

CHAPTER 6

ARRAYS

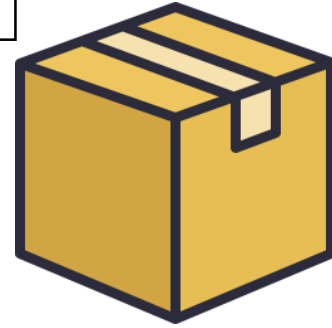
PART 1

DEFINE, ACCESS, BROWSE

A variable has a name and contain a data

bob = 45

I am Bob

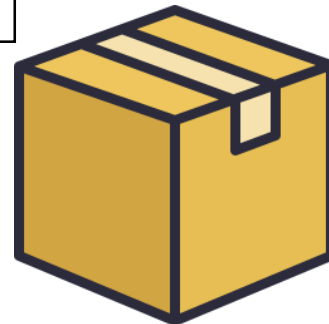


← 45



bob = “45”

I am Bob



← “45”





Simple types of data :



Integer

1 5



Float

1.45



Boolean

True

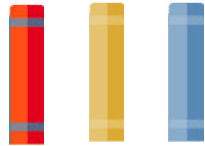


String

“blabla”



Array : data containing other data

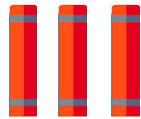


Array

[14, "ronan" , 18.4]

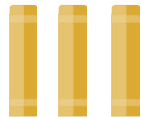


Arrays can contains any type of value



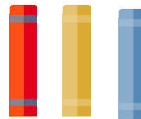
Array of integers

[14, 15 , 18]



Array of strings

["banana", "coconut"]



Array of mixed types

[14, "ronan", True]

Let's define an array !

Coma to separate each value

myArray = [14, 15, 18]



Start array

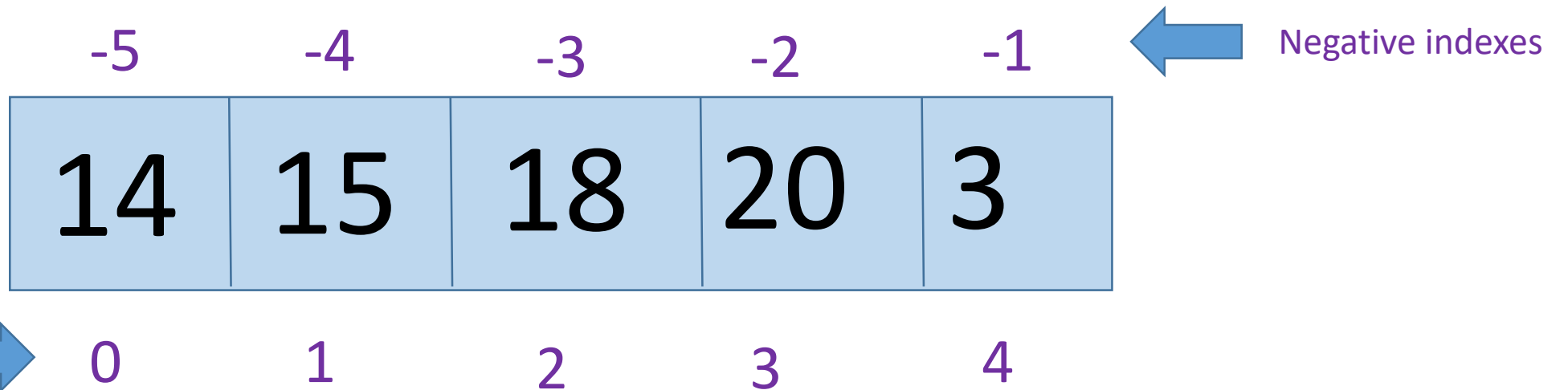



End array

Let's access to array elements !!

myArray = [14, 15, 18, 20, 3]

myElement = myArray[2]





What this code will print ?

```
myNumbers = [5, 2, 8]  
print( myNumbers[-1] )
```




What this code will print ?

```
myNumbers = [5, 2, 8]  
print( myNumbers[3] )
```



What this code will print ?

```
print(len("45"))  
print(len([45]))
```



What this code will print ?

```
print(len(45))
```



What this code will print ?

```
print( len("ronan, pnc") )  
print( len( ["ronan", "pnc"] ) )
```

2 ways to loop on array !!

