April 23rd, 2004, 11:48 AM

cjard
Sarcastic Member
Power Poster

TIP: A little tutorial about String.split();

As of version 1.4, the Java API now recommends that StringTokenizer no longer be used to cut a string up, **split()** method be used instead.

This is probably a good thing for the beginners, because arrays are a thing they can get their head around. a loop to scan over an array too (`for(int i=0; i<array.length; i++)`), whereas the subtleties of an Er difficult to grasp. (By the way, the process of using a loop like the one mentioned, to crawl over an array is actually a process of "enumera Java classes of Enumeration or Iterator.
This tutorial doesnt deal with Enumeration or Iterator either, so if you wish to know about them, pleas emake a separate thread)

So what is split() all about?

Very simply, it is a method you can call on a string, that causes the string to be chopped up and the bits pu StringTokenizer, it is NOT capable of returning the delimiters, so if you have a particular interest in the del

What is a delimiter? Delimit is maybe a shorter way of saying "denoting the limit" or "indicating the edge". If you have a string like this:
hello,this,is,a,string
you could choose a comma as your delimiter. the comma indicates the end of one word and the start of the next. it would split like:
hello
this
is
a
string
Equally, you could choose the letter **i** to be your demiliter. In this case the split string would run as:
hello,th
s,
s,astr
ng

Split returns you a chopped up string, and as mentioned in the small text above, it chops up based on delir
fence posts, and everything else is a fence panel. The only thing youre likely to be interested is the panel s
string:

Code:
```
string:  the cat sat on the mat
fence : |===|===|===|==|===|===|

| is a fence post
= are the panel sections
```

if we split that string around the "posts" we would end up with the following, separated strings:

Code:
```
the
cat
sat
on
the
mat
```

You may wonder, what would we get if there were multiple spaces, and we were using space as a delimiter

Code:
```
string:  the     cat  sat on the   mat
fence : |===|||||===||===|==|===|||===|
```

see the multiple fence posts with nothing between them.. no panels? well you just get zero length strings c

Code:
```
the
                    <-- a zero length string
                    <-- another 0-length
                    <-- third 0-length
                    <-- fourth 0-length
cat

sat
on
the


mat
```

but, there were 5 fence posts.. but only 4 blank strings? why is that? because if you hammered 5 fence pos
```
# # # # #
_1 2 3 4
```

there would be 4 "somethings" between them. In this case, the 4 "somethings" are zero length strings.. an

so it's pretty easy, that's how strings break up, and so far nothing is really different to string tokenizer.

The Difference

there are 2 methods:

split(String)
split(String, int)

the two are the same, except that if you provide an **int** then the split method will stop after it made that m
string, but you know all the information you really want is in the first 7 chunks (after splitting) you can pas
that you get an array of chunks back that is, at most, 7 long.

remember that split is a method of a String. We call it on the string we want to split, like this:
```
String myString = "hello world";
myString.split( String )
myString.split( String, int )
```

So, unlike the string tokenizer, we dont pass in the String we want to split. The String we pass in, is the de

..and here is where the real difference comes.

Delimiters

The delimiters are nothing like StringTokenizer's delimiters. With StringTokenizer you gave it a simple list:

```
new StringTokenizer( "hello,this.is:a;test", ",.:;")
```

Whenever StringTokenizer found just one of those delimiters, it would cut a chunk off the string. Very simp
more complex than one character.

Now the delimiters are a Regular Expression.. nothing to be afraid of, just something to be aware of.

Gentle Intro to Regular Expressions

Regular Expressions are used for matching text, and if your text editor that you use to write java, is fairly
TextPad.com and others), it will support RegularExpressions for finding data.
Youve probably used a find/replace dialog in your life, and if I asked you to find the next occurrence of the
would be easy.
But what if i asked you to find "a letter A, followed by any character, followed by (a number 1 or a number
Wow, suddenly a lot harder..

Well Regular Expressions let you do just that. There are special symbols like:

. means "any character"
\s means "any whitespace (tab, newline, space, new paragraph etc) character"
[] are or-group markers. anything within them is OR

so that search i asked for before would be:

Code:

```
 find: A.[12]Z
means: A, followed by ANY CHARACTER (the fullstop is like a wildcard),
       followed by [1 OR 2], followed by Z

it would match:
AA1Z
AB1Z
AC2Z

and hundreds more, but it wouldnt match:
aa1z
ab1z
abc1z
aa3z
```

Theres a lot in the grammar of a Regular Expression, and i wont detail it here.. there's an entire tutorial pr
thinking of writing a basic one for here, but for now you should at least be aware that split() uses a regular
delimiter. This causes some problems for the classical users of stringtokenizer..

