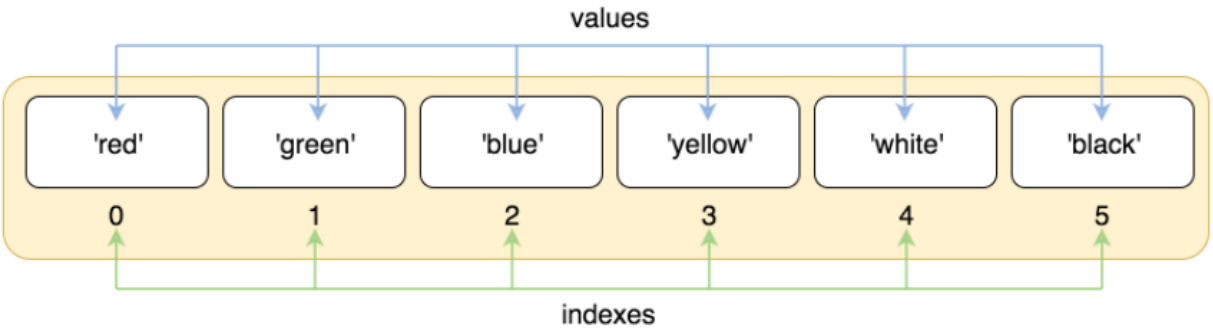
Indexing:

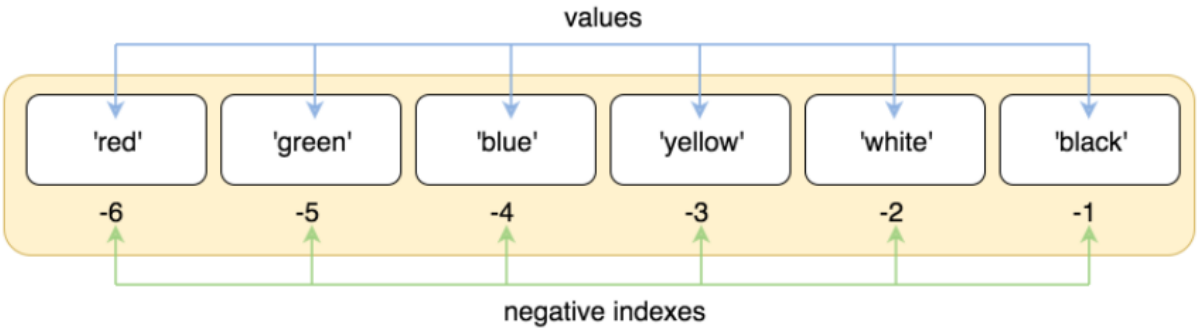
Let's take a simple example:

```
colors = ['red', 'green', 'blue', 'yellow', 'white', 'black']
```

Python uses zero-based indexing. That means, the first element(value 'red') has an index 0, the second(value 'green') has index 1, and so on.

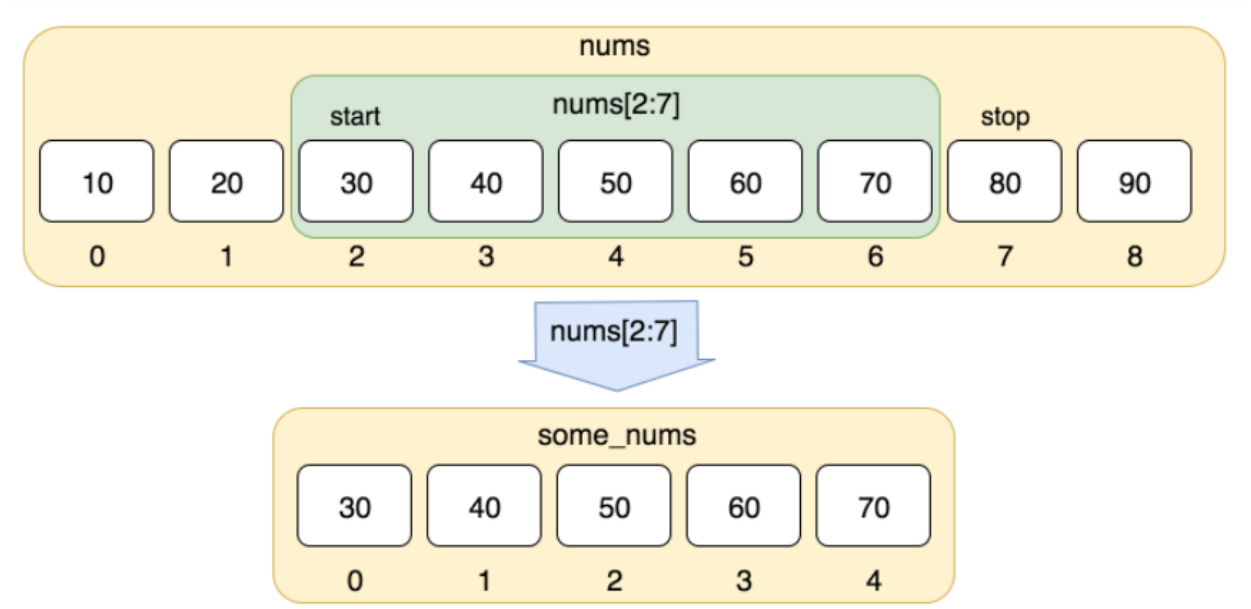


To address this requirement there is negative indexing. So, instead of using indexes from zero and above, we can use indexes from -1 and below.



