

Python

문자열(2), List

“

철수는 2000년 11월 3일에 태어났다.
오늘(2020년 11월 21일)은 태어난지 20개월째 되는 날일까?

”

운년을 고려해야 한다.

윤년 ?

1. 연도를 4로 나누어 떨어진다.
2. 단, 100으로 나누어 떨어지는 경우는 제외한다.
3. 하지만 연도가 400으로 나누어지면 포함시킨다.

윤년 ?

1. 연도를 4로 나누어 떨어진다.

$$\text{year} \% 4 == 0$$

2. 단, 100으로 나누어 떨어지는 경우는 제외한다.

$$\text{year} \% 100 != 0$$

3. 하지만 연도가 400으로 나누어지면 포함시킨다.

$$\text{year} \% 400 == 0$$

```
is_leap_year =  
(year % 4 == 0 and year % 100 != 0) or year %  
400 == 0
```

이전 코드에 적당한 위치가 어디일까요?
해당 위치에 윤년을 반영하도록 수정하시오.

```
currentMonth = monthOfBirthday + 1
while (currentMonth <= 12) :
    if (currentMonth == 1 ) :
        days = days + 31
    elif (currentMonth == 2 ) :
        days = days + 28
        is_leap_year = (year % 4 == 0 and year % 100 != 0)
                        or year % 400 == 0
        if is_leap_year:
            days += 1
    elif (currentMonth == 3 ) :
        days = days + 31
    elif (currentMonth == 4 ) :
        days = days + 30
...
currentMonth+=1
```


“

세계의 정수 값을 입력 받아 세계의 값 중에 최대값을 찾는
프로그램을 작성하시오.

”

문자열
(string)

자료형 (data type)

2. 문자열(string) : 문자, 단어 등으로 구성된 문자들의 집합

“Life is too short, You need Python”

“a”

“123”

자료형 (data type)

문자열 표현

“Life is too short, You need Python”

‘Life is too short, You need Python’

“Life is too short,\n You need Python”

‘Life is too short,\n You need Python’

“ Life is too short,
You need Python”

““ Life is too short,
You need Python””

문자열 연산

String concatenation

```
>>> A = "Life is too short"
```

```
>>> B = "You need Python"
```

```
>>> C = A + B
```

```
>>> print(C)
```

```
"Life is too shortYou need Python"
```

```
>>> D = A + ", " + B
```

```
>>> print(D)
```

```
"Life is too short, You need Python"
```

문자열 연산

String Replication

문자열 * 정수

or

정수 * 문자열

~~문자열 * 문자열~~ # error

문자열을 반복할 횟수

```
>>> A = "Life is too short"
```

```
>>> B = A * 3
```

```
>>> print(B)
```

```
"Life is too shortLife is too shortLife is too short"
```

문자열 연산

len() function

문자열의 길이 구하기

len(문자열)

```
>>> A = "Life is too short"
```

```
>>> B = len(A)
```

```
>>> print(B)
```

```
7
```

문자열 연산

문자열 안의 문자 참조 []

문자열의 앞에서 부터 → 0 1 2 3 4 5 6 7 8 9 10

H	e	l	l	o		w	o	r	l	d
---	---	---	---	---	--	---	---	---	---	---

-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 ← 문자열의 뒤에서 부터

```
>>> A = "Hello world"
```

```
>>> print(A[6])
```

w

```
>>> print(A[-5])
```

w

```
>>> print(A[1:7]) # 1 ≤ i < 7
```

ello w

```
>>> print(A[2:])
```

llo world

“

Palindrome string

앞으로 읽어도, 뒤로 읽어도 같은 문자열이 되는 경우 인지 판단
하는 프로그램을 작성하시오.

“eye”, “madam” 은 둘 다 True

“hello”는 False

* 팰린드롬(palindrom)이란 eye, madam처럼 역순으로 읽어도 같은 말이나 구절 또는 숫자를 말한다.

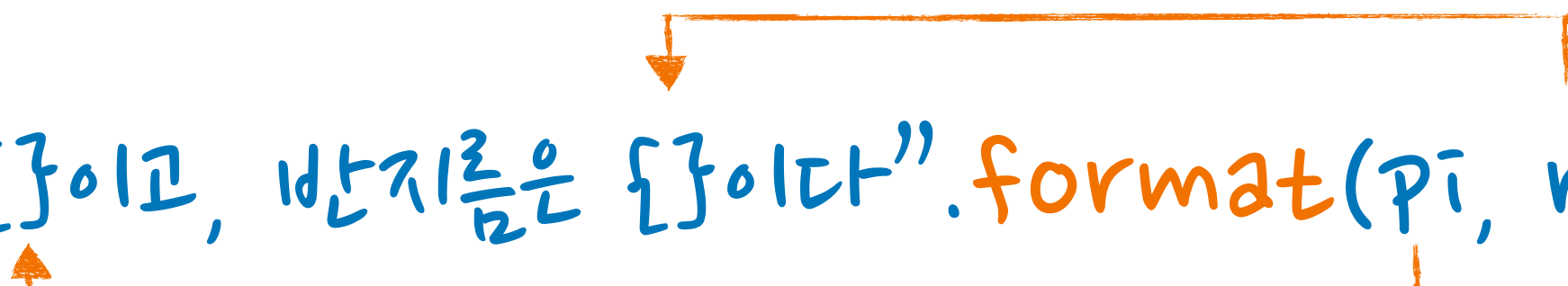
”

