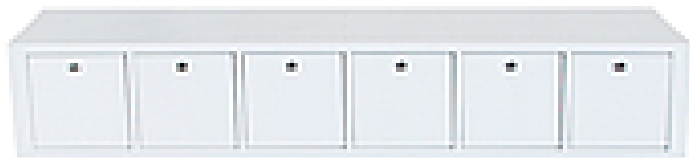


МАСИВИ

(вступ)



Для чого потрібні масиви?

Задача. Дано вік 2-х учнів. Знайти вік старшого учня.

Для чого потрібні масиви?

Задача. Дано вік 2-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age1 – вік 1-го учня

age2 – вік 2-го учня

Для чого потрібні масиви?

Задача. Дано вік 2-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age1 – вік 1-го учня

age2 – вік 2-го учня

```
const age1 = parseInt(prompt('Введіть вік 1-го учня'))
const age2 = parseInt(prompt('Введіть вік 2-го учня'))

let maxAge
if (age1 > age2) maxAge = age1
else maxAge = age2

document.write(`Max age = ${maxAge}`)
```

Для чого потрібні масиви?

Задача. Дано вік 3-х учнів. Знайти вік старшого учня.

Для чого потрібні масиви?

Задача. Дано вік 3-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age1 – вік 1-го учня

age2 – вік 2-го учня

age3 – вік 3-го учня

Для чого потрібні масиви?

Задача. Дано вік 3-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age1 – вік 1-го учня

age2 – вік 2-го учня

age3 – вік 3-го учня

```
const age1 = parseInt(prompt('Введіть вік 1-го учня'))
const age2 = parseInt(prompt('Введіть вік 2-го учня'))
const age3 = parseInt(prompt('Введіть вік 3-го учня'))

let maxAge = age1
if (age2 > maxAge) maxAge = age2
if (age3 > maxAge) maxAge = age3

document.write(`Max age = ${maxAge}`)
```

Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age1 – вік 1-го учня

age2 – вік 2-го учня

age3 – вік 3-го учня

age4 – вік 4-го учня

age5 – вік 5-го учня

age6 – вік 6-го учня

age7 – вік 7-го учня

Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age1 – вік 1-го учня

age2 – вік 2-го учня

age3 – вік 3-го учня

age4 – вік 4-го учня

age5 – вік 5-го учня

age6 – вік 6-го учня

age7 – вік 7-го учня

```
const age1 = parseInt(prompt('Введіть вік 1-го учня'))
const age2 = parseInt(prompt('Введіть вік 2-го учня'))
const age3 = parseInt(prompt('Введіть вік 3-го учня'))
const age4 = parseInt(prompt('Введіть вік 4-го учня'))
const age5 = parseInt(prompt('Введіть вік 5-го учня'))
const age6 = parseInt(prompt('Введіть вік 6-го учня'))
const age7 = parseInt(prompt('Введіть вік 7-го учня'))

let maxAge = age1
if (age2 > maxAge) maxAge = age2
if (age3 > maxAge) maxAge = age3
if (age4 > maxAge) maxAge = age4
if (age5 > maxAge) maxAge = age5
if (age6 > maxAge) maxAge = age6
if (age7 > maxAge) maxAge = age7

document.write(`Max age = ${maxAge}`)
```

Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

```
const age1 = parseInt(prompt('Введіть вік 1-го учня'))
const age2 = parseInt(prompt('Введіть вік 2-го учня'))
const age3 = parseInt(prompt('Введіть вік 3-го учня'))
const age4 = parseInt(prompt('Введіть вік 4-го учня'))
const age5 = parseInt(prompt('Введіть вік 5-го учня'))
const age6 = parseInt(prompt('Введіть вік 6-го учня'))
const age7 = parseInt(prompt('Введіть вік 7-го учня'))

let maxAge = age1
if (age2 > maxAge) maxAge = age2
if (age3 > maxAge) maxAge = age3
if (age4 > maxAge) maxAge = age4
if (age5 > maxAge) maxAge = age5
if (age6 > maxAge) maxAge = age6
if (age7 > maxAge) maxAge = age7

document.write(`Max age = ${maxAge}`)
```

Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

```
const age1 = parseInt(prompt('Введіть вік 1-го учня'))
const age2 = parseInt(prompt('Введіть вік 2-го учня'))
const age3 = parseInt(prompt('Введіть вік 3-го учня'))
const age4 = parseInt(prompt('Введіть вік 4-го учня'))
const age5 = parseInt(prompt('Введіть вік 5-го учня'))
const age6 = parseInt(prompt('Введіть вік 6-го учня'))
const age7 = parseInt(prompt('Введіть вік 7-го учня'))

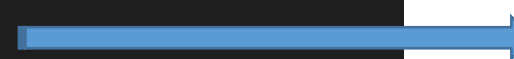
let maxAge = age1
if (age2 > maxAge) maxAge = age2
if (age3 > maxAge) maxAge = age3
if (age4 > maxAge) maxAge = age4
if (age5 > maxAge) maxAge = age5
if (age6 > maxAge) maxAge = age6
if (age7 > maxAge) maxAge = age7

document.write(`Max age = ${maxAge}`)
```



```
for (let i = 1; i <= 7; i++) { ...
  const age i = parseInt(prompt(`Введіть вік ${i}-го учня`))
}
```

Red arrow points to `age i` with a red question mark.



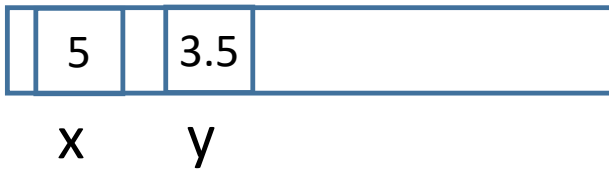
```
let maxAge = age1
for (let i = 1; i <= 7; i++) { ...
  if (age i > maxAge) maxAge = age i
}
```

Red arrows point to `age i` and `maxAge = age i` with red question marks.

Для чого потрібні масиви?

```
let x = 5  
let y = 3.5
```

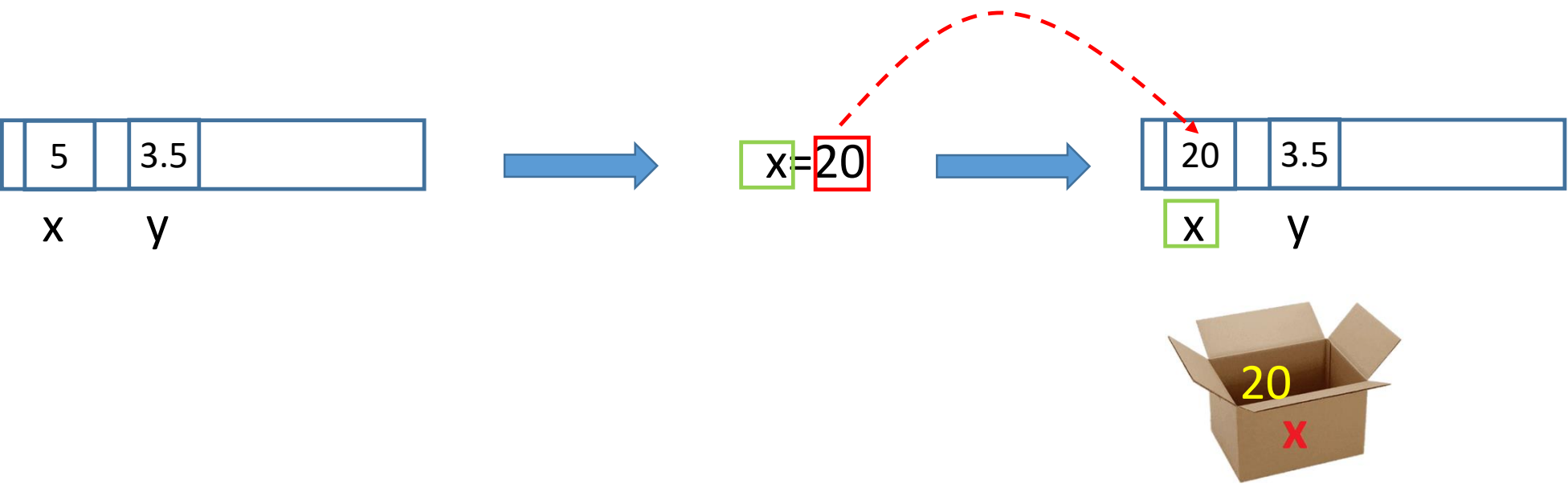
У змінній зберігаємо тільки одне значення



Для чого потрібні масиви?

```
let x = 5
let y = 3.5
```

Якщо присвоїти нове значення, то попереднє втрачається, і маємо тільки нове



Масив – нумерована послідовність елементів даних

Опис масиву

```
let x = 5  
let y = 3.5  
  
let a = [11, 4, 78]
```

(потрібно у квадратних дужках через кому перерахувати елементи масиву)

-> нумерація елементів починається з 0

let *назва_масиву* = [елемент_0, елемент_1, ..., елемент_N]

```
let a = [11, 4, 78]
```