A Problem

Suppose you wanted to split a whole document up into sentences. What separates one sentence from anotl

.

so with the new split() you could take the whole document as a String, and split it around full-stops (perioc

myDocument.split(".");

nope
Because remember, the delimtier is a regular expression, and a full stop in regex speak means "any single
put fenceposts everywhere, then give you a big load of nothing!

Code:

```
 text: This is my text. I will split it. Will it work?
split(".")
fence: ||||||||||||||||||||||||||||||||||||||||||||||||
```

So because we want to use literal . instead of special regular expression . we must do something different..

RegEx Compilation

Regular Expressions are compiled, something like java programs are compiled. It is a formal language, and
certain way. If you managed to write a pattern of regex that broke the language grammar, you would get a
your program was running. One such breakage may be using "*" as a regular expression.
* means "match the preceding character zero or more times", but if you dont precede it with anything, its i

There is a special character in regular expressions, which means "treat the following special character as a
That character is \

It is similar to the way you may wish to put a " character in a java string. You must put a \ before it, to let
end there:

```
System.out.print("to use a \" character in a string, you must put a slash \\ before it");
```

There, we used \" to indicate literal speech mark, and later \\ to indicate a literal slash.

If we put the pattern to split our document, as:
myDocument.split("\.");

that will make the regular expression compiler ignore the following character as a special character

BUT

the java compiler sees that \. before the regular expression compiler does, and the java compiler will try a
character' like it does with \n \b \r \t (newline, backspace, carriage return and tab)
So to make it turn that slash, into a literal slash so that the regex compiler sees a literal slash followed by
in our java source code:
myDocument.split("\\.");

\\. becomes \. (after javac) and
\. becomes "a literal full stop" (in the regular expression compiler)

super 

note, to get a literal slash into a regex requires the regex compiler to see \\, which in turn requires you to
\\\\ -> \\ -> \
talk about long winded.

How the RegEx is Used

In using a regular expression as a delimiter, the split() method basically starts at the start of the string an
regular expression. The following regular expression means "a or b or c, followed by 1 or 2 or 3":
"[abc][123]"

Here is how it would split some text:

```
text: hey, this is a1 a little demonstra3tion of the b2 power of a rec1gulab1r expression
is a0 hard at a3 first, but you get used b2 it
```

and the split array contains:

Code:

```
hey, this is
 a little demonstr
tion of the
 power of a re
gula
r expression when sp
litting text c4. it is a0 hard at
 first, but you get used
 it
```

Notice how the strings can start with spaces too.. thats because many times the regular text occurred as a
regular expressions dont know, or care about spaces, or words, or language!just literal text!

Oops..10000 char limit hit!
[/font]

_____
"it's a fax from your dog, Mr Dansworth. It looks like your cat" - Gary Larson...DW1: Data Walkthroughs 1.1...DW2: Data Walkthroughs 2.0...DDS: The DataSet Designer Surface...ANO: ADO.NET

Last edited by cjard; May 19th, 2004 at 09:03 AM.

April 23rd, 2004, 11:49 AM #**2**

cjard
Sarcastic Member
Power Poster

Join Date: Oct 2003
Location: .NET2.0 / VS2005 Developer
Posts: 7,070

if you get a real burning for Regular Expressions, here is the full grammar of special characters

And here is the regular expression trail.. a full blown tutorial by Sun, about their regexp packages.

