Subject: regular expressions (parsing strings)
Posted by setthivoine you on Wed, 12 Nov 2003 20:04:20 GMT
View Forum Message <> Reply to Message

Hi,

My IDL programs need to parse a text file which contains variables that would need replacing at read-time. Specifically:

```
#os          =unix
#windows_var1 =g:
#windows_var2 =/rootdir/
#unix_var1    =/mnt/groupserver/
#unix_var2    =/rootdir/
#var3        =< <os>_var1 >< <os>_var2 >data/
#var4        =<var3>counterfile.txt
```

So #var3 eventually becomes '/mnt/groupserver/rootdir/data/' and #var4 becomes 'mnt/groupserver/rootdir/data/counterfile.txt'. And if #os was set to 'windows', #var4 becomes 'g:/rootdir/data/counterfile.txt'.

I am trying to use regular expressions to replace the text (specifically using strepex.pro from
 http://astro.uni-tuebingen.de/software/idl/aitlib/misc/strep ex.html)
but am having problems with nested tags.

Could anyone point me to somewhere that could help me out ?

Thanks
--Sett

Subject: Re: Regular expression
Posted by Foldy Lajos on Fri, 04 May 2007 15:21:11 GMT
View Forum Message <> Reply to Message

On Fri, 4 May 2007, Lasse Clausen wrote:

```
>  Hi there,
>
>  why does
>
>  print, stregex('[', '[\[]')
>
>  work, i.e. produce 0, whereas
>
```

You are searching for \ or [==> found.

> print, stregex(']', '[\]]')
>
> prints -1?
>

You are searching for \ followed by] ==> not found.

> print, stregex(']', '\]')
>
> works (i.e. prints 0).
>

You are searching for] ==> found.

\ loses its 'escape char' meaning in a bracket expression, and becomes an ordinary character.

regards,
lajos

Subject: Re: Regular expression
Posted by Allan Whiteford on Fri, 04 May 2007 15:41:17 GMT

View Forum Message <> Reply to Message

Fï¿½LDY Lajos wrote:
>
> On Fri, 4 May 2007, Lasse Clausen wrote:
>

<snip>

>
>
> \ loses its 'escape char' meaning in a bracket expression, and becomes
> an ordinary character.
>

Note, however, that this is different from the implementation inside
other languages such as Perl. General discussions of regular expressions
(outside of an IDL context) will typically assume that the above isn't
true. IDL is missing a lot of the functionallity that other regular

expression engines have.

Thanks,

Allan

Subject: Re: Regular expression
Posted by lasse on Fri, 04 May 2007 15:46:19 GMT
View Forum Message <> Reply to Message

On 4 May, 16:21, FÖLDY Lajos <f...@rmki.kfki.hu> wrote:
> On Fri, 4 May 2007, Lasse Clausen wrote:
>> Hi there,
>
>> why does
>
>> print, stregex('[', '[\[]')
>
>> work, i.e. produce 0, whereas
>
> You are searching for \ or [==> found.
>
>> print, stregex(']', '[\]]')
>
>> prints -1?
>
> You are searching for \ followed by] ==> not found.
>
>> print, stregex(']', '\]')
>
>> works (i.e. prints 0).
>
> You are searching for] ==> found.
>
> \ loses its 'escape char' meaning in a bracket expression, and becomes an
> ordinary character.
>
> regards,
> lajos

mhmm, don't understand. Ok, here we go: I have a string like this

bb[23]

where bb can be any combination of alphanumerics and the number can be
anything. I am looking for the regular expression that will match the
whole thing. My first idea was (at the moment I am not bothered about

the order of the different parts):

```
regex = '[a-zA-Z0-9\[\]]+'
```

but alas!

```
print, stregex('bb[23]', regex)
       4
```

What?! And any combination of omitting or changing the \ character
will result in either IDL complainign about non-balanced brackets, a
match at position 4 or it won't match.

Help?

Cheers
Lasse

Subject: Re: Regular expression
Posted by Allan Whiteford on Fri, 04 May 2007 15:56:00 GMT
View Forum Message <> Reply to Message

Lasse,

Either:

```
regex='[a-zA-Z0-9]+\[[0-9]+\]'
```

or:

```
regex='[a-zA-Z0-9]{2}\[[0-9]{2}\]'
```

depending on whether your 'bb' and '23' need to be exactly two
characters long or not.

Note also you may want to check whether you're matching a substring
inside your search string or the complete string. I'm not sure what you
want to do.

