



# Java Strings

[< Previous](#)[Next >](#)

## Java Strings

Strings are used for storing text.

A `String` variable contains a collection of characters surrounded by double quotes:

### Example

Create a variable of type `String` and assign it a value:

```
String greeting = "Hello";
```

[Try it Yourself »](#)

## String Length

A String in Java is actually an object, which contain methods that can perform certain operations on strings. For example, the length of a string can be found with the `length()` method:

☐ Dark mode



```
String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
System.out.println("The length of the txt string is: " + txt.length());
```

Try it Yourself »

## More String Methods

There are many string methods available, for example `toUpperCase()` and `toLowerCase()` :

### Example

```
String txt = "Hello World";  
System.out.println(txt.toUpperCase()); // Outputs "HELLO WORLD"  
System.out.println(txt.toLowerCase()); // Outputs "hello world"
```

Try it Yourself »

## Finding a Character in a String

The `indexOf()` method returns the **index** (the position) of the first occurrence of a specified text in a string (including whitespace):

### Example

```
String txt = "Please locate where 'locate' occurs!";  
System.out.println(txt.indexOf("locate")); // Outputs 7
```

☐ Dark mode



Java counts positions from zero.

0 is the first position in a string, 1 is the second, 2 is the third ...

---

## Complete String Reference

For a complete reference of String methods, go to our [Java String Methods Reference](#).

The reference contains descriptions and examples of all string methods.

---

## Test Yourself With Exercises

### Exercise:

Fill in the missing part to create a `greeting` variable of type `String` and assign it the value `Hello` .

```
greeting =      ;
```

[Submit Answer »](#)

[Start the Exercise](#)