

# Implementation of Selection Sort

The selection sort improves on the bubble sort by making only one exchange for every pass through the list. In order to do this, a selection sort looks for the largest value as it makes a pass and, after completing the pass, places it in the proper location. As with a bubble sort, after the first pass, the largest item is in the correct place. After the second pass, the next largest is in place. This process continues and requires  $n-1$  passes to sort  $n$  items, since the final item must be in place after the  $(n-1)$  st pass.

## Resources for Review

Check out the resources below for a review of Selection sort!

- [Wikipedia](#)
- [Visual Algo](#)
- [Animation](#)
- [Sorting Algorithms Animcation with Pseudocode](#)

```
In [1]: def selection_sort(arr):  
  
    # For every slot in array  
    for fillslot in range(len(arr)-1,0,-1):  
        positionOfMax=0  
  
        # For every set of 0 to fillslot+1  
        for location in range(1,fillslot+1):  
            # Set maximum's Location  
            if arr[location]>arr[positionOfMax]:  
                positionOfMax = location  
  
        temp = arr[fillslot]  
        arr[fillslot] = arr[positionOfMax]  
        arr[positionOfMax] = temp
```

```
In [2]: arr = [3,5,2,7,6,8,12,40,21]  
        selection_sort(arr)  
        arr
```

```
Out[2]: [2, 3, 5, 6, 7, 8, 12, 21, 40]
```

## Good Job!