

Java Program to Check Palindrome String

Java – Check Palindrome String

A string is said to be palindrome, if it is same as that of its reversed value.

In this tutorial, we shall check if a string is palindrome in two ways. Firstly, by reversing the string and comparing it with the original value. Secondly, we shall traverse through string characters from start and end of the string, and kind of meet in the middle, checking if the characters have equal value.

Check Palindrome String using StringBuilder

String could be a palindrome if it equals the reverse of it.

In the following example, we shall reverse the given string in Java using `StringBuilder`, and compare it with its original value. If both are equal, then we can say that the given string is a palindrome.

PalindromeString.java

```
/**
 * Java Program - Check if String is Palindrome
 */

public class PalindromeString {

    public static void main(String[] args) {

        String str = "tattarrattat";

        //reverse the string
        String rev = (new StringBuilder(str)).reverse().toString();

        //check if str is palindrome
        if(str.equals(rev)) {
            System.out.println(str+" is Palindrome.");
        } else {
            System.out.println(str+" is not Palindrome.");
        }
    }
}
```

Output

```
tattarrattat is Palindrome
```

Check if String is Palindrome by Checking at Character Level

Instead of reversing the string and checking for equality, we can check if the first and last characters are equal and progress to the middle of the string. If at any point, the characters are not equal, then it is not a palindrome.

Algorithm

We shall implement following algorithm in Java and write a program to check if given string is palindrome.

1. Start.
2. Take string in **str**. We need to check if this is palindrome string or not.
3. Take a boolean variable **isPalindrome** to store if the string is palindrome or not. Initialize it with `true`.
4. Initialize variable **i** with `0`.
5. Check if **i** is less than half the length of string **str**. If yes, go to **step 6**, else go to **step 8**.
6. Check if character in **str** at index **i** is equal to that of at `length-1-i`. If not set **isPalindrome** to `false` and go to **step 8**.
7. Increment **i**. Go to **step 5**.
8. Based on the value of `isPalindrome`, print the result.
9. Stop.

PalindromeString.java

```
/**
 * Java Program - Check if String is Palindrome
 */

public class PalindromeString {

    public static void main(String[] args) {

        String str = "tattarrattat";

        boolean isPalindrome = true;

        //check if ith character is same from start and end
        for(int i=0;i<str.length()/2;i++) {
            if(str.charAt(i)!=str.charAt(str.length()-1-i)) {
                isPalindrome = false;
                break;
            }
        }

        //check if str is palindrome
        if(isPalindrome) {
            System.out.println(str+" is Palindrome.");
        } else {
            System.out.println(str+" is not Palindrome.");
        }
    }
}
```

```
}
```

Output

```
tattarrattat is Palindrome.
```

Conclusion

In this [Java Tutorial](#), we have written Java program using different techniques on how to check if given string is a palindrome or not.

Java Examples

- ◆ [Java Tutorial](#)
- ◆ [Check Positive Negative Number](#)
- ◆ [Read Integer From Console](#)
- ◆ [Add Two Integers](#)
- ◆ [Count Digits in Number](#)
- ◆ [Largest of Three Numbers](#)
- ◆ [Smallest of Three Numbers](#)
- ◆ [Display Even Numbers](#)
- ◆ [Display Odd Numbers](#)
- ◆ [Reverse a Number](#)
- ◆ [Check Prime Number](#)
- ◆ [Factorial of Number](#)
- ◆ [Print All Prime Numbers](#)
- ◆ [All Factors of a Number](#)
- ◆ [Check Palindrome Number](#)
- ⇒ **[Check Palindrome String](#)**
- ◆ [Swap Two Numbers](#)
- ◆ [Even or Odd Number](#)

Array Examples

- ◆ [Print Array Elements](#)

◆ Find Smallest Number in Array

◆ Find Largest Number in Array