

ex1

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
([known limitations](#))

```
1 array = [5,7,9,11,13]
2 print(array)
3 array_Length=len(array)
4 print(array_Length)
5 ans = sum(array)
→ 6 print("Sum of array ", ans)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

<< First < Prev Next > Last >>

[Customize visualization](#)

Done running (6 steps)

Print output (drag lower right corner to resize)

```
[5, 7, 9, 11, 13]
5
Sum of array 45
```

Frames

Global frame	
array	[5, 7, 9, 11, 13]
array_Length	5
ans	45

Objects

list					
0	1	2	3	4	
5	7	9	11	13	

ex2

Python Tutor: visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
([known limitations](#))

```
10 array_Length = len(array_A)
11 print("Length:",end=" ")
12 print(array_Length)
13 print("Find index of 7 in 2 arrays:",end=" ")
14 #array_A
15 LocationA = array_A.index(7)
16 LocationB = array_A.index(2)
17 print(LocationA,LocationB)
18
19 array_A.sort()
20 print("Sort 3rd Array :",array_A)
21
22 for i in array_A.copy():
23     if i is 7:
24         array_A.remove(i)
25     print("Remove 7 ->",array_A)
26
27 array_F = array_A.copy()
28 print("4th Array :",array_F)
29 array_F.reverse()
30 print("Reverse : ",array_F)
```

[Edit this code](#)

Print output (drag lower right corner to resize)

ARRAY_A : [7, 5, 10, 14, 3, 9, 7]
ARRAY_B : [9, 10, 3, 4, 2, 5, 7, 1]

Print both arrays : [7, 5, 10, 14, 3, 9, 7, 9, 10, 3, 4, 2, 5, 7, 1]
Length: 15
Find index of 7 in 2 arrays: 0 11
Sort 3rd Array : [1, 2, 3, 3, 4, 5, 5, 7, 7, 9, 9, 10, 10, 14]
Remove 7 -> [1, 2, 3, 3, 4, 5, 5, 9, 9, 10, 10, 14]
4th Array : [1, 2, 3, 3, 4, 5, 5, 9, 9, 10, 10, 14]
Reverse : [14, 10, 10, 9, 9, 5, 5, 4, 3, 3, 2, 1]
Array 3rd : [1, 2, 3, 3, 4, 5, 5, 9, 9, 10, 10, 14] Array 4th : [14, 10, 10, 9, 9, 5, 5, 4, 3, 3, 2, 1]

Frames

Global frame
array_A
array_B
array_Length
LocationA
LocationB
i
array_F

Objects

list
0 1 2 3 4 5 6 7 8 9 10 11
1 2 3 3 4 5 5 9 9 10 10 14

list
0 1 2 3 4 5 6 7
9 10 3 4 2 5 7 1

list
0 1 2 3 4 5 6 7 8 9 10 11
14 10 10 9 9 5 5 4 3 3 2 1

→ line that just executed
→ next line to execute

<< First < Prev Next > Last >>

ex3

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
([known limitations](#))

```
1 arr = [1630902151,"piraya",60]
2 print(arr)
3 arr_length = len(arr)
4 print("Length >",arr_length)
5 locaP = arr.index("piraya")
6 print(locaP)
7 arr.append("Status")
→ 8 print(arr)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Done running (8 steps)

[Customize visualization](#)

Print output (drag lower right corner to resize)

```
[1630902151, 'piraya', 60]
Length > 3
1
[1630902151, 'piraya', 60, 'Status']
```

Frames

Global frame	
arr	→
arr_length	3
locaP	1

Objects

list				
0	1	2	3	
1630902151	"piraya"	60	"Status"	

ex4

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
([known limitations](#))

```
1 arr = [[1630902151,"piraya"],[60,"Status"]]
2 print(arr)
3 arr_length = len(arr)
4 print("Length >",arr_length)
5
6 arr_a = ["Rubber",0,"Out of stock"]
7 arr_a.insert(2,arr)
8 print(arr_a)
9 arr_b = ["Ruler",5,"In stock"]
10 arr_b.insert(2,arr)
11 print(arr_b)
12 arr_c = ["Pencil",1,"In stock"]
13 arr_c.insert(2,arr)
14 print(arr_c)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Done running (13 steps)

[Customize visualization](#)

Print output (drag lower right corner to resize)

```
[[1630902151, 'piraya'], [60, 'Status']]
Length > 2
['Rubber', 0, [[1630902151, 'piraya'], [60, 'Status']], 'Out of stock']
['Ruler', 5, [[1630902151, 'piraya'], [60, 'Status']], 'In stock']
['Pencil', 1, [[1630902151, 'piraya'], [60, 'Status']], 'In stock']
```

Frames Objects

Global frame

arr	→
arr_length	2
arr_a	→
arr_b	→
arr_c	→

list

0	1
1630902151	"piraya"

list

0	1
60	"Status"

list

0	1	2	3
"Rubber"	0	"Out of stock"	

list

0	1	2	3
"Ruler"	5	"In stock"	

list

0	1	2	3
"Pencil"	1	"In stock"	

ex5

← → ↺ pythontutor.com/render.html#mode=display

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
([known limitations](#))

```
5
6 arr_a = ["Rubber",0,"Out of stock"]
7 arr_a.insert(2,arr)
8 print(arr_a)
9
10 arr_b = ["Ruler",5,"In stock"]
11 arr_b.insert(2,arr)
12 print(arr_b)
13 arr_c = ["Pencil",1,"In stock"]
14 arr_c.insert(2,arr)
15 print(arr_c)
16
17 arr_d = ["Pen",10,"In stock"]
18 arr_d.insert(2,arr)
19 print(arr_d)
20 arr_e = ["Colour pencil",5,"In stock"]
21 arr_e.insert(2,arr)
22 print(arr_e)
23 arr_f = ["A4 Paper",0,"Out of stock"]
24 arr_f.insert(2,arr)
25 print(arr_f)
```

[Edit this code](#)

→ line that just executed

→ next line to execute

<< First < Prev Next > Last >>

Done running (22 steps)

[Customize visualization](#)

Print output (drag lower right corner to resize)

```
[[1630902151, 'piraya'], [60, 'Status']]
Length > 2
['Rubber', 0, [[1630902151, 'piraya'], [60, 'Status']], 'Out of stock']
['Ruler', 5, [[1630902151, 'piraya'], [60, 'Status']], 'In stock']
['Pencil', 1, [[1630902151, 'piraya'], [60, 'Status']], 'In stock']
['Pen', 10, [[1630902151, 'piraya'], [60, 'Status']], 'In stock']
['Colour pencil', 5, [[1630902151, 'piraya'], [60, 'Status']], 'In stock']
['A4 Paper', 0, [[1630902151, 'piraya'], [60, 'Status']], 'Out of stock']
```

