The Sentential Loop Sandwich

While Loops vs. For Loops

- For Loops are great for “counting” situations – if you know that you need to do something a specific number of times, use a for loop.
- However, if the user gets to pick how many times it happens, use a while loop.
- While loops can easily stop when:
  - The user chooses to quit
  - The game is over
  - There is a tie
  - The user answers correctly
  - The input is valid (no longer an error)

```
char end = 'n';
while (end == 'n')
{
  System.out.println ("Hello");

  end = IO.inputChar ("End? (n/y) ");
}
```

1. Initialize the loop variable

2. Test the stopping condition

3. Steps to repeat

4. Move towards the stopping condition

More about the pieces of a loop!

1. Initialize the loop variable

```
char end = 'n';

while (end == 'n')
{
    System.out.println ("Hello");

    end = IO.inputChar ("End? (n/y) ");
}
```

- Happens ONCE
- Happens before the loop is run
- Declares the loop stopping variable.
- Gives the loop stopping variable a value that makes the loop stopping condition true… so that the loop will run.

More about the pieces of a loop!

```
char end = 'n';

while (end == 'n')
{
    System.out.println ("Hello");

    end = IO.inputChar ("End? (n/y) ");
}
```

2. Test the stopping condition

- Uses the loop stopping variable (end in this case)
- Is also a Boolean Expression
- If the expression is TRUE, the loop runs
- If the expression is FALSE, the loop stops

More about the pieces of a loop!

```
char end = 'n';

while (end == 'n')
{
   System.out.println ("Hello");

   end = IO.inputChar ("End? (n/y) ");
}
```

3. Steps to repeat

- Can be many statements… even an entire game
- Runs over and over again when the loop stopping condition is true
- Inside the { }

More about the pieces of a loop!

```
char end = 'n';

while (end == 'n')
{
   System.out.println ("Hello");

   end = IO.inputChar ("End? (n/y) ");
}
```

4. Move towards the stopping condition

- If this part is missing, the loop will run forever
- This part is where the user has a chance to get out
- The loop stopping variable (end in this case) is changed on this line
- It must be inside the { }

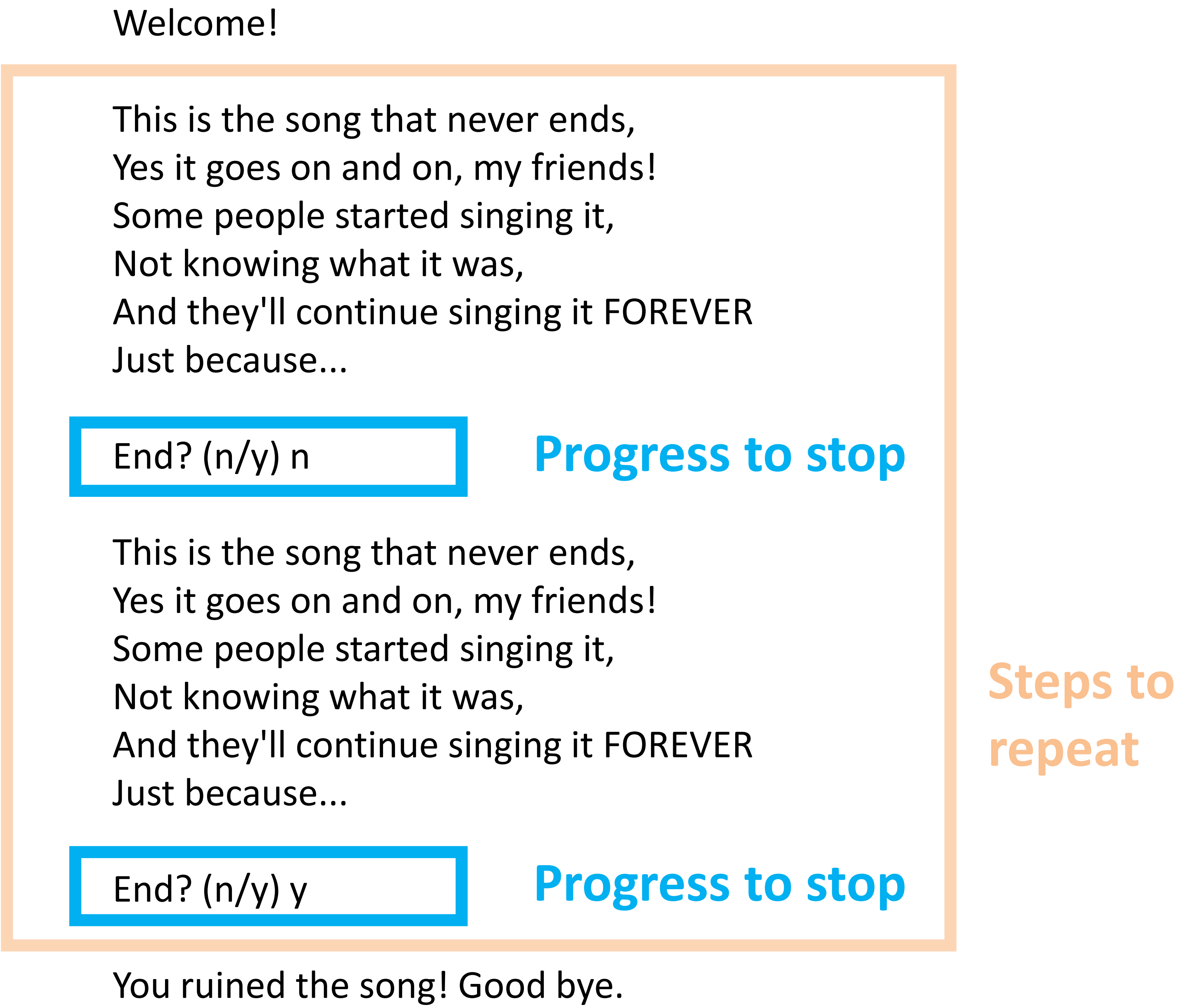
```
System.out.println ("Welcome!\n");

char end = 'n';

while (end == 'n')
{
  System.out.println ("\nThis is the song that never ends,");
  System.out.println ("Yes it goes on and on, my friends!");
  System.out.println ("Some people started singing it,");
  System.out.println ("Not knowing what it was,");
  System.out.println ("And they'll continue singing it FOREVER");
  System.out.println ("Just because...\n");

  end = IO.inputChar ("End? (n/y) ");
}

System.out.println ("\nYou ruined the song! Good bye.");
```

