Java → Basic syntax and simple programs → Strings → Processing strings

Theory: Processing strings

52 minutes 0 / 5 problems solved

Skip this topic Start practicing

5803 users solved this topic. Latest completion was 7 minutes ago.

As you already know, a string is a sequence of characters with a single data type, and so is an array. Strings are similar to arrays: in some sense, a string looks like an array of characters. Moreover, you can iterate both over strings and arrays. Sometimes you may need to process the string and convert it to an array. In this topic, we will discuss the ways you can process them.

§1. Strings and arrays

It's possible to convert between strings and character arrays using special methods like `valueOf()` or `toCharArray()`:

```
char[] chars = { 'A', 'B', 'C', 'D', 'E', 'F' };

String stringFromChars = String.valueOf(chars); // "ABCDEF"


char[] charsFromString = stringFromChars.toCharArray(); // { 'A', 'B', 'C', 'D', 'E', 'F' }

String theSameString = new String(charsFromString); // "ABCDEF"
```

There is another way to turn a string into an array. Take a look:

```
String text = "Hello";
String[] parts = text.split(""); // {"H", "e", "l", "l", "o"}
```

Here we used a much more concise method that *splits* a string into parts. Let's consider it in more detail!

§2. Splitting the string

A string can be separated by delimiters to an array of strings. To perform this, call the method `split`, it divides a string into substrings by a **separator**. In the previous example, we used the `""` delimiter, which automatically splits a string into the smallest elements, substrings, that consist of one char.

If the delimiter is specified, the method returns an array of all the substrings and, notably, the delimiter itself is not included in any of the substrings:

```
String sentence = "a long text";
String[] words = sentence.split(" "); // {"a", "long", "text"}
```

Let's try to split an American phone number into country code, area code, central office code, and other remaining digits:

```
String number = "+1-213-345-6789";
String[] parts = number.split("-"); // {"+1", "213", "345", "6789"}
```

Note that all the parts are still strings no matter how they look!

Choose your delimiter wisely, otherwise, you can receive some sentences that start with a space:

```
String text = "That's one small step for a man, one giant leap for mankind.";
String[] parts = text.split(","); // {"That's one small step for a man", " one giant leap for mankind."}
```

You can choose any delimiter you prefer, even the combination of spaces and words:

Current topic:

Processing strings Stage 2

Topic depends on:

✓ Branching statements Stage 2

✓ Array Stage 2

✓ String Stage 2

```
String text = "I'm gonna be a programmer";
String[] parts = text.split(" gonna be "); // {"I'm", "a programmer"}
```

As you can see, the `split` method is also a good tool to get rid of something you don't need or don't want to use.

§3. Iterating over a string

It's possible to iterate over characters of a string using a loop **(while, do-while, for-loop).**

See the following example.

```
String scientistName = "Isaac Newton";

for (int i = 0; i < scientistName.length(); i++) {
    System.out.print(scientistName.charAt(i) + " "); // print the current character
}
```

The code outputs:

```
I s a a c   N e w t o n
```

§4. Conclusion

As you see, there are various methods to convert a string like `valueOf()`, `toCharArray()`, `split()`. You may also iterate over characters of a string in a loop. Using these methods will help you to solve different tasks for picking up your programming experience.

Report a typo

560 users liked this theory. **14** didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

Start practicing

Comments (5)   Hints (0)   Useful links (0)   Show discussion

Table of contents: