

This exercise will help you to practice manipulating Strings and using JUnit. This exercise will not be collected or graded. Visit us during office hours if you have questions.

1. Open a window where you can view the documentation for the String class API. You may use any String methods that you find convenient.
2. Create a project called “StringPractice”.
3. Create a class called “StringTools”.
4. Create JUnit test case called “StringToolsTesting”
5. Write the following methods in the StringTools class. After completing each method, write a THOROUGH JUnit test for it in the StringToolsTesting class. Be sure that you are testing several “typical” cases, and also any “corner cases” you can think of. Run your tests each time to see if your method is working properly in all cases.
  - a. `public static int count(String a, char c)`  
Returns the number of times the character c appears in the String a.
  - b. `public static int countSubstring(String a, String b)`  
Returns the number of times the String b occurs inside of the String a. For example, if a is “Mississippi”, and b is “issi”, then the return value would be 2. (Note that the occurrences of the substring may overlap!)
  - c. `public static String reverse(String a)`  
Returns a copy of the parameter, but with the letters reversed. For example, if the parameter is “monkey”, then the return value would be “yeknom”.
  - d. `public static String expand(String a)`  
Returns a String comprised of: One copy of the first character from a, two copies of the second character from a, three copies of the third character, etc. For example, if the parameter is “Fred” then the return value would be “Ffreeeedddd”.
  - e. `public static String alternating(String a, String b)`  
Returns a string formed using alternating letters from a and b. For example, if a is “cow” and b is “buffalo”, then the return value would be “cbouwffalo”. Note that either parameter could be the longer of the two.
  - f. `public static String linkExtractor(String a)`  
Looks inside the parameter for a substring of the form:  
`href = “something”`  
Where there could be 0 or more spaces on either side of the equals sign, and there could be anything between the quotes. If such a substring exists, return it. Otherwise, return null.