서식을 이용해서 문자열 완성하기

$\pi = 3.14$

$R = 4$

`str_var = "π의 값은 {}이고, 반지름은 {}이다".format(π , r)`



`print(str_var)`

`str_var = "π의 값은 {}이고, 반지름은 {}이다".format(r, π)`


`print(str_var)`

서식을 이용해서 문자열 완성하기(2)

$\pi = 3.14$

$R = 4$

`str_var = "π의 값은 %.f이고, 반지름은 %d이다" % (π, r)`



`print(str_var)`

`%d` 정수값을 정수로

`%x` 정수값을 16진수로

`%f` 실수값을

`%s` 문자열을

배열

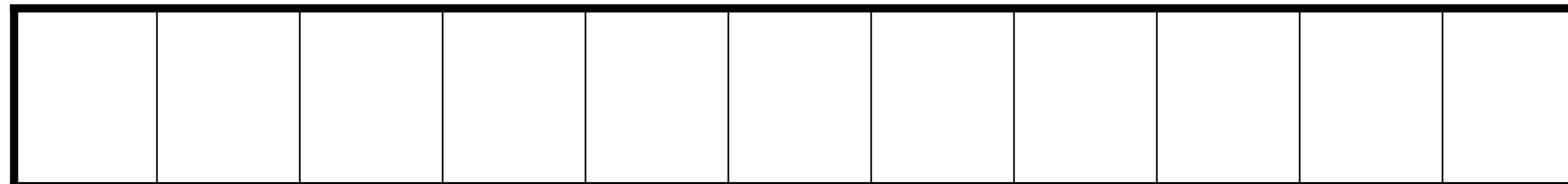
(list, tuple)

배열(array)은 자료구조(data structure)의 한 형태이다.

배열 안에 있는 각 항은 요소(element)라고 한다.

이 요소를 참조하기 위해서는 인덱스(index)를 사용한다.

배열의 앞에서 부터 index값 \rightarrow 0 1 2 3 4 5 6 7 8 9 10



-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1

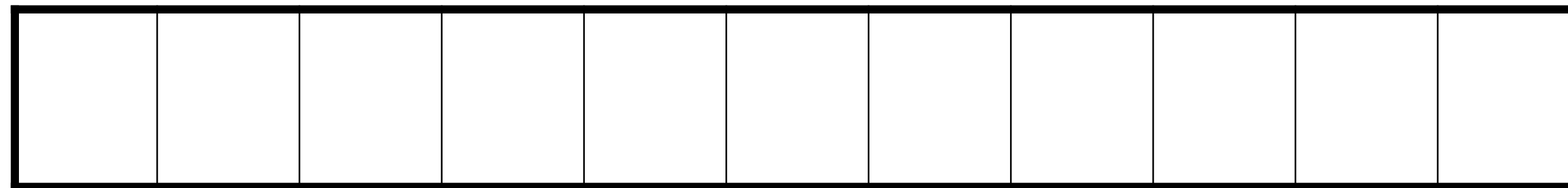
문자열의 뒤에서 부터
index표현 \leftarrow

Python에서

배열은 list와 tuple이란 것으로 표현 할 수 있다.

요소는 정수, 실수, 문자열 뿐 아니라 data structure도 될 수 있다.

배열의 앞에서 부터 index가 $\rightarrow 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10$