1. Initialize the loop variable

2. Test the stopping condition

4. Steps to repeat

3. Move towards the stopping condition



```
Welcome!

This is the song that never ends,
Yes it goes on and on, my friends!
Some people started singing it,
Not knowing what it was,
And they'll continue singing it FOREVER
Just because...

End? (n/y) n

This is the song that never ends,
Yes it goes on and on, my friends!
Some people started singing it,
Not knowing what it was,
And they'll continue singing it FOREVER
Just because...

End? (n/y) y

You ruined the song! Good bye.
```

Progress to stop

Progress to stop

Steps to repeat

```
System.out.println ("Welcome!\n");
```

Before the loop runs....

```
char end = 'n';

while (end == 'n')
{
   System.out.println ("\nThis is the song that never ends,");
   System.out.println ("Yes it goes on and on, my friends!");
   System.out.println ("Some people started singing it,");
   System.out.println ("Not knowing what it was,");
   System.out.println ("And they'll continue singing it FOREVER");
   System.out.println ("Just because...\n");

   end = IO.inputChar ("End? (n/y) ");
}
```

After the loop runs....

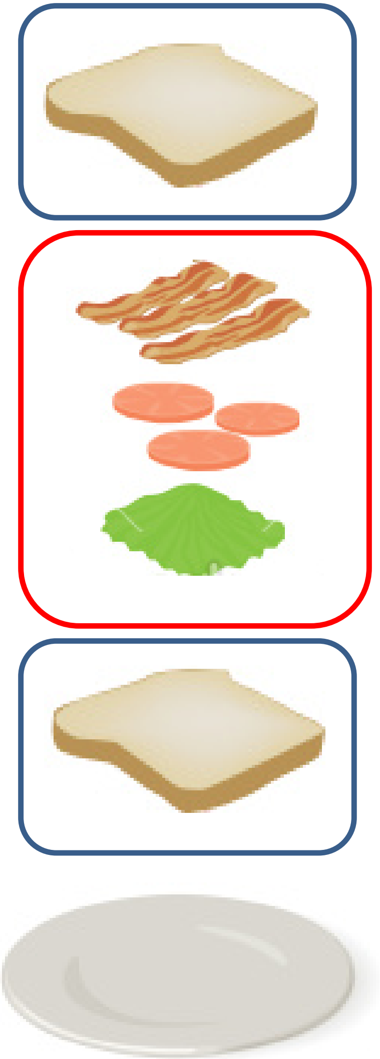
```
System.out.println ("\nYou ruined the song! Good bye.");
```

Before the loop runs....

```
Welcome!

This is the song that never ends,
Yes it goes on and on, my friends!
Some people started singing it,
Not knowing what it was,
And they'll continue singing it FOREVER
Just because...

End? (n/y) n

This is the song that never ends,
Yes it goes on and on, my friends!
Some people started singing it,
Not knowing what it was,
And they'll continue singing it FOREVER
Just because...

End? (n/y) y

You ruined the song! Good bye.
```

After the loop runs....

