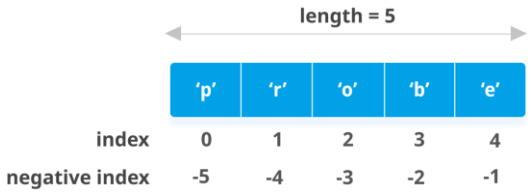
LIST

Definition of List: A List in python is used to store the sequence of various data types.

- Lists are **MUTABLE**—we can modify or change the elements present in the list.
- Elements in list are separated by comma (,) and enclosed by square bracket[ ].
- The elements in the list are identified by index.
- In a list there are n elements then the starting index is 0 and ending index is n-1.
- List has negative index also



- List allows duplication of elements.

Structure of list: list_var=[int,float,"str",bool……]

Ex: lst=[2, 3, 7.53, 3.14, True, 'veena']

- It allows duplicaton of elements.

  Ex: n=[2 ,3 ,7.53 ,3.14 ,True ,2 ,'veena', 3.14]
- Length (len) of list describes number of elements in the list**.**

```
n=[2 ,3 ,7.53 ,3.14 ,True ,2 ,'veena', 3.14]
print(len(n))
```

output: 8

Extract particular element in the list:

```
lst=[2,3,"veena",True,"grace",9.31,2,3.14]
print("lst[4]:",lst[4])   # positive index
print("lst[6]:",lst[6])
print("lst[-1]:",lst[-1]) # negative index
print("lst[-5]:",lst[-5])
```

output:

lst[4]: grace

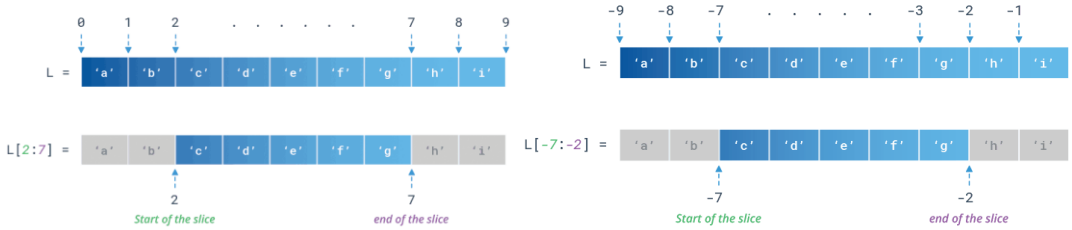
lst[6]: 2

lst[-1]: 3.14

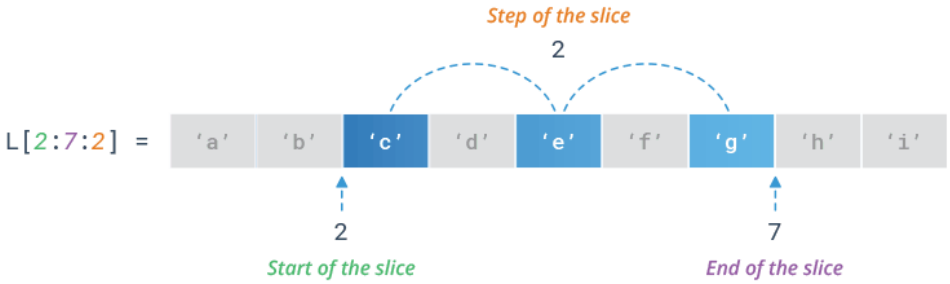
lst[-5]: True

SLICING: To access a range of elements in a list

Syntax: List_var[start:stop:step]



Positive index slicing **Negative index slicing**



Slicing with step size 2

Ex: lst=[2, "vinni", True, "rina", 9.31, 2, 3.14]

Print(lst[0:7:2])

Output: [2, True, 9.31, 3.14]

Nested list: we can create list within the list

Ex: lst=[2, "vinni", True, "rina", 9.31, 2, 3.14, [1,3,5,7,9]]

List methods:

- **Append :** adding new element in the last of the list.

  Syntax: lst.append("new element")

- **Extend:** we can add more than one element as a list

  Syntax: lst.extend([new_element1,new_element2,…..])

Difference b/w append and extend: When we want to add more than one value

- by using append it will add as another list
- By using extend it will add as different elements

- **Copy:** it will copy the list and returns it, after copy function we can't modify the list

  Syntax: lst.copy( )

- **Clear:** it removes all elements from the list

  Syntax: lst.clear( )

- **Count:** it will count the repeated elements in the list

  Syntax: lst.count(repeated element in the list)

- **Index:** it will print the index value of the particular element

  Syntax: lst.index(any element in the list)

- **Insert:** add an element at desired position

  Syntax: lst.insert(index,'new_element')

- **Pop:** delete element in a particular index

  Syntax: lst.pop(index number)

- **Remove:** delete particular element in the list

  Syntax: lst.remove(any element in the list)

- **Reverse:** it returns the total list in reverse order

  Syntax: lst.reverse( )

- **Sort:** it will arrange the elements in the specific order(ascending to decending or decending to ascending)

  Syntax: lst.sort( )

List comprehension: x for x in range( )

```
list=[1,2,3,4,5,6,7,8,9,10]
list=["EVEN" if i%2==0 else "ODD" for i in range(10)]
```