0 1 2

індекси елементів

11 4 78

a

```
let a = [11, 4, 78]
```

5

3.5

x

y

Масив – нумерована послідовність елементів даних

Опис масиву

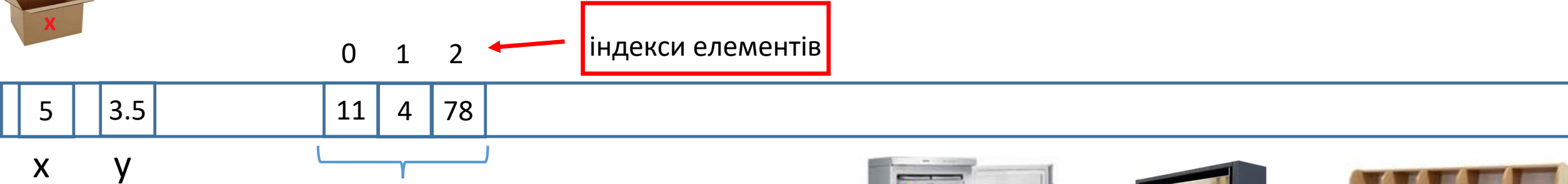
```
let x = 5
let y = 3.5

let a = [11, 4, 78]
```

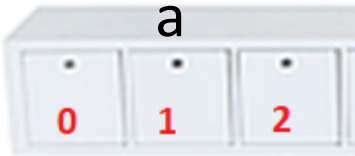
(потрібно у квадратних дужках через кому перерахувати елементи масиву)
-> нумерація елементів починається з 0

let назва_масиву = [елемент_0, елемент_1, ..., елемент_N]

```
let a = [11, 4, 78]
```



```
let a = [11, 4, 78]
```



Масив – нумерована послідовність елементів даних

```
let x = 5  
let y = 3.5  
  
let a = [11, 4, 78]
```

Довжина масиву
(кількість елементів)

назва_масиву . length

(насправді, номер останнього + 1)

a.length ← 3

0 1 2

індекси елементів



x

y

a

```
let a = [11, 4, 78]
```

Масив – нумерована послідовність елементів даних

Опис масиву

```
let x = 5
let y = 3.5

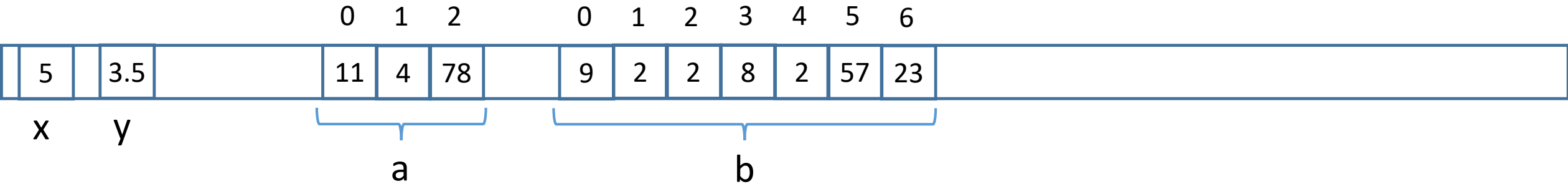
let a = [11, 4, 78]

let b = [9, 2, 2, 8, 2, 57, 23]
```

(потрібно у квадратних дужках через кому перерахувати елементи масиву)
-> нумерація елементів починається з 0
-> елементи можуть повторюватись

let назва_масиву = [елемент_0, елемент_1, ..., елемент_N]

```
let a = [11, 4, 78]
```



Масив – нумерована послідовність елементів даних

Опис масиву

```
let x = 5  
let y = 3.5  
  
let a = [11, 4, 78]  
  
let b = [9, 2, 2, 8, 2, 57, 23]
```

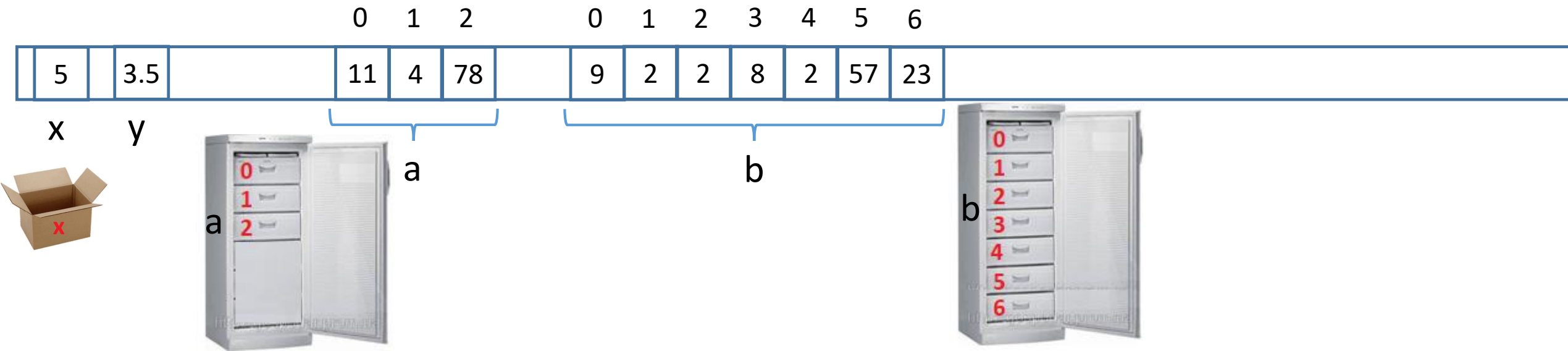
(потрібно у квадратних дужках через кому перерахувати елементи масиву)

-> нумерація елементів починається з 0

-> елементи можуть повторюватись

let *назва_масиву* = [елемент_0, елемент_1, ..., елемент_N]

let a = [11, 4, 78]



Масив – нумерована послідовність елементів даних

Опис масиву

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

let b = [9, 2, 2, 8, 2, 57, 23]

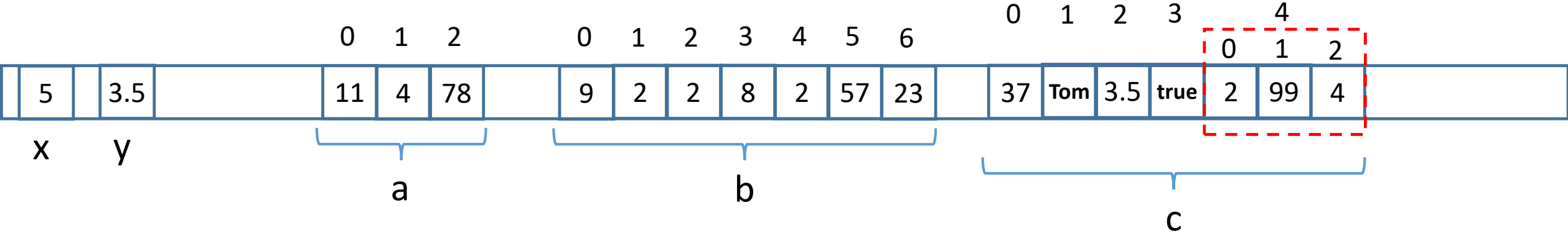
let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

(потрібно у квадратних дужках через кому перерахувати елементи масиву)

- > нумерація елементів починається з 0
- > елементи можуть повторюватись
- > елементи довільного типу

let назва_масиву = [елемент_0, елемент_1, ..., елемент_N]

```
let a = [11, 4, 78]
```



Масив – нумерована послідовність елементів даних

Опис масиву

(потрібно у квадратних дужках через кому перерахувати елементи масиву)