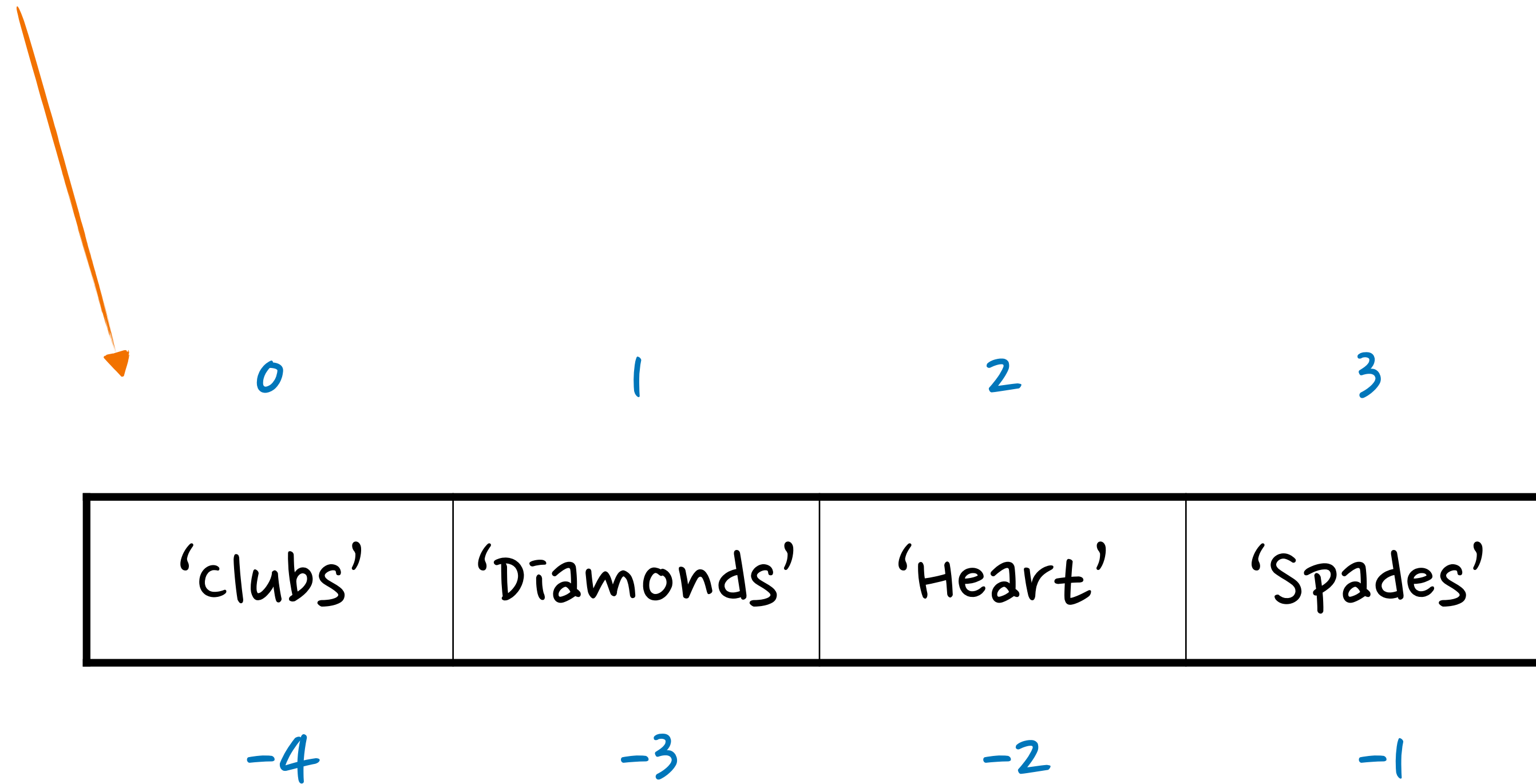


$-11 \quad -10 \quad -9 \quad -8 \quad -7 \quad -6 \quad -5 \quad -4 \quad -3 \quad -2 \quad -1$

