

How to iterate through a String [duplicate]

[Ask Question](#)

12

This question already has an answer here:
[What is the easiest/best/most correct way to iterate through the characters of a string in Java?](#)
13 answers

How can I iterate through a string in Java?

I'm trying to use a foreach style for loop

```
for(char x : examplestring)
{
    //action
}
```

javastringloops

shareimprove this question

edited Sep 26 '10 at 18:15

BalusC
794k27129723078

asked Sep 26 '10 at 18:05

muttley91
4,3702178126

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign UpOR SIGN IN WITH

GoogleFacebook

This question has been asked before and already has an answer. If those answers do not fully address your question, please [ask a new question](#).

add a comment

4 Answers

activeoldestvotes

35

If you want to use enhanced loop, you can convert the string to charArray

```
for (char ch : exampleString.toCharArray()){
    System.out.println(ch);
}
```

shareimprove this answer

answered Sep 26 '10 at 18:13

surajz
2,6412329

7

This is a lot worse that my version as it copies the `String` to a temporary array that is returned by the function. – cletus Sep 26 '10 at 18:25

@cletus - you can't access original array - String is immutable so same should be the source array. But in Java you can always modify an array, which would cause to break Java optimizations (internig). Anyway one copy is faster than many `length()` and `charAt()`. – gertas Sep 26 '10 at 18:43

4 @gertas that's exactly what I was saying. Because arrays are mutable it must be defensively copied. So you have the cost of that copy for what? Syntactic sugar. That's why this is a bad idea. – cletus Sep 26 '10 at 18:48

@cletus: but here it isn't syntactic sugar. Generally we have rather memory vs cpu problem. I wouldn't mind if JVM optimizes access to String methods somehow, but still you call `.length()` and `.charAt()` for each char. – gertas Sep 26 '10 at 18:56

@gertas: You need to give more credit to Java's optimizing compiler. Iterating by index is 2% faster on my machine (jre7-32bit-single) than iterating through a `char[]` copy. – Gunslinger47 Sep 26 '10 at 19:54

show 2 more comments

27

Java `String` s aren't cahracter `Iterable` . You'll need:

```
for (int i=0; i<examplestring.length(); i++) {
    char c = examplestring.charAt(i);
    ...
}
```

Awkward I know.

shareimprove this answer

answered Sep 26 '10 at 18:11

cletus
477k134818900

add a comment

2

Using [Guava](#) (r07) you can do this:

```
for(char c : Lists.charactersOf(someString)) { ... }
```

This has the convenience of using foreach while *not* copying the string to a new array. `Lists.charactersOf` returns a view of the string as a `List` .

shareimprove this answer

answered Sep 26 '10 at 19:03

ColinD
86.7k24169182

+1. I didn't know that r07 was out. Though, interestingly, this is the slowest of the available options. It took 49% longer to complete than an equivilant `charAt` test. – Gunslinger47 Sep 26 '10 at 20:26

@Gunslinger47: I imagine the need to box and unbox each char for this would slow it down a bit. The method is probably more intended to adapt strings for use with various `Collection` -based APIs than just for iteration like this. – ColinD Sep 27 '10 at 0:47

add a comment

0

How about this

```
for (int i=0;i < str.length();i++)
{
    System.out.println(str.substring(i,i+1));
}
```

shareimprove this answer

edited Mar 6 at 0:03

Shadow
3,63711934

answered Sep 28 '10 at 8:14

Dead Programmer
8,2891765105

add a comment

asked 7 years, 7 months ago

viewed 68,817 times

active 1 month ago

EVBOX EVBox Automotive

cloudjava-c

Embedded Firmware Engineer

Amsterdam, Netherlands Visa sponsor

c linux

View all 2 job openings!

Linked

216 What is the easiest/best/most correct way to iterate through the characters of a string in Java?

3 What does `str[newLength] = '0'` mean?

Related

5118 What is the difference between String and string in C#?

2371 How to efficiently iterate over each entry in a 'Map'?

2561 The most elegant way to iterate the words of a string

3118 Read/convert an InputStream to a String

1999 How do I loop through or enumerate a JavaScript object?

1492 Breaking out of nested loops in Java

2726 Iterate through a HashMap

7439 How to check whether a string contains a substring in JavaScript?

2415 How do I convert a String to an int in Java?

3621 For-each over an array in JavaScript?

Hot Network Questions

Is Picard actually fluent in French?

What is the difference between a qubit and a quantum state?

Is there a command to get the number of frames in a gif?

Which king of Burgundy sheltered two Robertians?

Does Commander's Strike reset Reaction?

How to create a villain for level 1 players

Please help me find myself

How to swap elements in a vector using an anonymous function in Octave?

Who counts as a companion for Suggestion?

Can my familiar aid me?

Made a mistake on check need help to fix

What should I do if someone submit a review of a manuscript in Elsevier system sooner than I know and I already prepared a detailed review?

The order of template specializations in C++

pgfplots How to add marker defined by x value and existing function?

Stacking multiple TikZ pictures

Why is automatic key detection hard?

How to prevent metagaming induced by trying to not metagame?

Probability of choosing ace of spades before any club

Role derivative of sigmoid function in neural networks

What do you wish you had known when you started as a writer?

Detect heat waves

Dry and Tough Rumproast

How to monetize uploaded consciousness?

Digit Product Sequences

Not the answer you're looking for? Browse other questions tagged [java](#) [string](#) [loops](#) or [ask your own question](#).

STACK OVERFLOW

QuestionsJobsDeveloper Jobs DirectorySalary CalculatorHelpMobile

PRODUCTS

TeamsTalentAdsEnterprise

COMPANY

AboutPressWork HereLegalPrivacy PolicyContact Us

STACK EXCHANGE NETWORK

TechnologyLife / ArtsCulture / RecreationScienceOther

BlogFacebookTwitterLinkedIn

site design / logo © 2018 Stack Exchange Inc; user contributions licensed under cc by-sa 3.0 with attribution required. rev 2018.4.23.30016