- > нумерація елементів починається з 0
- > елементи можуть повторюватись
- > елементи довільного типу

```
let x = 5
let y = 3.5

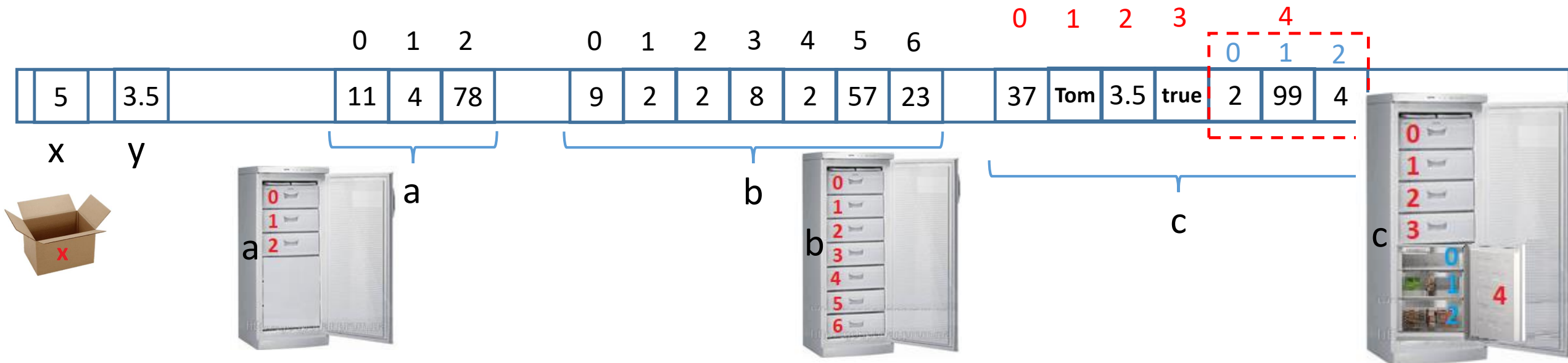
let a = [11, 4, 78]

let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

let назва_масиву = [елемент_0, елемент_1, ..., елемент_N]

let a = [11, 4, 78]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

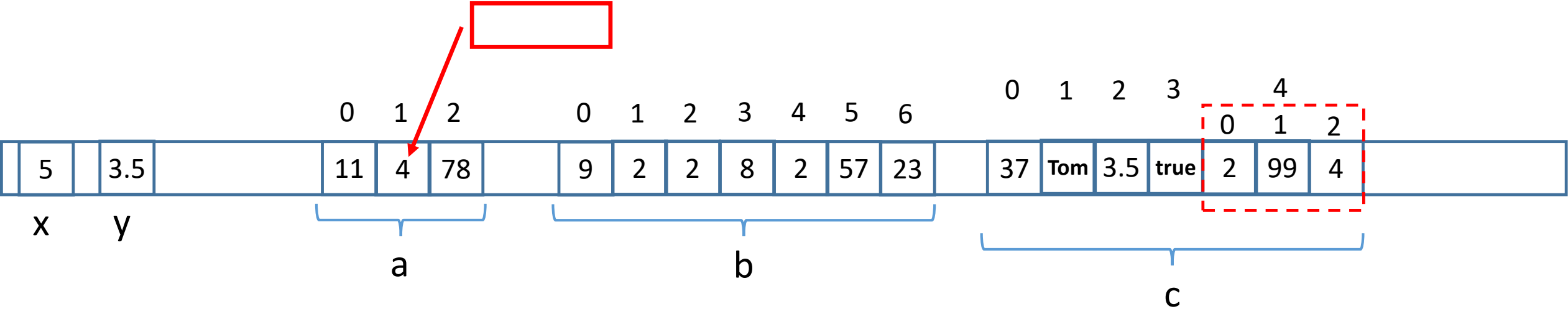
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента у квадратних дужках

назва_масиву [індекс_елемента]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

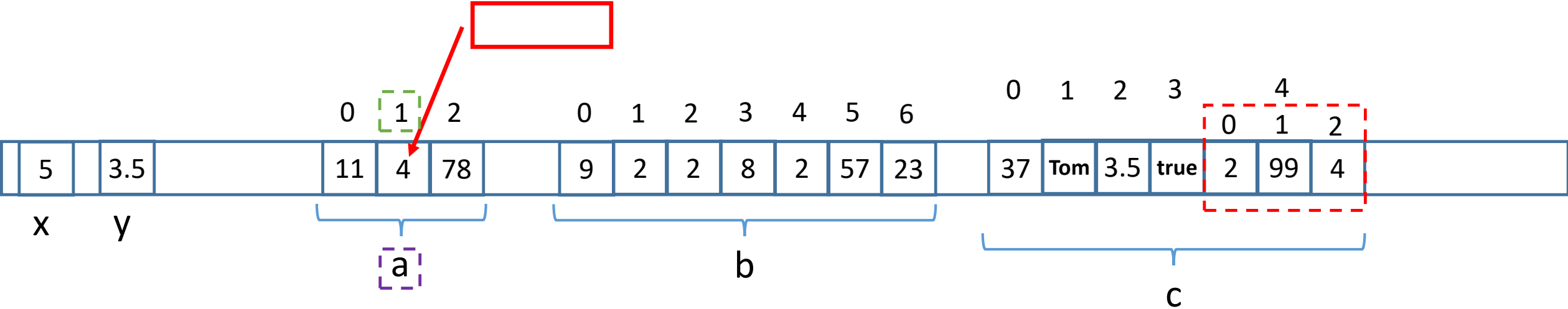
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента у квадратних дужках

назва_масиву [індекс_елемента]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

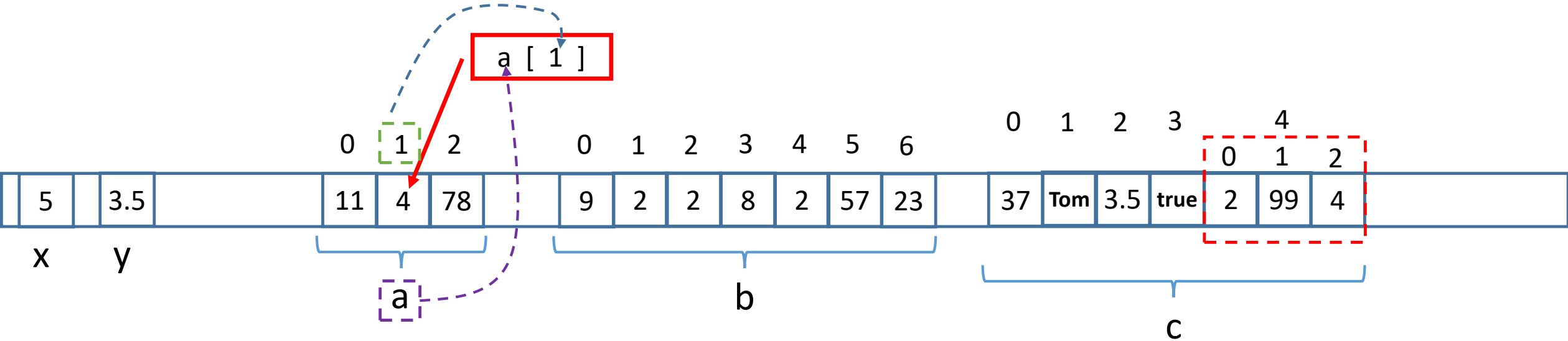
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента у квадратних дужках

назва_масиву [індекс_елемента]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

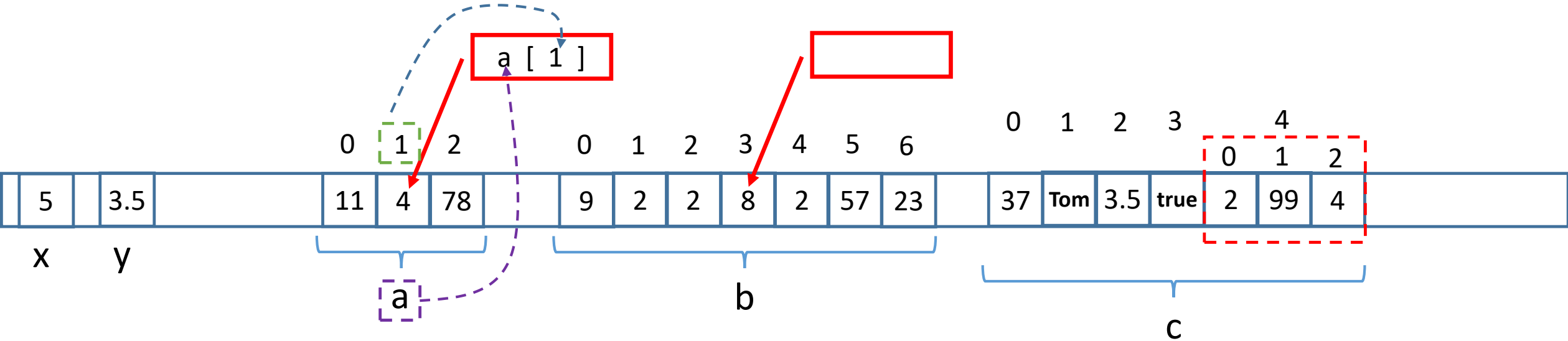
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [індекс_елемента]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

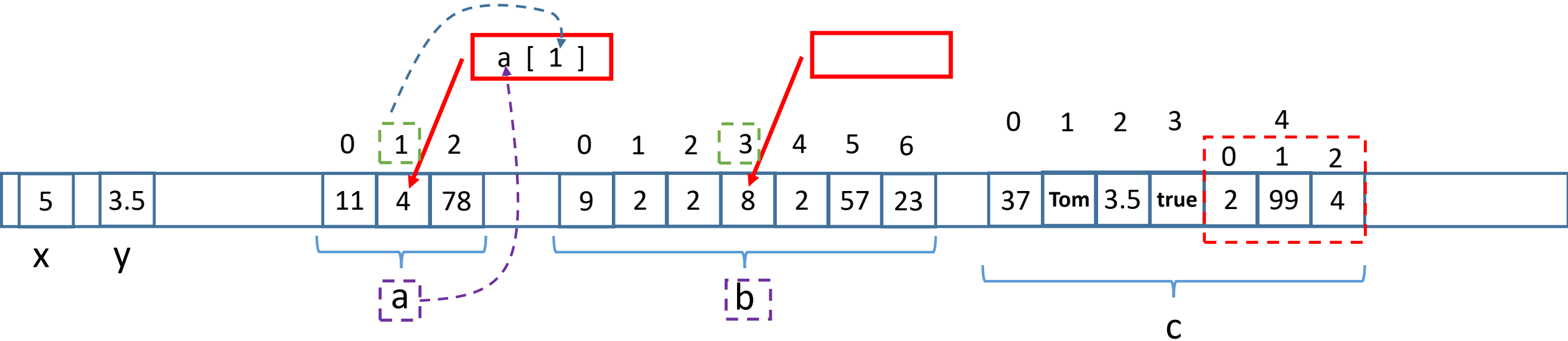
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [індекс_елемента]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

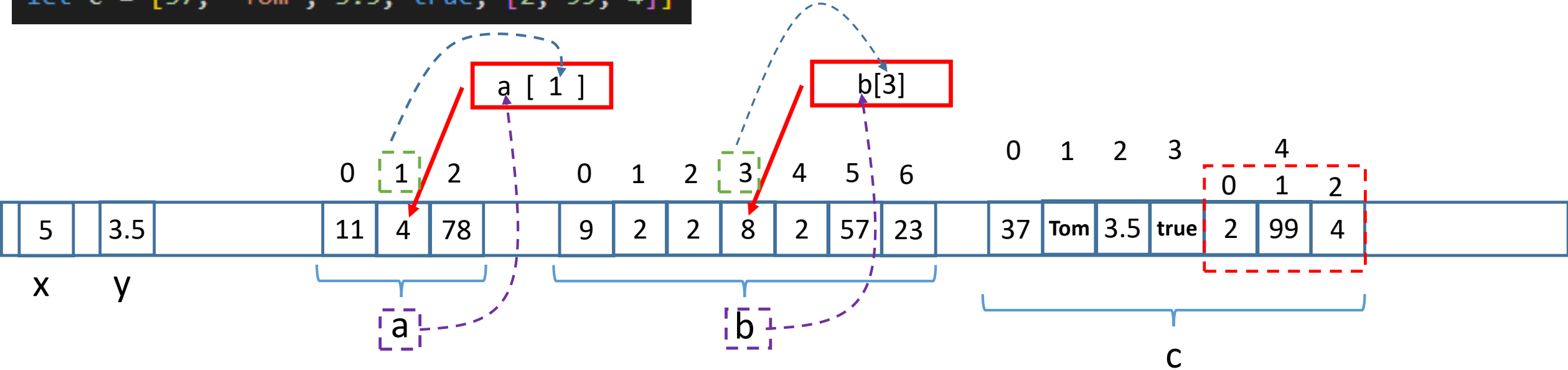
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [індекс_елемента]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

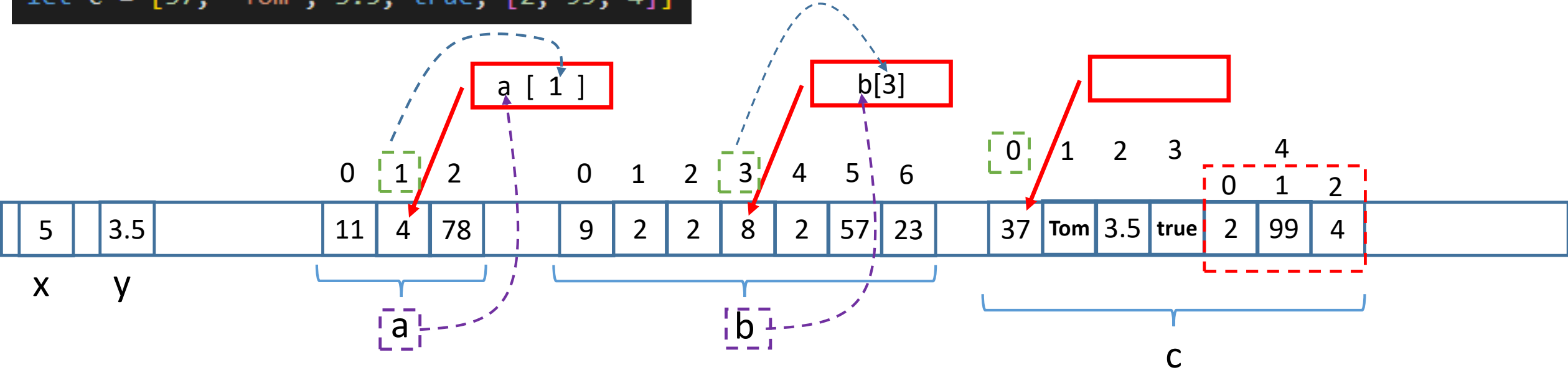
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [індекс_елемента]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

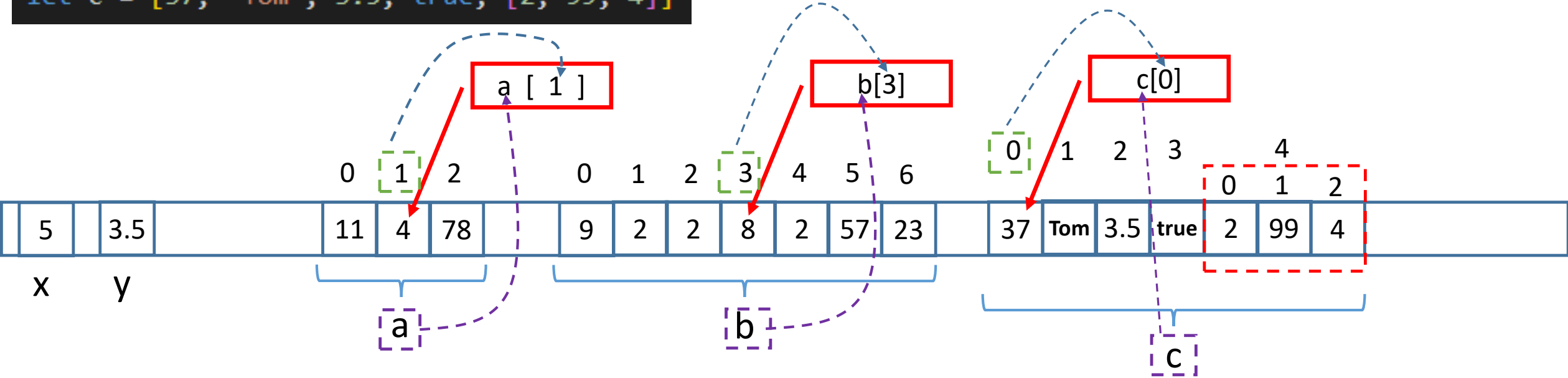
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [індекс_елемента]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

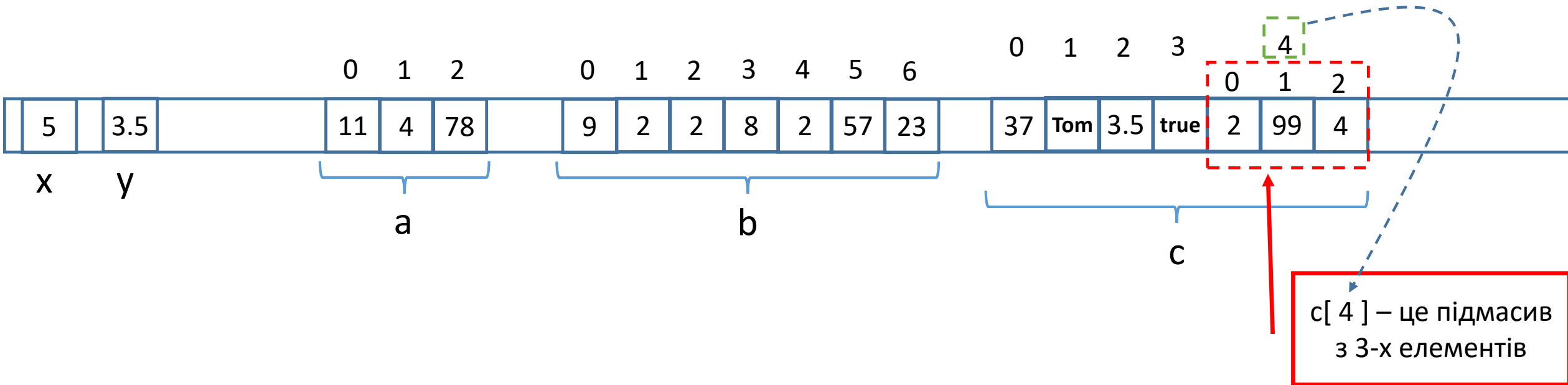
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [індекс_елемента]

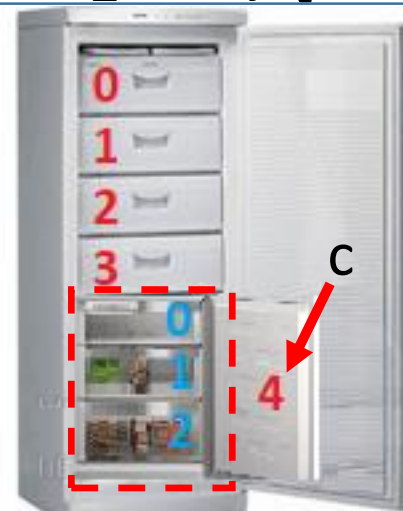


Звертання до елементів масиву