문자열의 뒤에서 부터
index표현 \leftarrow

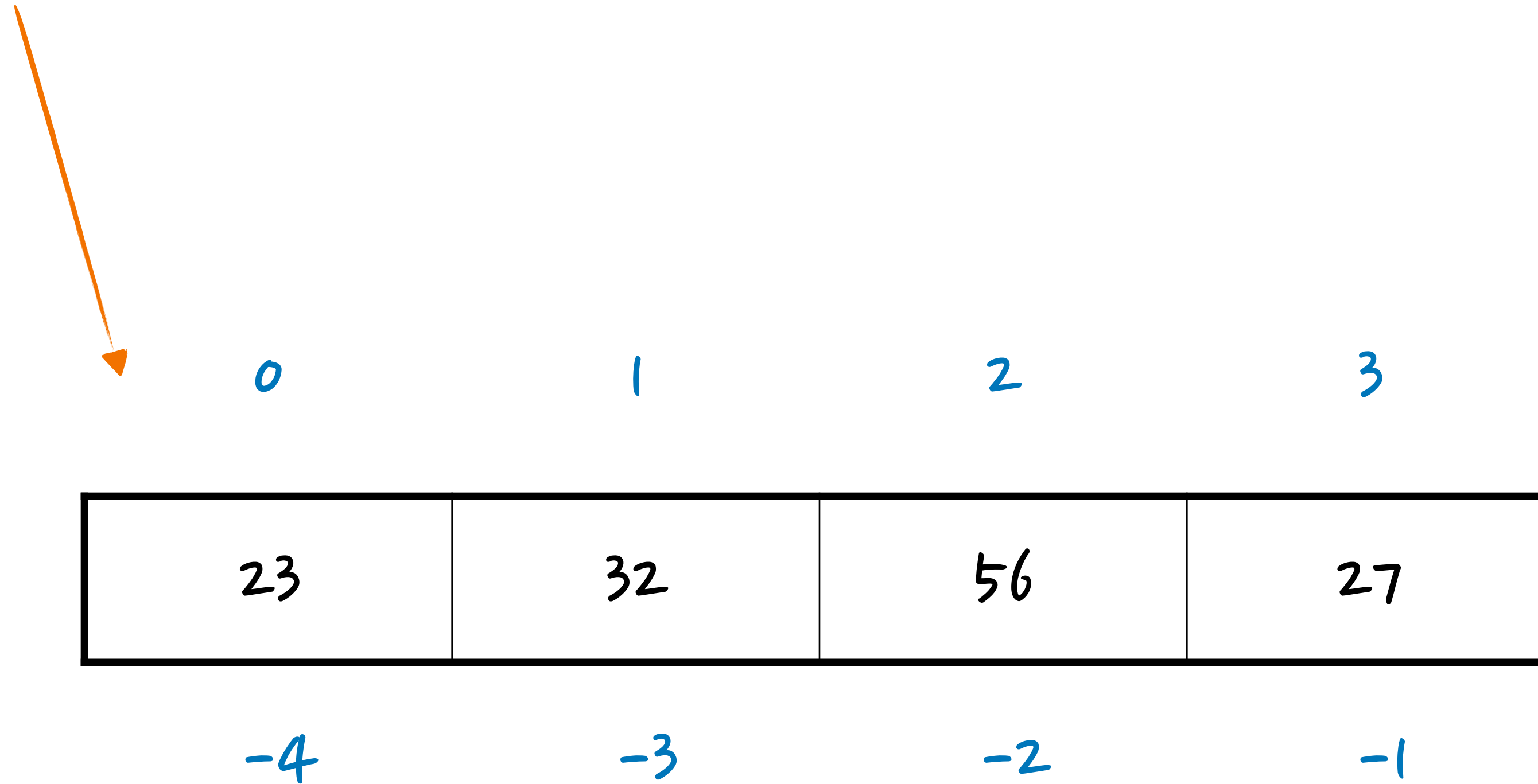
List

Suits = ['clubs', 'Diamonds', 'Heart', 'Spades']



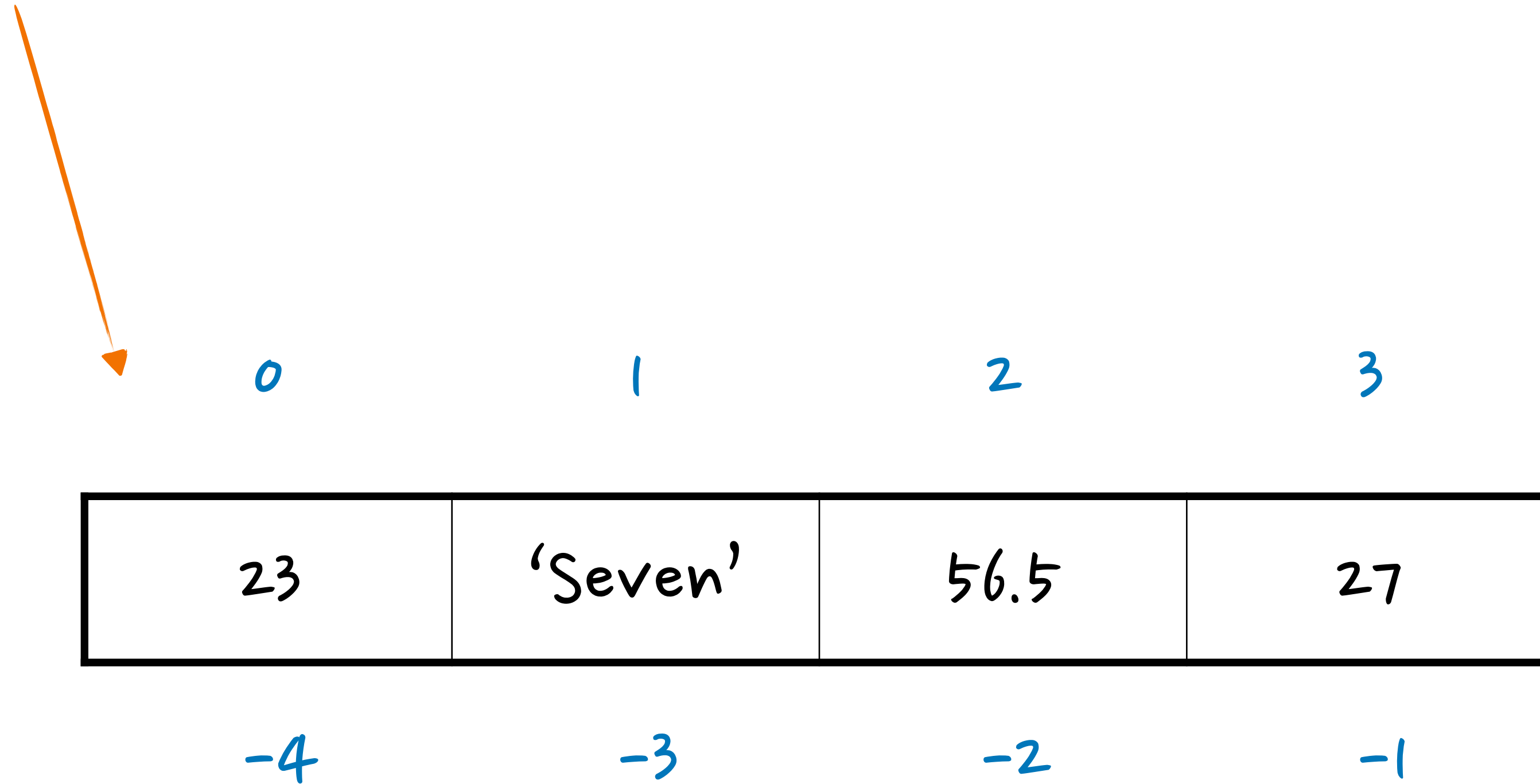
List

Ages = [23, 32, 56, 27]