While you want to continue….
It’s like a loop sandwich around the code!



```
char continu = 'y';
while (continu == 'y')
{

    System.out.println ("\nHi");
    System.out.println ("Hello");
    System.out.println ("Howdy");

    continu = IO.inputChar ("Continue?");
}
```

Consider the Magic 8 Ball.... If you want it to loop, what parts do you want to loop?

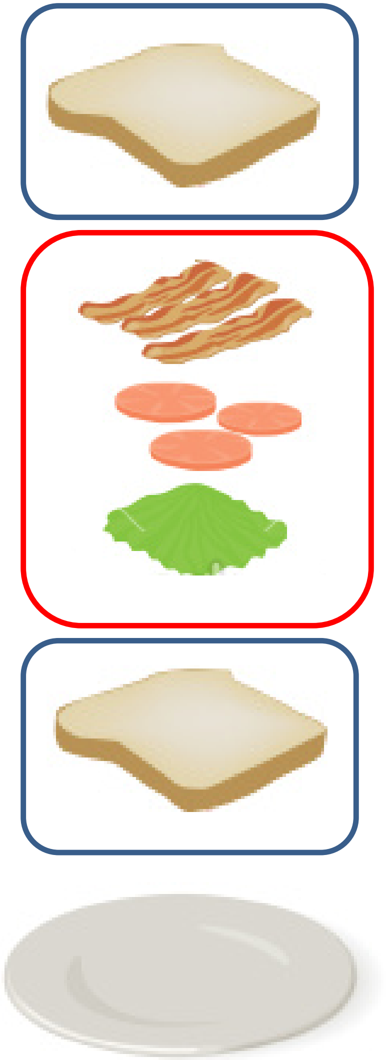
```
System.out.println ("Welcome to the Magic 8 Ball");

String q = IO.inputString ("Enter question: ");
int n = ((int) (Math.random () * 3)) + 1;
if (n == 1)
    System.out.println ("Not going to happen.");
else if (n == 2)
    System.out.println ("It's possible.");
else
    System.out.println ("It is 100% certain.");

System.out.println ("Have a nice day.");
System.out.println ("Welcome to the Magic 8 Ball");
```

Top
Here

```
String q = IO.inputString ("Enter question: ");
int n = ((int) (Math.random () * 3)) + 1;
if (n == 1)
   System.out.println ("Not going to happen.");
else if (n == 2)
   System.out.println ("It's possible.");
else
   System.out.println ("It is 100% certain.");
```

Bottom
Here

```
System.out.println ("Have a nice day.");
```



```
System.out.println ("Welcome to the Magic 8 Ball");
char continu = 'y';
while (continu == 'y')
{

  String q = IO.inputString ("Enter question: ");
  int n = ((int) (Math.random () * 3)) + 1;
  if (n == 1)
     System.out.println ("Not going to happen.");
  else if (n == 2)
     System.out.println ("It's possible.");
  else
     System.out.println ("It is 100% certain.");

  continu = IO.inputChar ("Continue?");
}
System.out.println ("Have a nice day.");
```

Top Here

Bottom Here

```
Welcome to the Magic 8 Ball
Enter question: Will I pass?
It's possible.
Continue? y
Enter question: Is it sunny?
It is 100% certain.
Continue? n
Have a nice day.
```

Compare these two sets of output.

The second one is easier on the user because they have to answer fewer questions.

```
Welcome to the Magic 8 Ball
Enter question (q to quit): Will I pass?
It is 100% certain.
Enter question (q to quit): Is it sunny?
Not going to happen.
Enter question (q to quit): q
Have a nice day.
```

A more clever loop sandwich – only one question:

```
System.out.println ("Welcome to the Magic 8 Ball");
String q = IO.inputString ("Enter question (q to quit): ");
while (!q.equals ("q"))
{

  int n = ((int) (Math.random () * 3)) + 1;
  if (n == 1)
     System.out.println ("Not going to happen.");
  else if (n == 2)
     System.out.println ("It's possible.");
  else
     System.out.println ("It is 100% certain.");

 q = IO.inputString ("Enter question (q to quit): ");
}
System.out.println ("Have a nice day.");
```