```
let x = 5  
let y = 3.5  
  
let a = [11, 4, 78]  
  
let b = [9, 2, 2, 8, 2, 57, 23]  
  
let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [*індекс_елемента*]



0 1 2 3

4

0 1 2

0 1 2

0 1 2 3 4 5 6

11 4 78

9 2 2 8 2 57 23

37 Tom 3.5 true

2 99 4

5 3.5

x

y

a

b

c

c[4] – це підмасив
з 3-х елементів

Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

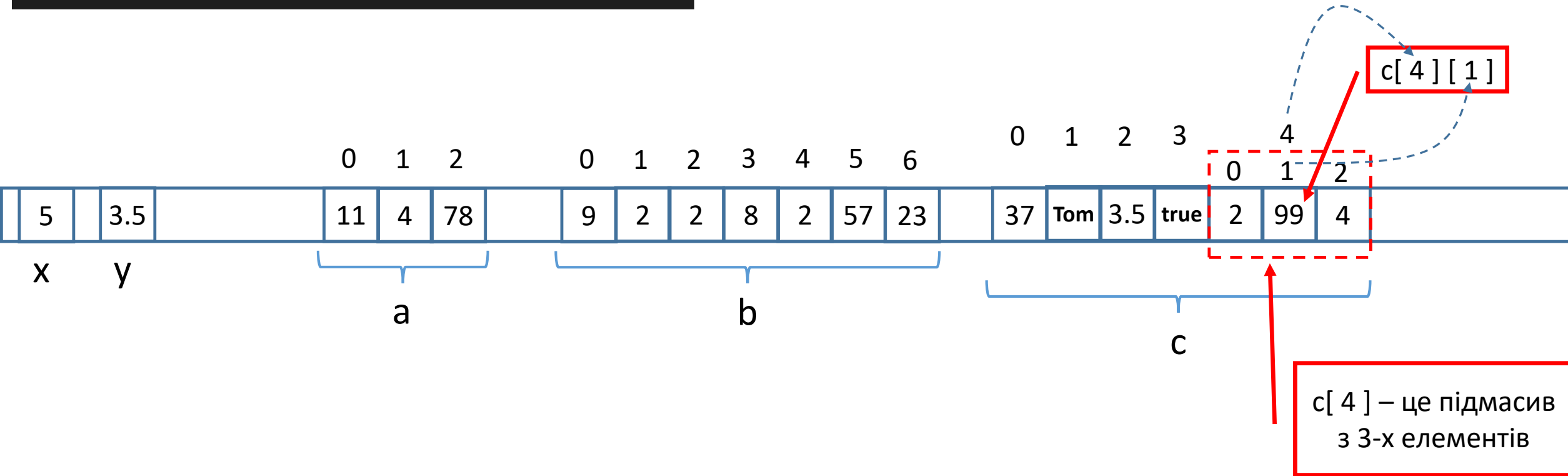
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [індекс_елемента]



Звертання до елементів масиву

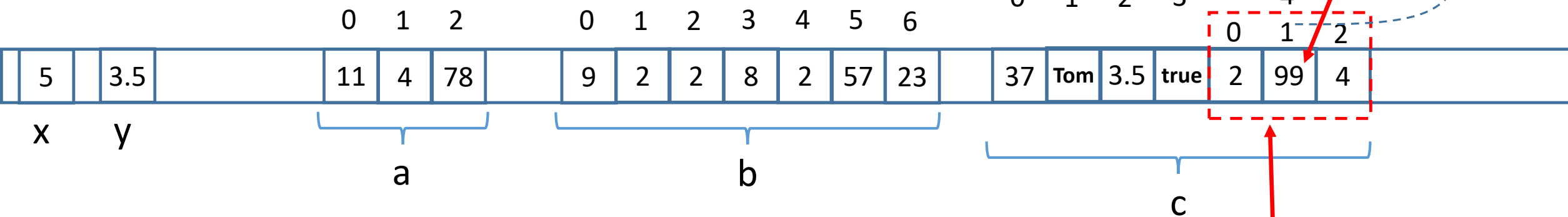
```
let x = 5  
let y = 3.5  
  
let a = [11, 4, 78]  
  
let b = [9, 2, 2, 8, 2, 57, 23]  
  
let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [індекс_елемента]



c[4][1]



c[4] – це підмасив з 3-х елементів

Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

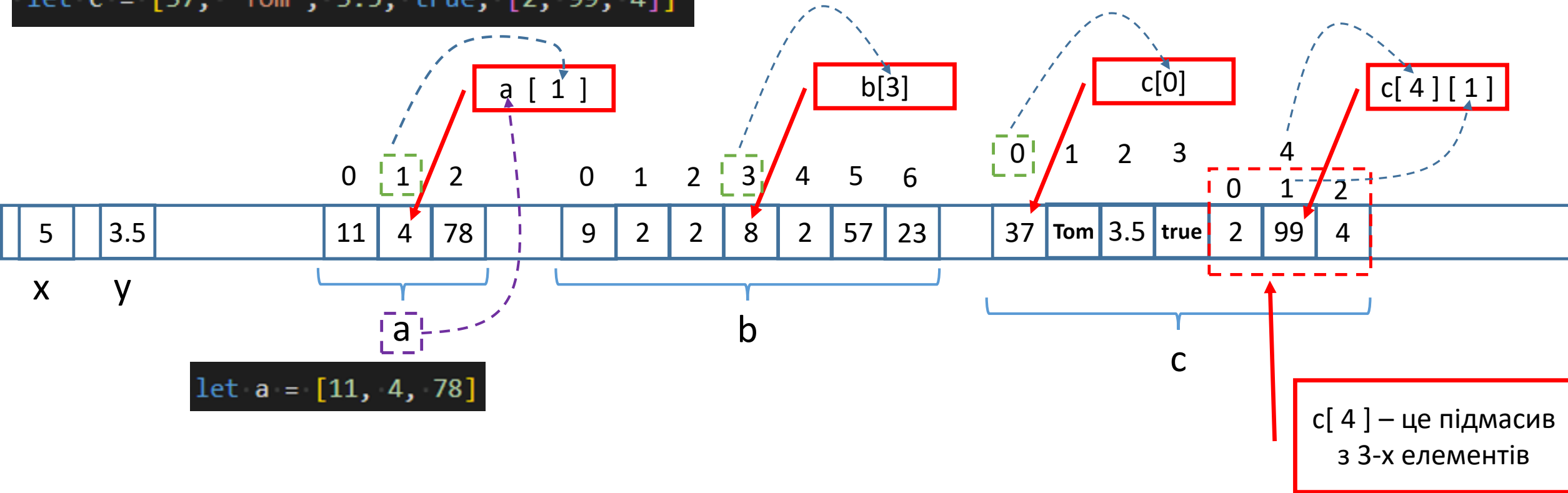
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента

назва_масиву [індекс_елемента]



Масив – нумерована послідовність елементів даних

```
let x = 5
let y = 3.5

let a = [11, 4, 78]

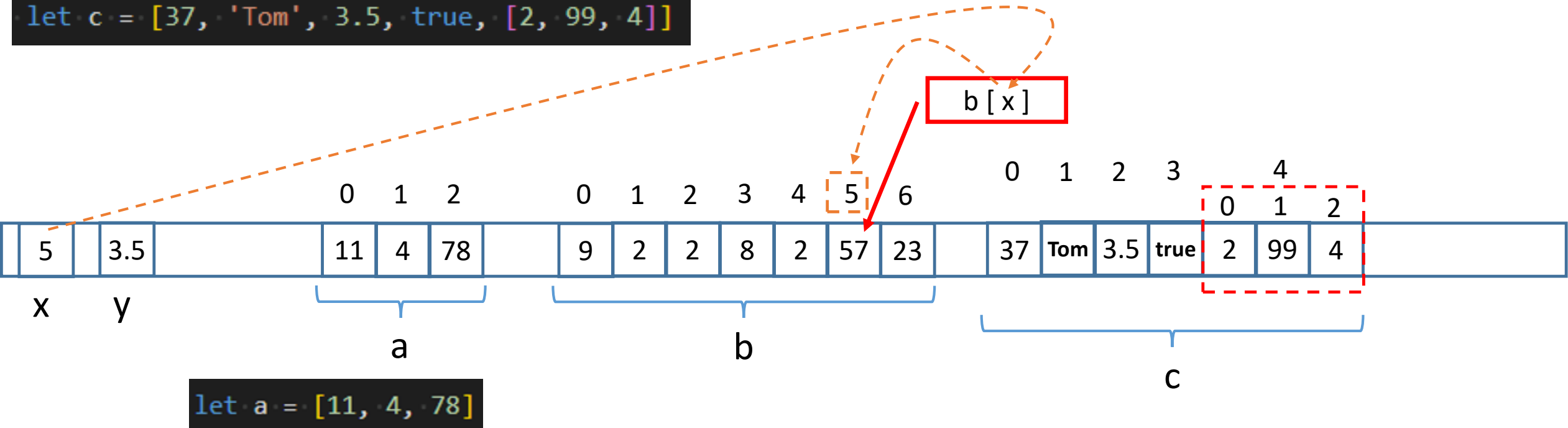
let b = [9, 2, 2, 8, 2, 57, 23]

let c = [37, 'Tom', 3.5, true, [2, 99, 4]]
```

Звертання до елементів масиву

потрібно вказати назву масиву і номер (індекс) потрібного елемента
(індекс може зберігатися у деякій змінній)

назва_масиву [змінна_що_містить_індекс_елемента]



Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age1 – вік 1-го учня

age2 – вік 2-го учня

age3 – вік 3-го учня

age4 – вік 4-го учня

age5 – вік 5-го учня

age6 – вік 6-го учня

age7 – вік 7-го учня

