Tutorial: Java String Methods

Goals:

- Understand the **functionality** of Java **String methods**
- Learn to **call String methods**

Description:

Java provides a **String library** with plenty of methods that you can use for your program. In this tutorial, you will learn some of the common **String methods** and how to use them to manipulate a given string.

Contents:

*(**Note:** It will help with your understanding if you **code along** with this tutorial.)*

String Methods

When you are programming, there are times when you will come across problems that require String manipulation. The most basic problems can easily be solved with the methods provided by the Java **String library**. The table below lists the commonly used **String** methods, taken from the Java 8 String API documentation.

No.	Return Type	Method	Description
1	`int`	`charAt(int index)`	Returns the char value at the specified index.
2	`boolean`	`contains(CharSequence s)`	Returns true if and only if this string contains the specified sequence of char values.
3	`boolean`	`equals(Object anObject)`	Compares this string to the specified object.
4	`int`	`indexOf(int ch)`	Returns the index within this string of the first occurrence of the specified character.
5	`int`	`length()`	Returns the length of this string.
6	`String`	`replace(char oldChar, char newChar)`	Returns a string resulting from replacing all occurrences of oldChar in this string with newChar.
7	`String`	`substring(int beginIndex, int endIndex)`	Returns a string that is a substring of this string.
8	`String`	`toLowerCase()`	Converts all of the characters in this String to lower case using the rules of the default locale.
9	`String`	`valueOf(int i)`	Returns the string representation of the int argument.

(***Source:*** https://docs.oracle.com/javase/8/docs/api/java/lang/String.html)

String Methods

Before we begin, note that the index of the first letter of a String is 0.

Index:	0	1	2	3	4	5	6	7	8	9	10
	B	o	i	l	e	r	m	a	k	e	r

However, the string has the length of 11, because it is made of 11 letters.

Length:	1	2	3	4	5	6	7	8	9	10	11
	B	o	i	l	e	r	m	a	k	e	r

1. **charAt(int index)**
   Returns the char value at the specified index.

```
String str = "abcde";
System.out.println(str.charAt(1));
/* Output: b */
```

2. **contains(CharSqeuence s)**
   Returns true if and only if this string contains the specified sequence of char values.

```
String str = "hello";
System.out.println(str.contains("ll"));
/* Output: true */
```

3. **equals(Object anObject)**
   Compares this string to the specified object.
   (***Note:*** *This method is case sensitive! For case insensitive comparison, use* ***equalsIgnoreCase(String anotherString)*** *method.*)

```
String str1 = "hello";
String str2 = "hello";
String str3 = "Hello";

System.out.println(str1.equals(str2));
/* Output: true */

System.out.println(str1.equals(str3));
/* Output: false */

System.out.println(str1.equalsIgnoreCase(str3));
/* Output: true */
```

4. **indexOf(int ch)**
   Returns the index within this string of the first occurrence of the specified character.

```
String str = "hello";
System.out.println(str.indexOf("l"));
/* Output: 2 */
```

5. **length()**
   Returns the length of this string.

```
String str = "hello";
System.out.println(str.length());
/* Output: 5 */
```

6. **replace(char oldChar, char newChar)**
   Returns a string resulting from replacing all occurrences of oldChar in this string with newChar.

```
String str = "hello";
str = str.replace('l', 'z');
System.out.println(str);
/* Output: "hezzo" */
```

7. **substring(int beginIndex, int endIndex)**
   Returns a string that is a substring of this string.
   (***Note**: This method follows the [inclusive, exclusive) principle.*)

```
String str = "hello";
System.out.println(str.substring(2,4));
/* Output: "ll" */
```

8. **toLowerCase()**
   Converts all of the characters in this String to lower case.

```
String str = "HeLLo";

System.out.println(str.toLowerCase());
/* Output: "hello" */

System.out.println(str.toUpperCase());
/* Output: "HELLO" */
```

9. **valueOf(int i)**
   Returns the string representation of the int argument.
   There are also overloaded methods that perform the similar function for boolean, char, double, float, and long.

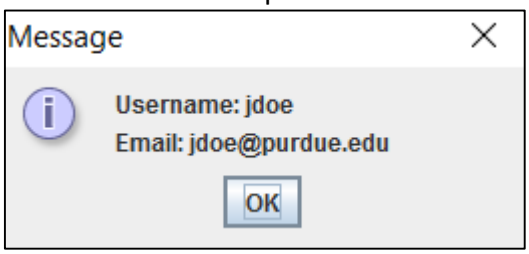
```
int i = 1;
String str = String.valueOf(i);
System.out.println(str);
/* Output: "1" */

double d = 1.23;
str = String.valueOf(d);
System.out.println(str)
/* Output: "1.23" */
```

Example:
This example shows how a username and email address is generated from a given string.

Username & Email Generator

```
String name = "John Doe";
char firstChar = name.charAt(0);
int startIndex = name.indexOf(" ") + 1;
int stopIndex = name.length();
String username = firstChar + name.substring(startIndex, stopIndex);
username = username.toLowerCase();
String email = username + "@purdue.edu";
JOptionPane.showMessageDialog(null, "Username: " + username + "\nEmail: " + email);
```

Output:



For more information on Java String methods, visit the Java 8 String API documentation: https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#equalsIgnoreCase-java.lang.String-