

Reading from a Text File

Reading text from a file is simple, but this time we'll use a `File` object to represent the file, a `FileReader` to do the actual reading, and a `BufferedReader` to make the reading more efficient.

The read happens by reading lines in a *while* loop, ending the loop when the result of a `readLine()` is null. That's the most common style for reading data (pretty much anything that's not a `Serialized` object): read stuff in a while loop (actually a while loop *test*), terminating when there's nothing left to read (which we know because the result of whatever read method we're using is null).

A file with two lines of text.

```
What's 2 + 2?/4
What's 20+22/42
```

MyText.txt

```
import java.io.*; Don't forget the import
```

```
class ReadAFile {
    public static void main (String[] args) {
```

```
        try {
            File myFile = new File("MyText.txt");
            FileReader fileReader = new FileReader(myFile);
```

```
            BufferedReader reader = new BufferedReader(fileReader);
```

Make a String variable to hold each line as the line is read

```
            String line = null;
```

```
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
            reader.close();
```

```
        } catch (Exception ex) {
            ex.printStackTrace();
```

```
        }
```

```
    }
```

```
}
```

A FileReader is a connection stream for characters, that connects to a text file

Chain the FileReader to a BufferedReader for more efficient reading. It'll go back to the file to read only when the buffer is empty (because the program has read everything in it).

This says, "Read a line of text, and assign it to the String variable 'line'. While that variable is not null (because there WAS something to read) print out the line that was just read."

Or another way of saying it, "While there are still lines to read, read them and print them."