```
const age1 = parseInt(prompt('Введіть вік 1-го учня'))!  
const age2 = parseInt(prompt('Введіть вік 2-го учня'))!  
const age3 = parseInt(prompt('Введіть вік 3-го учня'))!  
const age4 = parseInt(prompt('Введіть вік 4-го учня'))!  
const age5 = parseInt(prompt('Введіть вік 5-го учня'))!  
const age6 = parseInt(prompt('Введіть вік 6-го учня'))!  
const age7 = parseInt(prompt('Введіть вік 7-го учня'))!
```

```
let maxAge = age1  
if (age2 > maxAge) maxAge = age2  
if (age3 > maxAge) maxAge = age3  
if (age4 > maxAge) maxAge = age4  
if (age5 > maxAge) maxAge = age5  
if (age6 > maxAge) maxAge = age6  
if (age7 > maxAge) maxAge = age7 .....
```

```
document.write(`Max age = ${maxAge}`)
```

Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age1 – вік 1-го учня
age2 – вік 2-го учня
age3 – вік 3-го учня
age4 – вік 4-го учня
age5 – вік 5-го учня
age6 – вік 6-го учня
age7 – вік 7-го учня

Без масивів – дуже погано!!!

```
const age1 = parseInt(prompt('Введіть вік 1-го учня'))  
const age2 = parseInt(prompt('Введіть вік 2-го учня'))  
const age3 = parseInt(prompt('Введіть вік 3-го учня'))  
const age4 = parseInt(prompt('Введіть вік 4-го учня'))  
const age5 = parseInt(prompt('Введіть вік 5-го учня'))  
const age6 = parseInt(prompt('Введіть вік 6-го учня'))  
const age7 = parseInt(prompt('Введіть вік 7-го учня'))
```

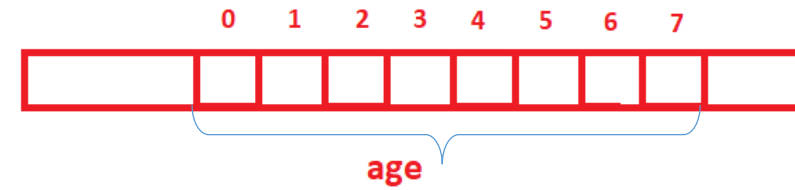
```
let maxAge = age1  
if (age2 > maxAge) maxAge = age2  
if (age3 > maxAge) maxAge = age3  
if (age4 > maxAge) maxAge = age4  
if (age5 > maxAge) maxAge = age5  
if (age6 > maxAge) maxAge = age6  
if (age7 > maxAge) maxAge = age7
```

```
document.write(`Max age = ${maxAge}`)
```

3 масивами - КРАЩЕ

Крок 0. Позначення змінних

age – масив, де будемо зберігати вік учнів



```
let age = []  
age[1] = parseInt(prompt('Введіть вік 1-го учня'))  
age[2] = parseInt(prompt('Введіть вік 2-го учня'))  
age[3] = parseInt(prompt('Введіть вік 3-го учня'))  
age[4] = parseInt(prompt('Введіть вік 4-го учня'))  
age[5] = parseInt(prompt('Введіть вік 5-го учня'))  
age[6] = parseInt(prompt('Введіть вік 6-го учня'))  
age[7] = parseInt(prompt('Введіть вік 7-го учня'))
```

```
let maxAge = age[1]  
if (age[2] > maxAge) maxAge = age[2]  
if (age[3] > maxAge) maxAge = age[3]  
if (age[4] > maxAge) maxAge = age[4]  
if (age[5] > maxAge) maxAge = age[5]  
if (age[6] > maxAge) maxAge = age[6]  
if (age[7] > maxAge) maxAge = age[7]
```

```
document.write(`Max age = ${maxAge}`)
```

Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age – масив, де будемо зберігати вік учнів

```
let age = []
age[1] = parseInt(prompt('Введіть вік 1-го учня'))
age[2] = parseInt(prompt('Введіть вік 2-го учня'))
age[3] = parseInt(prompt('Введіть вік 3-го учня'))
age[4] = parseInt(prompt('Введіть вік 4-го учня'))
age[5] = parseInt(prompt('Введіть вік 5-го учня'))
age[6] = parseInt(prompt('Введіть вік 6-го учня'))
age[7] = parseInt(prompt('Введіть вік 7-го учня'))

let maxAge = age[1]
if (age[2] > maxAge) maxAge = age[2]
if (age[3] > maxAge) maxAge = age[3]
if (age[4] > maxAge) maxAge = age[4]
if (age[5] > maxAge) maxAge = age[5]
if (age[6] > maxAge) maxAge = age[6]
if (age[7] > maxAge) maxAge = age[7]

document.write(`Max age = ${maxAge}`)
```

Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age – масив, де будемо зберігати вік учнів

Можливість використання циклів



```
let age = []
age[1] = parseInt(prompt('Введіть вік 1-го учня'))
age[2] = parseInt(prompt('Введіть вік 2-го учня'))
age[3] = parseInt(prompt('Введіть вік 3-го учня'))
age[4] = parseInt(prompt('Введіть вік 4-го учня'))
age[5] = parseInt(prompt('Введіть вік 5-го учня'))
age[6] = parseInt(prompt('Введіть вік 6-го учня'))
age[7] = parseInt(prompt('Введіть вік 7-го учня'))

let maxAge = age[1]
if (age[2] > maxAge) maxAge = age[2]
if (age[3] > maxAge) maxAge = age[3]
if (age[4] > maxAge) maxAge = age[4]
if (age[5] > maxAge) maxAge = age[5]
if (age[6] > maxAge) maxAge = age[6]
if (age[7] > maxAge) maxAge = age[7]

