Loop Intuition

The concept of looping is somewhat new to the beginners, so I will try to make it as easily understandable as possible.

In general programming, we need to do a thing repeatedly to reach a result.

Let's see an example, say we want to add 10 numbers from 1 - 10. Easy right? We just go like this:

1+2+3+4+5+6+7+8+9+10 = 55 Easy right?

If we are asked to do this with our basic programming knowledge, we would do the same thing, we would just add the numbers in java and print them out:

```
int totalSum = 0;
totalSum = 1+2+3+4+5+6+7+8+9+10;
System.out.println(totalSum);
```

But is it a really good way of adding such numbers. Now instead of just 10 numbers, if we want to add a a 100 number or a 1000 numbers together, then this suddenly becomes a really herculin task to do.

So we use the concept of looping, where we repeatedly do a certain thing again and again till we hit a stopping condition or a stopping criteria.

Now what is this **stopping criteria** we are talking about? This in fact is a very important **condition** or **criteria** for loops

In the above example, we stop when we have added our last number **10**. Now in loops we apply such similar approach, where we stop our **loops** with such stopping criteria.

so The gist we get is , so we can add as many numbers as we like by using loops in the following way:

```
loop condition{
    body of loop to do something
}
```

So we get the basic intuition that we can loop over and over till we reach the stopping criteria that would end the loop and inside the loop, we have the actions that we perform or take.

In the next tutorial we will see how loops are written and we would see this with a couple of examples.