**STRING METHODS**

**1.What is String contains( ) method?**

**Definition :** The **Java String class contains()** method searches the sequence of characters in this string. It returns true if the sequence of char values is found in this string otherwise returns false.

***Features :***

The contains() method is helpful to find a char-sequence in the string. We can use it in the control structure to produce the search-based result.

***Limitations:***

* The contains() method should not be used to search for a character in a string. Doing so results in an error.
* The contains() method only checks for the presence or absence of a string in another string. It never reveals at which index the searched index is found. Because of these limitations, it is better to use the indexOf() method instead of the contains() method.

**Example:**

1. **class** ContainsExample{
2. **public** **static** **void** main(String args[]){
3. String name="what do you know about me";
4. System.out.println(name.contains("do you know"));  //true
5. System.out.println(name.contains("about")); //true
6. System.out.println(name.contains("hello"));  //false
7. }}

**2. What is String endsWith( ) method?**

**Definition :** The **Java** **String class endsWith()** method checks if this string ends with a given suffix. It returns true if this string ends with the given suffix; else returns false.

***Features :***

The contains() method is helpful to find a char-sequence in the string. We can use it in the control structure to produce the search-based result.

***Limitations:***

* The internal implementation shows that the endWith() method is dependent on the startsWith() method of the String class.
* Since the endsWith() method returns a boolean value, the method can also be used in an if statement.
* The endsWith() method takes care of the case sensitiveness of the characters present in a string.

**Example:**

1. **public** **class** EndsWithExample{
2. **public** **static** **void** main(String args[]){
3. String s1="I am Learning Java";
4. System.out.println(s1.endsWith("e"));  //false
5. System.out.println(s1.endsWith("Java"));  //true
6. }}

**3. What is String equals( ) method?**

**Definition :** The **Java String class equals()** method compares the two given strings based on the content of the string. If any character is not matched, it returns false. If all characters are matched, it returns true.

***Features :***

* The String equals() method overrides the equals() method of the Object class.
* The equals() method compares two strings and can be used in if-else control structure.

***Example:***

1. **public** **class** EqualsExample {
2. **public** **static** **void** main(String args[]){
3. String s1="javafullstack";
4. String s2="javafullstack";
5. String s3="JAVATPOINT";
6. String s4="python";
7. System.out.println(s1.equals(s2)); //true because content and case is same
8. System.out.println(s1.equals(s3)); //false because case is not same
9. System.out.println(s1.equals(s4)); //false because content is not same
10. }}

**4.What is String contains( ) method?**

**Definition :** The **Java String class contains()** method searches the sequence of characters in this string. It returns true if the sequence of char values is found in this string otherwise returns false.

***Features :***

The contains() method is helpful to find a char-sequence in the string. We can use it in the control structure to produce the search-based result.

***Limitations:***

* The contains() method should not be used to search for a character in a string. Doing so results in an error.
* The contains() method only checks for the presence or absence of a string in another string. It never reveals at which index the searched index is found. Because of these limitations, it is better to use the indexOf() method instead of the contains() method.

**Example:**

1. **public** **class** ContainsExample2 {
2. **public** **static** **void** main(String[] args) {
3. String str = "Hello Javatpoint readers";
4. **boolean** isContains = str.contains("Javatpoint");
5. System.out.println(isContains);   // Case Sensitive
6. System.out.println(str.contains("javatpoint"));  // false
7. }
8. }

**5. What is String endsWith( ) method?**

**Definition :** The **Java** **String class endsWith()** method checks if this string ends with a given suffix. It returns true if this string ends with the given suffix; else returns false.

***Features :***

The contains() method is helpful to find a char-sequence in the string. We can use it in the control structure to produce the search-based result.

***Limitations:***

* The internal implementation shows that the endWith() method is dependent on the startsWith() method of the String class.
* Since the endsWith() method returns a boolean value, the method can also be used in an if statement.
* The endsWith() method takes care of the case sensitiveness of the characters present in a string.
* When an empty string is passed in the parameter of the method endsWith(), the method always returns a true value. The reason behind this is that a string never changes when we append an empty string to it.

**Example:**

1. **public** **class** EndsWithExample4
2. {
3. **public** **static** **void** main(String args[])   // main method
4. {
5. String str = "Welcome to JavaTpoint";
6. System.out.println(str.endsWith(""));  // prints true
7. System.out.println(str.endsWith(" "));  // prints false as there is no white space after the string
8. }
9. }