A “Loop”

- A simple but powerful mechanism for “making lots of things happen!”
- Performs a statement (or block) over & over
- Usually set up to repeat an action until some condition is satisfied
- Computing Scenarios Examples
  - Run an application until user hits “quit” button
  - Deal card hands until game over
Syntax of the **while** statement

while (condition){
    statement(s)
}

- **condition** is a true/false (boolean) expression
- If **condition** is initially false, the statement is never executed
- If **condition** is true, **statement** is executed and **condition** is re-evaluated
- The **statement** should eventually make the loop stop

*more Infinite Loops*

// Some infinite loops are intentional
boolean notQuitKey = true;
while (notQuitKey){
    statement(s)
}

// Others are not
int x = 5;
while (x < 10){
    statement(s) which don't change x
}

Note: if the beo-bot needs to continuously read/output something then it will be in a
while(true){ } loop
- To stop the action simply bring the switch to 0 position

*a while Loop to Print Numbers*

// Print the numbers 1 thru 10
int x = 1;
while (x <= 10){
    System.out.println(x);
    x = x + 1;
}

- What happens if you forget the statement x = x + 1 ?
- We print value 1 forever
- Known as **infinite** loop

*Compute Square of first 5 numbers*

// In Square.java
int num = 1;
int sqNum = 0;
while (num <= 5){
    sqNum = num * num;
    System.out.println(num + "  " + sqNum);
    num = num + 1;
}
**For Loop**

```java
for (init; end-test; re-init){
    statement
}
```

- Executes loop body (statements within `{}`) as long as `end-test` evaluates to TRUE
- Initialization and re-initialization code included in loop statement
- Note: Test is evaluated **before** executing loop body

**While vs. For**

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| int x = 1; while (x <= 10){
   System.out.println(x);
   x = x + 1;
} | An example of a while loop that has this pattern |
| for (int x = 1; x <= 10; x = x + 1){
   System.out.println(x);
} | A for loop that does the same thing |

Note: *For* loops are used generally for bounded iteration

**Summary of Loops**

<table>
<thead>
<tr>
<th>Type of Loop</th>
<th>Syntax</th>
</tr>
</thead>
</table>
| while        | while (condition){
   statement(s)
} |
| for          | for (expr1; condition; expr3){
   statement(s)
} |