Thanks,

Allan

Lasse Clausen wrote:
> On 4 May, 16:21, Fï¿½LDY Lajos <f...@rmki.kfki.hu> wrote:
>
>> On Fri, 4 May 2007, Lasse Clausen wrote:

>>
>>> Hi there,
>>
>>> why does
>>
>>> print, stregex('[', '[\[]')
>>
>>> work, i.e. produce 0, whereas
>>
>> You are searching for \ or [==> found.
>>
>>
>>> print, stregex(']', '[\]]')
>>
>>> prints -1?
>>
>> You are searching for \ followed by] ==> not found.
>>
>>
>>> print, stregex(']', '\]')
>>
>>> works (i.e. prints 0).
>>
>> You are searching for] ==> found.
>>
>> \ loses its 'escape char' meaning in a bracket expression, and becomes an
>> ordinary character.
>>
>> regards,
>> lajos
>
>
> mhmm, don't understand. Ok, here we go: I have a string like this
>
> bb[23]
>
> where bb can be any combination of alphanumerics and the number can be
> anything. I am looking for the regular expression that will match the
> whole thing. My first idea was (at the moment I am not bothered about
> the order of the different parts):
>
> regex = '[a-zA-Z0-9\[\]]+'
>
> but alas!
>
> print, stregex('bb[23]', regex)
>   4
>

> What?! And any combination of omitting or changing the \ character
> will result in either IDL complainign about non-balanced brackets, a
> match at position 4 or it won't match.
>
> Help?
>
> Cheers
> Lasse
>

Subject: Re: Regular expression
Posted by Foldy Lajos on Fri, 04 May 2007 16:04:41 GMT

View Forum Message <> Reply to Message

On Fri, 4 May 2007, Lasse Clausen wrote:

> mhmm, don't understand. Ok, here we go: I have a string like this
>
> bb[23]
>
> where bb can be any combination of alphanumerics and the number can be
> anything. I am looking for the regular expression that will match the
> whole thing. My first idea was (at the moment I am not bothered about
> the order of the different parts):
>
> regex = '[a-zA-Z0-9\[\]]+'
>

This regexp searches for a bracket expression (a-zA-Z0-9\[\) followed by one or more]'s. (\ behaves as an ordinary character after the opening bracket [, so the first] is the closing bracket.)

> but alas!
>
> print, stregex('bb[23]', regex)
>   4
>

3 matches the bracket expr. and] matches itself. So the answer is 4.

> What?! And any combination of omitting or changing the \ character
> will result in either IDL complainign about non-balanced brackets, a
> match at position 4 or it won't match.
>

Try something like this:

```
  [a-zA-Z0-9]+   one ore more alphanumeric char
  \[             [
  [0-9]+         one or more digits
  ]              }
```

ie:

regex = '[a-zA-Z0-9]+\[[0-9]+]'

regards,
lajos

Subject: Re: Regular expression
Posted by lasse on Fri, 04 May 2007 16:36:22 GMT

View Forum Message <> Reply to Message

On 4 May, 16:56, Allan Whiteford
<allan.rem...@phys.remove.strath.ac.remove.uk> wrote:
> Lasse,
>
> Either:
>
> regex='[a-zA-Z0-9]+\[[0-9]+\]'
>
> or:
>
> regex='[a-zA-Z0-9]{2}\[[0-9]{2}\]'
>
> depending on whether your 'bb' and '23' need to be exactly two
> characters long or not.
>
> Note also you may want to check whether you're matching a substring
> inside your search string or the complete string. I'm not sure what you
> want to do.
>
> Thanks,
>
> Allan
>
> Lasse Clausen wrote:
>> On 4 May, 16:21, FÖLDY Lajos <f...@rmki.kfki.hu> wrote:
>
>>> On Fri, 4 May 2007, Lasse Clausen wrote:
>
>>>> Hi there,

>
>>>> why does
>
>>>> print, stregex('[', '[\[]')
>
>>>> work, i.e. produce 0, whereas
>
>>> You are searching for \ or [==> found.
>
>>>> print, stregex(']', '[\]]')
>
>>>> prints -1?
>
>>> You are searching for \ followed by] ==> not found.
>
>>>> print, stregex(']', '\]')
>
>>>> works (i.e. prints 0).
>
>>> You are searching for] ==> found.
>
>>> \ loses its 'escape char' meaning in a bracket expression, and becomes an
>>> ordinary character.
>
>>> regards,
>>> lajos
>
>> mhmm, don't understand. Ok, here we go: I have a string like this
>
>> bb[23]
>
>> where bb can be any combination of alphanumerics and the number can be
>> anything. I am looking for the regular expression that will match the
>> whole thing. My first idea was (at the moment I am not bothered about
>> the order of the different parts):
>
>> regex = '[a-zA-Z0-9\[\]]+'
>
>> but alas!
>
>> print, stregex('bb[23]', regex)
>>   4
>
>> What?! And any combination of omitting or changing the \ character
>> will result in either IDL complainign about non-balanced brackets, a
>> match at position 4 or it won't match.
>
>> Help?