_____
"it's a fax from your dog, Mr Dansworth. It looks like your cat" - Gary Larson...DW1: Data Walkthroughs 1.1...DW2: Data Walkthroughs 2.0...DDS: The DataSet Designer Surface...ANO: ADO.NET2 Orientation...DAN: Deeper ADO.NET...DNU...PQ

Quote

April 23rd, 2004, 11:55 AM

cjard
Sarcastic Member
Power Poster

here is the source for a little program i wrote to test splits i was working with:

Code:

```
public class SplitTest{
  public static void main(String[] argv){
    System.out.println("Usage: java SplitTest \"string to split\" \"regex to split by\"

    String[] ary = argv[0].split(argv[1]);

    for(int i=0;i<ary.length;i++){
      System.out.println("array["+i+"]\tlength("+ary[i].length()+")\t>"+ary[i]+"<");
    }
  }
}
```

here is the output sample (splitting on space.. in red):

Quote:

```
C:\javawork>java SplitTest " Date : 22. 4. 4 Time: 10.48 ,00009952,000.00," " "
Usage: java SplitTest "string to split" "regex to split by"
array[0] length(0) ><
array[1] length(0) ><
array[2] length(0) ><
array[3] length(0) ><
array[4] length(0) ><
array[5] length(0) ><
array[6] length(4) >Date<
array[7] length(1) >:<
array[8] length(3) >22.<
array[9] length(2) >4.<
array[10] length(1) >4<
array[11] length(5) >Time:<
array[12] length(5) >10.48<
array[13] length(0) ><
array[14] length(0) ><
array[15] length(0) ><
```

```
array[16] length(0) ><
array[17] length(0) ><
array[18] length(17) >,00009952,000.00,<
```

<attachment has been removed.. a new version that is more capable, appears later in the thread>

_____
"it's a fax from your dog, Mr Dansworth. It looks like your cat" - Gary Larson...DW1: Data Walkthroughs 1.1...DW2: Data Walkthroughs 2.0...DDS: The DataSet Designer Surface...ANO: ADO.NET

Last edited by cjard; April 28th, 2004 at 08:29 AM.

April 23rd, 2004, 12:43 PM

cjard
Sarcastic Member
Power Poster

Heres some more advanced examples. Im trying to get this line:

Code:

```
"      Date : 22. 4. 4 Time: 10.48      ,00009952,000.00,"
```

split so that theres a small array produced of JUST the date, 22nd of the 4 of the 4 (that's 2004)

the | character means OR, not quite the same as []
[] means an or group

[abc] means "a or b or c"
[abc][123] means "a or b or c" followed by "1 or 2 or 3" (it will find a1 a2 a3 b1 b2 b3 c1 c2 c3)

But as you can see, its hard to express "(a followed by b) or (a followed by c)" with an or group. the followi
[abac] simply means "a or b or a or c"
[ab][ac] means "a or b followed by a or c"
a[ba]c means "a followed by (b or a) followed by c"
which is the same text as what im trying to express:

"(a followed by b) or (a followed by c)"
"a followed by (b or a) followed by c"

but as any mathematician will know, the brackets are a BIG DEAL

So im using | to OR the expressions.

Heres an attempt that crashed:

Quote:

```
"*Date : |\. | Time.* "
Usage: java SplitTest "string to split" "regex to split by"
Exception in thread "main" java.util.regex.PatternSyntaxException: Dangling meta character '*' near index 0
*Date : |\. | Time.*
^
at java.util.regex.Pattern.error(Unknown Source)
at java.util.regex.Pattern.sequence(Unknown Source)
at java.util.regex.Pattern.expr(Unknown Source)
at java.util.regex.Pattern.compile(Unknown Source)
at java.util.regex.Pattern.<init>(Unknown Source)
at java.util.regex.Pattern.compile(Unknown Source)
at java.lang.String.split(Unknown Source)
```

```
at java.lang.String.split(Unknown Source)
at SplitTest.main(SplitTest.java:5)
```

Its that "Cant use a special character than needs something before it, at the start of a line" problem i ment

"*Date : |\. | Time.* "

means: "zero or more occurrences of <nothing> Folloewd by D followed by a, followed by t... blah blah"

let's try again:

Quote:

```
C:\javawork>java SplitTest " Date : 22. 4. 4 Time: 10.48 ,00009952,000.00," " *Date : |\. | Time.* "
Usage: java SplitTest "string to split" "regex to split by"
array[0] length(0) ><
array[1] length(2) >22<
array[2] length(1) >4<
array[3] length(1) >4<
array[4] length(17) >,00009952,000.00,<
```

hmm.. not bad, but why is there a blank at the start?
Remember that split hammers its first fence post in at the start of the line, it then looks for a delimiter. On OR THIS | OR THIS is:

Code:

```
" *Date : " //space, star, "Date", space, colon, space
```

so way hey.. it found a delimiter starting at character 0. remember that the start of the line is conceptually

Code:

```
 H e l l o  //characters
 0 1 2 3 4  //character indexes
|           //the first fence post is before the start
```

if we then provide a delimiter that matches at character 0(e.g. "H" will match the H at position 0):

Code:

```
 H e l l o  //characters
 0 1 2 3 4  //character indexes
|           //the first fence post is before the start
 |          //the second fence post is on char 0
```

whats between the first 2 posts? nothing.. thats why we are still getting a 0 length string at the first place regex that matches right from character 0.

