

พัฒนพงศ์ 6670774 1630902191

2.1

2.1.1

Bubble sort

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

```
Python 3.6
(known limitations)
1 def printlist(arr):
2     for i in range(len(arr)):
3         print(arr[i], end=" ")
4     print()
5
6 def bubbleSort(arr):
7     for i in range(len(arr)):
8         for j in range(0, len(arr)-i-1):
9             if arr[j] > arr[j+1]:
10                temp = arr[j]
11                arr[j] = arr[j+1]
12                arr[j+1] = temp
13
14
15 data = [1,25,-1,0,6]
16 arr = [1,25,-1,0,6]
17 print("Array :",end=" ")
18 printlist(arr)
19 bubbleSort(data)
20 print("Sorted Array :",end=" ")
21 print(data)
```

[Edit this code](#)

it just executed

Print output (drag lower right corner to resize)

Array : 1 25 -1 0 6  
Sorted Array : [-1, 0, 1, 6, 25]

Frames

Global frame

- printlist
- bubbleSort
- data
- arr

Objects

- function printlist(arr)
- function bubbleSort(arr)
- list
- list

0	1	2	3	4
-1	0	1	6	25

0	1	2	3	4
1	25	-1	0	6

code%

<https://onlinegdb.com/dcTkL6DRD>

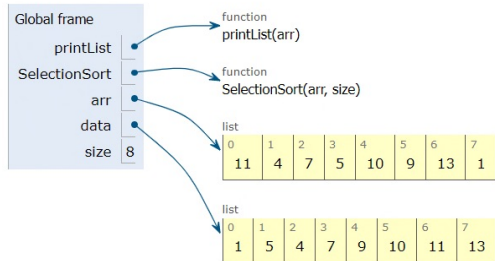
# Selection Sort

```
Python 3.6 (known limitations)
1 #selection sort
2 def printList(arr):
3     for i in range(len(arr)):
4         print(arr[i], end=" ")
5
6 def SelectionSort(arr,size):
7     for step in range(size):
8         min_index = step
9         for i in range(step + 1, size):
10            if arr[i] < arr[min_index]:
11                min_index = i
12            (arr[step], arr[min_index]) = (arr[min_index], arr[step])
13 arr = [11,4,7,5,10,9,13,1]
14 data = [11,4,7,5,10,9,13,1]
15 size = len(data)
16 print("Array :",end=" ")
17 printList(arr)
18 print()
19 SelectionSort(data,size)
20 print("Selection Sort Array :",end=" ")
21 print(data)
```

Print output (drag lower right corner to resize)

Array : 11 4 7 5 10 9 13 1  
Selection Sort Array : [1, 5, 4, 7, 9, 10, 11, 13]

Frames Objects



code: <https://onlinegdb.com/L3eHQD8SY>

# Insertion Sort

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

```
Python 3.6 (known limitations)
8 def insertionSort(arr):
9
10     for step in range(1, len(arr)):
11         key = arr[step]
12         j = step - 1
13
14         while j >= 0 and key < arr[j]:
15             arr[j + 1] = arr[j]
16             j = j - 1
17
18         arr[j + 1] = key
19
20
21 data = [1,25,-1,0,6]
22 arr = [1,25,-1,0,6]
23 print("Array :",end=" ")
24 printList(arr)
25
26 insertionSort(data)
27 print("Insertion sort :",end=" ")
28 print(data)
```

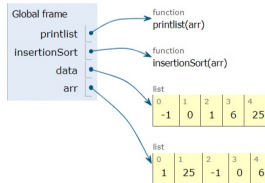
[Edit this code](#)

→ line that just executed

Print output (drag lower right corner to resize)

Array : 1 25 -1 0 6  
Insertion sort : [-1, 0, 1, 6, 25]

Frames Objects



code: <https://onlinegdb.com/F5XTaEltg>

## 2.2 Explanation

2.2.1 From all above method which one is the best in term of performance?  
Ans: Selection sort

2.2.2 From all above method which one is the best in term of resource management?  
Ans: Insertion sort

2.2.3 From all above method which one is the best in your opinion?  
Ans: Bubble sort