

A *literal* is a text representation of a value and is used to assign a particular value to a variable in Java code. For example

```
int million = 1000000;  
String name = "Alice";
```

Here, 1000000 is an integer literal representing one million and "Alice" is a string literal representing a particular string that is five characters long.

In string literals certain characters can not be typed because they would be misinterpreted by the reader or the Java compiler. These special characters are represented in Java code as *escape sequences*: a backslash followed by a character. The Java compiler replaces the escape sequence with the special character it represents during compilation.

- 1) In your own words, what do each of the following escape sequences represent? What effect does each have when it appears in a string literal and the string is printed?

```
\n  
\t  
\  
\\
```

- 2) Consider the following small program. Write down what you think it will output, then type it in and run it. Does running it confirm what you thought about \t? If not, can you explain the difference?

```
public class TestProgram {  
    public static void main(String[] args) {  
        System.out.println("\t!");  
        System.out.println(".\t!");  
        System.out.println(".. \t!");  
        System.out.println"... \t!");  
        System.out.println(".... \t!");  
    }  
}
```

- 3) Tab stops on Java systems are typically 4 or 8 characters long. Does the above program tell you what the tab stop size is on your system? Is there any problem with tab stops being system-dependent?
- 4) Go to <https://codestepbystep.com> and create an account if you don't already have one. Follow Practice! > Java > console output. Do the PrintProgram exercise. To get practice using the `\n` escape sequence, use only `System.out.print` instead of `System.out.println` in this exercise.
- 5) Could you use `\t` for the indents? Why or why not?
- 6) If time remains, go to <https://codestepbystep.com>. Follow Practice! > Java > console output. Do the FearTheTree exercise.