I tried everything I could think of, including \A meaning "start of input" and ^ meaning "start of line" but i start.. i'll just have to ignore it

eventually I refined my regex to this:

Quote:

```
C:\javawork>java SplitTest " Date : 22. 4. 4 Time: 10.48 ,00009952,000.00," ".*Date : |`
```

```
Usage: java SplitTest "string to split" "regex to split by"
array[0] length(0) ><
array[1] length(2) >22<
array[2] length(1) >4<
array[3] length(1) >4<
```

".*Date : |\. | Time.*" means:

zero or more characters, followed by the word Date, followed by a space, a colon and a space
OR
a literal character fullstop (period), followed by a space
OR
a space, followed by the word Time, followed by any number of any kind of characters

heres a colorcoded matching of where the "fence post" (delimiters) are found(ive turned the spaces into un
them:

Code:

```
!______Date_:_22._4._4_Time:_10.48______,00009952,000.00,
```

what i get out of the string after it is split, is shown in pink. ive also shown a ! to indicate the start of the li
index in my array.

I recommend using a little app similar to SplitTest, if youre going to be using split() in any major way.. it w
guessing where input will come, and show you actual results because you may well forget about some thin
may cause array[0] to be a zero length string" thing..

Just remember that when you write your split string in on the command line, the program has already bee
suffer the "javac.exe compiler turns \\ into \" issue.. just write your regex in using characters in exactly th
help file link i gave in post number 2.. So when it says "use \W to mean a word character" you literally wri
you would write \\W in a source code

_____
"it's a fax from your dog, Mr Dansworth. It looks like your cat" - Gary Larson...DW1: Data Walkthroughs 1.1...DW2: Data Walkthroughs 2.0...DDS: The DataSet Designer Surface...ANO: ADO.NET

Last edited by cjard; April 28th, 2004 at 08:27 AM.

April 23rd, 2004, 01:09 PM

cjard
Sarcastic Member
Power Poster

Here is an updated version that will print at the end, the exact code to put in your program (It takes care c

Code:

```
public class SplitTest{
  public static void main(String[] argv){
    System.out.println("Usage: java SplitTest \"string to split\" \"regex to split by\"");

    String[] ary = argv[0].split(argv[1]);

    for(int i=0;i<ary.length;i++){
      System.out.println("array["+i+"]\tlength("+ary[i].length()+")\t>"+ary[i]+"<");
    }

    String regex = argv[1];
    if(regex.indexOf("\\")>=0){
      regex = regex.replaceAll("\\\\", "\\\\\\\\");
      System.out.println("\n\nYour delimiter expression contains at least one \\ character.\n"+
                         "To prevent the java compiler interpreting this as an escape\n"+
                         "code, you should use the following split() command in your \n.java sourcecode:\n\n"+
                         "myString.split(\""+regex+"\");");
    }else{
      System.out.println("\n\nUse the following split() command in your .java sourcecode:\n\n"+
                          "myString.split(\""+regex+"\");");
    }
  }
}
```

Attached Files

splittest.txt (1,007 Bytes, 987 views)

Last edited by cjard; April 23rd, 2004 at 01:19 PM.

April 23rd, 2004, 01:14 PM #6

mikeBarr81
Member +

Join Date: Dec 2003
Location: Colchester, England
Posts: 619

wow, i really started something here didn't i! well done again

Quote

April 23rd, 2004, 01:20 PM #7

Vow
Junior Member

Join Date: Jan 2004
Posts: 19

Nice work
Always wondered what all the fuss was about.

Quote

November 13th, 2008, 03:18 AM

Panajev
Junior Member

Re: TIP: A little tutorial about String.split();

Hello, great tutorial and great test harness/tool .

