

## 365 DataScience Insertion sort in Python

### Step 1 Create a function that performs an insertion sort on a list

*# Insertion sort relies on the assumption that the sub-list to the left of the current item is sorted.*

```
def insertion_sort(my_list):
    # Create a variable storing the length of the list
    n = len(my_list)

    # This first iteration starts from the second item (index 1) and goes all the way to the end of the list.
    for i in range(1,n):

        # Create a variable storing the value of the current item.
        value = my_list[i]

        # The while-loop that we will implement shortly will start looping from the index of the current item,
        # so we store it in a separate variable.
        # This is the index that will keep track of the items to our left.
        j = i

        # While this index is larger than 0 and while the value to the left is larger than the current value,
        # perform the operations in the body of the loop.
        while j > 0 and my_list[j-1] > value:

            # Swap the current item with the one to the left.
            my_list[j] = my_list[j-1]

            # Decrease j by one
            j = j - 1

        # Update the value sitting at index j with the value that we're on.
        my_list[j] = value

    # Once all loops have been exhausted, return the sorted list.
    return my_list
```

### Step 2 Check if the function performs as expected

*# Create a list of numbers*

```
test = [6, 5, 8, 2, 3, 45, 87, 24, 70]
```

```
# Sort the List using insertion sort  
insertion_sort(test)
```