>
>> Cheers
>> Lasse

Thanks for the reply. I realized that I could do it the way you (Allan) proposed, without including the brackets in the character group, but I was being more academic and looking for a way to include them in the character group. The following works

```
print, stregex('bb[23]', '[][0-9a-b]+', length=length) & print, length
           0
           6
```

however, order is, not surprisingly, essential:

```
print, stregex('bb[23]', '[[]0-9a-b]+', length=length) & print, length
          -1
          -1
```

Cheers
Lasse

Subject: Re: Regular expression
Posted by James Kuyper on Fri, 04 May 2007 17:44:40 GMT
View Forum Message <> Reply to Message

Subject: Re: Regular expression
Posted by Foldy Lajos on Fri, 04 May 2007 18:13:18 GMT
View Forum Message <> Reply to Message

On Fri, 4 May 2007, kuyper@wizard.net wrote:

> FÖLDY Lajos wrote:
>> On Fri, 4 May 2007, Lasse Clausen wrote:
>>
>>> Hi there,
>>>
>>> why does
>>>
>>> print, stregex('[', '[\[]')
>>>
>>> work, i.e. produce 0, whereas
>>>

>>
>> You are searching for \ or [==> found.
> ...
>> \ loses its 'escape char' meaning in a bracket expression, and becomes an
>> ordinary character.
>
> In other cases, such as the unix vi command, the regular expression \
> [[^\]]*] matches any string that starts with '[', has an arbitrarily
> long string of characters that are not ']', followed by a ']'
> character. In IDL, however, stregex("ab[23]", "\[[^\]]*]") returns -1.
> Is there any simple way to perform a similar search using IDL regular
> expression?
>

It's easy, just omit the backslash: print, stregex("ab[23]", "\[[^]]*]")
If you want to put a] in the non-matching list, put it right after the ^.

regards,
lajos

Subject: Re: Regular expression
Posted by Allan Whiteford on Tue, 08 May 2007 12:10:37 GMT
View Forum Message <> Reply to Message

Lasse Clausen wrote:
> On 4 May, 16:56, Allan Whiteford
> <allan.rem...@phys.remove.strath.ac.remove.uk> wrote:
>
>> Lasse,
>>
>> Either:
>>
>> regex='[a-zA-Z0-9]+\[[0-9]+\]'
>>
>> or:
>>
>> regex='[a-zA-Z0-9]{2}\[[0-9]{2}\]'
>>
>> depending on whether your 'bb' and '23' need to be exactly two
>> characters long or not.
>>
>> Note also you may want to check whether you're matching a substring
>> inside your search string or the complete string. I'm not sure what you
>> want to do.
>>
>> Thanks,
>>

>> Allan
>>
>> Lasse Clausen wrote:
>>
>>> On 4 May, 16:21, Fï¿½LDY Lajos <f...@rmki.kfki.hu> wrote:
>>
>>>> On Fri, 4 May 2007, Lasse Clausen wrote:
>>
>>>> >Hi there,
>>
>>>> >why does
>>
>>>> >print, stregex('[', '[\[]')
>>
>>>> >work, i.e. produce 0, whereas
>>
>>>> You are searching for \ or [==> found.
>>
>>>> >print, stregex(']', '[\]]')
>>
>>>> >prints -1?
>>
>>>> You are searching for \ followed by] ==> not found.
>>
>>>> >print, stregex(']', '\]')
>>
>>>> >works (i.e. prints 0).
>>
>>>> You are searching for] ==> found.
>>
>>>> \ loses its 'escape char' meaning in a bracket expression, and becomes an
>>>> ordinary character.
>>
>>>> regards,
>>>> lajos
>>
>>> mhmm, don't understand. Ok, here we go: I have a string like this
>>
>>> bb[23]
>>
>>> where bb can be any combination of alphanumerics and the number can be
>>> anything. I am looking for the regular expression that will match the
>>> whole thing. My first idea was (at the moment I am not bothered about
>>> the order of the different parts):
>>
>>> regex = '[a-zA-Z0-9\[\]]+'
>>
>>> but alas!

```
>>
>>> print, stregex('bb[23]', regex)
>>>           4
>>
>>> What?! And any combination of omitting or changing the \ character
>>> will result in either IDL complainign about non-balanced brackets, a
>>> match at position 4 or it won't match.
>>
>>> Help?
>>
>>> Cheers
>>> Lasse
>
>
>  Thanks for the reply. I realized that I could do it the way you
>  (Allan) proposed, without including the brackets in the character
>  group, but I was being more academic and looking for a way to include
>  them in the character group. The following works
>
>  print, stregex('bb[23]', '[][0-9a-b]+', length=length) & print, length
>            0
>            6
>
>  however, order is, not surprisingly, essential:
>
>  print, stregex('bb[23]', '[[]0-9a-b]+', length=length) & print, length
>           -1
>           -1
>
>  Cheers
>  Lasse
>
```

Lasse,

That regular expression will pretty much match anything though:

```
IDL> print, stregex('bb[23]', '[][0-9a-b]+', length=length) & print, length
           0
           6
IDL> print, stregex('bba23a', '[][0-9a-b]+', length=length) & print, length
           0
           6
```

You can't put the square brackets in the range of characters to match
unless you're willing for them to be optional which I'd presume you
don't want. In the example above an 'a' is just as good as a '[' or a ']'.

Thanks,

Allan