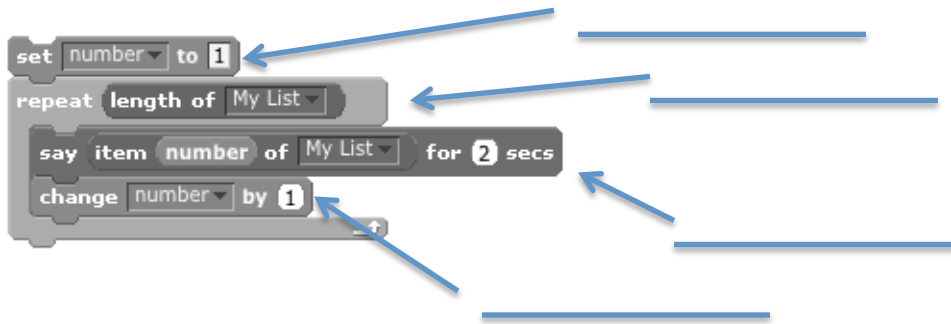


## Iteration In a Nutshell

1. Iteration is a short and easy way for doing the same action on every item in a list. There are four parts:

- Initialization--This tells you where to start, by setting a variable to the first item.
- Loop--This runs the Action and Increment multiple times.
- Action—This is the thing you wanted done with each element of the list.
- Increment—This moves the variable to be the number of the next item.

2. Identify the four parts of iteration:



3. In your own words, what happens when the part listed is missing from iteration?

<p>No Initialization</p> <p>_____</p> <p>_____</p>	<p>No Loop</p> <p>_____</p> <p>_____</p>
<p>No Action</p> <p>_____</p> <p>_____</p>	<p>No Increment</p> <p>_____</p> <p>_____</p>

4. The action can be much than just “say”. One example is an accumulation. The blocks below sum a list.

The code consists of the following blocks:

- set number to 1
- set sum to 0
- repeat length of My List
  - change sum by item number of My List
  - change number by 1
- say sum for 2 secs

Annotations with arrows:

- Just as if we were adding it ourselves, we need to remember a running total. (points to 'set sum to 0')
- Note the changed Action. (points to 'change sum by item number of My List')
- Because the action doesn't say anymore, we have to say the answer at the end. (points to 'say sum for 2 secs')

5. Write blocks to say the smallest number in a list.

6. Write blocks that find any items named “marco” in a list, and replace it with “polo”.

7. What does the following print out?

The code consists of the following blocks:

- think What is 9 times 6? for 2 secs
- set count to 1
- set Answer to 0
- repeat until count > length of myList
  - change Answer by item count of myList
  - change count by 3
- say join The Answer is Answer for 2 secs

The list object **myList** contains the following items:

Index	Value
1	17
2	29
3	14
4	25
5	12

myList length: 5