Selection_sort :

This code implements the selection sort algorithm to sort a given list `arr` in ascending order. Here's an explanation of the code:

1. `selection_sort(arr)`: This function takes a list `arr` as input and sorts it using the selection sort algorithm.

2. `n = len(arr)`: Gets the length of the input list `arr`.

3. The outer loop `for i in range(n)` iterates over each element of the list, starting from the first element.

4. Inside the outer loop, the inner loop `for j in range(i+1, n)` is used to find the index of the minimum element in the unsorted part of the list. It starts from `i+1` because elements before index `i` are already sorted.

5. `min_idx = i`: Initializes `min_idx` to `i`, assuming the minimum element is at index `i`.

6. If `arr[j]` is less than `arr[min_idx]`, `min_idx` is updated to `j`, which means `arr[j]` is the new minimum element in the unsorted part of the list.

7. After finding the index of the minimum element, the values at index `i` and `min_idx` are swapped using tuple unpacking and simultaneous assignment (`arr[i], arr[min_idx] = arr[min_idx], arr[i]`).

8. The outer loop continues until all elements are sorted, and the sorted list is returned.

9. `Example usage`: An example list `arr = [64, 25, 12, 22, 11]` is used, and the `selection_sort` function is called to sort it. The sorted array is then printed.

When you run this code, it will output the sorted array `[11, 12, 22, 25, 64]`, demonstrating the selection sort algorithm's functionality.