document.write(`Max age = ${maxAge}`)
```

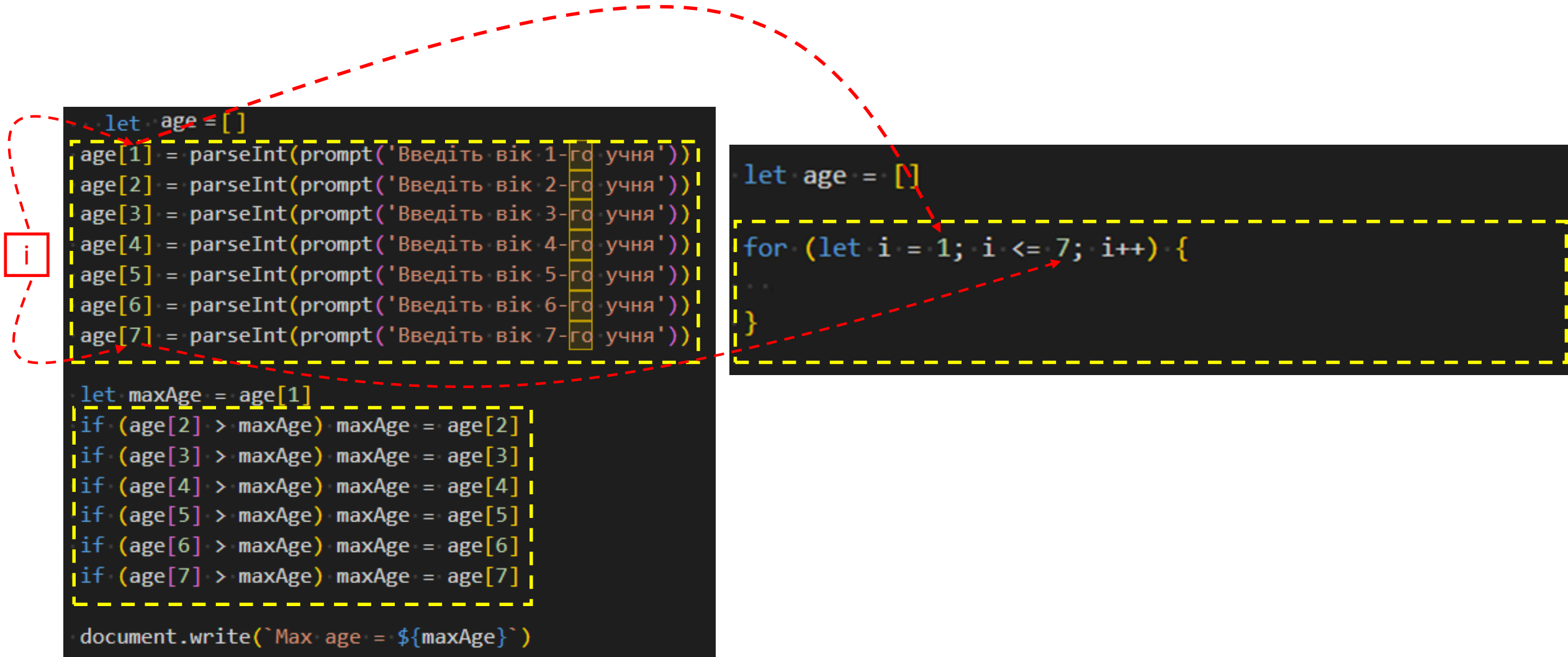
Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age – масив, де будемо зберігати вік учнів

Можливість використання циклів



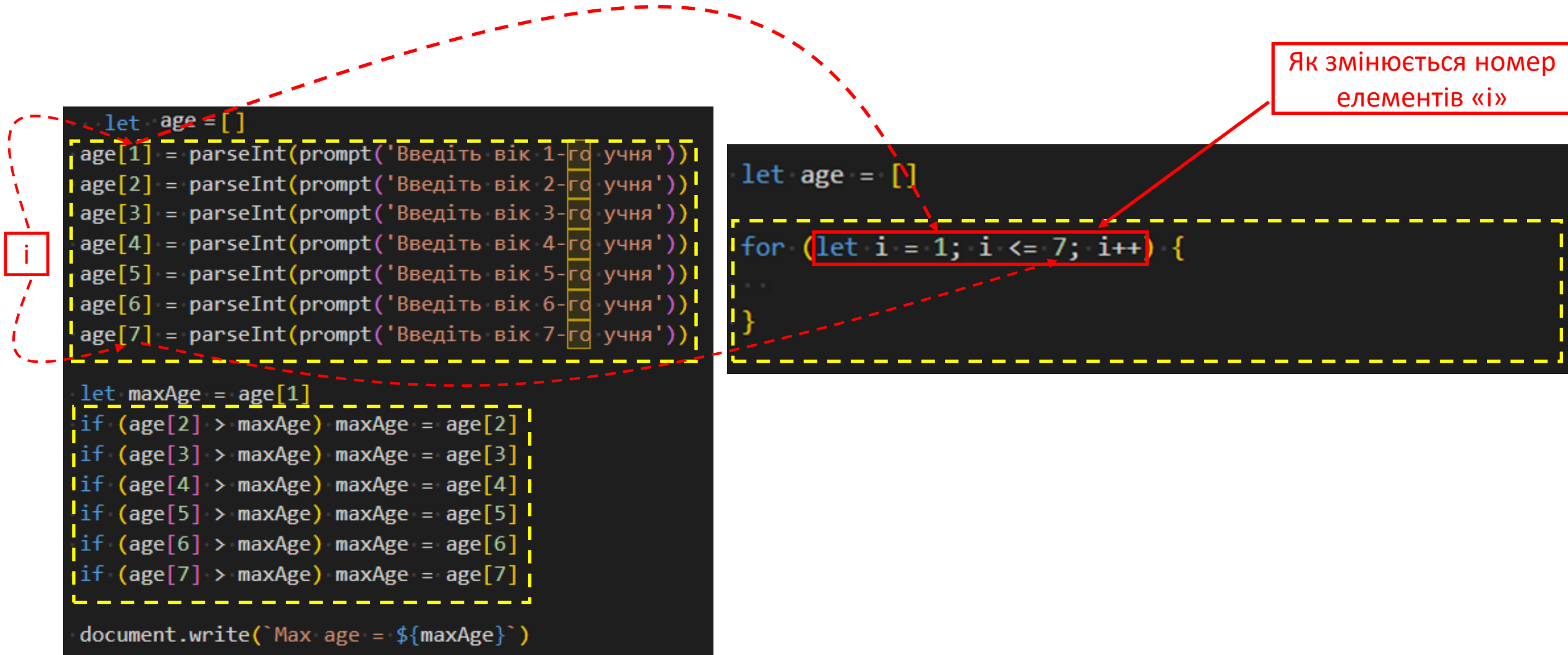
Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age – масив, де будемо зберігати вік учнів

Можливість використання циклів



Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age – масив, де будемо зберігати вік учнів

Можливість використання циклів



Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age – масив, де будемо зберігати вік учнів

Можливість використання циклів

```
let age = []
age[1] = parseInt(prompt('Введіть вік 1-го учня'))
age[2] = parseInt(prompt('Введіть вік 2-го учня'))
age[3] = parseInt(prompt('Введіть вік 3-го учня'))
age[4] = parseInt(prompt('Введіть вік 4-го учня'))
age[5] = parseInt(prompt('Введіть вік 5-го учня'))
age[6] = parseInt(prompt('Введіть вік 6-го учня'))
age[7] = parseInt(prompt('Введіть вік 7-го учня'))

let maxAge = age[1]
if (age[2] > maxAge) maxAge = age[2]
if (age[3] > maxAge) maxAge = age[3]
if (age[4] > maxAge) maxAge = age[4]
if (age[5] > maxAge) maxAge = age[5]
if (age[6] > maxAge) maxAge = age[6]
if (age[7] > maxAge) maxAge = age[7]

document.write(`Max age = ${maxAge}`)
```

```
let age = []
for (let i = 1; i <= 7; i++) {
  age[i] = parseInt(prompt(`Введіть вік ${i}-го учня`))
}
```

Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age – масив, де будемо зберігати вік учнів

Можливість використання циклів



Для чого потрібні масиви?

Задача. Дано вік 7-х учнів. Знайти вік старшого учня.

Крок 0. Позначення змінних

age – масив, де будемо зберігати вік учнів

Можливість використання циклів

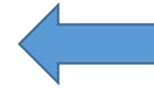


Задача. Дано вартості товарів, яку купив користувач. Знайти загальну вартість усіх товарів

Введення елементів масиву

```
let age = []

for (let i = 1; i <= 7; i++) {
  age[i] = parseInt(prompt(`Введіть вік ${i}-го учня`))
}
```



Так недуже добре

Приклад:

```
let age = [] //age = []

//      0      <-- індекси
age[0] = 5 //age = [5]

//      0      1      <-- індекси
age[1] = 8 //age = [5, 8]

//      0      1      2      <-- індекси
age[2] = 23 //age = [5, 8, 23]
```

Якщо елемента з таким індексом не існувало,
то його буде створено

```
//      0      1      2      3      4      5      6      7      <-- індекси
age[7] = 56 //age = [5, 8, 23, , , , , , 56]
```

Небезпека виникнення порожніх елементів
(які дорівнюють undefined, «дірок»)

Введення масиву

Виділення пам'яті для елементів масиву наперед
(спочатку елементи є порожніми (undefined))

назва_масиву = **new Array**(кількість_елементів)

```
let studentsNumber = parseInt(prompt('Введіть кількість учнів'))
let studentsAges = new Array(studentsNumber)
for (let i = 0; i < studentsAges.length; i++) {
  let studentsAge = parseInt(prompt(`Введіть вік `${i}`-го учня:`))
  studentsAges[i] = studentsAge
}
```

Виділяємо пам'ять для масиву

Вводимо вік i-го учня

Зберігаємо у i-вій позиції

Приклад:

```
// .....0.....1.....2<---індекси
let studentsNumber = new Array(3) //studentsNumber=[.....,.....,.....]

// .....0.....1.....2<---індекси
studentsNumber[0] = 5 //studentsNumber=[5,.....,.....]

// .....0.....1.....2<---індекси
studentsNumber[1] = 8 //studentsNumber=[5,8,.....]

// .....0.....1.....2<---індекси
studentsNumber[2] = 23 //studentsNumber=[5,8,23]
```


Введення масиву

Приклад:



Виділили пам'ять для 3-х елементів (поки невизначені – undefined)

```
let studentsNumber = new Array(3) //studentsNumber = [ , , ]
```

0 1 2 <-- індекси

Введення масиву

Приклад:



```
//.....0.....1.....2<-- індекси
let studentsNumber = new Array(3) //studentsNumber=[.....,.....,.....]

//.....0.....1.....2<-- індекси
studentsNumber[0] = 5 //studentsNumber=[5,.....,.....]
```

Записали 5 у позицію під номером 0

Введення масиву

Приклад:



```
// ..... 0 ..... 1 ..... 2 ..... індекси
let studentsNumber = new Array(3) ..... //studentsNumber = [ ....., ....., ..... ]

// ..... 0 ..... 1 ..... 2 ..... індекси
studentsNumber[0] = 5 ..... //studentsNumber = [ 5, ....., ..... ]

// ..... 0 ..... 1 ..... 2 ..... індекси
studentsNumber[1] = 8 ..... //studentsNumber = [ 5, 8, ..... ]
```

Записали 8 у позицію під номером 1

Введення масиву

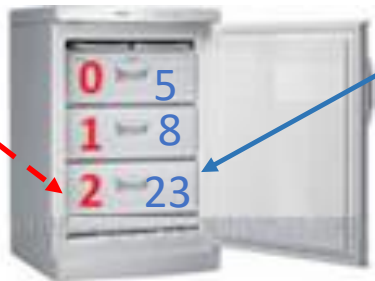
Приклад:

```
// .....0.....1.....2<-- індекси
let studentsNumber = new Array(3) //studentsNumber=[.....,.....,.....]

// .....0.....1.....2<-- індекси
studentsNumber[0] = 5 //studentsNumber=[5,.....,.....]

// .....0.....1.....2<-- індекси
studentsNumber[1] = 8 //studentsNumber=[5,8,.....]