I took the tool and expanded on it a bit for ease of use needs, or so I hope, and I wanted to contribute back here.

It can read from the command line, it can read from a file, and is designed to work when run from Eclipse c command line (well, if you are honest at the menu selection part :P).

(I attached the whole Eclipse project in a .zip file as well as the .java file itself [which I am quoting in this p

Code:

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FilenameFilter;
import java.io.IOException;
import java.io.InputStreamReader;

public class SplitTest{

        public void splitting(String arr_s, String regex) {

                String[] ary = arr_s.split(regex);

                for(int i=0;i<ary.length;i++){
                        System.out.println("array["+i+"]\tlength("+ary[i].length()+")\t>
                }

                //String regex = argv[1];
                if(regex.indexOf("\\")>=0){
                        regex = regex.replaceAll("\\\\", "\\\\\\\\");
                        System.out.println("\n\nYour delimiter expression contains at le
                                        "To prevent the java compiler interpreting this
                                        "code, you should use the following split() comm
                                        "myString.split(\""+regex+"\");");
                }else{
                        System.out.println("\n\nUse the following split() command in you
                                        "myString.split(\""+regex+"\");");
                }

                return;

        }

        public static void main(String[] argv){
```

Attached Files

PAO_REGEXP_Test.zip (5.7 KB, 59 views)

File Type: java SplitTest.java (3.8 KB, 61 views)

Last edited by Panajev; November 13th, 2008 at 04:10 AM. Reason: (updated/fixed the menu selection part and re-uploaded

November 13th, 2008, 03:22 AM #9

Panajev
Junior Member

Join Date: Nov 2008
Posts: 2

Re: TIP: A little tutorial about String.split();

I know there are sloppy parts like the menu selection one, in which you can crash the program feeding the Integer's method with a non-number... I guess I should have parsed the input better, but it was intended as a quick and easy to use tool for people who wanted to test String.split() and learn regexp... ok, I get the point :P. I guess I will have to work on it a little bit more and maybe add a GUI too :P.

Edit: fixed the NumberFormatException issue and added a nice while(true) loop to allow you to try again to use the menu correctly .

Last edited by Panajev; November 13th, 2008 at 03:33 AM.

Quote

February 10th, 2009, 01:50 AM #**10**

newbie1
Junior Member

Join Date: Feb 2009
Posts: 1

Re: TIP: A little tutorial about String.split();

HELP NEEDED:
i would like split a string and place it in to an array[][].
can i do that?

for example i have this string.

String a = {"cat dog rat duck owl bird fish lion"};

i would to place it in an array[][]
so that it would look like this.

array[][] = {{"cat", "dog"},
{ "rat", "duck"},
....and so on...

how do i do that?

Last edited by newbie1; February 10th, 2009 at 02:06 AM.

Quote

February 10th, 2009, 06:04 AM #**11**

keang
Senior Member

Join Date: May 2006
Location: UK
Posts: 2,019

Re: TIP: A little tutorial about String.split();

Please start your own thread when posting a question, don't just add a post onto someone else's thread.

As to your question, the split method returns a String[] so you can't use it to directly create a 2D array. However you can use it to extract the names into a 1D array and then use whatever the required logic is to convert the 1D array into a 2D array by iterating over the array and copying the names into an appropriate 2D array.

BTW what is the logic for splitting the names into a 2D array from the example given it looks like the first 2 names go into the first element, the next 2 go into the next element etc etc.

Quote

Post Reply

Bookmarks

Digg del.icio.us StumbleUpon Google

CodeGuru Forums > Java Programming > Java Programming

TIP: A little tutorial about String.split();

« Previous Thread | Next Thread »

Posting Rules

You **may not** post new threads
You **may not** post replies
You **may not** post attachments
You **may not** edit your posts

BB code is **On**
Smilies are **On**
[IMG] code is **On**
HTML code is **Off**

Forum Jump

Java Programming Go

All times are GMT -5. The time now is 08:58 AM.

Contact Us - CodeGuru Forums - Archive - Privacy Statement - Top

Acceptable Use Policy

WebMediaBrands

internet.com | mediabistro.com | JustTechJobs | graphics.com

Search: Find

WebMediaBrands Corporate Info

Advertise | Newsletters | Shopping | E-mail Offers