Step One: We define a function called bubbleSort with a parameter called arr:

```
function bubbleSort(arr) {

  return arr;
}

bubbleSort([14, 33, 27, 35, 10, 56, 101]);
```

...it will return for us arr.

...the arr is an array of numbers.

Step Two: We add a statement to keep the loop constant. It will retrieve the array length:

```
function bubbleSort(arr) {
    const loop = arr.length;
  return arr;
}

bubbleSort([14, 33, 27, 35, 10, 56, 101]);
```

...this will tell us there are 7 items in this array.

Step Three: We add a for loop:

```
function bubbleSort(arr) {
    //loop length
    const loop = arr.length;

    //loop for loop length
    for(let i = 0; i < loop; i++) {

    }

  return arr;
}

bubbleSort([14, 33, 27, 35, 10, 56, 101]);
```

Step Four: We want to cycle through the array items:

So this will be another for loop:

```
function bubbleSort(arr) {
    //loop length
    const loop = arr.length;

    //loop for loop length
    for(let i = 0; i < loop; i++) {
      //cycle through arr items
      for(let j = 0; j < loop; j++) {

      }
    }

  return arr;
}

bubbleSort([14, 33, 27, 35, 10, 56, 101]);
```

...the reason for this is we compare 2 items with each other.

Step Five: We compare the 2 items with each other:

```
function bubbleSort(arr) {
    //loop length
    const loop = arr.length;

    //loop for loop length
    for(let i = 0; i < loop; i++) {
      //cycle through arr items
      for(let j = 0; j < loop; j++) {
        //compare adjacent items
        if(arr[j] > arr[j+1]) {
          let temp = arr[j];
          arr[j] = arr[j+1];
          arr[j+1] = temp;
        }
      }
    }

  return arr;
}

bubbleSort([14, 33, 27, 35, 10, 56, 101]);
```

...basically, it's saying, "if j is bigger than the item next to it (ie, j+1), let the temp be j."

For example, let's say:

```
33 = arr[j];
27= arr[j+1];
33 = temp;
```

33 is j, 27 is j+1... therefore temp will be 33... simply put: we are SWAPPING the 2 items.

So far, this is what we have:

```
function bubbleSort(arr) {
    //loop length
    const loop = arr.length;

    //loop for loop length
    for(let i = 0; i < loop; i++) {
      //cycle through arr items
      for(let j = 0; j < loop; j++) {
        //compare adjacent items
        if(arr[j] > arr[j+1]) {
          //swap
          let temp = arr[j];
          arr[j] = arr[j+1];
          arr[j+1] = temp;
        }
      }
    }

  return arr;
}

bubbleSort([14, 33, 27, 35, 10, 56, 101]);
```

If we check our console log at this point:

```
function bubbleSort(arr) {
    //loop length
    const loop = arr.length;
    //loop for loop length
    for(let i = 0; i < loop; i++) {
      //cycle through arr items
      for(let j = 0; j < loop; j++) {
        //compare adjacent items
        if(arr[j] > arr[j+1]) {
          //swap
          let temp = arr[j];
          arr[j] = arr[j+1];
          arr[j+1] = temp;
        }

        console.log(arr)
      }
    }

  return arr;
}

bubbleSort([14, 33, 27, 35, 10, 56, 101]);
```

...this is what we see... the array being sorted:

```
[ 14, 27, 10, 33, 35, 56, 101 ]
[ 14, 27, 10, 33, 35, 56, 101 ]
[ 14, 27, 10, 33, 35, 56, 101 ]
[ 14, 10, 27, 33, 35, 56, 101 ]
[ 14, 10, 27, 33, 35, 56, 101 ]
[ 14, 10, 27, 33, 35, 56, 101 ]
[ 14, 10, 27, 33, 35, 56, 101 ]
[ 14, 10, 27, 33, 35, 56, 101 ]
[ 14, 10, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
[ 10, 14, 27, 33, 35, 56, 101 ]
=> [ 10, 14, 27, 33, 35, 56, 101 ]
```