In negative indexing system -1 corresponds to the last element of the list(value 'black'), -2 to the penultimate (value 'white'), and so on.

Slicing:




```
[start : stop : steps]

which means that slicing will start from index start
 will go up to stop in step of steps.
 Default value of start is 0, stop is last index of list
 and for step it is 1
```

append() and extend() in Python

Last Updated: 01-04-2020

Append: Adds its argument as a single element to the end of a list. The length of the list increases by one.

```
syntax:
# Adds an object (a number, a string or a
# another list) at the end of my_list
my_list.append(object)
```

```
my_list = ['geeks', 'for']
my_list.append('geeks')
print my_list
```

Output:

```
['geeks', 'for', 'geeks']
```

NOTE: A list is an object. If you append another list onto a list, the parameter list will be a single object at the end of the list.

```
my_list = ['geeks', 'for', 'geeks']
another_list = [6, 0, 4, 1]
my_list.append(another_list)
print my_list
```

Output:

```
['geeks', 'for', 'geeks', [6, 0, 4, 1]]
```

Tuple

Tuples are used to store multiple items in a single variable.

Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage.

A tuple is a collection which is ordered and **unchangeable**.

Tuples are written with round brackets.

Example

Create a Tuple:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

1. Printing a empty list:
   a=list()
   print(a)
   Output: []

2. Check index:
   a=[10, 20, 30, 40]
   c=a[1:2]
   print(c)
   Output: 20

3. Reverse a string
   L = [1,2,3]
   print(list(reversed(L))
   Output: [3,2,1]

4. Check any number
   L = [1,2,3]
   print(5 in L)
   Output: False

5. Sort the list
   L= [1,3,5,2,7,8]
   print(sorted(L))
   Output: [1,2,3,5,7,8]

6. Value at a index:
   L= [1,3,5,2,7,8]
   print(L.index(5))
   Output: 2

7. Pop the value
   L= [1,2,3]
   L.pop()
   Print(L)

Output: [1,2]

8. Copy the list:
   a= [10,20,30,40]
   c=a
   print(c)
   Output: [10,20,30,40]

9. Pick the last index:
   L= [1,2,3]
   print(L[-1])
   Output: 3

10. I= (10,20,30)

    print(I)

    print(I[1:2])

    print(I.index(20))

    del()

    Output:
    (10, 20, 30)
    (20,)
    1

11. List = [['Python ', 'is'], ['Easy']]

    print("\nValue in Multi-Dimensional List: ")

    print(List[0][0], List[1][0][0])

    List = [1, 2, 'Python', 4, 'is', 6, 'Easy']

    print("\nList with the use of Mixed Values: ")

    print(List[1], List[2][2])

    Output:

    Value in Multi-Dimensional List:

    Python E

List with the use of Mixed Values:

2 t

12. **Predict the output: (difference between append, insert and extend)**

```
List = [1,2,3,4]
List.append(12)
print(List)
List.insert(3, 12)
print(List)
List.extend([ 'Bennett', 'University'])
print(List)
```

Output:

[1, 2, 3, 4, 12]

[1, 2, 3, 12, 4, 12]

[1, 2, 3, 12, 4, 12, 'Bennett', 'University']

13. Predict the output

```
List = [1, 2, 3, 9, 5, 6, 4, 7, 8, 7, 10, 11, 12]
List.remove(7)
print(List)
for i in range(3, 5):
   List.remove(i)
print(List)
List = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
List.remove(7)
print(List)
```

List.pop(7)

print(List)

Output:

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

[1, 2, 9, 5, 6, 8, 7, 10, 11, 12]

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

[1, 2, 3, 4, 5, 6, 8, 10, 11, 12]

14. **Predict the output:**

```
round1 = ['chuck norris', 'bruce lee', 'sonny chiba']
round2 = round1.copy()
round1.remove('sonny chiba')
print(round1)
print(round2)
```

Output:

['chuck norris', 'bruce lee']

['chuck norris', 'bruce lee', 'sonny chiba']

15. **Predict the output:**

```
lst = ['python', 'is', 'cool', 'language']
for i in range(len(lst)):
   print(lst[i])
```

Output:
python
is
cool
language

16. Create a matrix and display the elements of it, using python program.

Solution:

```
r=int(input("Enter row size"))
```

```
c=int(input("Enter column size"))
m=[]
for i in range(r):
    n=[]
    print("Enter elements in row")
    for j in range(c):
        l=int(input())
        k=[l]
        n.extend(k)
    m.append(n)
    print("Matrix is" ,m)
```

17. **A list a provided to you, print a sub-list of a given range using python Program:**

```
r=int(input("Enter list size"))
m=[]
for i in range(r):
    print("Enter elements in list")
    l=input()
    k=[l]
    m.extend(k)
print("The input list" ,m)
s_i=int(input("enter start index"))
e_i=int(input("enter start index"))
if((s_i>0 and e_i<=r)):
   sub=m[s_i-1:e_i-1]
print("Sublist is:",sub)
```

18. Write a program to pack and unpack a tuple.

```
x=(10,20,30,40) #Tuple packing
print("Type of x:",type(x))
print("Values of x:", x)
(v1,v2,v3,v4)=x #Tuple unpacking
print("Value1:",v1)
print("Value2:",v2)
```

```
print("Value3:",v3)
print("Value4:",v4)
```