

Week 03 Research Assignment

Prompt: Select five methods from the String JavaDocs and describe the following for each:

- *What is the method signature?*
- *What does the method do?*
- *Why would this method be useful (how could you use it)?*

NOTE: *Everything presented here is a synthesis of information from the Javadoc entry on Strings:* (<https://docs.oracle.com/en/java/javase/20/docs/api/java.base/java/lang/String.html>) *and my own experience, except where I have explicitly linked other sources.*

1. compareTo()

Signature: `compareTo(String anotherString)`

What does the method do? `compareTo()` returns an int which represents the difference in unicode character values of each character in the string on which `compareTo()` is being called and the string provided as a parameter. If the strings contain the same characters at the same index positions, but are of different lengths, `compareTo()` will instead return the difference in length between the two strings.

Use Scenario: This method could be useful in an application that takes a large variety of user inputs that could easily be mistyped, misspelled, and so on. In this case, when user input throws an error, the developer may want to then automatically provide the user with suggestions for what they could have meant, taken from a list of accepted inputs that are "close to" what the user typed erroneously. This may not be a good method to use if the user misremembers an input--such as typing "circle" instead of "ellipse"--but could be effective if they type "ellispe" or "elipse" instead of "ellipse", for example. An implementation of something similar to what I'm describing is the "friendly error" system" in the p5js library: <https://www.almichu.com/p5-js-Friendly-Error-System>, though this was written in Javascript, not Java.

2. startsWith()

Signature: `startsWith(String prefix)` -or- `startsWith(String prefix, int toffset)`

What does the method do? `startsWith()` returns a boolean value depending on whether or not the prefix provided as a parameter matches the string on which it is called up til the prefix's length. This method has an additional signature with the added int-type parameter `toffset`. If provided, the method

will test the string for the matching prefix starting at the index provided by this parameter.

Use Scenario: If user input of a date needs to conform to a certain format, `startsWith()` could be used to verify that format before further processing of the input. For example, if the format is mm/dd/yyyy, a developer could use the method to ensure the first character in the date is either "0" or "1" (since there are no months past 12), and could then display an error message to the user if the input doesn't pass this test.

3. `toUpperCase()`

Signature: `toUpperCase()`

What does the method do? `toUpperCase()` converts all lowercase characters in a String to their uppercase counterpart.

Use Scenario: Let's say the developer needs to obtain some user data which needs to conform to a standard format. For example, such standardizing exists for street addresses in the city of Philadelphia, as can be seen here: <https://www.phila.gov/it/PDF/City%20Addressing%20Standards.pdf> (***note: this link may prompt you to download a pdf!***). Though street names in this standard are required to be all uppercase, user input of a Philadelphia street address may not be uppercase, and it seems counter-intuitive to require that kind of formatting from users when street addresses aren't typically written completely in upper case lettering. In this case, it is more user-friendly to automate such formatting through a method like `toUpperCase()` (or its equivalent in whatever programming language the city of Philadelphia uses to handle data storage).

4. `trim()`

Signature: `stripLeading()`

What does the method do? `stripLeading()` removes all leading characters from a string, returning the substring starting at the index of the first non-whitespace character .

Use Scenario: This could be useful when transferring data from sources where it is represented in multiple formats; for example, when writing a program attempting to gauge the frequency of certain words at the beginning of paragraphs across similar sources. One sample from which such a comparison program might draw its data could have a full tab character at the beginning of each paragraph, while another sample might have several spaces, and a third might have simply a new line character. After probably using that information to help identify the first word of each paragraph, the programmer could then use `stripLeading()` before adding it to a frequency count (probably also normalizing the case, etc.). This would help normalize the data without counting strings like:

- " Today" ,

- "Today", or
- "Today"

as different words.

5. `valueOf()`

Signature: `valueOf(boolean b)`

What does the method do? This method will convert the value of its sole parameter to a String representation of the same value, returning that String representation. This method is overloaded and can be provided a value representing any of Java's basic types as a parameter--not just a boolean, as shown in the signature above.

Use Scenario: If we're preparing to serialize some data output generated by a Java program, it might make sense to convert all non-String types to Strings first. That way, if we tried to work with the same values in a different language, the data would be in a more universally recognized format: Javascript or Python could read a String output by a Java program, and a Javascript or Python developer could convert any values to an appropriate type when needed. For example, if we had a Java double, we could take its String value (obtained by `valueOf()`) and serialize it in a JSON object (which we'd presumably also represent as a longer string). We could then store the JSON object in a text file and pick it back up with Javascript (using `JSON.parse()`). If that original Java double was important, it would be easy to convert it from what is now a Javascript string to a Javascript number, but only because we obtained its string representation with `valueOf()` while we were still working in Java.

Prompt: What is your favorite thing you learned this week?

I really enjoyed using the ice-cream shop example to play with reading and displaying user input! I even implemented my own class (a "Pet") and then created a small program that populates an instance of the "Pet" class based on prompts to the user. You can find it on my github.