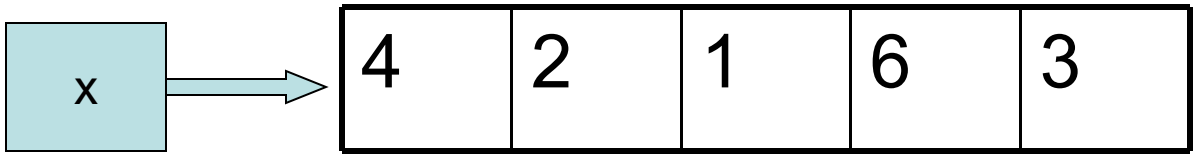
Bubble Sort

- A bubble sort accomplishes this by swapping values with the values next to each other and repeating until the values are in order

Problem

- For the following array



- Arrange the elements in the array from smallest to largest

Round 1

- {4, 2, 1, 6, 3}
  - Swap index 0 and 1 in the array if first number is larger
- {2, 4, 1, 6, 3}
  - Swap index 1 and 2 in the array if first number is larger
- {2, 1, 4, 6, 3}
  - Swap index 2 and 3 in the array if first number is larger
- {2, 1, 4, 6, 3}
  - Swap index 3 and 4 in the array if first number is larger
- {2, 1, 4, 3, 6}
  - We have reached the end of the array. We now repeat this process over again until we make no more swaps

Round 2

- {2, 1, 4, 3, 6}
  - Swap index 0 and 1 in the array if first number is larger
- {1, 2, 4, 3, 6}
  - Swap index 1 and 2 in the array if first number is larger
- {1, 2, 4, 3, 6}
  - Swap index 2 and 3 in the array if first number is larger
- {1, 2, 3, 4, 6}
  - Swap index 3 and 4 in the array if first number is larger

  - We have reached the end of the array. We now repeat this process over again until we make no more swaps

Round 3

- {1, 2, 3, 4, 6}
  - Swap index 0 and 1 in the array if first number is larger
- {1, 2, 3, 4, 6}
  - Swap index 1 and 2 in the array if first number is larger
- {1, 2, 3, 4, 6}
  - Swap index 2 and 3 in the array if first number is larger
- {1, 2, 3, 4, 6}
  - Swap index 3 and 4 in the array if first number is larger

  - Since no swaps were made, we know that the list is now sorted from smallest to largest

On the computer

- Create a variable of type boolean to keep track of whether the a swap was made in the round
- Use a conditional loop to loop through the rounds until no swap has been made
  - For each round, loop through the array comparing the current index in the array to the index above
  - If the current index is larger, swap the two values (create a temporary variable to hold a value while you swap)

```
public static void bubbleSort(String[] list) {
        boolean sorted = false;
        for (int top = list.length - 1; top > 0 && !sorted; top--) {
                sorted = true;
                for (int i = 0; i < top; i++)
                        if (list[i].compareTo(list[i + 1]) > 0) {
                                sorted = false;
                                String temp = list[i];
                                list[i] = list[i + 1];
                                list[i + 1] = temp;
                        }
        }
}
```