Using array index

```
array = [1, 2, 3, 4, 5]
for index in range(len(array)):
    value = array[index]
    print("hello " + str(value))
```

More customizable

Using array values

```
array = [1, 2, 3, 4, 5]
for value in array:
    print("hello " + str(value))
```

Shorter way to loop



What this the meaning of this function?

```
def fonction1(myArray):  
    m = myArray[0]  
    for value in myArray:  
        if value > m :  
            m = value  
    return m
```



What this code will print ?

```
list1 = [5, 4, 7, 3, 1]  
list2 = list1
```

```
list1[0] = 99
```

```
print(list1[0])  
print(list2[0])
```

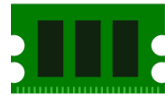


And why ?!!

list1 and **list2** are referring to
The same value in memory !!



list1



MEMORY



list2

[5, 4, 7, 3, 1]



What this code will print ?

```
a = 5  
b = a  
a = 7  
print(b)
```

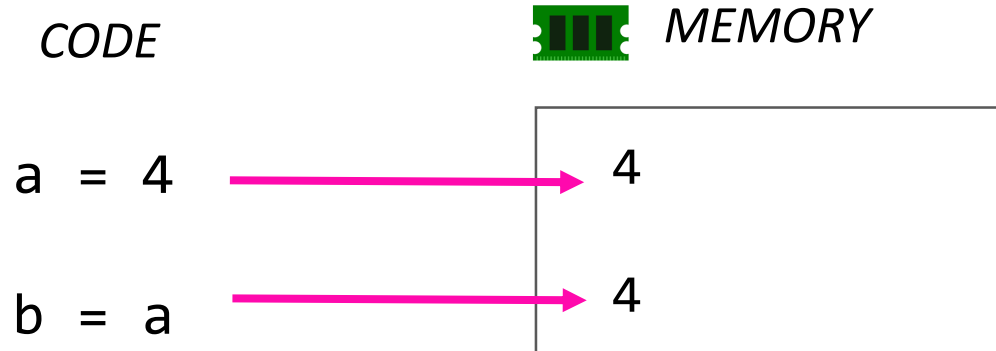


And why ?!!

Make the difference !

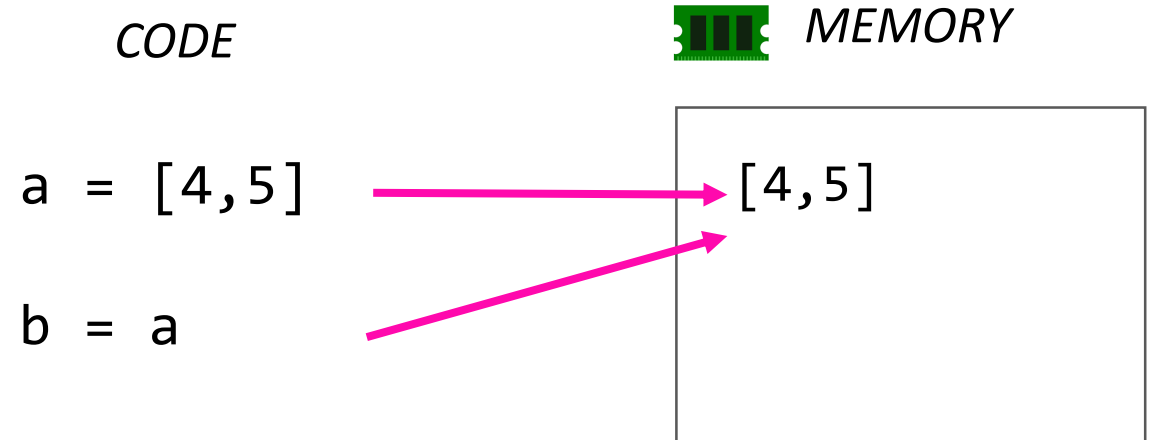
Integer, string, boolean

→ We **duplicate** the value
In memory



Array

→ We **refer to the same** value
In memory





What this code will print ?

```
a = [18, 24]  
b = a  
a[1]=3  
print(b[1])
```



And why ?!!