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