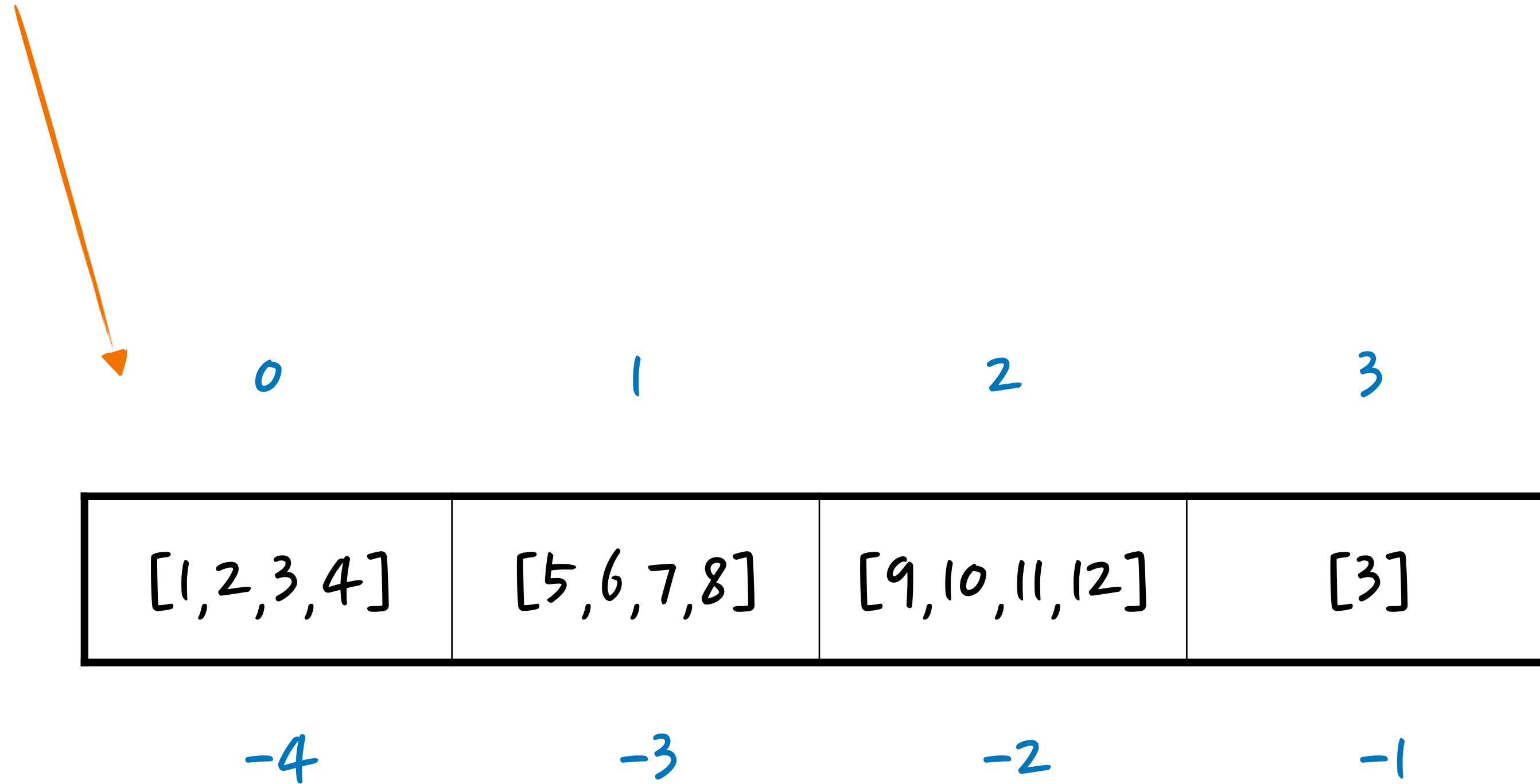
List

aList = [23, 'Seven', 56.5, 27]



