String Methods

This is just a quick summary of some commonly-used string methods.

Given:

```
String s = "hello there";
```

1. We can get the length of the string with **length()** . We use the string variable we want to get the length of, followed by the name of the method, in this case, length:

   System.out.println(s.length()); // Outputs 11

2. Each character in the string is indexed with a number starting at zero. The 'h' is at index 0, the 'e' is at index 1, and so forth, up to the last 'e' which is at index 10. We can get a specific character using the **charAt(index)** function:

   s.charAt(1); // The character e

3. **toUpperCase()** and **toLowerCase()** return a new string in all upper or lower case. The original string variable is unchanged!

   System.out.println(s.toUpperCase()); // Outputs HELLO THERE

4. **trim()** returns a new string with any leading or trailing whitespace removed

   s = " hello world ";
   System.out.println(s.trim()); // hello world

5. **equals(otherString)** returns true if the strings are the same, and false if they are not. Don't be tempted to use == with strings.

   if (s.equals("hello there")) {
   }

6. **equalsIgnoreCase(otherString)** compares the strings but is case agnostic.

7. **compareTo(otherString)** returns a number that is positive, zero, or negative depending upon whether one is alphabetically before the other.

8. **substring(index)** returns everything in the string from index to the end of the string.

   System.out.println(s.substring(6)); // “there”

9. **substring(startIndex, endIndex)** returns everything in the string from startIndex, including the startIndex, up to endIndex, but doesn’t include the character at endIndex

   System.out.println(s.substring(0,4)); // “hell”

10. **indexOf(char)** returns the index of the first character that matches char or -1 if not found.

    System.out.println(s.indexOf(‘ ‘)); // 5

11. We can convert a string containing all digits to a int or double:

    int i = Integer.parseInt(“123”); // i =123
    double d = Double.parseDouble(“3.14”); // d = 3.14

12. We can convert an int to a String by just concatenating with a blank string:

    String s = “” + 3; // “3”