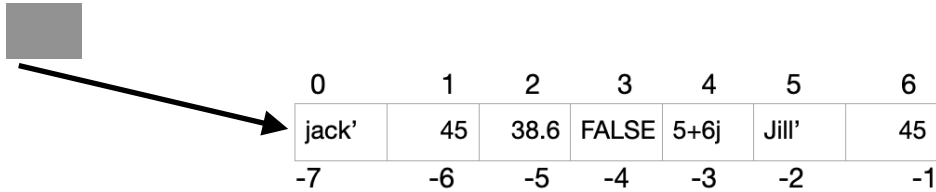


## Tuple Iteration and Operators

**Iteration:** traversing and visiting all the elements of a tuple .

t1=('jack', 45, 38.6, false, 5+6j,'Jill', 45)#this is a tuple

t1



It looks like this in memory.

Iteration can be done using for loop

Creating a tuple using for loop

```
t1=('jack', 45, 38.6, False, 5+6j, 'jill', 45)
t1
('jack', 45, 38.6, False, (5+6j), 'jill', 45)
for x in t1:
    print(x)

jack
45
38.6
False
(5+6j)
jill
45
```

```
>>> for i in range(len(t1)):
      print(t1[i])

Jack
45
38.6
False
(5+6j)
Jill
45
>>>
```

This can be done using while loop also

## Operators in tuple

[]	Index
[:]	Slicing
+	Concatenation
*	Repeat
In	Membership
Not in	Membership

```
>>>
>>> t1 = ('Jack', 45, 38.6, False, 5+6j, 'Jill', 45)
>>>
>>> t1[1]
45
>>> t1[4]
(5+6j)
>>> t1[4]=25
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    t1[4]=25
TypeError: 'tuple' object does not support item assignment
>>>
>>> t1[-1]
45
>>> t1[-2]
'Jill'
>>> t1[:]
('Jack', 45, 38.6, False, (5+6j), 'Jill', 45)
>>> t1[3:]
(False, (5+6j), 'Jill', 45)
>>> t1[3:len(t1):-1]
0
>>> t1[-3:-len(t1):-1]
((5+6j), False, 38.6, 45)
>>> t1[::-1]
(45, 'Jill', (5+6j), False, 38.6, 45, 'Jack')
>>> t1[::2]
('Jack', 38.6, (5+6j), 45)
>>>
>>>
>>>
>>>
>>> |
...
>>> t1=(1,2,3)
>>> t2=(4,5,6)
>>>
>>> t1+t2
(1, 2, 3, 4, 5, 6)
>>> t1
(1, 2, 3)
>>> t2
(4, 5, 6)
>>> t1 + [4,5,6]
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    t1 + [4,5,6]
TypeError: can only concatenate tuple (not "list") to tuple
>>>
>>> t1 + tuple([4,5,6])
(1, 2, 3, 4, 5, 6)
>>> t1 * 2
(1, 2, 3, 1, 2, 3)
>>> t1 * 3
(1, 2, 3, 1, 2, 3, 1, 2, 3)
>>> temp = t1*3
>>> temp
(1, 2, 3, 1, 2, 3, 1, 2, 3)
>>> 3 in t1
True
>>> 5
```