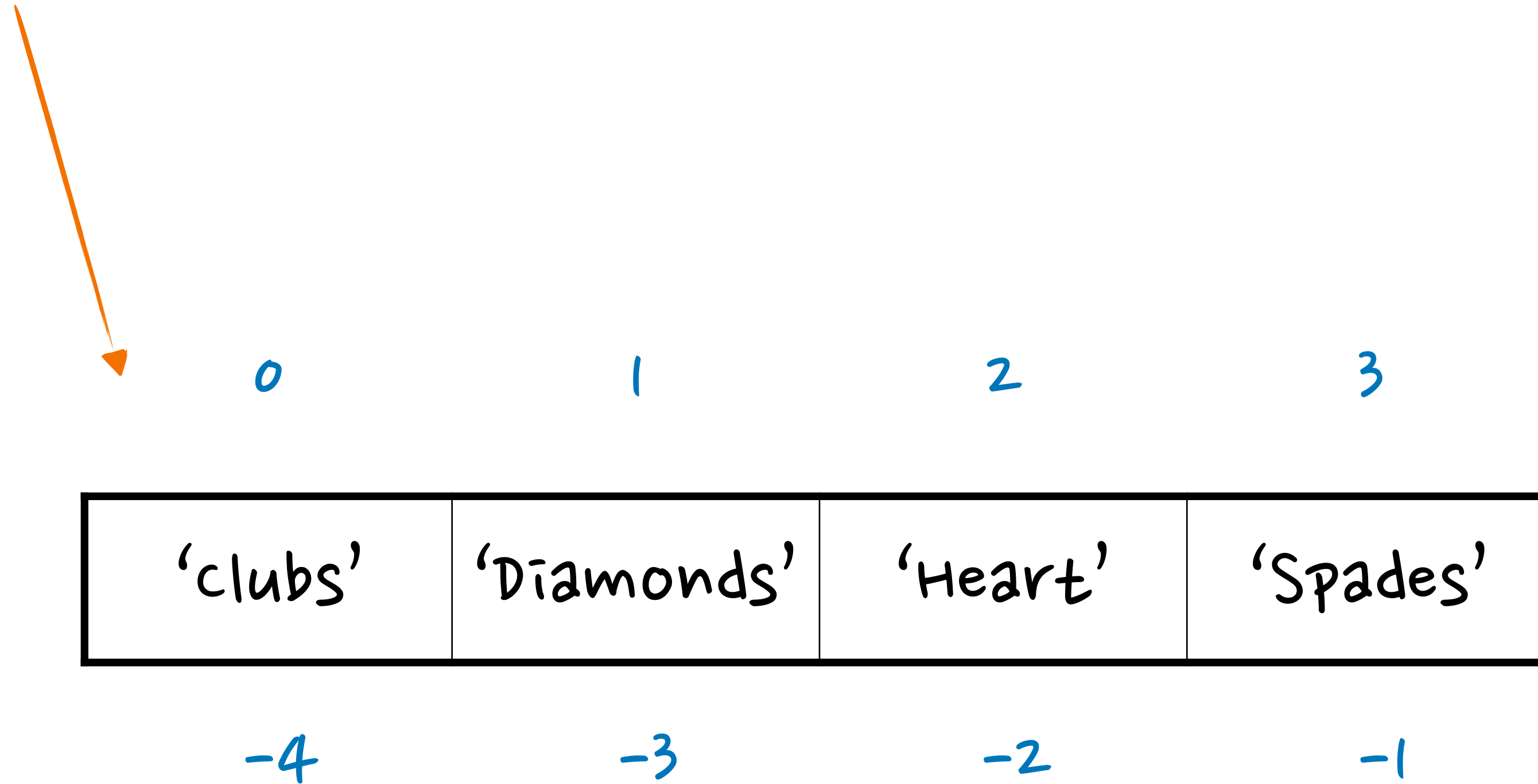
List

list_in_list = [[1,2,3,4], [5,6,7,8], [9,10,11,12], [3]]



