**48. Practising String Methods**

1. You can get methods easily.
2. String in Java are immutable.
3. Below function does not affect the current passed string but return the modified string.
4. When we assigned new string to older reference variable, the older string before assignment is lost and collected by the garbage collection mechanism.
5. **int str.length()**: It is used to get string of the number.  
   Example:   
   **String str = “Shreyash”;**  
   **int length = str.length(str);  
   System.out.println(“length”);  
   O/P: 8**
6. **String str.toUpperCase():**  It is used to get modified string to upper case.  
   Example:  
   **String str = “Shreyash”;**  
   **String str1 = str.toUpperCase();  
   System.out.println(“str1”);  
   O/P: SHREYASH**
7. **String str.trim():** It is used to remove blank spaces(White Spaces).  
   Example:  
   **String str = “ Shreyash ”;**  
   **System.out.println(“str1”);  
   String str1 = str.trim (str);  
   System.out.println(“str1”);  
   O/P 1: Shreyash  
   O/P 2: Shreyash**
8. **String str.substring(int begin):** It is used get the substrings(cut out small string from main string).  
   1. Example:  
      **String str = “Shreyash”;  
      String str1 = str.substring(2);  
      System.out.println(“str1”);  
      O/P:reyash**
   2. Example for (Overloaded Function)  
      **Syntax: str.substring(starting index, ending index);   
      \*Ending index is not included.**   
      **String str = “Shreyash”;  
      String str1 = str.substring(2,5);  
      System.out.println(“str1”);  
      O/P: rey**
9. **String str.replace(char old, char new):** It is used to replace a character with any other character.  
   Example:  
   **String str = “Shreyash”;  
   String str1 = str.replace (‘e’,’M’);  
   System.out.println(“str1”);  
   O/P:ShMyash**

**49. Practising String Methods #2**

1. **-1** means something is not found or invalid.
2. **boolean str.startsWith(String s):** It is used to check whether the string starts with give string or a character.  
   It does take space or tab as parameter.  
   Example:  
   **String name = ‘Shreyash’;  
   System.out.println(name.startsWith(‘Sh’));  
   O/P: true**
3. **boolean str.endsWith(String s):** It is used to check whether the string ends with given string or a character. Example:  
   **String name = ‘Shreyash’;  
   System.out.println(name.endsWith (‘sh’));  
   O/P: true**
4. **char str.charAt(int index):**  It is used to get the character at the given index number.   
   Example: **String name = ‘Shreyash’;  
   System.out.println(name.charAt (0));  
   O/P: S**
5. **int indexOf(String s):**  It is used to get the starting index of a given particular character or of the given String.  
   Example:  
   **String name = “Shreyash”;**  
   **System.out.println(name.indexOf(‘h’));  
   O/P: 1**
6. **int lastIndexOf(Strign s):**  It is used to get the ending index of a given particular character or of the given String.  
   Example:  
   **String name = “Shreyash”;  
   System.out.println(name.lastIndexOf(‘h’));  
   O/P: 7**

**50.Practising String Methods 2**

1. Example:

class Two  
{  
 public static void main(String[] args)  
 {  
 String str1 = "Mr. Shreyash Lokhande";  
 System.*out*.println(str1.startsWith("Mr"));  
 System.*out*.println(str1.endsWith("."));  
 System.*out*.println(str1.charAt(4));  
 System.*out*.println(str1.indexOf(".", 4));  
 System.*out*.println(str1.lastIndexOf("."));  
 }  
}

**51.String Methods #3**

1. **boolean equals(String s):** It is used to compare two strings.

It compare only the literals(value).

\***Unlike == compares references.**

2. **boolean equalsIgnoreCase(String s):** It is used to equate strings without considering case of the Strings.

Example:

**str1.equalsIgnoreCase(str2);**

3. **int compareTo(String s):** It is used to compare two strings in dictionary order.

It compares only the literal (value).  
Example:

**str1.compareTo(String s);**

**(Everything happens according to ASCII code)**

**\* smaller = comes first.**

**\* bigger = comes last.**

\* **-1 means first string is smaller than second string.(context dictionary)**

\* **1 means second string is smaller than first string. (context dictionary)**

\* **0 means both string are same.**

\* **Upper case of a string is smaller than Lower case of string.**

\* **The equals() and == are used to compare two strings.**

**str1.equals(str2) means only string value is compared.**

**str1 == str2 means string value and data-type of the object is compared.**

4. **boolean contains(String s):**  It is used to check whether string contains the given substring.