// .....0.....1.....2<-- індекси
studentsNumber[2] = 23 //studentsNumber=[5,8,23]
```



Записали 23 у позицію під номером 2

Введення масиву

Додавання елементів у кінець

(спочатку виділяється пам'ять для елемента у кінці масиву,
потім зберігається значення елемента)

назва_масиву.push(значення)

```
let studentsNumber = parseInt(prompt('Введіть кількість учнів'))
let studentsAges = []
for (let i = 0; i < studentsNumber; i++) {
  let studentsAge = parseInt(prompt(`Введіть вік -${i}-го учня:`))
  studentsAges.push(studentsAge)
}
```

Спочатку масив порожній

Вводимо вік i-го учня

Додаємо в кінець масиву

Приклад:

```
let studentsNumber = [] //studentsNumber = []
// ..... // ..... 0 <-- індекси
studentsNumber.push(5) //studentsNumber = [5]
// ..... // ..... 0 1 <-- індекси
studentsNumber.push(8) //studentsNumber = [5, 8]
// ..... // ..... 0 1 2 <-- індекси
studentsNumber.push(23) //studentsNumber = [5, 8, 23]
```



Створили порожній масив (кількість елементів **length** = 0)

```
let studentsNumber = [] //studentsNumber = []
```



```
let studentsNumber = [] //studentsNumber = []  
..... // ..... 0 <-- індекси  
studentsNumber.push(5) //studentsNumber = [5]
```

Додали один елемент (він є першим, під номером 0) і зберегли у ньому 5



```
let studentsNumber = [] //studentsNumber = []  
  
..... // ..... 0 ..... <-- індекси  
studentsNumber.push(5) //studentsNumber = [5]  
  
..... // ..... 0 ..... 1 ..... <-- індекси  
studentsNumber.push(8) //studentsNumber = [5, 8]
```

Додали в кінець ще один елемент (він є другим, під номером 1) і зберегли у ньому 8


```
let studentsNumber = [] //studentsNumber = []  
  
..... // ..... 0 ..... <-- індекси  
studentsNumber.push(5) //studentsNumber = [5]  
  
..... // ..... 0 ..... 1 ..... <-- індекси  
studentsNumber.push(8) //studentsNumber = [5, 8]  
  
..... // ..... 0 ..... 1 ..... 2 ..... <-- індекси  
studentsNumber.push(23) //studentsNumber = [5, 8, 23]
```



Додали в кінець ще один елемент (він є третім, під номером 2) і зберегли у ньому 23

Задача. Дано розміри доходу магазину за кожен місяць протягом року. Знайти:

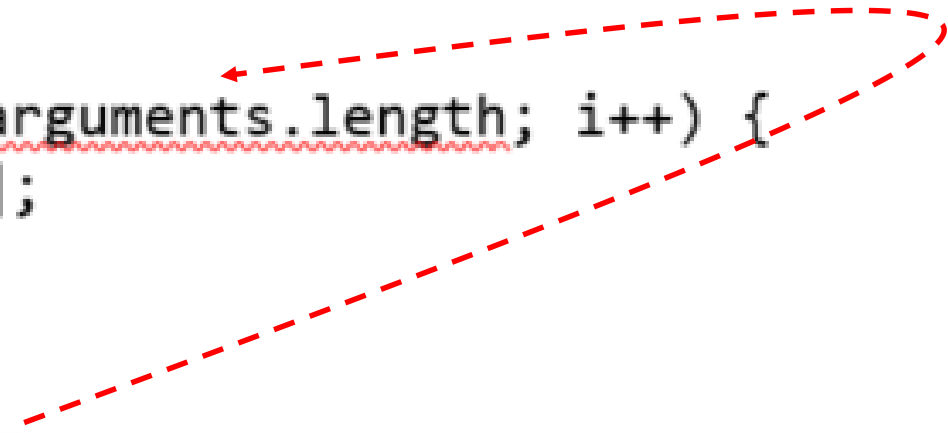
- 1) загальний дохід за рік
- 2) загальний дохід за 1-ше півріччя
- 3) загальний дохід за 2-ге півріччя
- 4) загальний дохід за 2-гий квартал
- 5) загальний дохід за 2-гий і 3-тій квартал

Стосовно функцій

Передача довільної кількості параметрів (усі значення, які передаємо у функцію заносяться у псевдо-масив arguments)

===== Функція для знаходження суми довільної кількості аргументів =====

```
function sum() {  
    var s = 0;  
    for (var i = 0; i < arguments.length; i++) {  
        s += arguments[i];  
    }  
    return s;  
}  
var r = sum(2, 34, 23, 45);  
alert(r);
```



```
//-----  
// Знайти розмір заробітної плати (ставка + премія - податок)  
function getFinalSalary() {  
    //arguments[0] - ставка, arguments[1] - премія, arguments[2] - податок  
    return (arguments[0] + arguments[1]) * (1 - arguments[2])  
}
```

Задача. Створити функцію, яка для довільної кількості цін товарів знаходить суму цін товарів, які більші за 100.

Оператор spread «...»

Дозволяє деякі значення передавати в іменовані параметри, а усі інші у змінну-масив

```
function назва(параметр1, параметр2, ... , параметрN,  
  . . . акумулячий масив ) {  
  тіло функції;  
  return результат;  
}
```

Може бути тільки один і обов'язково повинен знаходитись у кінці

```
// Визначити скільки елементів належать заданому діапазону
```

```
function getCount(min,max,...numbers) {  
  var count = 0;  
  for (var i = 0; i < numbers.length; i++) {  
    if (numbers[i] >= min && numbers[i] <= max) {  
      count++;  
    }  
  }  
  return count;  
}  
  
alert(getCount ( 1,    10,    3, 21, 45, 3, 90, 7));  
      //min=1, max=10, numbers=[3, 21, 45, 3, 90, 7]
```

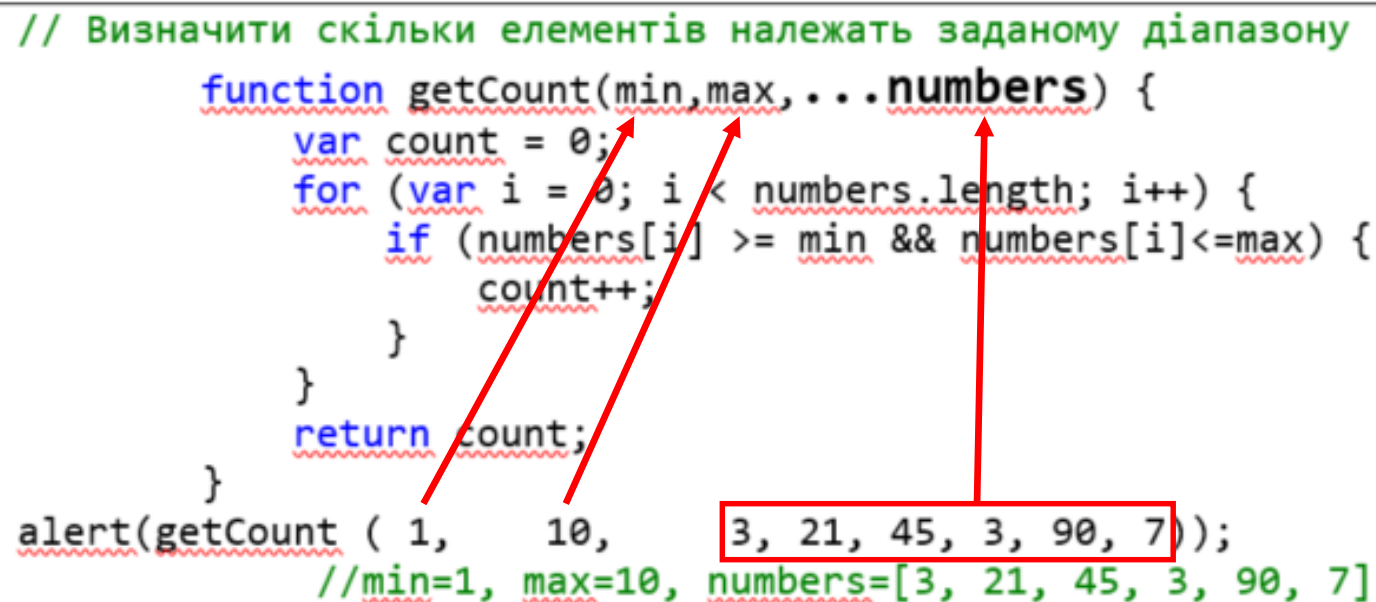
Оператор spread «...»

Дозволяє деякі значення передавати в іменовані параметри, а усі інші у змінну-масив

```
function назва(параметр1, параметр2, ... , параметрN,  
  . . . акумулячий масив ) {  
  тіло функції;  
  return результат;  
}
```

Може бути тільки один і обов'язково повинен знаходитись у кінці

```
// Визначити скільки елементів належать заданому діапазону  
function getCount(min,max,...numbers) {  
  var count = 0;  
  for (var i = 0; i < numbers.length; i++) {  
    if (numbers[i] >= min && numbers[i] <= max) {  
      count++;  
    }  
  }  
  return count;  
}  
alert(getCount ( 1, 10, 3, 21, 45, 3, 90, 7));  
//min=1, max=10, numbers=[3, 21, 45, 3, 90, 7]
```



Задача. Створити функцію, яка приймає рік виходу на пенсію, та довільну кількість значень років працівників. Визначити кількість пенсіонерів.