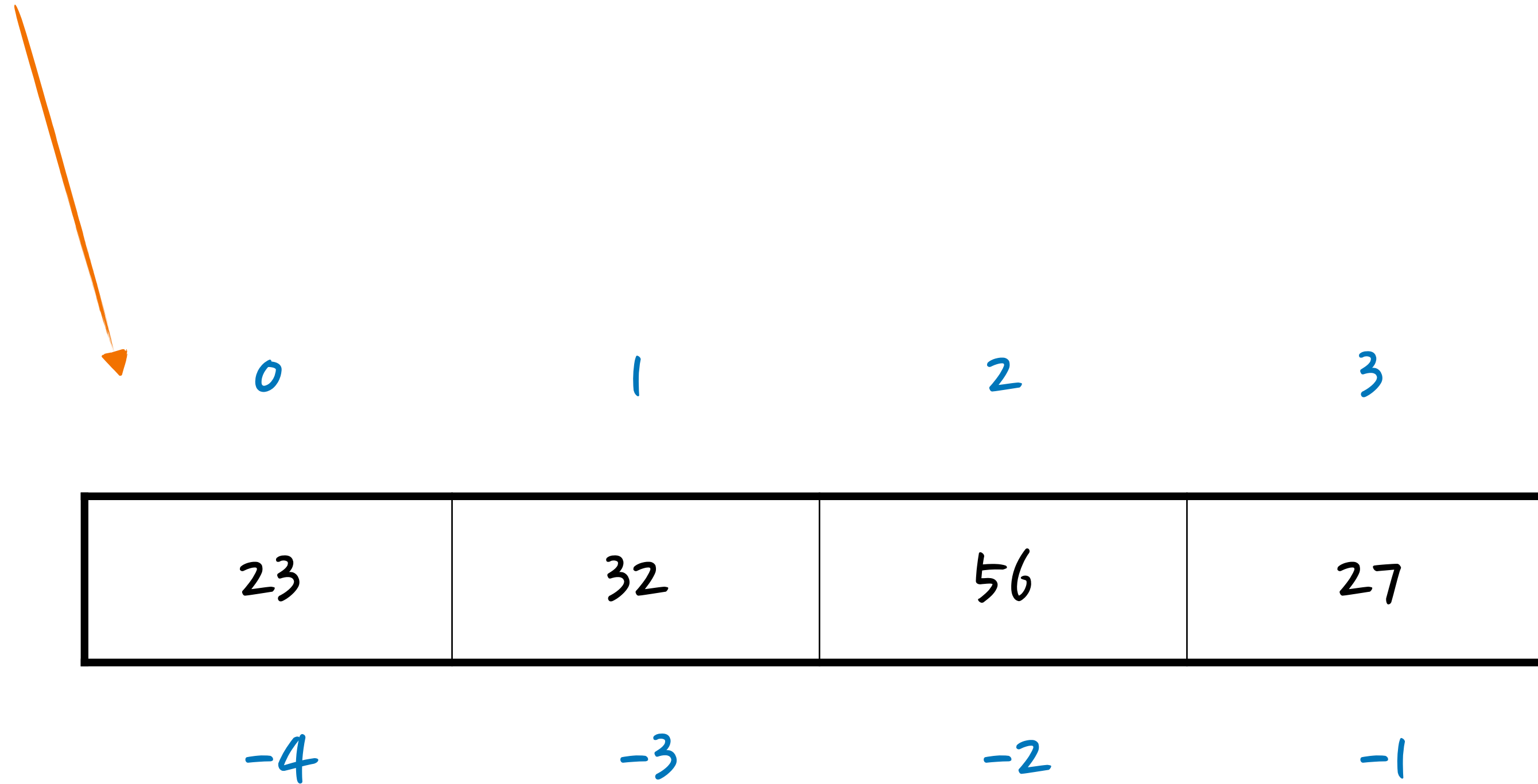
Tuple

Suits = ('clubs', 'Diamonds', 'Heart', 'Spades')



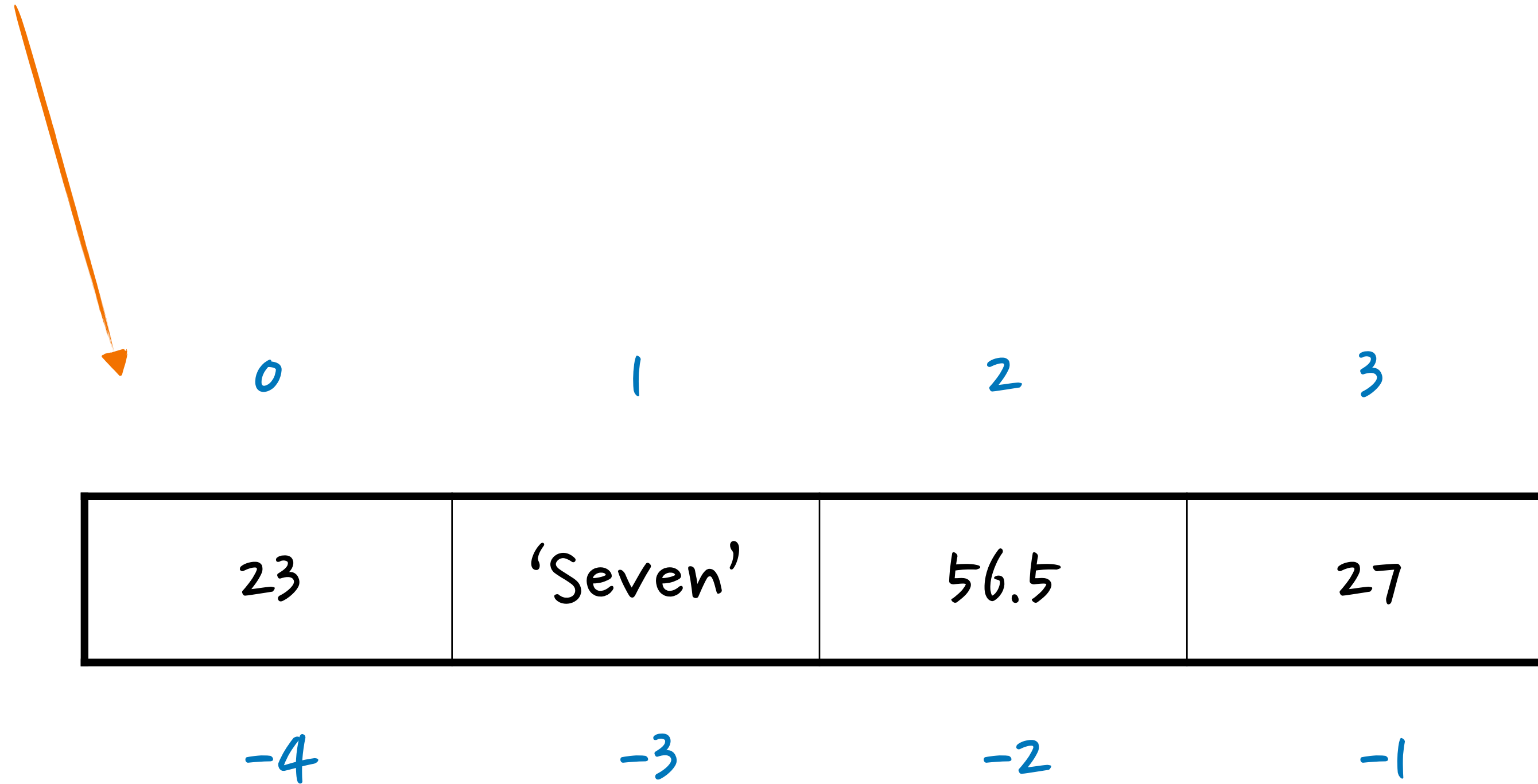
Tuple

Ages = (23, 32, 56, 27)