Example:

**str = “Hello wall”;**

**System.out.println(str.contains(“all”));**

**O/P: true**

5. **String concat(String s) / ‘+’ :** It is used to get two concatenated strings.

6. **String valueOf(int i):** It is used converting any value of any datatype into String. This is a static method.

Example:

public class Three  
{  
 public static void main(String[] args)  
 {  
 String str2="java";  
 String str1 = new String("java");  
 String str3="python";  
 String str4="Python";  
 System.*out*.println(str2.equals(str3));  
 System.*out*.println(str2.equalsIgnoreCase(str3));  
 System.*out*.println(str4.compareTo(str3));  
 System.*out*.println(str2.*valueOf*(21));  
  
 // reference  
 System.*out*.println(str1 == str2);  
 System.*out*.println(str2.equals(str1));  
  
 System.*out*.println(str1.*valueOf*(21).getClass());  
  
  
 }  
}

**51. Regular Expressions**

1. Regular expressions are used for defining patterns for strings.

2. **Boolean matches(String s):**

3. **Matching Symbol**

**Boolean matches():** It is used to find out patterns.

|  |  |
| --- | --- |
| **Regular Expression** | **Description** |
| . | Any character |
| [abc] | Exactly given letters. Between a to c |
| [abc][vc] | Either first or second test. a to c and v to z.  **\*Used for multiple symbols.** |
| [^abc] | Except abc |
| A|B | A or B |
| XZ | Exactly XZ (Exact string) |

**Meta Characters(Used for only one character):**

[\\d](file:///\\d) is used.

|  |  |
| --- | --- |
| **Regular Expression** | **Description** |
| \d | Digits |
| \D | Not Digits |
| \s | Space |
| \S | Not Space |
| \w | Alphabets or digit |
| \w | Neither alphabet or digit |

**54. String matching with regular expressions.**

1. **Quantifiers(To select one or multiple):**

|  |  |
| --- | --- |
| **Regular Expression** | **Description** |
| **\*** | 0 or more time |
| **+** | One or more |
| **?** | 0 or 1 time |
| **{X}** | X times. (length of strings be X) |
| **{X,Y}** | Between X and Y. (X = minimum and Y = maximum \***Range**) |

**55. Student Challenge: String methods.**

1. W.A.P to separate username and domain name from e-mail Id.

public class Six  
{  
 public static void main(String[] args)  
 {  
 //useing methods of string class find out whether gmail id is upon gmail or not with domain name and username.  
 String str = “shreyashlokhande1224@gmail.com”;  
 String sername = str.substring(0,str.indexOf(‘@’));  
 String domainName = str.substring(str.indexOf(‘@’) +1, str.length());  
 System.*out*.println(sername);  
 System.*out*.println(domainName);  
  
 if(domainName.matches(“gmail”))  
 {  
 System.*out*.println(“Yes it is a gmail.”);  
 }  
 }  
}

**56. Student Challenge: Regular Expressions #1**

Program: Write Regular for the following.

1. Check whether given number is a binary number or not.
2. Check whether the number or string in the form of number is a hexa-decimal or not.
3. Check whether the date is in the Date format (dd/mm/yyyy) or not.
4. import java.util.\*;  
   class Seven  
   {  
    public static void main(String[] args)  
    {  
    Scanner sc = new Scanner(System.*in*);  
    System.*out*.println("Enter your number:");  
    String number = "0110";  
    String number1 = "AB873C";  
    String dateFormat = "13/12/2004";  
     
    System.*out*.println(number.matches("[01]+"));  
    System.*out*.println(number1.matches("([A-F0-9])\*")); // Done it in ([A-Z]|[0-9])+ way.  
    System.*out*.println(dateFormat.matches("[0-3][0-9]/[01][0-9]/[0-9]")); // year we have done it in a way [0-9][0-9][0-9][0-9].  
    }  
   }

**\*A way to convert integer into a string.**

int b = 10110001;

String str = b+””;

Or

int b = 10110001;

String str = String.valueOf(b);

**57. Student Challenge: Regular Expression #2**

1. W.A.P

a. Remove Special characters from a string.

b. Remove extra spaces from string.

c. Find number of words in a String

2. **String replaceAll(String s):** It is used to replace string with a sub-string of

a give string.

3. **Split(String s):** It is used to split the string according to the string and return the array.

**\*Notes:**

**1. We can’t use meta characters in normal String functions except matches().**