Java Strings

‹ Previous Next ›

Java Strings

Strings are used for storing text.

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

Example

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

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

Run example »

Quotes Inside a String

To use quotes inside a string, use single quotes ('):

Example

```
String answer1 = "It's alright";
String answer2 = "He is called 'Johnny'";
```

Run example »

String Length

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

Example

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

Run example »

Other 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"
```

Run example »

Finding a String 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
```

Run example »

Java counts positions from zero.
0 is the first position in a string, 1 is the second, 2 is the third ...

String Concatenation

The `+` operator can be used between strings to add them together to make a new string. This is called **concatenation**:

Example

```
String firstName = "John";
String lastName = "Doe";
System.out.println(firstName + " " + lastName);
```

Run example »

Note that we have added an empty text (" ") to create a space between firstName and lastName on print.

You can also use the `concat()` method to concatenate two strings:

Example

```
String firstName = "John ";
String lastName = "Doe";
System.out.println(firstName.concat(lastName));
```

Run example »

Adding Numbers and Strings

> WARNING!
>
> Java uses the + operator for both addition and concatenation.
>
> Numbers are added. Strings are concatenated.

If you add two numbers, the result will be a number:

Example

```
int x = 10;
int y = 20;
int z = x + y;        // z will be 30 (an integer/number)
```

Run example »

If you add two strings, the result will be a string concatenation:

Example

```
String x = "10";
String y = "20";
String z = x + y;     // z will be 1020 (a String)
```

Run example »

If you add a number and a string, the result will be a string concatenation:

Example

```
String x = "10";
int y = 20;
String z = x + y;      // z will be 1020 (a String)
```

Run example »

Using the Java Keyword new

You can also create a String and assign values to it with the `new` keyword:

```
// The following example (a "string literal"):
String greeting = "Hello";

// Can also be created with the new keyword:
String greeting = new String("Hello");
```

Run example »

Strings in Java are objects, and an object in Java is declared with the `new` keyword. However, instead of using the `new` keyword, you can easily just write the string text inside double quotes. This is called a "String literal". You will learn more about objects in a later chapter.

‹ Previous Next ›