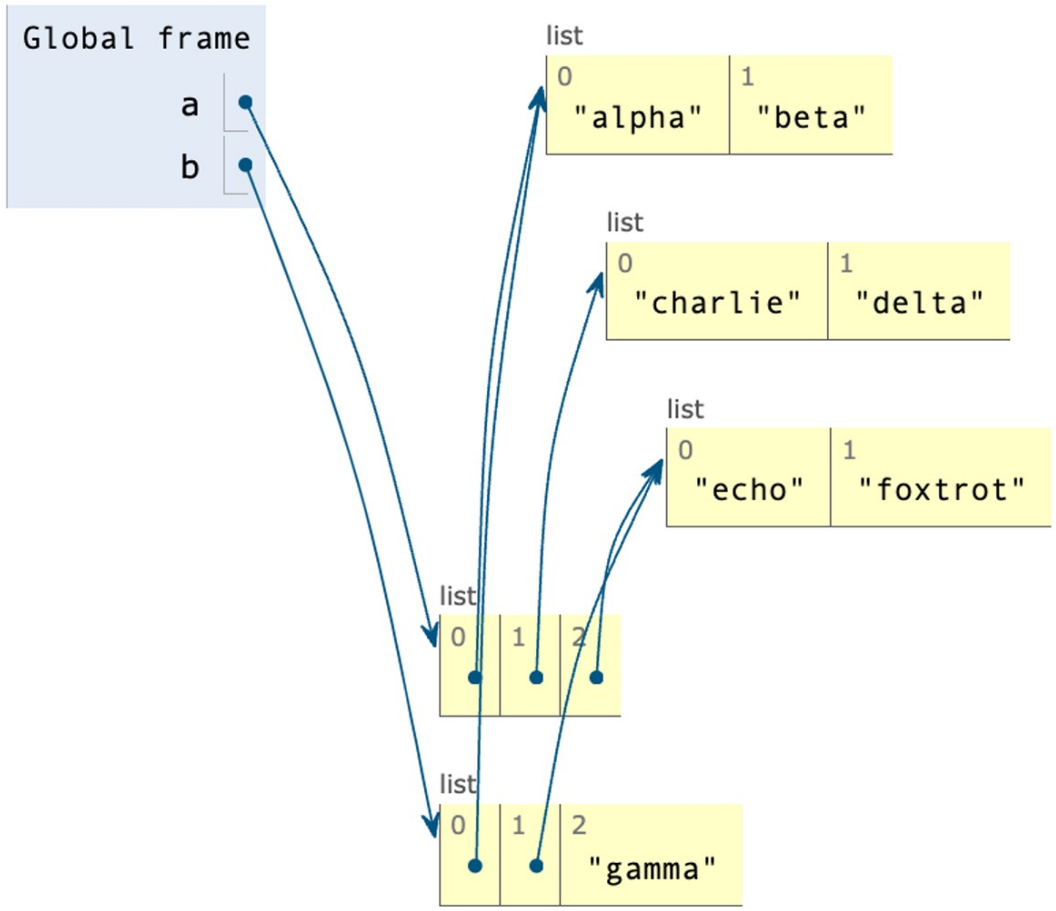
PYTHON TUTOR

- Gratis, reklamefinansiert visualiseringsverktøy

https://pythontutor.com/

EKSAMENSOPPGAVE HØST 2023

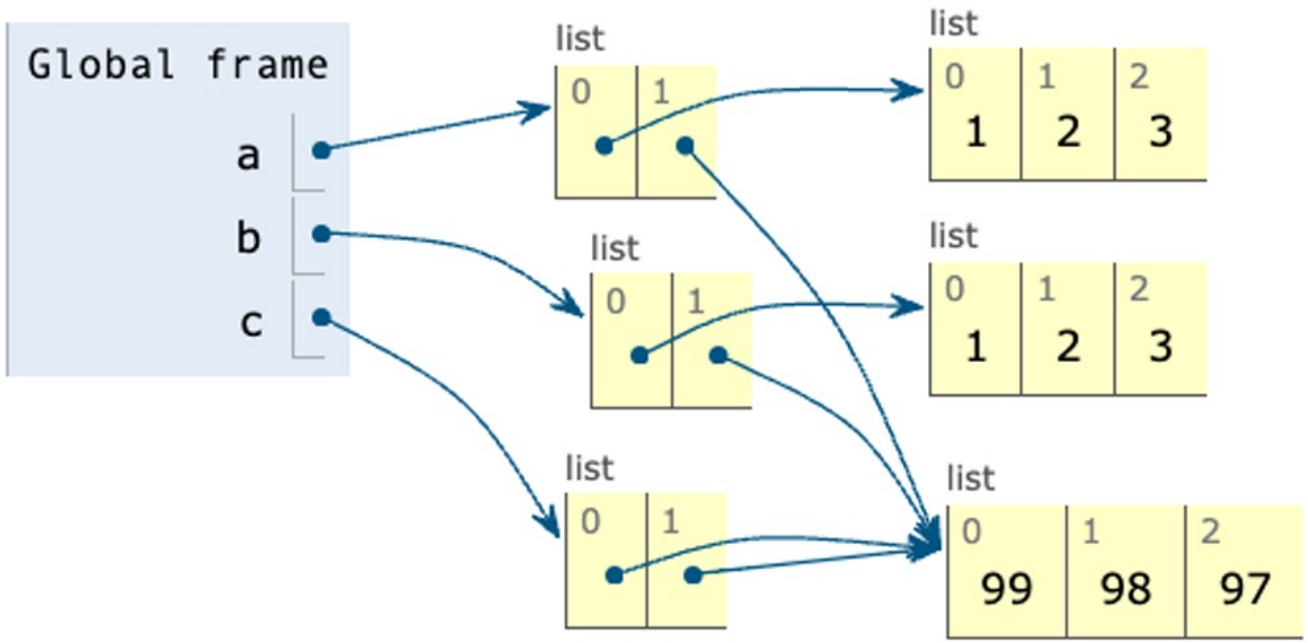
1(I)



Gitt at minnet har tilstanden vist over, hva blir skrevet ut etter setningen **print(a[1][1] + b[1][1])**?
(hvis programmet krasjer, skriv kun 'Error')

EKSAMENSOPPGAVE HØST 2023

3(a)



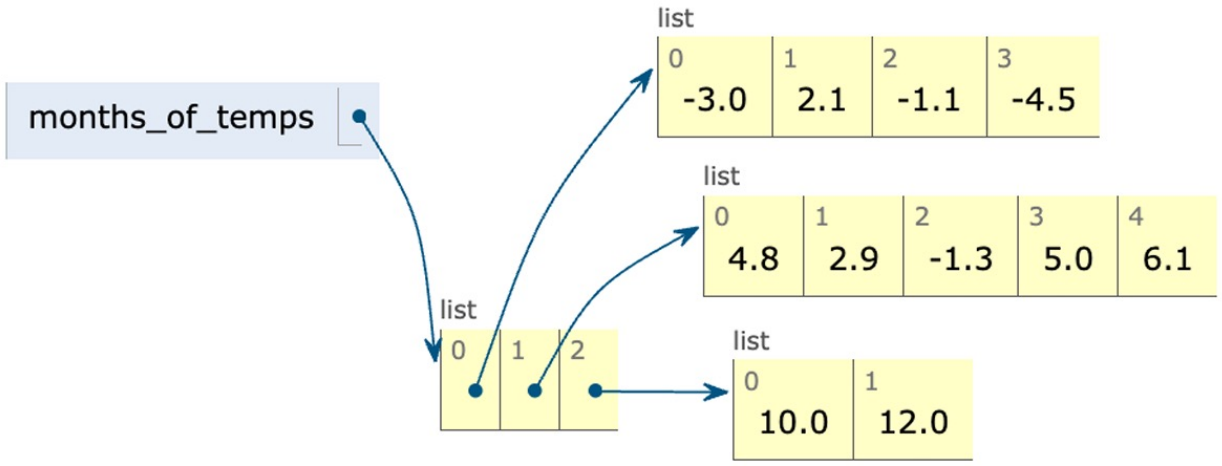
Skriv en kodesnutt slik at variabler og minnets tilstand for variablene a, b og c blir som vist over.

EKSAMENSOPPGAVE HØST 2023

3(b) Anta at **months_of_temps** er en variabel som peker på en to-dimensjonal liste av flyttall. De indre listene inneholder flyttall som representerer gjennomsnittstemperaturen for hver dag i en måned; mens den ytterste listen inneholder flere ulike måneder. Forskjellige måneder kan ha ulikt antall dager.

Skriv en funksjon **average_temp** med en parameter **months_of_temps** som beskrevet over. La funksjonen returnere gjennomsnittstemperaturen for alle dagene, uansett måned.

Eksempel på months_of_temps:



Hvis funksjonen du skriver kalles med eksempelet vist over som argument, skal returverdien bli 3.0: i eksempelet er det 11 ulike måleverdier (dager med temperaturmålinger), og summen av alle måleverdiene er 33.0.