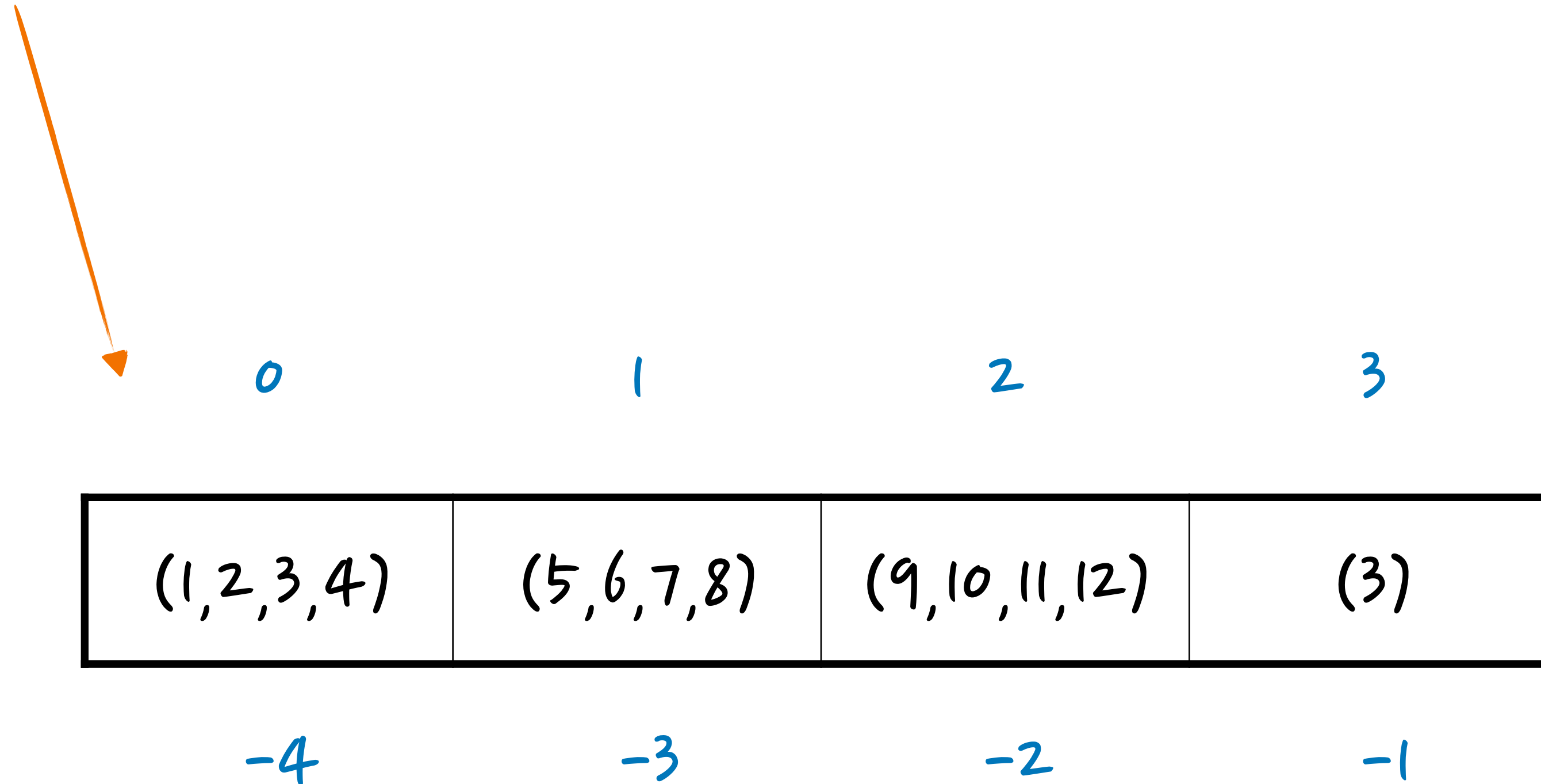
Tuple

aTuple = (23, 'Seven', 56.5, 27)



Tuple

`tuple_in_tuple = ((1,2,3,4), (5,6,7,8), (9,10,11,12), (3))`



list는 되고 Tuple 은 안되는 것

```
aList = []
```

```
aList.append(100)
```

```
aList += [200]
```

```
aList[0] = 200
```

```
print(aList)
```

```
Del aList[0] #해당 항목 삭제
```

```
print(aList)
```

list, Tuple 둘 다 되는것

```
aList = [100, 200, 300, 400]
```

```
print(aList.index(200))
```

```
aTuple = (100, 200, 300, 400)
```

```
print(aTuple.index(200))
```

“

aList = [23, 56, 12, 7, 9, 201, 33]과 같은 list가 주어졌을 때, list안에 최대값을 찾아 해당 index를 출력하시오.

”

“

주어진 배열 $nums = [2, 7, 11, 15]$ 가 있을 때 배열 안의
두 개의 원소를 더해 주어진 $target = 9$ 에 만족할 때, 해당
인덱스 $[0, 1]$ 을 보여주는 프로그램을 작성하라.

”

“

철수는 2000년 11월 3일에 태어났다.
오늘(2020년 11월 21일)은 태어난지 20개월째 되는 날일까?

”