What is Iteration?

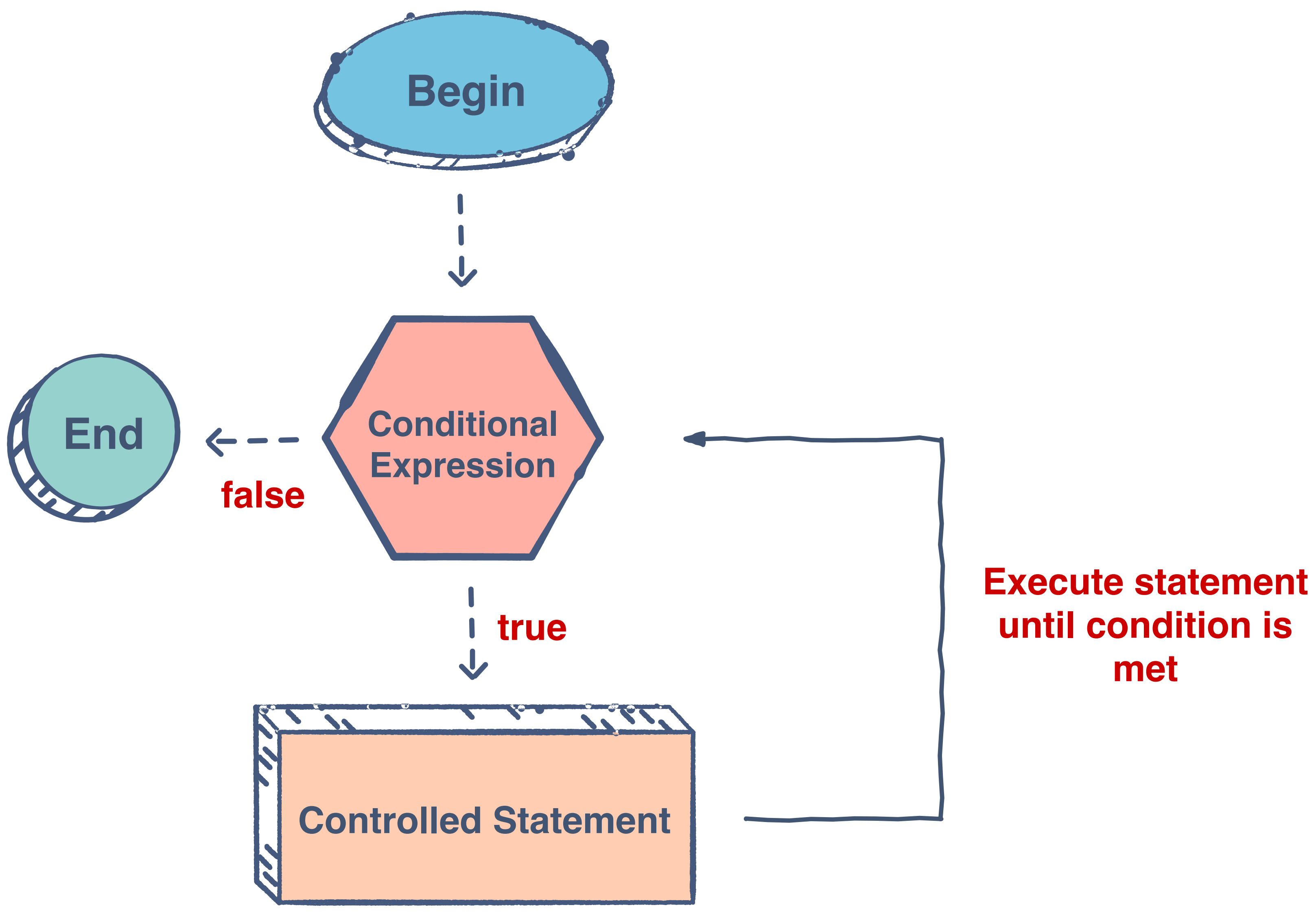
Iterative code is a block of code that runs in a **loop**. In other words, the same code is repeated over and over again. The idea is similar to that of recursion.

As you get comfortable with the recursion concept, you might not use iteration that frequently as you solve your problems, but there are definitely cases where you will find it valuable. This is also one of the easiest ways to ease yourself into recursion because we already understand how to iterate over arrays and lists. Recursion is basically taking the same concept forward but with some modifications.

For example,

- Print a Linked List in reverse order
- Factorial of a number
- Fibonacci series
- A problem where you expect to use a variable number of loops

Conceptual illustration of iteration



Example

The code snippet below provides you with an iterative and recursive code to find the **factorial of a number**.

Iterative | Recursive

```
class MainClass {

    private static int factorialIterative(int n) {

      int factorialResult = 1;

      for (int i = 1; i <= n; i++) {
        factorialResult = factorialResult * i;
      }
      return factorialResult;

    }

    public static void main( String args[] ) {
        int result = factorialIterative(5);
        System.out.println("The factorial is: " + result);
    }
}
```

Run | Save | Reset

In general, any time you want to use a loop, consider replacing that loop with a recursive method.

Now that we have understood the basic concept of an iterative code, let’s move on to the next lesson, which discusses the differences between these two approaches.